

---

安徽科大讯飞信息科技股份有限公司

**ANHUI USTC IFLYTEK CO.,LTD**

---

科大讯飞 MSC 新手指南

---

## 重要声明

### 版权声明

版权所有 © 2013, 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在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/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 © 安徽科大讯飞信息科技股份有限公司

Copyright © 2013 ANHUI USTC iFLYTEK CO., LTD.

## 1. 概述

本文档是开发科大讯飞 Android 语音程序的用户指南，定义了语音听写、语音合成、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。

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.CHANGE_NETWORK_STATE" />
<uses-permission android:name="android.permission.READ_PHONE_STATE"/>
```

工程混淆设置：如需打包或者生成 APK 的时候进行混淆，在 proguard.cfg 中添加如下代码

```
-keep class com.iflytek.**{*;};
```

## 2. MSC 概述

科大讯飞 MSC SDK 可以为开发者提供语音听写、语音合成、语义理解等功能，如图 3-1 所示。

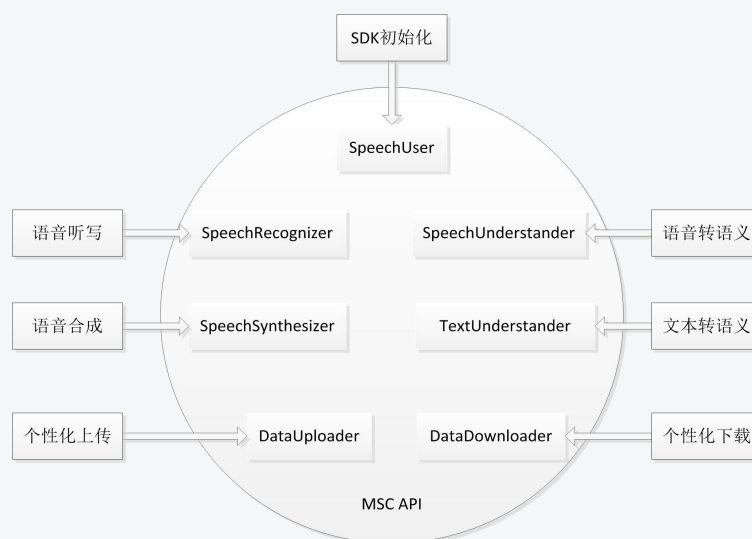
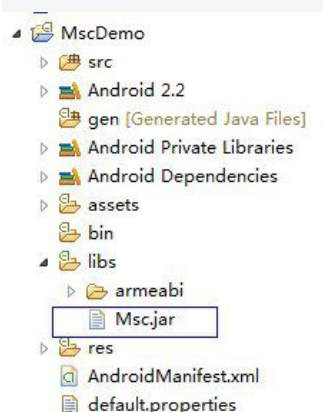


图 3-1 MSC 功能结构图

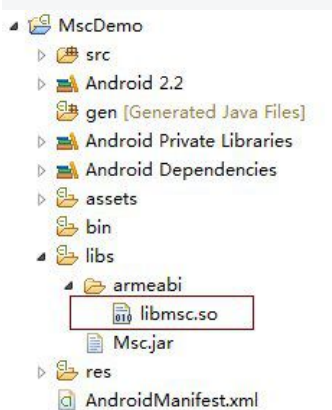
## 3. 搭建开发环境

[1] 在 Eclipse 中建立你的 Android 工程。

[2] 将开发工具包中 libs 目录下的 Msc.jar 复制到新建工程的 libs 目录中（如下图所示）。



[3] 将开发工具包中 armeabi（armeabi-v7a, mips, x86）目录下的 libmsc.so 复制到新建工程的 armeabi 目录中（如下图所示）。



[4] 在你需要使用 MSC 服务的文件中导入相应的类。

例如：`import com.iflytek.cloud.speech.SpeechRecognizer;`

## 4. 在代码中使用开发工具包

### [1] 语音转写

使用示例如下图所示。

