
SPEECH 文档

2017-06-28



百度云
cloud.baidu.com

目录

1 前言	1
1.1 意图表示协议	1
1.1.1 意图表示协议描述语言	1
1.1.2 意图表示协议	2
1.1.3 领域取值	3
1.1.4 意图取值	4
1.1.5 属性值类型	5
1.2 各领域意图表示	9
1.2.1 天气	9
1.2.2 日历	11
1.2.3 列车	13
1.2.4 航班	16
1.2.5 地图	18
1.2.6 电话	20
1.2.7 通讯录	22
1.2.8 短信	24
1.2.9 手机应用	27
1.2.10 网站	29
1.2.11 日程提醒	30
1.2.12 社交网络	32
1.2.13 手机设置	33

1.2.14 音乐	36
1.2.15 笑话	39
1.2.16 故事	40
1.2.17 酒店	41
1.2.18 旅游	43
1.2.19 通用指令	44
1.2.20 翻译	46
1.2.21 话费流量查询	47
1.2.22 视频	49
1.2.23 电视节目	51
1.2.24 人物	52
1.2.25 电视指令	53
1.2.26 股票	55
1.2.27 小说	56
1.2.28 播放器	57
1.2.29 记账	59
1.2.30 搜索	61
1.2.31 车载指令	62
1.2.32 收音机控制指令	63
1.2.33 菜谱	65
1.2.34 影讯	66
1.2.35 导航指令	68
2 概念解释	73
2.1 简介	73
2.1.1 功能介绍	73
2.1.2 支持的语音格式	73
2.2 集成指南	74

2.2.1 获取 Access Token	74
2.3 API请求方式基本说明	75
2.3.1 语音上传模式	75
2.3.2 下行接口定义	77
2.3.3 错误码解释	78
2.4 注意事项	78
3 简介	79
3.0.1 概念解释	79
3.0.2 功能介绍	79
3.0.3 兼容性	80
3.0.4 开发包说明	80
3.0.5 总体框图	80
3.1 集成指南	81
3.1.1 创建应用	81
3.1.2 添加BDVRClient到工程	81
3.1.3 配置AndroidManifest.xml	81
3.2 重要接口说明	82
3.2.1 语音识别器	82
创建实例	82
设置语音识别监听器	83
开始识别	83
停止录音	84
取消识别	84
销毁	85
3.2.2 语音识别监听器	85
准备就绪	86
开始说话	86

音量变化	87
获取原始语音	87
说话结束	88
识别出错	88
识别最终结果	88
识别临时结果	89
识别事件返回	89
3.3 参数和错误码说明	90
3.3.1 识别参数	90
3.3.2 离线识别支持语法说明	94
槽说明 (导航)	95
支持说法	95
槽说明 (音乐)	95
支持说法	95
槽说明 (应用)	96
支持说法	96
槽说明 (打电话/发短信)	97
支持说法	97
槽说明 (联系人)	97
支持说法	98
槽说明 (手机设置)	98
支持说法	98
槽说明 (电视指令)	98
支持说法	98
槽说明 (播放器指令)	99
支持说法	99
槽说明 (收音机指令)	99
支持说法	99

槽说明 (用户命令)	100
支持说法	100
3.3.3 错误码说明	100
通用错误码	100
详细错误信息	100
业务错误码	101
3.4 完整示例	101
3.4.1 API方式识别示例	101
3.4.2 语音识别控件示例	103
3.4.3 离线识别参数设置示例	104
注意事项	104
3.5 语音唤醒	105
3.5.1 接入唤醒功能	105
3.5.2 自定义唤醒词	105
3.5.3 接入示例	105
3.5.4 完整示例	106
3.6 其他说明	108
3.6.1 结果解析	108
3.6.2 常见问题	108
DEMO工程无法使用离线识别 ?	108
如何获取录音数据	109
4 简介	111
4.0.3 概念解释	111
4.0.4 功能介绍	111
4.0.5 兼容性	112
4.0.6 开发包说明	112
4.1 集成指南	113

4.1.1 创建应用	113
4.1.2 引入编译需要的Framework	113
4.1.3 引入BDVRClient的头文件	114
4.1.4 引入静态库文件	115
4.1.5 添加第三方开源库	115
4.1.6 引入库所需的资源文件	115
4.1.7 引入授权文件和数据文件	116
4.2 重要接口说明	117
4.2.1 语音识别控件	118
创建识别控件对象	118
启动识别	119
取消本次识别	119
调整控件坐标	120
4.2.2 语音识别控件回调对象	120
语音识别最终结果返回回调	120
录音数据返回	121
返回中间识别结果	121
发生错误	122
4.2.3 语音识别客户端	122
创建语音识别客户对象	123
释放语音识别客户端对象	123
判断是否可以录音	123
开始语音识别	124
主动说话结束	124
结束语音识别	125
获取当前识别的采样率	125
指定语音采样率	126
设置识别类型列表	126

获取当前识别类型列表	127
设置播放提示音	127
监听当前音量级别	128
获得当前音量级别	128
取消监听音量级别	128
设置识别语言	129
设置开发者申请的api key和secret key	129
验证apiKey和secretKey	130
设置参数	130
关闭标点	131
设置本地VAD开关	131
设置服务器VAD开关	132
设置是否使用压缩开关	132
设置是否使用AMR压缩开关	133
是否进行车载环境下的噪声消除	133
设置是否获取原始json结果	134
判断是否开启获取原始json结果	134
加载离线识别引擎	135
设置在线识别时等待超时时间	136
是否正在进行离线识别	136
4.2.4 语音识别客户端回调对象	137
识别状态变化回调	137
识别出错回调	137
识别网络状态变化回调	138
4.3 参数和错误码说明	138
4.3.1 语音识别控件参数	138
4.3.2 语音识别器参数	140
播放提示音	140

语音识别类型	140
采样率	141
语言	141
4.3.3 语音识别状态	141
4.3.4 语音识别错误码	142
4.4 完整示例	144
4.4.1 语音识别控件完整示例	144
4.4.2 语音识别器完整示例	146
 5 简介	149
5.0.3 概念解释	149
5.0.4 功能介绍	149
5.0.5 兼容性	150
5.0.6 开发包说明	150
5.1 集成指南	150
5.1.1 创建应用	150
5.1.2 添加BDVRClient到工程	150
5.1.3 添加权限	151
5.1.4 Proguard配置	151
5.2 重要接口说明	152
5.2.1 识别器	152
创建对象	152
设置授权	152
开始识别	153
说话结束	153
停止识别	154
释放对象	154
5.2.2 识别参数配置器	155

创建识别参数配置器	155
设置采样率	155
启用语音识别开始音效	156
禁用语音识别开始音效	156
启用语音识别结束音效	157
禁用语音识别结束音效	157
禁用标点符号	158
启用语义理解	158
启用通讯录识别	159
设置垂直领域	159
设置资源类型	160
设置是否使用缺省的录音	160
设置语种	162
5.2.3 识别状态监听器	162
更新识别状态变化	162
更新网络状态变化	163
出错	163
5.2.4 语音识别对话框	164
创建对话框	164
设置对话框监听器	165
获取识别结果	165
5.3 参数和错误码说明	166
5.3.1 识别器参数说明	166
采样率	166
语种或方言	166
语言模型垂类	166
5.3.2 开始识别返回错误码说明	167
5.3.3 出错回调返回错误码说明	167

5.3.4 语音识别对话框参数说明	168
5.4 完整示例	168
5.4.1 自调用接口方式示例	168
5.4.2 语音识别对话框模式示例	169
 6 简介	
6.0.3 概念解释	171
6.0.4 功能介绍	171
6.0.5 兼容性	172
6.0.6 开发包说明	172
6.0.7 总体框图	173
 6.1 集成指南	173
6.1.1 创建应用	173
6.1.2 引入编译需要的Framework	173
6.1.3 引入BDVRClient的头文件	174
6.1.4 引入静态库文件	175
6.1.5 添加第三方开源库	176
6.1.6 引入库所需的资源文件	176
 6.2 重要接口说明	177
6.2.1 语音识别控件	177
创建识别控件对象	177
启动识别	178
取消本次识别	179
调整控件坐标	179
6.2.2 语音识别控件回调对象	180
语音识别最终结果返回回调	180
录音数据返回	181
返回中间识别结果	181

发生错误	182
6.2.3 语音识别客户端	182
创建语音识别客户对象	182
释放语音识别客户端对象	183
判断是否可以录音	183
开始语音识别	183
主动说话结束	184
结束语音识别	184
获取当前识别的采样率	185
指定语音采样率	185
设置识别类型列表	186
获取当前识别类型列表	186
设置播放提示音	187
监听当前音量级别	187
获得当前音量级别	188
取消监听音量级别	188
设置识别语言	189
设置开发者申请的api key和secret key	189
验证apiKey和secretKey	190
设置参数	190
关闭标点	191
设置本地VAD开关	191
设置服务器VAD开关	192
设置是否使用压缩开关	192
设置是否使用AMR压缩开关	193
是否进行车载环境下的噪声消除	193
设置是否获取原始json结果	194
判断是否开启获取原始json结果	194

6.2.4 语音识别客户端回调对象	195
识别状态变化回调	195
识别出错回调	195
识别网络状态变化回调	196
6.3 参数和错误码说明	196
6.3.1 语音识别控件参数	196
播放提示音	198
语音识别类型	198
采样率	199
语言	199
6.3.3 语音识别状态	199
6.3.4 语音识别错误码	200
6.4 完整示例	201
6.4.1 语音识别控件完整示例	201
6.4.2 语音识别器完整示例	203
7 常见问题	205
8 概念解释	209
8.1 简介	209
8.1.1 功能介绍	209
8.2 集成指南	209
8.2.1 获取 Access Token	209
8.3 API请求方式基本说明	210
8.3.1 文本上传模式	211
8.3.2 GET调用方式	211
8.3.3 POST调用方式	211
8.3.4 下行接口定义	212

8.3.5 错误码解释	212
8.4 注意事项	212
9 简介	213
9.0.1 概念解释	213
9.0.2 功能介绍	213
9.0.3 兼容性	213
9.0.4 开发包说明	214
9.1 集成指南	214
9.1.1 创建应用	214
9.1.2 添加BDTTSClient到工程	214
9.1.3 添加语音合成资源文件	214
9.1.4 添加权限	215
9.1.5 Proguard配置	215
9.2 重要接口说明	216
9.2.1 合成器	216
创建对象	216
设置Context	216
设置监听器	217
初始化TTS引擎	217
获取版本信息	218
设置在线授权	218
设置离线授权	219
设置参数	219
设置播放器的音频流类型	221
暂停任务	221
恢复任务	222
停止任务	222

释放TTS资源	223
加载离线自定义模型文件	223
释放离线自定义模型文件	223
切换离线发音人接口	224
加载英文模型文件	225
合成并播放文本	225
合成文本	227
批量合成并播放	228
授权验证	229
9.2.2 合成状态监听器	229
合成开始	229
获取合成数据	230
合成完成状态	230
播放开始状态	231
播放进度状态	231
播放完成	232
出错	232
9.3 参数和错误码说明	234
9.3.1 合成器参数设置说明	234
9.3.2 错误码说明	235
9.4 完整示例	237
 10 简介	239
10.0.1 概念解释	239
10.0.2 功能介绍	239
10.0.3 兼容性	240
10.0.4 开发包说明	240
10.1 集成指南	240

10.1.1 创建应用	240
10.1.2 添加BDSSpeechSynthesizer到工程	240
10.1.3 添加授权文件	241
10.1.4 添加语音合成资源文件	241
10.1.5 引入BDSSpeechSynthesizer的头文件	241
10.1.6 引入静态库文件	241
10.2 重要接口说明	242
10.2.1 语音合成器	242
获取合成器唯一实例	242
释放合成器唯一实例	242
设置合成器代理	243
设置合成参数	243
获取合成参数	244
设置认证信息	244
设置回调队列	245
获得当前回调队列	245
设置合成线程优先级	246
启动合成引擎	246
加载英文合成数据文件及模型文件	247
重新加载文本分析数据文件或者声学模型数据文件	247
加载定制库	248
卸载定制库	248
验证音库文件的有效性	249
获取音库文件相关参数	249
批量开始文本合成但不朗读或添加文本至当前合成过程	250
批量开始文本合成并朗读或添加文本至当前合成过程	250
取消本次合成并停止朗读	251
暂停文本合成并朗读	251

继续文本合成并朗读	252
获取合成器状态	252
设置播放器音量	253
设置AudioSessionCategory类型	253
10.2.2 语音合成器委托对象	254
开始合成	254
结束合成	254
开始朗读	255
结束朗读	255
新的语音数据已经合成	256
播放进度变更	256
合成器发生错误	257
10.3 参数和错误说明	257
10.3.1 语音合成器参数	257
合成参数	257
语音合成文本语言	258
语音合成文本编码格式	258
语音合成音频编码格式	258
10.3.2 语音合成错误码	259
开始合成错误	259
合成错误	260
10.4 完整示例	261
10.4.1 语音合成完整示例	261
11 概念解释	263
11.1 简介	263
11.1.1 兼容性	263
11.1.2 开发包说明	264

11.2 集成指南	264
11.2.1 引入编译需要的 Framework	264
11.2.2 引入BDVRClient的头文件	265
11.2.3 引入静态库文件	265
11.3 语音合成	266
11.3.1 合成同时播放	266
11.3.2 参数设置	266
11.3.3 音频流类型设置	267
11.3.4 播放控制	267
11.3.5 状态监听	267
11.3.6 获取合成数据自行播放	267
11.3.7 获取合成数据	268
11.3.8 播放音频数据	268
11.3.9 播放控制	268
11.3.10 播放器状态监听	268
11.4 日志	268
12 常见问题	269

第1章

前言

百度语义理解开放平台面向互联网开发者提供对自然语言文本的解析服务。自然语言文本是用户意图的表述，意图解析的目的就是将文本解析成意图表示。

百度语义理解开放平台的特点，并计划逐步开放：

支持多意图解析

query 可能有多种意图，因此意图表示应该是多个。

容错

语音输入环境下，有错别字时能正确解析的能力非常重要。依托百度搜索行为大数据和 query 纠错技术，开放平台具备强大的纠错能力。

具备基于知识库的智能推理能力

依托百度知道等社区等产品上积累的知识构建的知识库，开发平台能够做到智能推理、“不言而明”。

具备上下文理解能力

基于上文的解析下文，具体包括省略补全和指代消解等。

1.1 意图表示协议

1.1.1 意图表示协议描述语言

为了易于人阅读，同时也方便机器解析和生成，意图表示协议采用 json 语言进行描述，采用 gb18030 编码。json 语言的基本概念：

属性名/属性值

即键值对 (key-value)

数组

在 json 中是 “[]” 括起来的内容，数据结构为 [“value1” , “value2” , …]，取值方式和所有语言中一样，使用索引获取，属性值的类型可以是数字、字符串、数组、对象。

对象

在 json 中表示为 “{ }” 括起来的内容，数据结构为 {key1:value1,key2:value2,…} 的键值对的结构。在面向对象的语言中，key 为对象的属性，value 为对应的属性值，所以很容易理解，取值方法为对象.key 获取属性值，这个属性值的类型可以是数字、字符串、数组、对象几种。

1.1.2 意图表示协议

自然语言文本（以下简称 query）。query 的意图表示由如下键值对和数组构成：

1. raw_text: 用户的原始输入文本
2. parsed_text: 经过分词，纠错，改写处理后的文本
3. version: 协议版本号
4. results: 意图表示数组

意图表示数组中存放多个意图表示对象。每个意图表示对象含有领域、意图、置信度三个键值对和一个意图对象。

领域 (domain)：同一类型的数据或资源，以及围绕数据或资源提供的服务称为一个领域。领域数据一般是结构化的表格数据，有一个主键（主属性）；领域一般以名词命名。如：列车：围绕车次数据的列车时刻查询和预订服务，主键是车次 天气：围绕天气数据的查询服务，主键是日期 电话：围绕联系人-号码数据的电话拨打、通话记录查询服务，主键是电话号码

意图 (intent)：代表用户对领域数据的操作，如查询、查询某一个属性的值、预订、拨打等。一般以动词命名。

置信度 (score)：意图表示的置信度，多个意图表示对象按置信度降序排序。

意图对象 (object)：描述意图的数据（键值对的集合），也是实现意图所需要的参数。

意图表示以 json 语言描述，json 语言中“对象”指的是键值对的集合。

语义理解示例：

query: 北京西站到无锡的高铁

```
{  
  "raw_text": "北京西站到无锡的高铁",  
  "parsed_text": "北京 西站 到 无锡 的 高铁",  
  "version": 1,  
  "results": [  
    {  
      "domain": "train",  
      "intent": "query",  
      "score": 1.0,  
      "object": {"start": "北京西站", "end": "无锡", "mode": "高铁"}  
    }  
  ]  
}
```

```

"results": [
{
  "domain": "train",
  "intent": "query",
  "score": 1,
  "object": {
    "arrival_city": "无锡",
    "start_city": "北京",
    "start_station": "北京西站",
    "train_type": "高速动车"
  }
}
]
}

```

1.1.3 领域取值

本表格详细描述了领域 (domain) 的取值。

领域名称	描述	示例 query
weather	天气	后天天气如何；明天几度
calendar日历	日历	到国庆节还有几天
train	列车	早上 8 点前发车去无锡的动车
flight	航班	飞上海最晚的国航航班
map	地图	西二旗到新街口坐几号线
telephone	电话	打电话给李四；查看有没有张三的未接来电
contact	联系人	找老爸的办公电话
message	短信	发短信给小明说下午 3 点文档评审
app	手机应用	打开相机；安装百度手机助手
website	网站	打开百度
alarm	提醒	提醒我周五晚上看中国好声音；工作日早上 8 点半叫醒我
sns	社交网络	发新浪微博，内容是我结婚了
setting	手机设置	打开 wifi

领域名称	描述	示例 query
music	音乐	听刘德华的歌曲
joke	笑话	讲一个笑话
story	故事	讲一个故事
hotel	酒店	外滩附近速八酒店，商务大床
travel	旅游	豫园开放时间、门票？
instruction	通用指令	打开，关闭
video	视频（现在只包含影视）	观看电影阿甘正传
translation	翻译	hello 的中文翻译
phone_charges	话费流量查询	我这个月的话费
tv_show	电视节目	我想看快乐大本营
person	人物	赵本山的女儿叫什么
tv_instruction	电视指令	回看刚才的节目
stock	股票	招商银行股票多少钱
novel	小说	我想看小说
player	播放器	继续播放
account	记账	昨天吃饭花了 10 块钱
search	搜索	搜索步步惊心
vehicle_instruction	车载指令	查看胎压
radio	收音机控制指令	收听 FM2.1
recipe	菜谱	西红柿炒鸡蛋怎么做
navigate_instruction	导航指令	报告超速
movie_news	影讯	最近有啥热门电影

1.1.4 意图取值

意图和领域相关，不同领域下有多种意图，我们将意图抽样为一般查询（query）和具体属性值的查询（get）两类，部分领域还有特殊的意图。详见第三节各领域意图表示。

一般查询是指用户对某一个领域的通用查询，不针对特定属性，如“北京到上海的飞机”；具体属性值的查询（下文简称属性查询）是用户针对一个领域的特定属性进行查询，如“ho1252 航班什么时候起飞”，用户查询的是航班 ho1252 的起飞时间属性。

1.1.5 属性值类型

基本数据类型，如 string、int、float 之外，还定义了时间（time）、日期（date）、折扣（discount）等属性值类型。

在以上类型定义的基础上，还定义了组合类型和范围类型，如时间日期（date_time）、时间段（time_range）、日期段（date_range）、时间日期段（date_time_range）、金额范围（amount_range）、折扣范围（discount_range）等。

日期（date）

格式是：yyyy-mm-dd，4位阿拉伯数字代表年份，2位数字代表月份，2位数字代表日期。支持多种自然语言表述的日期到上述标准格式的归一化解析。

- 一般日期或农历日期。如果没有年或月，则默认当年当月。示例：
 - 2013年1月10号：2013-01-10
 - 8月15：2013-08-15
 - 农历二月初一：2013-03-12
- 相对日期，示例：
 - 明天：2013-09-26
 - 下周一：2013-09-30
- 节日，示例：
 - 春节：2013-02-10
 - 明年春节：2014-01-31
 - 父亲节：2013-06-16

时间（time）

格式是：hh:mm:ss，24小时制，分别用2位数字代表时、分、秒。支持多种自然语言表述的时间到上述标准格式的归一化解析。示例：

- 十点整：10:00:00
- 下午三点半：15:30:00

时间日期（date_time）

时间和日期的组合类型。示例：

query：明天早上八点

```
{  
    "date": "2013-09-26",  
    "time": "08:00:00",  
}
```

日期段 (date_range)

以逗号间隔的两个日期表示日期段的开始和结束。示例：

- 明后天：2013-09-26,2013-09-27
- 下周末：2013-10-05,2013-10-06

时间段 (time_range)

以逗号间隔的两个时间表示时间段的开始和结束。如“上午”、“明天下午”等常见的时间段表述，起始和结束的时间点见表格 2-2 常见时间段的开始和结束时间点定义。示例：