```
// 创建SpeechRecognizer对象，使用前要先登录
// 详细内容请参考MscDemo代码
SpeechUser.getUser().login(this,null,null, "appid=" + APP_ID,listener);
SpeechRecognizer recognizer = SpeechRecognizer.createRecognizer(this);
recognizer.setParameter(SpeechConstant.DOMAIN,"iat");
recognizer.setParameter(SpeechConstant.LANGUAGE,"zh_cn");
recognizer.setParameter(SpeechConstant.ACCENT,"accent");
recognizer.startListening(mRecoListener);
//转写监听器
private RecognizerListener mRecoListener = new RecognizerListener()
{
    // 识别结果回调接口(返回Json格式结果，用户可参见附录)；
    // 一般情况下会通过onResults接口多次返回结果，完整的识别内容是多次结果的累加；
    //关于解析Json的代码可参见MscDemo中JsonParser类；
    //isLast等于true时会话结束。
    public void onResult(RecognizerResult results, boolean isLast) {
        results.getResultString ( );
    }
    // 会话发生错误回调接口
    public void onError(SpeechError error) {
    }
    //开始录音
    public void onBeginOfSpeech() {
    }
    //音量值0~30
    public void onVolumeChanged(int volume){
    }
    //结束录音
    public void onEndOfSpeech() {
    }
    //扩展用接口
    public void onEvent(int eventType,int arg1,int arg2,String msg) {
    }
};
```

## [2] 语音合成

使用示例如下图所示。

```
// 创建 SpeechSynthesizer 对象，使用前要先登录
// 详细内容请参考 sample 代码
SpeechSynthesizer mSpeechSynthesizer=SpeechSynthesizer.createSynthesizer(this);
//设置发音人
mSpeechSynthesizer.setParameter(SpeechConstant.VOICE_NAME,"xiaoyu");
//设置语速
mSpeechSynthesizer.setParameter(SpeechConstant.SPEED, "50");
//开始合成
mSpeechSynthesizer.startSpeaking("安徽科大讯飞信息科技股份有限公司",mSynListener);
//合成监听器
private SynthesizerListener mSynListener = new SynthesizerListener()
{
    // 会话结束回调接口，没有错误时，error 为 null
    public void onCompleted(SpeechError error) {
    }
    // 缓冲进度回调
    //percent 为缓冲进度 0~100，beginPos 为缓冲音频在文本中开始位置，endPos 表示缓冲音频在文本中结束位置，info 为附加信息。
    public void onBufferProgress(int progress, int beginPos,int endPos, String info) {
    }
    //开始播放
    public void onSpeakBegin() {
    }
    //暂停播放
    public void onSpeakPaused() {
    }
    //播放进度回调
    //percent 为播放进度 0~100,beginPos 为播放音频在文本中开始位置，endPos 表示播放音频在文本中结束位置。
    public void onSpeakProgress(int percent, int beginPos, int endPos) {
    }
    //恢复播放回调接口
    public void onSpeakResumed() {
    }
};
```

### [3] 登录示例

使用示例如下图所示。

```
// 用户只有在登录状态才能进行数据上传、下载、语义搜索、密码文本下载等操作
// 注册用户登录需要上传用户名和密码，非匿名方式
// 用户名和密码传 null，表示使用匿名方式进行登录,每台设备具备唯一性，建议使用此种方式
SpeechUser.getUser().login(MainActivity.this, null, null
    , SpeechConstant.APPID + "=123456789", loginlistener);
// 登录回调监听器
SpeechListener loginListener = new SpeechListener()
{
    // 若 error 为 null，表示登录成功，反之登录失败
    public void onCompleted(SpeechError error) {
    }
    // 登录操作服务端不会返回数据，忽略此接口
    public void onData(byte[] buffer) {
    }
    // 扩展消息，忽略此接口
    public void onEvent(int eventType, Bundle params) {
    }
};
```

### [4] 语义示例

使用示例如下图所示。