- 下午：12:00:00,18:59:59
- 早上：04:00:00,11:59:59

常见时间段的开始和结束时间点定义如下表

时间范围表述	开始	结束	时间点
早晨、早上、黎明	04:00	11:00	06:00
上午	08:00	12:00	08:00
中午、晌午	10:00	14:00	12:00
下午、午后	12:00	18:00	13:00
晚上	17:00	24:00	18:00
凌晨	22:00	04:00	02:00
半夜、深夜、午夜	22:00	04:00	02:00

时间日期段 (date_time_range)

以两个时间日期类型的值分别表示时间范围和起始(begin)和结束(end)。示例：

query: 明天下午 3 点半到后天早上 8 点

```
{  
  "begin": {  
    "date": "2013-09-26",  
    "time": "15:00:00"  
  },  
  "end": {  
    "date": "2013-09-27",  
    "time": "08:00:00"  
  }  
}
```

query: 中秋节假期

```
{  
  "begin": {  
    "date": "2013-09-19",  
    "time": "00:00:00"  
  },  
  "end": {  
    "date": "2013-09-21",  
    "time": "23:59:59"  
  }  
}
```

query: 明天晚上

```
{  
  "begin": {  
    "date": "2013-09-26",  
    "time": "17:00:00"  
  },  
  "end": {  
    "date": "2013-09-26",  
    "time": "23:59:59"  
  }  
}
```

重复时间 (repeat_time)

提醒、闹钟等设置需要用到：示例：

query: 每天早上十点

```
{  
  "time": "10:00:00",  
  "repeat": "0000111111",  
}
```

注意: repeat 是 11 个字符的字符串, 从左到右分别是月份(0₁)日期(2)和星期(4~10)。对月份和日期位, 非 0 值代表该月或日期进行提醒, 对 7 个星期位, 每位对应一天, 值为 1 代表该天进行提醒。

query: 每天中午十二点整

```
{  
    "time": "12:00:00",  
    "repeat": "00001111111",  
}
```

query: 每个月 11 号 12 点

```
{  
    "time": "12:00:00",  
    "repeat": "00110000000",  
}
```

query: 每年的元旦

```
{  
    "repeat": "01010000000",  
}
```

金额范围 (amount_range)

以两个浮点数代表金额范围。示例:

```
{  
    "begin": 100.00,  
    "end": 200.00  
}
```

query: 300 元左右

```
{  
    "begin": 240.00,  
    "end": 360.00  
}
```

query: 200 元以下

```
{  
  "begin":0.00,  
  "end":200.00  
}
```

query: 300 元以上

```
{  
  "begin":300.00,  
  "end":-1  
}
```

注意: -1 代表无穷大。

折扣 (discount)

价格的折扣, 如一折、八五折、半价等, 解析成 0 到 1 之间的浮点数。示例:

- 三折:0.3
- 半价:0.5

折扣范围 (discount_range)

以两个折扣类型的值分别表示折扣范围的最小值(begin)和最大值(end)。示例:

query: 三折到八五折:

```
{  
  "begin":0.3,  
  "end":0.85,  
}
```

1.2 各领域意图表示

1.2.1 天气

天气 (weather) 能支持对天气的一般查询和温度、风速等属性的查询的解析。天气意图表示如下:

领域	weather			
意图	取值	说明	示例	
	query	一般查询	北京后天天气如何	
	get	属性查询	后天有雨吗 明天会刮风吗 明天温度几度	
意图对象	属性	类型	取值	说明
	date	date_range	yyyy-mm-dd,yyyy-mm-dd	日期范围
	region	string	北京、鄂尔多斯等	地名， 默认本地城市
	weather	string	雨，雪，雾，等	天气基本要素
	wind	string	风力	8 级大风（一期不解析）
	temp	string	温度	25 度（一期不解析）
	focus	string	date、region、weather、wind、temp	用户查询的属性字段名

示例：

query: 北京明天天气

```
{
  "raw_text": "北京明天天气",
  "parsed_text": "北京 明天 天气",
  "results": [
    {
      "domain": "weather",
      "intent": "query",
      "score": 1,
      "object": {
        "date": "2013-09-25,2013-09-25",
        "region": "北京"
      }
    }
  ]
}
```

query: 明天几度

```
{  
  "raw_text": "明天几度",  
  "parsed_text": "明天 几 度",  
  "results": [  
    {  
      "domain": "weather",  
      "intent": "get",  
      "score": 1,  
      "object": {  
        "focus": "temp",  
        "date": "2013-11-22,2013-11-22"  
      }  
    }  
  ]  
}
```

1.2.2 日历

日历 (calendar) 能够支持天数查询和对节日、农历、星座、生肖等特定内容的查询请求的解析。日历意图如下：

领域	calendar			
意图	取值	说明	示例	
	count_day	天数查询	到国庆节还有几天	
意图对象	get	属性查询	后天是农历几月几号	
			下周三是几号	
			父亲节是哪天	
			今天是什么年	
			五月十二号生日是什么星座	
			5月1日是什么节日	
意图对象	属性	类型	取值	说明
	date	date	yyyy-mm-dd	日期
	lunarday	date	yyyy-mm-dd	农历日期, 已转为阳历日期
	date_begin	date	yyyy-mm-dd	天数查询的开始日期
	date_end	date	yyyy-mm-dd	天数查询的结束日期
	zodiac	string	马, 牛, 羊	生肖名 (一期不解析)
	constellation	string	天平座	星座名 (一期不解析)
	featival	string	国庆节、中秋等	节日名称
	focus	string	一期支持: lunarday、 date、 zodiac、 constellation、 festival	用户查询的属性名: lunarday: 农历日期查询 date: 日期查询 zodiac: 生肖查询 constellation: 星座查询 festival: 节日查询

示例：

query: 今天是几号

```
{
  "raw_text": "今天是几号",
  "parsed_text": "今天 是 几 号",
  "results": [
    {
      "domain": "calendar",
      "intent": "get",
      "score": 1,
      "object": {
        "focus": "date",
        "value": "2023-09-28"
      }
    }
  ]
}
```

```
"date": "2013-11-21,2013-11-21"  
}  
}  
]  
}
```

1.2.3 列车

列车 (train) 能够支持通用列车信息的查询解析，也可以解析对坐席、列车类别、票价等特殊查询请求的解析。列车意图表示如下：

领域	train		
意图	取值	说明	示例
	query	一般查询	北京到上海的车票
	get	属性查询	北京到上海的动车多少钱 g3 几点发车 g3 几点到上海 g3 全程几个小时 g3 余票
	属性	类型	取值
	start_city	string	北京、上海等
	arrival_city	string	北京、上海等
	start_station	string	北京西、上海南等
	arrival_station	string	北京西、上海南等
	start_date	date_range	yyyy-mm-dd, yyyy-mm-dd
意图对象	arrival_date	date_range	yyyy-mm-dd, yyyy-mm-dd
	start_time	time_range	hh:mm:ss, hh:mm:ss
	arrival_time	time_range	hh:mm:ss, hh:mm:ss
	price	amount_range	数值, 数值, 数值
	ticket_type	string	硬座、硬卧上/中/下、软卧上/下、高级软卧、一/二等软座、特等座
意图对象	train_type	string	动车组、空调普快、直达特快、高速动车、临时旅客、空调快速、城际高速、空调特快等 8 种
	focus	string	一期支持： price、start_time、 arrival_time、 running_time、ticket_left
			用户查询的属性名： price:查询票价 start_time:查询发车时刻 arrival_time:查询到达时刻 running_time:查询运行时间 ticket_left:查询余票

示例：

query: 北京西站到上海的高铁

```
{
  "raw_text": "北京西站到上海的高铁",
  "parsed_text": "北京 西站 到 上海 的 高铁",
```

```
"results": [
{
  "domain": "train",
  "intent": "query",
  "score": 1,
  "object": {
    "arrival_city": "上海",
    "start_city": "北京",
    "start_station": "北京西站",
    "train_type": "高速动车"
  }
}
]
```

query: 北京西到无锡的动车几点发车

```
{
  "raw_text": "北京西到无锡的动车几点发车",
  "parsed_text": "北京 西 到 无锡 的 动车 几 点 发车",
  "results": [
{
  "domain": "train",
  "intent": "get",
  "score": 1,
  "object": {
    "focus": "starttime",
    "arrival_city": "无锡",
    "get>starttime": "几点发车",
    "start_city": "北京",
    "start_station": "北京西",
    "train_type": "动车组"
  }
}
]
```

query: 北京到上海的动车多少钱

```
{
  "raw_text": "北京到上海的动车多少钱",
  "parsed_text": "北京 到 上海 的 动车 多少 钱",
  "results": [
```

```
{  
  "domain": "train",  
  "intent": "get",  
  "score": 1,  
  "object": {  
    "focus": "price",  
    "arrival_city": "上海",  
    "start_city": "北京",  
    "train_type": "动车组"  
  }  
}  
]  
}
```

1.2.4 航班

航班 (flight) 支持基于航班类型、出发日期、航空公司、城市等的航班信息、机票信息的语义解析。航班意图表示：

领域	flight		
意图	取值	说明	示例
	query	一般查询	北京飞上海
	get	属性查询	北京到上海的机票多少钱 北京飞上海的东航航班几点起飞 北京飞上海的东航航班几点到 北京飞上海的东航航班全程几个小时
意图对象	属性	类型	取值
	start_city	string	北京、上海等
	arrival_city	string	北京、上海等
	start_station	string	首都机场等
	arrival_station	string	虹桥机场等
	start_date	date_range	yyyy-mm-dd, yyyy-mm-dd
	arrival_date	date_range	yyyy-mm-dd, yyyy-mm-dd
	start_time	time_range	hh:mm:ss, hh:mm:ss
	arrival_time	time_range	hh:mm:ss, hh:mm:ss
	price	amount_range	数值,数值,数值
	airline	string	东南航空等
	discount	discount_range	0,0.3
	ticket_type	string	经济舱、商务舱等
	flight_type	string	大型机等
focus		一期支持： price、start_time、 arrival_time、 running_time	用户查询的属性名： price:查询票价 start_time:查询起飞时刻 arrival_time:查询到达时刻 running_time:查询运行时间

示例：

query：明天从北京到上海的东航客机

```
{
  "raw_text": "明天从北京到上海的东航客机",
  "parsed_text": "明天 从 北京 到 上海 的 东航 客机",
  "results": [
    {
      "domain": "flight",
      "intent": "query",
      "score": 1,
      "parameters": [
        {
          "name": "start_time",
          "value": "2023-09-25T00:00:00"
        },
        {
          "name": "arrival_time",
          "value": "2023-09-25T12:00:00"
        },
        {
          "name": "airline",
          "value": "东航"
        }
      ]
    }
  ]
}
```

```
"object":{  
  "start_date":" 2013-09-26,2013-09-26",  
  "start_city":"北京",  
  "arrival_city":"上海",  
  "airline":"东方航空"  
}  
}  
]  
}
```

query: 明天北京飞上海东航的机票

```
{  
  "raw_text":"明天北京飞上海东航的机票",  
  "parsed_text":"明天 北京 飞 上海 东航 的 机票",  
  "results": [  
    {  
      "domain": "flight",  
      "intent": "query",  
      "score": 1,  
      "object": {  
        "start_date": "2013-09-26,2013-09-26",  
        "start_city": "北京",  
        "arrival_city": "上海",  
        "airline": "东方航空"  
      }  
    }  
  ]  
}
```

1.2.5 地图

地图 (map) 支持基于路线查询、附近信息查和地点查询的语义解析。地图意图表示：

领域	map			
意图	取值	说明	示例	
	route	路线查询	西二旗到新街口路线	
	nearby	附近查询	附近的餐厅 西二旗附近的餐厅	
	poi	地点查询	百度大厦	
意图对象	属性	类型	取值	说明
	start	string	我的位置、西二旗、北京等	路线查询的出发地，默认是“我的位置”
	arrival	string	新街口、上海等	路线查询的目的地
	route_sort	string	{ quick, few_transfer, few_walk, no_subway }	路线需求： quick: 较快捷 few_transfer: 少换乘 few_walk: 少步行 no_subway: 不坐地铁
	drive_sort	string	{less_time,less_distance,less_motorway}	驾车需求： less_time: 最短时间 less_distance: 最短距离 less_motorway: 不走高速
	route_type	string	{walk,public,drive}	出行方式： walk: 步行 public: 公共交通 drive: 驾车； taxi: 打车
	centre	string	西二旗、西单等	附近查询的中心点
	keywords	string	餐厅、停车场、poi 点等	附近查询的关键词，多个关键词以#间隔
	lbs_tag	string	{wifi, 停车, 优惠,...}	附近或 poi 查询的需求

示例：

query：从西二旗到天安门不走高速

```
{
  "raw_text": "从西二旗到天安门不走高速",
  "parsed_text": "从 西二旗 到 天安门 不走 高速",
  "results": [
    {
      "name": "附近查询"
    }
  ]
}
```

```
"domain": "map",
"intent": "route",
"score": 1,
"object": {
  "start": "西二旗",
  "arrival": "天安门",
  "drive_sort": "less_motorway",
  "route_type": "drive"
}
}
]
```

query: 西二旗附近的餐厅

```
{
  "raw_text": "西二旗附近的餐厅",
  "parsed_text": "西二旗 附近 的 餐厅",
  "results": [
    {
      "domain": "map",
      "intent": "nearby",
      "score": 1,
      "object": {
        "centre": "西二旗",
        "keywords": "餐厅"
      }
    }
  ]
}
```

1.2.6 电话

电话 (telephone) 提供常用的拨打电话、通话记录查看等操作的语义解析。电话意图表示：

领域	telephone			
意图	取值	说明	示例	
	call	拨打电话	打电话给张三 打电话给 13644444444 打电话给联通客服	
	view	查看未接电话	查看未接来电/已接来电/通话记录	
意图对象	属性	类型	取值	说明
	name	string	人名或者公司名	表示联系人的名字，可以是通讯录中的条目名，也可以是其他人名或者公司名
	number	string	由"+,-,(,)"和数值组成的电话号码或者电话号码的一部分	表示要拨打的电话号码，或者号码的一部分
	type	string	home:固定/住宅, office:办公室, mobile:手机	表示电话类型
	head	string	136,139,159...	号段
	tail	string	任意数字串	号尾
	location	string		号码归属地
	operator	string	移动, 联通, 电信...	运营商

示例：

query：打电话给妈妈北京移动的号码

```
{
  "raw_text": "打电话给妈妈北京移动的号码",
  "parsed_text": "打电话 给 妈妈 北京 移动 的 号码",
  "results": [
    {
      "domain": "telephone",
      "intent": "call",
      "score": 1,
      "object": {
        "name": "妈妈",
        "operator": "移动",
        "location": "北京"
      }
    }
  ]
}
```

query: 显示没接听的电话

```
{  
  "raw_text": "显示没接听的电话",  
  "parsed_text": "显示 没 接听 的 电话",  
  "results": [  
    {  
      "domain": "telephone",  
      "intent": "view",  
      "score": 1,  
      "object": {  
        "call_type": "view_missed"  
      }  
    }  
  ]  
}
```

1.2.7 通讯录

通讯录 (contacts) 提供常用的通讯录操作如新建、查看等操作的语义解析支持。通讯录意图表示：

领域	contacts			
意图	取值	说明	示例	
	view	查询通讯录	查看通讯录	
	create	新建联系人	新建联系人	
	modify	编辑联系人	编辑联系人	
	remove	删除联系人	删除联系人	
意图对象	属性	类型	取值	说明
	name	string	人名或者公司名	表示联系人的名字，可以是通讯录中的条目名，也可以是其他人名或者公司名
	number	string	由"+,-,()"和数值组成的电话号码或者电话号码的一部分	表示要拨打的电话号码，或者号码的一部分；若 name 对应多个电话号码，则以#间隔。
	type	string	home:固定/住宅, office:办公室,mobile:手机	表示电话类型
	head	string	136,139,159...	号段
	tail	string	任意数字串	号尾
	location	string		号码归属地
	operator	string	移动, 联通, 电信...	运营商

示例：

query: 查一下玉洁的手机号码

```
{
  "raw_text": "查一下玉洁的手机号码",
  "parsed_text": "查一下玉洁的手机号码",
  "results": [
    {
      "domain": "contacts",
      "intent": "view",
      "score": 1,
      "object": {
        "name": "玉洁",
        "type": "mobile"
      }
    }
  ]
}
```

query: 新建联系人姓名玉洁号码13641096029

```
{  
  "raw_text": "新建联系人姓名玉洁号码 13641096029",  
  "parsed_text": "新建 联系人 姓名 玉洁 号码 13641096029",  
  "results": [  
    {  
      "domain": "contacts",  
      "intent": "create",  
      "score": 1,  
      "object": {  
        "name": "玉洁",  
        "number": "13641096029"  
      }  
    }  
  ]  
}
```

1.2.8 短信

短信 (message) 提供常用的短信操作的语义解析支持。短信意图表示：

领域	message			
意图	取值	说明	示例	
	view	查看短信	查看未读短信	
	send	发送短信	给海雷发一条春节祝福短信	
	read	阅读短信	阅读未读短信	
意图对象	属性	类型	取值	说明
	name	string	人名或者公司名	表示联系人的名字，可以是通讯录中的条目名，也可以是其他人名或者公司名，为了支持群发，所以是列表
	number	string	由 "+,-,(,)" 和数值组成的电话号码或者电话号码的一部分	表示要拨打的电话号码，或者号码的一部分；若 name 对应多个电话号码，则以#间隔，为了支持群发，所以是列表
	phone_type	string	home:固定/住宅, office:办公室,mobile:手机	表示电话类型，默认 mobile
	sms_type	string	unread/未读,readed/已读,sent/已发送,unsent/未发送, recieived/全部接收	表示短信类型
	head	string	136,139,159...	号段
	tail	string	任意数字串	号尾
	location	string		号码归属地
	operator	string	移动, 联通, 电信...	运营商
	msgbody	string	短信内容	用户说的发送内容
	wishes	list	多条祝福短信	当用户想发送祝福短信时，给他多条候选供其选择

注意：祝福短信(wishes)的内容可扩展，现在主要有：七夕,万圣节,中秋节,乔迁,健康,儿童节,元宵节,元旦,劳动节,升职,周末,国庆节,圣诞节,复活节,女生节,妇女节,寒假,小年,平安夜,幽默,开业,开学,思念,思恋,情人节,情话,愚人节,感恩节,感谢,护士节,教师节,早安,春节,晚安,暑假,母亲节,毕业,求婚,清明节,爱情,父亲节,生子,生日,生活,生病,端午节,结婚,致谢,表白,记者节,送别,道歉,重阳节,除夕,青年节,高考。

示例：

query: 给玉洁发短信说我下周请你吃饭

```
{  
  "raw_text": "给玉洁发短信说我下周请你吃饭",  
  "parsed_text": "给 玉洁 发短信 说 我 下周 请 你 吃饭",  
  "results": [  
    {  
      "domain": "message",  
      "intent": "send",  
      "score": 1.0,  
      "object": {  
        "name": [  
          "玉洁"  
        ],  
        "msgbody": "我下周请你吃饭"  
      }  
    }  
  ]  
}
```

query: 给李明发一条春节祝福短信

```
{  
  "raw_text": "给李明发一条春节祝福短信",  
  "parsed_text": "给 李明 发 一 条 春节 祝福 短信",  
  "results": [  
    {  
      "domain": "message",  
      "intent": "send",  
      "score": 1.0,  
      "object": {  
        "name": [  
          "李明"  
        ],  
        "msgbody": "合家团圆",  
        "wishes": [  
          "春节好",  
          "春节快乐",  
          "新春快乐"  
        ]  
      }  
    }  
  ]  
}
```

```
}
```

query: 查看未读短信

```
{
  "raw_text": "查看未读短信",
  "parsed_text": "查看 未 读 短信",
  "results": [
    {
      "domain": "message",
      "intent": "view",
      "score": 1.0,
      "object": {
        "sms_type": "unread"
      }
    }
  ]
}
```

1.2.9 手机应用

手机应用 (app) 用于对手机应用的搜索、下载、安装、打开、分享等操作的语义解析支持。app 意图表示:

领域	app		
意图	取值	说明	示例
	open	打开 app	打开百度语音助手
	uninstall	卸载 app	卸载百度语音助手
	update	更新 app	更新百度语音助手
	download	下载 app	下载百度语音助手
	search	搜索 app	搜索百度语音助手
	recommend	推荐 app	推荐百度语音助手
意图对象	属性	类型	取值
	appname	string	任意文本
	category	string	游戏类型
	keyword	string	任意文本
	channel	string	{baidu,google,sina,sohu,qq,91,wandoujia,yingyonghui}
			下载或者查询的渠道: baidu:百度 google:谷歌 sina:新浪 sohu:搜狐 qq:腾讯 91:91移动 wandoujia:豌豆荚 yingyonghui:应用汇

示例：

query: 打开百度语音助手

```
{
  "raw_text": "打开百度语音助手",
  "parsed_text": "打开 百度 语音 助手",
  "results": [
    {
      "domain": "app",
      "intent": "open",
      "score": 1.0,
      "object": {
        "appname": "百度语音助手"
      }
    }
  ]
}
```

query: 通过 91 帮我下一款益智类的游戏

```
{
  "raw_text": "通过 91 帮我下一款益智类的游戏",
  "parsed_text": "通过 91 帮 我 下 一 款 益 智 类 的 游 戏",
  "results": [
    {
      "domain": "app",
      "intent": "download",
      "score": 1,
      "object": {
        "channel": "91",
        "category": [
          "休闲益智"
        ]
      }
    }
  ]
}
```