```
SpeechUnderstander speechUnderstander = new SpeechUnderstander();
//清空 Grammar_ID，防止语义理解后进行语义理解时 Grammar_ID 的干扰
speechUnderstander.setParameter(SpeechConstant.CLOUD_GRAMMAR, null);
//设置语义理解的引擎
speechUnderstander.setParameter(SpeechConstant.DOMAIN, "iat");
//设置采样率参数，支持 8K 和 16K
speechUnderstander.setParameter(SpeechConstant.SAMPLE_RATE, "16000");
//默认使用开放语义 2.0 协议，返回 JSON 结果；
//需要在开放语义平台 (http://osp.voicecloud.cn) 上进行配置
/*如需使用旧版开放语义协议，需添加如下代码：
*speechUnderstander.setParameter(SpeechConstant.NLP_VERSION, null);
speechUnderstander.setParameter(SpeechConstant.RESULT_TYPE, "xml");*/
speechUnderstander.startUnderstanding(understanderListener);
SpeechUnderstanderListener understanderListener = new SpeechUnderstanderListener()
{
    /**SpeechUnderstanderListener其它回调接口
    *出错情况下回调onError接口，正常返回结果通过onResult回调接口*/
```

```
public void onResult(UnderstanderResult result) {  
    mResultText.append(result.getResultString());  
    //光标定位到第一行  
    mResultText.setSelection(1);  
}  
}
```

## [5] 上传联系人

使用示例如下图所示。

```
//使用上传联系人前，请确保已经登录，否则会造成上传联系人不成功。  
ContactManager mgr = ContactManager.getManager();  
if(mgr == null)  
    //创建一个 ContactManager 实例化对象，  
    mgr = ContactManager.createManager(NLPDemo.this, new ContactListener() {  
  
        /**  
         * 联系人信息回调接口  
         * @param contactInfos 联系人信息，数据格式为：张三/n 李四/n 王五  
         * @param changeFlag 为 true 时表示联系人信息发生变更，为 false 表示未发生变更。  
         */  
        public void onContactQueryFinish(String contactInfos, boolean changeFlag) {  
            try {  
                // 若非首次上传且联系人信息未发生变更，则不进行上传操作  
                if(!changeFlag || TextUtils.isEmpty(contactInfos))  
                    return;  
                byte[] datas = contactInfos.getBytes("utf-8");  
                //创建上传对象  
                DataUploader dataUploader = new DataUploader( );  
                // 联系人上传的参数为 subject=uup,data_type=contact  
                dataUploader.uploadData(latDemoActivity.this, uploadListener, "contacts"  
                    , "subject=uup,data_type=contact", datas);  
            } catch (UnsupportedEncodingException e) {  
                e.printStackTrace();  
            }  
        }  
    });  
// 异步查询联系人接口，通过 onContactQueryFinish 接口回调  
mgr.asyncQueryAllContactsName();
```



## [6] 用户词表的上传与下载

使用示例如下图所示。

```
//在上传和下载用户词表前，先登录。
//用户词表
UserWords userwords = new UserWords();
userwords.putWord("词条 1");
userwords.putWord("词条 2");
userwords.putWord("词条 3");
//创建数据上传实例
DataUploader dataUploader = new DataUploader();
//调用 toString 获取用户词表的 json 字符串，再转换为二进制数组。
byte[] datas = userwords.toString().getBytes("utf-8");
//用户词表上传的参数为 subject=uup,data_type=userword
dataUploader.uploadData(IatDemoActivity.this, null, "userwords"
    , "subject=uup,data_type=userword", datas);
//词表数据下载
DataDownloader dataDownloader = new DataDownloader();
dataDownloader.downloadData(NLPDemo.this, downloadlistener
    , "subject=spp,data_type=userword");
/**
 * 用户词表下载监听器
 */
SpeechListener downloadlistener = new SpeechListener()
{
    // 返回下载数据
    public void onData(byte[] arg0) {
        // 获得用户词表内容
        String text = new String(arg0,"utf-8")
    }