1.2.10 网站

网站 (website) 用于支持基于网站名或 URL 网址的打开、搜索和查询的语义解析。网站意图表示:

领域	website			
意图	取值	说明	示例	
	open	打开网站	打开百度首页	
意图对象	属性	类型	取值	说明
	sitename	string	任意文本	网站名
	url	string	任意文本	网站的 url
	browser	string	{ie; uc; dolphin; 360; baidu; chrome; qq; maxthon; sogou; firefox; opera;safari}	打开所要使用的浏览器: ie:ie 浏览器 uc:uc 浏览器 dolphin:海豚浏览器 360:360 浏览器 baidu:百度浏览器 chrome:google chrome 浏览器 qq:qq 浏览器 maxthon:傲游浏览器 sogou:搜狗浏览器 firefox:火狐浏览器 opera:欧朋浏览器 safari:safari 浏览器

示例：

query: 用 ie 打开百度首页

```
{
  "raw_text": "用 ie 打开百度首页",
  "parsed_text": "用 ie 打开 百度 首页",
  "results": [
    {
      "domain": "website",
      "intent": "open",
      "score": 1.0,
      "object": {
        "sitename": "百度",
        "browser": "ie"
      }
    }
  ]
}
```

1.2.11 日程提醒

日程提醒 (alarm) 提供新建、查看和删除提醒等操作的语义解析。日程提醒意图表示:

领域	alarm			
意图	取值	说明	示例	
	view	查看日程	查看明天的日程	
	insert	新建日程	建立明天日程	
	remove	删除日程	删除昨天日程	
意图对象	属性	类型	取值	说明
	type	string	{absolute, relative, repeat}	某一个时间点的提醒
	time	time		提醒时间
	date	date		提醒日期
	interval	int		相对当前时间的秒数
	range	time_range		时间范围
	event	string	任意字符串	提醒事件

示例：

query: 提醒我早上 8 点准时起床去拜年

```
{  
  "raw_text": "提醒我早上 8 点准时起床去拜年",  
  "parsed_text": "提醒 我 早上 8 点 准时 起床 去 拜年",  
  "results": [  
    {  
      "domain": "alarm",  
      "intent": "insert",  
      "score": 1,  
      "object": {  
        "type": "absolute",  
        "date": "2013-09-04",  
        "time": "08:00:00",  
        "event": "准时起床去拜年"  
      }  
    }  
  ]  
}
```

query: 每天 10 点提醒我开会

```
{  
  "raw_text": "每天 10 点提醒我开会",  
  "parsed_text": "每天 10 点 提醒 我 开会",  
  "results": [  
    {  
      "domain": "alarm",  
      "intent": "insert",  
      "score": 1,  
      "object": {  
        "type": "repeat",  
        "time": "10:00:00",  
        "repeat": "0000111111",  
        "event": "开会"  
      }  
    }  
  ]  
}
```

query: 1 个小时之后提醒我吃饭

```
{  
  "raw_text": "1 个小时之后提醒我吃饭",  
  "parsed_text": "1 个 小时 之后 提醒 我 吃饭",  
  "results": [  
    {  
      "domain": "alarm",  
      "intent": "insert",  
      "score": 1,  
      "object": {  
        "type": "absolute",  
        "date": "2013-09-04",  
        "time": "11:00:00",  
        "event": "吃饭"  
      }  
    }  
  ]  
}
```

```

"results": [
{
  "domain": "alarm",
  "intent": "insert",
  "score": 1,
  "object": {
    "type": "relative",
    "interval": 3600,
    "event": "吃饭"
  }
}
]
}

```

1.2.12 社交网络

社交网络 (sns) 用于支持基于常用社交网络如微博、人人等的消息发送、查看和分享等操作的语义解析。社交网络意图表示:

领域	sns			
意图	取值	说明	示例	
	send	发送	发送一条微博	
	view	查看	查看微博	
	share	分享	分享一条微博	
	refresh	刷新	刷新微博	
意图对象	属性	类型	取值	说明
	channel	list	{sina; qq; sohu; renren; kaixin; weixin}	操作的微博名列表 sina:新浪微博 qq:腾讯微博 sohu:搜狐微博 renren:人人网 kaixin:开心网 weixin:微信朋友圈 默认是 sina
	content	string	任意字符串	微博内容
	at	list		@的昵称列表

示例:

query: 发送一条新浪微博埃特晓明就说今天天气不错

```
{  
  "raw_text": "发送一条新浪微博埃特晓明就说今天天气不错",  
  "parsed_text": "发送 一 条 新浪 微博 埃特 晓明 就 说 今 天 天 气 不 错",  
  "results": [  
    {  
      "domain": "sns",  
      "intent": "send",  
      "score": 0.6,  
      "object": {  
        "channel": [  
          "sina"  
        ],  
        "at": [  
          "晓明"  
        ],  
        "content": "今天天气不错"  
      }  
    }  
  ]  
}
```

query: 刷新腾讯微博

```
{  
  "raw_text": "刷新腾讯微博",  
  "parsed_text": "刷新 腾讯 微博",  
  "results": [  
    {  
      "domain": "sns",  
      "intent": "refresh",  
      "score": 1,  
      "object": {  
        "channel": [  
          "qq"  
        ]  
      }  
    }  
  ]  
}
```

1.2.13 手机设置

手机设置 (setting) 提供对手机各项设置的语义解析。手机设置意图表示:

领域	setting			
意图	取值	说明	示例	
	set	设置	手机声音设置	
属性	类型	取值	说明	
意图对象	settingtype	string	{sound_setting; screen_rotation_on; screen_rotation_off; display_setting; security_setting; date_setting; ring_shake; ring_silent; ring_normal; wifi_on; wifi_off; wifi_setting; gps_on; gps_off; data_on; data_off bluetooth_on; bluetooth_off; flight_on; flight_off; bluetooth_setting;}	定义的手机操作类型 sound_setting:进入系统声音设置 screen_rotation_on:打开旋转 screen_rotation_off:关闭旋转 display_setting:进入系统显示设置界面 security_setting:进入安全设置 date_setting:进入时间日期设置 ring_shake:设置手机为震动模式 ring_silent:设置手机为静音模式 ring_normal:设置手机振铃为正常模式 wifi_on:打开 wifi wifi_off:关闭 wifi wifi_setting:进入 wifi 设置 gps_on:打开 gps gps_off:关闭 gps data_on:打开数据连接 data_off:关闭数据连接 bluetooth_on:打开蓝牙 bluetooth_off:关闭蓝牙 flight_on:打开飞行模式 flight_off:关闭飞行模式 bluetooth_setting:进入蓝牙设置 lock_screen:关闭屏幕 shortcut:快捷方式 network_on:移动网络关 network_off:移动网络开 autolock_on 自动锁屏开 autolock_off:自动锁屏关 ync_account_on 同步账户关

				sync_account_off:同步账户开 usb_mode_on:usb 模式关 usb_mode_off:usb 模式关 touch_reflect_on::触感反应开 touch_reflect_off:触感反应 timeout_15/timeout_30/...:屏幕超时 media_voice_setting:媒体音量 warn_voice_setting:警告音量 date_setting:设时间 sound_setting:设铃声 add_setting:添加设置 typeface:换字体 theme:换主题 wallpaper:换壁纸 screen_effect:换滑屏特效 desktop_setting:打开桌面设置 system_setting:打开系统设置 ind_center:打开个人中心 restart_desktop:重启桌面 desktop_off:退出桌面 app_list_setting:应用列表设置 theme:更换桌面主题
--	--	--	--	--

示例：

query: 打开 gps

```
{  
  "raw_text": "打开 gps",  
  "parsed_text": "打开 gps",  
  "results": [  
    {  
      "domain": "setting",  
      "intent": "set",  
      "score": 1.0,  
      "object": {  
        "settingtype": "gps_on"  
      }  
    }  
  ]  
}
```

query: 进入时间日期设置

```
{  
  "raw_text": "进入时间日期设置",  
  "parsed_text": "进入 时间 日期 设置",  
  "results": [  
    {  
      "domain": "setting",  
      "intent": "set",  
      "score": 1,  
      "object": {  
        "settingtype": "date_setting"  
      }  
    }  
  ]  
}
```

1.2.14 音乐

音乐 (music) 提供对音乐播放控制、搜索和下载等的语义解析支持。音乐意图表示：

领域	music			
意图	取值	说明	示例	
	play	播放	播放刘德华的冰雨	
	search	搜索	搜索刘德华的冰雨	
	recommend	推荐	推荐首歌	
	download	下载	下载刘德华的冰雨	
意图对象	属性	类型	取值	说明
	name	string	任意字符串	歌曲名
	byartist	list	数组	歌手名
	album	string	任意字符串	专辑名
	genre	string	任意字符串	歌曲类型
	tag	list	数组	歌曲 tag
	country	string	任意字符串	地域
	audience	string	任意字符串	人群
	mood	string	任意字符串	心情
	movie	string	任意字符串	电影名
	tv	string	任意字符串	电视剧名
	program	string	任意字符串	节目名称
	inlanguage	string	任意字符串	语言
	opera	string	任意字符串	戏曲
	instrument	string	任意字符串	乐器

示例：

query：播放刘德华的歌曲冰雨

```
{
  "raw_text": "播放刘德华的歌曲冰雨",
  "parsed_text": "播放 刘德华 的 歌曲 冰雨",
  "results": [
    {
      "domain": "music",
      "intent": "play",
      "score": 1,
      "object": {
        "name": "冰雨",
        "byartist": [
          "刘德华"
        ]
      }
    }
  ]
}
```

```
]  
}
```

query: 天龙八部的片尾曲

```
{  
  "raw_text": "天龙八部的片尾曲",  
  "parsed_text": "天龙八部 的 片尾曲",  
  "results": [  
    {  
      "domain": "music",  
      "intent": "search",  
      "score": 1,  
      "object": {  
        "tv": "天龙八部",  
        "genre": "片尾曲"  
      }  
    }  
  ]  
}
```

query: 适合 80 后听的伤感英文歌曲

```
{  
  "raw_text": "适合 80 后听的伤感英文歌曲",  
  "parsed_text": "适合 80 后 听 的 伤感 英文 歌曲",  
  "results": [  
    {  
      "domain": "music",  
      "intent": "search",  
      "score": 1,  
      "object": {  
        "audience": "80 后",  
        "inlanguage": "英语",  
        "tag": [  
          "伤感"  
        ]  
      }  
    }  
  ]  
}
```

1.2.15 笑话

笑话 (joke) 提供对笑话的搜索和播放控制的语义解析支持。笑话意图表示：

领域	joke		
意图	取值	说明	示例
	play	播放	讲个笑话
	search	搜索	搜个笑话
意图对象	属性	类型	取值
	type	string	任意字符串
	tag	list	数组
	sence	string	任意字符串
	crowd	string	任意字符串
	说明		