    //会话结束接口
    //error 不为 null 表示上传失败
    public void onCompleted(SpeechError error) {
    }
    // 扩展接口，忽略此消息
    public void onEvent(int eventType, Bundle params) {
    }
};
```

## [7] abnf 语法上传示例

使用示例如下图所示。

```
// 上传语法需要先进行登录操作，请参考登录示例
SpeechUser.getUser().login(MainActivity.this, null, null
    , SpeechConstant.APPID + "=123456789",null);

// ABNF语法示例，可以说“北京到上海”
String grammar = "#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
DataUploader uploader = new DataUploader();
// subject=asr,data_type=abnf表示上传abnf语法参数
uploader.uploadData(this, uploadListener, "abnf", "subject=asr,data_type=abnf",
    grammar.getBytes("gb2312"));

// 语法ID
String grammarID = null;
// 上传数据回调监听器
SpeechListener uploadListener = new SpeechListener()
{
    // 会话结束回调
    // 若error不为null，对话框将显示错误提示
    public void onCompleted(SpeechError error) {
    }

    // 回调数据
    public void onData(byte[] buffer) {
        // 获得语法ID
        grammarID = new String(buffer);
    }

    // 扩展接口，忽略此消息
    public void onEvent(int eventType, Bundle params) {
    }
};
```

## 5. 附录

### 1. 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",
          "sc": 0
        }
      ]
    },
    {
      "bg": 0,
      "cw": [
        {
          "w": "的",
          "sc": 0
        }
      ]
    },
    {
      "bg": 0,
      "cw": [
        {
          "w": "天",
          "sc": 0
        }
      ]
    },
    {
      "bg": 0,
      "cw": [
        {
          "w": "气",
          "sc": 0
        }
      ]
    },
    {
      "bg": 0,
      "cw": [
        {
          "w": "怎么样",
          "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. 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	

### 3. 错误码列表

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

错误码	错误值	意义
ERROR_NO_NETWORK	20001	无有效的网络连接
ERROR_NETWORK_TIMEOUT	20002	网络连接超时
ERROR_NET_EXPECTATION	20003	网络连接发生异常
ERROR_INVALID_RESULT	20004	无有效的结果
ERROR_NO_MATCH	20005	无匹配结果
ERROR_AUDIO_RECORD	20006	录音失败
ERROR_NO_SPEECH	20007	未检测到语音
ERROR_SPEECH_TIMEOUT	20008	音频输入超时
ERROR_EMPTY_UTTERANCE	20009	无效的文本输入
ERROR_FILE_ACCESS	20010	文件读写失败
ERROR_PLAY_MEDIA	20011	音频播放失败
ERROR_INVALID_PARAM	20012	无效的参数
ERROR_TEXT_OVERFLOW	20013	文本溢出
ERROR_INVALID_DATA	20014	无效数据
ERROR_LOGIN	20015	用户未登陆
ERROR_PERMISSION_DENIED	20016	无效授权
ERROR_INTERRUPT	20017	被异常打断
ERROR_VERSION_LOWER	20018	版本过低
ERROR_COMPONENT_NOT_INSTALLED	21001	没有安装语音组件
ERROR_ENGINE_NOT_SUPPORTED	21002	引擎不支持
ERROR_ENGINE_INIT_FAIL	21003	初始化失败
ERROR_ENGINE_CALL_FAIL	21004	调用失败
ERROR_ENGINE_BUSY	21005	引擎繁忙
ERROR_LOCAL_NO_INIT	22001	本地引擎未初始化
ERROR_LOCAL_RESOURCE	22002	本地引擎无资源
ERROR_LOCAL_ENGINE	22003	本地引擎内部错误
ERROR_IVW_INTERRUPT	22004	本地唤醒引擎被异常打断
ERROR_UNKNOWN	20999	未知错误