示例：

query：适合约会时候讲的有趣笑话

```
{
  "raw_text": "适合约会时候讲的有趣笑话",
  "parsed_text": "适合 约会 时候 讲 的 有趣 笑话",
  "results": [
    {
      "domain": "joke",
      "intent": "search",
      "score": 0.916667,
      "object": {
        "tag": [
          "有趣"
        ],
        "sence": "约会"
      }
    }
  ]
}
```

query：讲一个睡觉前儿童听的笑话

```
{
  "raw_text": "讲一个睡觉前儿童听的笑话",
  "parsed_text": "讲 一 个 睡觉 前 儿童 听 的 笑话",
  "results": [
    {
      "domain": "joke",
      "intent": "search",
      "score": 0.916667,
      "object": {
        "tag": [
          "睡前"
        ],
        "sence": "儿童"
      }
    }
  ]
}
```

```
{
  "domain": "joke",
  "intent": "play",
  "score": 0.916667,
  "object": {
    "sence": "睡前",
    "audience": "儿童"
  }
}
]
```

1.2.16 故事

故事 (story) 提供对故事的搜索和播放控制的语义解析支持。故事意图表示：

领域	story			
意图	取值	说明	示例	
	play	播放	讲个故事	
	search	搜索	搜索故事	
意图对象	属性	类型	取值	说明
	type	string	任意字符串	故事类型
	tag	list	数组	故事 tag
	sence	string	任意字符串	故事适用场景
	crowd	string	任意字符串	故事适用人群
	title	string	任意字符串	故事名称

示例：

query：讲一个睡觉前儿童听的温馨的小故事

```
{
  "raw_text": "讲一个睡觉前儿童听的温馨的小故事",
  "parsed_text": "讲 一 个 睡觉 前 儿童 听 的 温馨 的 小 故事",
  "results": [
    {
      "domain": "story",
      "intent": "play",
      "score": 0.9375,
      "object": {
        "sence": "睡前",
        "audience": "儿童",
      }
    }
  ]
}
```

```
"tag": [  
    "温馨"  
]  
}  
}  
}  
]  
}
```

1.2.17 酒店

酒店（hotel）提供对酒店查询的语义解析支持，支持房间类型、酒店星级、所在城市、设施和服务等的细节解析。酒店意图表示：

领域	hotel			
意图	取值	说明	示例	
	query	一般查询	西二旗附近的酒店 速八酒店有什么房间 速八酒店商务大床房价格	
	属性	类型	取值	说明
意图对象	name	string	如家广安门店	酒店名称
	city	string	北京、上海等	酒店所在地城市
	district	string	海淀等	酒店所在的区县
	keywords	string	外滩、燕莎等	酒店所在的商业区
	brand	string	速八、7天等	酒店品牌
	facility	string	wifi、接机服务等	酒店设施、服务
	checkin_date	date_range	yyyy-mm-dd,yyyy-mm-dd	入住日期范围
	checkout_date	date_range	yyyy-mm-dd,yyyy-mm-dd	离店日期范围
	price	amount_range	数值,数值,数值	金额范围
	days	int	{1,2,3,⋯}	入住天数
	room_type	string	单间、标间、总统套房等	房间类型
	star	int	{1,2,3,4,5}	酒店星级: 1:一星级、经济型、快捷酒店 2:二星级 3:三星级、舒适型 4:四星级、高档 5:五星级、豪华
	focus	string	price、room_type	用户关注点: price: 查询酒店房间价格 room_type: 查询酒店房间类型

示例：

query: 北京五星级酒店

```
{
  "raw_text": "北京五星级酒店",
  "parsed_text": "北京 五星级 大酒店",
  "results": [
    {
      "domain": "hotel",
      "intent": "query",
      "query": "北京五星级酒店"
    }
  ]
}
```

```

    "score":1,
    "object":{
      "city":"北京",
      "star":"5"
    }
  }
]
```

1.2.18 旅游

旅游 (travel) 支持对旅游景点的相关信息 (如门票价格、开放时间等) 查询的语义解析。
旅游意图表示：

领域	travel			
意图	取值	说明	示例	
	query	查询	北京好玩的地方 豫园门票多少钱 故宫开放时间	
意图对象	属性	类型	取值	说明
	destination	string	北京、上海等	旅游目的地
	view_spot	string	豫园、故宫等	景点名称
	topic	string	博物馆、宗教、家庭、蜜月等旅游主题词	旅游主题词
	type	string	古镇、名山、古村	旅游类型词
	date	date_range	yyyy-mm-dd,yyyy-mm-dd	旅游日期范围
	focus	string	price、open_time	用户关注点： price:查询门票价格 open_time:查询开放时间

示例：

query: 北京有什么博物馆

```
{
  "raw_text": "北京有什么博物馆",
  "parsed_text": "北京 有 什 么 博物馆",
  "results": [
    {
      "domain": "travel",
      "intent": "query",
      "score": 1
    }
  ]
}
```

```
"score":1,  
"object":{  
"destination":"北京",  
"topic":"博物馆"  
}  
}  
]  
}
```

query: 故宫门票价格

```
{  
"raw_text": "故宫门票价格",  
"parsed_text": "故宫 门票 价格",  
"results": [  
{  
"domain": "travel",  
"intent": "get",  
"score": 1,  
"object": {  
"focus": "price",  
"view_spot": "故宫"  
}  
}  
]  
}
```

1.2.19 通用指令

通用指令 (instruction) 提供对各场景中通用命令的语义解析支持。通用指令意图表示：

领域	instruction		
意图	取值	说明	示例
	next	下一个	
	previous	上一个	
	pause	暂停	
	stop	停止	
	close	关闭	
	open	打开	
	quit	退出	
	back_home	返回首页	
	return	返回	
	yes	是/确认	是、确认、好、对、行、 okey 、 ok 、好吧、是的、可以
	no	否/取消	否、取消、算了、不是、不好、不对、不要、不行、 no 、 cancel
	select	选择第 x 个	选第二个，倒数第一个，开头那个，最后那个
	refresh	刷新	刷新
	clear	清空	清空、清除
意图对象	view	查看 x , 见字段 item	查看电量、还剩多少电、剩余电量、电池电量
	volume_up	调高音量	大声点、大点声、
	volume_down	降低音量	小声点、调低声音
	mute	静音	静音、别说话、别出声、音量调到最低
意图对象	字段	类型	取值
	item		{电量}
	option	int	{1,2,3,...} {-1,-2,-3,...}
			查看指令的项目，目前支持的取值：{电量}
			选择指令的选项编号,负数表示倒数的第 x 个

示例：

query: 代开

```
{
  "raw_text": "代开",
  "parsed_text": "代 开",
  "results": [
    {
      "domain": "instruction",
      "intent": "open",
      "score": 1,
      "object": {}
    }
  ]
}
```

query: 还剩多少电量

```
{
  "raw_text": "还剩多少电量",
  "parsed_text": "还剩 多少 电量",
  "results": [
    {
      "domain": "instruction",
      "intent": "view",
      "score": 1,
      "object": {
        "item": "电量"
      }
    }
  ]
}
```

1.2.20 翻译

翻译 (translation) 提供句子翻译需求的语义解析支持。翻译意图表示：

领域	translation			
意图	取值		说明	示例
	translate		翻译	翻译今天的天气如何
意图对象	字段	类型	取值	说明
	source	string	原始语言	1:中文
	target	string	目标语言	0:英文
	transbody	string	翻译内容	
	tool	string	翻译工具	百度翻译

注意：

- 为了统一表示语言，分别用数字代号表示语言，语言的代号见附录1。
- 支持所有语言方向之间，但是系统只探测中英日三种翻译字符串语言。

示例：

query: 成王败寇的泰语翻译

```
{
  "raw_text": "成王败寇的泰语翻译",
  "parsed_text": "成王败寇 的 泰语翻译",
```

```
"results": [
  {
    "domain": "translation",
    "intent": "translate",
    "score": 0.555556,
    "object": {
      "transbody": "成王败寇",
      "source": "1",
      "target": "6"
    }
  }
]
```

query: 成王败寇什么意思

```
{
  "raw_text": "成王败寇什么意思",
  "parsed_text": "成王败寇 什么 意思",
  "results": [
    {
      "domain": "translation",
      "intent": "translate",
      "score": 0.5,
      "object": {
        "transbody": "成王败寇",
        "source": "1"
      }
    }
  ]
}
```

1.2.21 话费流量查询

话费流量 (phone_charges) 提供手机流量、话费、账户余额、欠费查询的语义解析支持。
话费流量查询意图表示:

领域	phone_charges			
意图	取值		说明	示例
	search		查看	
意图对象	字段	类型	取值	说明
	number	string	查询的号码	默认本机("0")
	name	array	韩愈姐	人名或公司名
	date	DATE_RANGE	YYYY-MM-DD, YYYY-MM-DD	时间
	subject	string	flow、bill、balance、arrears	流量、话费、账户余额、欠费
	sperator	string	10086、10010等	运营商
	item	string	left,spend	剩余量，花费量

示例：

query: 我还剩多少话费

```
{
  "raw_text": "我还剩多少话费",
  "parsed_text": "我 还剩 多少 话费",
  "results": [
    {
      "domain": "phone_charges",
      "intent": "search",
      "score": 1,
      "object": {
        "phonenumber": "0",
        "item": "left",
        "subject": "charge"
      }
    }
  ]
}
```

query: 13641096027 这个月用了多少流量

```
{
  "raw_text": "13641096027 这个月用了多少流量",
  "parsed_text": "13641096027 这个 月 用 了 多 少 流 量",
  "results": [
    {
      "domain": "phone_charges",
      "intent": "search",
      "score": 1,
      "object": {
        "phonenumber": "13641096027",
        "item": "left",
        "subject": "charge"
      }
    }
  ]
}
```

```

    "intent": "search",
    "score": 1,
    "object": {
        "phononenumber": "13641096027",
        "date": "2015-01-01,2015-01-31",
        "item": "spend",
        "subject": "flow"
    }
}
]
}
}

```

1.2.22 视频

视频 (video) 提供视频下载、推荐、在线播放、查询等方面的语义解析。视频意图表示

领域	video		
意图	取值	说明	示例
	search	视频查询	电影星际迷航
	recommend	推荐视频	推荐个搞笑视频
	download	下载视频	下载星际迷航
	play	播放视频	播放电影星际迷航
意图对象	字段	类型	取值
	name	string	钢铁侠...
	type	array	恐怖, 剧情...
	date	date	2013 年
	keywords	array	好看, 冷血...
	sence	string	和家人一起看的...
	crowd	string	宝宝、老人...
	channel	string	爱奇艺、百度影音...
	actor	array	刘德华, 刘诗诗...
	director	string	张艺谋...
	country	string	中国, 美国...
	season	string	1,2,3,4,5...
	episode	string	1,2,3,4,5...
	datepublish	DATE_RANGE	YYYY-MM-DD,YYYY-MM-DD
	inlanguage	string	中文, 英语...
	sub_type	string	movie,tv
	focus	string	inlanguage, datepublish,actor, director...

注意：目前只支持影视。

示例：

query: 2013 年好看的恐怖电影

```
{  
  "raw_text": "2013 年好看的恐怖电影",  
  "parsed_text": "2013 年 好看 的 恐怖 电影",  
  "results": [  
    {  
      "domain": "video",  
      "intent": "search",  
      "score": 1,  
      "object": {  
        "datepublish": "2013-01-01,2013-12-31",  
        "sub_type": "movie",  
        "keywords": [  
          "好看"  
        ],  
        "type": [  
          "恐怖"  
        ]  
      }  
    }  
  ]  
}
```

query: 我想看韩剧继承者们的最后一集

```
{  
  "raw_text": "我想看韩剧继承者们的最后一集",  
  "parsed_text": "我想看 韩剧 继承者 们 的 最后 一 集",  
  "results": [  
    {  
      "domain": "video",  
      "intent": "play",  
      "score": 1,  
      "object": {  
        "country": "韩国",  
        "sub_type": "serial",  
        "name": "继承者们",  
        "episode": "-1"  
      }  
    }  
  ]  
}
```

```

    }
}
]
}

```

1.2.23 电视节目

电视节目 (tv_show) 提供电视节目查询、下载、在线观看、播出时间查询等方面的语义解析。电视节目意图表示：

领域	tv_show		
意图	取值	说明	示例
	get	电视节目查询	快乐大本营什么时候播出
	search	推荐电视节目	推荐娱乐类电视节目
	download	下载电视节目	下载本期快乐大本营
	play	播放电视节目	观看快乐大本营
意图对象	字段	类型	取值
	name	string	快乐大本营...
	type	string	娱乐、新闻等常用类型表述
	sence	string	约会的时候...
	crowd	string	宝宝、老人...
	station	string	北京卫视...
	datepublish	DATE_RANGE	YYYY-MM-DD,YYYY-MM-DD
	period	string	1,2,3...
	focus	string	datepublish...

示例：

query: 观看上一期的快乐大本营

```

{
  "raw_text": "观看上一期的快乐大本营",
  "parsed_text": "观看 上 一 期 的 快乐大本营",
  "results": [
    {
      "domain": "tv_show",
      "intent": "play",
      "score": 1,
      "object": {
        "period": "-1",

```

```

    "name": "快乐大本营"
}
}
]
}
}
```

query: 看一看昨天的新闻联播

```
{
  "raw_text": "看一看昨天的新闻联播",
  "parsed_text": "看一 看 昨天 的 新闻 联播",
  "results": [
    {
      "domain": "tv_show",
      "intent": "play",
      "score": 1,
      "object": {
        "datepublish": "2014-01-09,2014-01-09",
        "name": "新闻联播"
      }
    }
  ]
}
```

1.2.24 人物

人物 (person) 提供人物身高、体重、年龄的方面的语义解析支持。人物意图表示：

领域	person		
意图	取值	说明	示例
	get	人物属性查询	刘德华多大了
	search	人物查询	歌手刘德华
意图对象	字段	类型	取值
	name	string	刘德华
	title	string	演员
	focus	string	birthday, birthplace, sex...

示例：

query: 歌手刘德华身高是多少

```
{
  "raw_text": "歌手刘德华身高是多少",
  "parsed_text": "歌手 刘德华 身高 是 多少",
  "results": [
    {
      "domain": "person",
      "intent": "get",
      "score": 1,
      "object": {
        "title": "歌手",
        "person": "刘德华",
        "focus": "height"
      }
    }
  ]
}
```

1.2.25 电视指令

电视指令 (tv_instruction) 提供电视调台、屏幕调节、截屏等电视指令的语义解析。电视指令意图表示：

领域	tv_instruction		
意图	取值	说明	示例
	change_channel	调频道	调台到 15 频道
	set	设置	自动搜台
	change_station	调台	我要看湖南卫视
	look_back	回看	返回刚才的节目
	get	属性查询	中央台接下来演什么
意图对象	字段	类型	取值
	channel	string	1、2、3、+1、-1
	station	string	北京卫视
	settingtype	string	channel_search
	value	string	1,2...
			说明
			频道值
			电视台
			指令集
			settingtype 的参数

示例：

query: 调台到 15 频道

```
{
  "raw_text": "调台到 15 频道",
  "parsed_text": "调 台 到 15 频道",
  "results": [
```

```
{  
  "domain": "tv_instruction",  
  "intent": "change_channel",  
  "score": 1,  
  "object": {  
    "channel": "15"  
  }  
}  
}
```

query: 返回刚才的节目

```
{  
  "raw_text": "返回刚才的节目",  
  "parsed_text": "返回 刚才 的 节目",  
  "results": [  
    {  
      "domain": "tv_instruction",  
      "intent": "look_back",  
      "score": 1,  
      "object": {}  
    }  
  ]  
}
```

query: 中央台接下来演什么

```
{  
  "raw_text": "中央台接下来演什么",  
  "parsed_text": "中央台 接下来 演 什么",  
  "results": [  
    {  
      "domain": "tv_instruction",  
      "intent": "get",  
      "score": 1,  
      "object": {  
        "station": "cctv-1",  
        "focus": "programs"  
      }  
    }  
  ]  
}
```

1.2.26 股票

股票 (stock) 支持股票查询语义解析。股票意图表示：

领域	stock		
意图 意图对象	取值	说明	示例
	search	股票查询	招商银行股票
	get	属性查询	
	字段	类型	取值
	name	string	股票名称 现在只有港股，沪市和深市的股票
	code	string	股票代号
	company	string	上市公司名称
	invest_type	string	股票，基金 投资类型
	stock_type	string	a股，b股，h股 股票类型
	exchange	string	深圳证券交易所，上海证券交易所 上市交易所
	date	DATE_RANGE	YYYY-MM-DD,YYYY-MM-DD 日期

示例：

query: 股市行情

```
{
  "raw_text": "股市行情",
  "parsed_text": "股市 行情",
  "results": [
    {
      "domain": "stock",
      "intent": "search",
      "score": 1,
      "object": {
        "invest_type": "股票"
      }
    }
  ]
}
```

query: 招商银行股票多少钱

```
{
  "raw_text": "招商银行股票多少钱",
  "parsed_text": "招商 银行 股票 多少 钱",
```

```

"results": [
{
  "domain": "stock",
  "intent": "get",
  "score": 1,
  "object": {
    "code": "03968",
    "invest_type": "股票",
    "focus": "price"
  }
}
]
}

```

1.2.27 小说

小说 (novel) 提供对小说下载、在线阅读、推荐、搜索等的语义解析支持。小说意图表示：

领域	novel		
意图	取值	说明	示例
	download	下载小说	下载小说步步惊心
	play	在线阅读小说	小说步步惊心在线阅读
	recommend	推荐小说	推荐一个武侠小说 我想看一本武侠小说
	search	搜索小说	搜索小说步步惊心
意图对象	字段	类型	取值
	name	string	步步惊心
	genre	json arrary	武侠、爱情等常用类型
	chapter	int	1,2,3,4...
	sence	string	睡前看的小说
	crowd	string	儿童、初中生
	author	string	小说作者
	limit	string	txt/all/latest/pdf 等
	keywords	json arrary	小说描述关键字

示例：

query: 在线阅读步步惊心第一部

```
{
  "raw_text": "在线阅读步步惊心第一部",
  "parsed_text": "在线 阅读 步步惊心 第一部",
```

```
"results": [
  {
    "domain": "novel",
    "intent": "read",
    "score": 1,
    "object": {
      "name": "步步惊心",
      "season": "1"
    }
  }
]
```

query: 我想看一部武侠小说

```
{
  "raw_text": "我想看一部武侠小说",
  "parsed_text": "我想看 一 部 武侠小说",
  "results": [
    {
      "domain": "novel",
      "intent": "search",
      "score": 1,
      "object": {
        "genre": [
          "武侠"
        ]
      }
    }
  ]
}
```

1.2.28 播放器

播放器（player）提供对音乐的暂停、播放等控制指令的语义解析。播放器意图表示：

领域	player		
意图	取值	说明	示例
	set	设置指令	继续播放
意图对象	字段	类型	取值
	action_type	string	{ exitplayer,play,locate,pause,previou s,next, setting}
	minute	string	{1,2,3...}
	second	string	{1,2,3...}

示例：

query: 暂停播放

```
{
  "raw_text": "暂停播放",
  "parsed_text": "暂停 播放",
  "results": [
    {
      "domain": "player",
      "intent": "set",
      "score": 1,
      "object": {
        "action_type": "pause"
      }
    }
  ]
}
```

query: 从第十三分钟开始播放

```
{
  "raw_text": "从第十三分钟开始播放",
  "parsed_text": "从 第 十三 分钟 开始 播放",
  "results": [
    {
      "domain": "player",
      "intent": "locate",
      "score": 1,
      "object": {
        "minute": "13"
      }
    }
  ]
}
```

```
]
}
```

1.2.29 记账

记账 (account) 提供对记账口语表述的语义解析支持，能提供收支信息等。记账意图表示：

领域	account		
意图	取值	说明	示例
	income	收入	今天发了 5000 元工资
	expend	支出	我今晚从机场打车回家，花了 80 元；
	query	查询	帮我记一下 8 月花销多少
意图对象	字段	类型	取值
	amount	PRICE_RANGE	float,float
	date	DATE_RANGE	YYYY-MM-DD,YYYY-M M-DD
	time	TIME_RANGE	HH:MM:SS,HH:MM:SS
	location	string	地名、地标、商户
	expend_on	string	消费类型，两级：一 级>二级
	account_query	string	收入/支出
	income_from	string	从以下取值：工资： 工资、奖金、分红、 还款 生意：公司、淘宝店 理财：股票、基金、 定期存款
	expend_by	string	从以下取值{现金,刷 信用卡,刷卡}
	expend_with	string	从以下取值{自己,爱 人,子女,父母,朋友,同 学,同事}
keywords	string	query 中的关键词	输入中代表消费类型的 关键词

示例：

query: 昨天 11 点半吃饭花了 30

```
{  
  "raw_text": "昨天 11 点半吃饭花了 30",  
  "parsed_text": "昨天 11 点 半 吃饭 花 了 30",  
  "results": [  
    {  
      "domain": "account",  
      "intent": "expend",  
      "score": 1,  
      "object": {  
        "_amount": "30",  
        "_date": "昨天",  
        "_expend_on": "吃饭",  
        "_time": "11 点半",  
        "amount": "30",  
        "date": "2014-01-15,2014-01-15",  
        "expend_on": "餐饮>其他",  
        "keywords": "吃饭",  
        "time": "11:30:00,11:30:59"  
      }  
    }  
  ]  
}
```

query: 今天工资发了 5000

```
{  
  "raw_text": "今天工资发了 5000",  
  "parsed_text": "今天 工资 发 了 5000",  
  "results": [  
    {  
      "domain": "account",  
      "intent": "income",  
      "score": 1,  
      "object": {  
        "_amount": "5000",  
        "_date": "今天",  
        "amount": "5000",  
        "date": "2014-01-16,2014-01-16",  
        "income_from": "工资"  
      }  
    }  
  ]  
}
```

```
]
}
```

1.2.30 搜索

搜索 (search) 提供知识搜索方面的语义解析。搜索意图表示

领域	search			
意图	取值		说明	示例
	search		设置指令	继续播放
意图对象	字段	类型	取值	说明
	channel	string	{baidu,360,soso,sogou,google,bing,yahoo, 百科}	搜索平台
	content	string	任意字符串	搜索内容
	source	string	空或者 net	一般情况忽略，只有用户需要区分是否网络搜索时候有用

示例：

query: 百度一下明天天气

```
{
  "raw_text": "百度一下明天天气",
  "parsed_text": "百度一下 明天 天气",
  "results": [
    {
      "domain": "search",
      "intent": "search",
      "score": 0.5,
      "object": {
        "channel": "baidu",
        "content": "明天天气"
      }
    }
  ]
}
```

query: 上网搜搜去天安门的路线

```
{
```

```

"raw_text": "上网搜搜去天安门的路线",
"parsed_text": "上网 搜搜 去 天安门 的 路线",
"results": [
{
  "domain": "search",
  "intent": "search",
  "score": 0.363636,
  "object": {
    "source": "net",
    "content": "去天安门的路线"
  }
}
]
}
}

```

1.2.31 车载指令

车载指令 (vehicle_instruction) 提供用于车载环境的打开、关闭设备、蓝牙、空调控制等功能的语义解析。车载指令意图表述：

领域	vehicle_instruction		
意图	取值	说明	示例
	open	打开	打开空调；打开蓝牙；打开导航
	close	关闭	关闭空调；关闭蓝牙；关闭收音
	view	查看设备状态	查看胎压
	set_ac	设置空调	空调调到 XX 度
	call	拨打蓝牙电话	蓝牙打电话给 XXX
	answer	接听蓝牙电话	接听蓝牙电话
	hang	挂断蓝牙电话	挂断蓝牙电话
意图对象	字段	类型	取值
	equipment	string	打开关闭的设备名称
	view_item	string	查看的车辆数据项目
	temp	string	表示温度的具体数值或代表最大和最小的 MAX, MIN
	wind	string	表示风力等级的具体数值或代表最大和最小的 MAX, MIN
	phone_number	string	电话号码
			说明
			空调、收音机等
			胎压等
			当前版本仅支持具体号码

示例

query: 打开空调

```
{  
  "raw_text": "打开空调",  
  "parsed_text": "打开 空调",  
  "results": [  
    {  
      "domain": "vehicle_instruction",  
      "intent": "open",  
      "score": 1,  
      "object": {  
        "_equipment": "空调",  
        "equipment": "air_conditioner"  
      }  
    }  
  ]  
}
```

query: 空调调到 35 度

```
{  
  "raw_text": "空调调到 35 度",  
  "parsed_text": "空调 调到 35 度",  
  "results": [  
    {  
      "domain": "vehicle_instruction",  
      "intent": "set_ac",  
      "score": 1,  
      "object": {  
        "temp": "35"  
      }  
    }  
  ]  
}
```

1.2.32 收音机控制指令

收音机控制指令 (radio) 提供对收音机调台、模式切换等命令的解析支持。收音机控制指令需求表示：

领域	radio		
意图	取值	说明	示例
	open	打开	打开收音机；听收音；收音机；收音
	close	关闭	关闭收音
	search	搜索频道	自动搜索频道；搜索频道
	tune	调频到 XX 频道或电台	收 听 AM109.1 ; 调 频 到 AM109.1; AM109.1
	change_mode	切换收听模式(AM, FM)	AM; 收听 AM; 切换到 AM; 打开 AM;
意图对象	属性	类型	取值
	radio_channel	string	电台频道
	radio_station	string	电台名称
	mode	string	AM/FM

示例

query: 打开收音机

```
{
  "raw_text": "打开收音机",
  "parsed_text": "打开收音机",
  "results": [
    {
      "domain": "radio",
      "intent": "open",
      "score": 1,
      "object": {
        }
      }
    ]
  }
}
```

query: 收听 am101.1

```
{
  "raw_text": "收听 am101.1",
  "parsed_text": "收听 am101.1",
  "results": [
    {
      "domain": "radio",
      }
    ]
  }
}
```

```

    "intent": "tune",
    "score": 1,
    "object": {
        "channel": "am101.1"
    }
}
]
}
}

```

1.2.33 菜谱

菜谱 (recipe) 提供菜谱查询、推荐等语义解析的支持。菜谱意图表示：

领域	recipe		
意图	取值	说明	示例
	search	菜谱查询	西红柿炒鸡蛋的做法
	recommend	推荐菜谱(泛需求相关)	推荐一个京菜菜谱
	字段	类型	说明
	name	string	菜名
	genre	string	菜谱类型
	sence	string	菜谱场景
	material	string	食材
	tool	string	炊具
	taste	string	口味
	action	string	做法
	need	string	展现形式上的特殊需求
意图对象	crowd	string	菜谱适用人群

示例：

query：推荐一道春节吃的川菜

```

{
    "raw_text": "推荐一道春节吃的川菜",
    "parsed_text": "推荐 一 道 春节 吃 的 川菜",
    "results": [
        {
            "domain": "recipe",
            "intent": "recommend",
            "score": 1,
            "object": {
                "sence": "春节",
                "genre": "川菜"
            }
        }
    ]
}

```

```
    }  
    }  
]  
}
```

query: 西红柿炒鸡蛋的做法

```
{  
  "raw_text": "西红柿炒鸡蛋的做法",  
  "parsed_text": "西红柿 炒鸡蛋 的 做法",  
  "results": [  
    {  
      "domain": "recipe",  
      "intent": "search",  
      "score": 1,  
      "object": {  
        "name": "西红柿炒鸡蛋"  
      }  
    }  
  ]  
}
```

1.2.34 影讯

影讯 (movie_news) 提供电影、电影院相关信息的查询 (如电影上映时间、影院电影排期等)。影讯意图表示：

领域	movie_news		
意图	取值	说明	示例
	get_schedule	某地、电影院的影讯查询	最近上映了什么好看的电影
	get_info	具体电影的上映时间、地点查询	星际迷航什么时候上映
	get	获取票价	星际迷航电影票多少钱
意图对象	字段	类型	取值
	movie	string	金刚…
	cinema	string	北京东都国际影城…
	region	string	城市、区县等
	vtag	string	电影类型等
	director	string	张艺谋…
	actor	string	刘德华
	time	TIME_RANGE	
	date	DATE_RANGE	YYYY-MM-DD,YYYY-MM-DD
	dub	string	电影语种
	landmark	string	影院所在地的地标, 比如: 六里桥店

示例：

query: 明天北京的影院有啥好看的电影

```
{
  "raw_text": "明天北京的影院有啥好看的电影",
  "parsed_text": "明天 北京 的 影院 有啥 好看 的 电影",
  "results": [
    {
      "domain": "movie_news",
      "intent": "get_schedule",
      "score": 0.583994,
      "object": {
        "_date": "明天",
        "date": "2014-01-17,2014-01-17",
        "region": "北京"
      }
    }
  ]
}
```

query: 平顶山星美影讯

```
{  
  "raw_text": "平顶山星美影讯",  
  "parsed_text": "平顶山 星美 影讯",  
  "results": [  
    {  
      "domain": "movie_news",  
      "intent": "get_schedule",  
      "score": 1,  
      "object": {  
        "cinema": "星美",  
        "region": "平顶山"  
      }  
    }  
  ]  
}
```

1.2.35 导航指令

导航指令 (navigate_instruction) 提供用于导航需求的语义解析支持。导航指令意图表示：

领域	navigate_instruction		
意图	取值	说明	示例
	open	打开 X, 见字段 item	打开路况、打开收藏、前面堵不堵、我要看路况
	close	关闭 X, 见字段 item	关闭路况、关闭收藏
	follow	跟随	跟随
	view_map	查看地图	查看地图
	clear_map	清空地图	清空图区
	view_route	查看路线	查看路线、路线详情
	clear_route	清空路线	清除路线
	view_previous_route	查看上一段路线	返回上一段路线
	view_next_route	查看下一段路线	换一个路线
	route_home	回家路线	回家、我要回家、回家的路线
	route_work	上班路线	上班、回公司、去单位
	zoom_in	放大地图	放大地图、看地图细节
	zoom_out	缩小地图	缩小地图
	favorite	收藏地点	收藏这个点、保存这个地方、把这个点添加到收藏、收藏
	location	定位、当前路名	我的位置、定位、我在哪里、这是哪条街？
	navigate	开始导航, 设置或更改导航目的地, 见字段 start,arrival	开始导航、导航, 导航去西单, 改去天安门
	navigate_simulation	开始模拟导航	开始模拟导航、模拟导航
	repeat_reminder	重复上一个提醒	重复上个提醒、没听清
	set_route_sort	设置 / 切换 路线 方式, 见 字段 route_drive_sort	换成避免拥堵、路线方式切换成最少拥堵
	detour	绕开某段路线, 见字段 street	绕开北四环
	tour	经过某段路线、设置途径点, 见字段 waypoint	走北四环, 先去天安门
	quit	退出导航	退出导航、我到了, 谢谢
	speed_limit	查询限速	限速、这段路限速多少、这段路限多少公里

	rest_time	查询到达终点的时间	还要多久到、还有多久到目的地
	rest_distance	查询到达终点的剩余里程	还差几公里
	navi_street	诱导路名，见字段 navi_action	哪条街转弯？
	navi_direction	诱导方向，见字段 navi_action	往哪边转
	navi_distance	诱导距离，见字段 navi_action	多少吗左转
	switch_mode	导航相关的各类模式切换，见字段 item	话多一点，我不是在辅路上，切换成实景导航
	字段	类型	取值
意图对象	item	string	<p>打开关闭的项目： {route_condition,setting,share_poi,metro_network,satellite,favorite,overspeed_alarm,straight_tip}</p> <p>模式：</p> <p>(1) 展示模式{hud,map,text,real}</p> <p>(2) 切换主辅路 {main_road,frontage_road}</p> <p>(3) 语言模式{safe,mute,simple_clear}</p> <p>(4) 全览切换 {full_route,rest_route}</p> <p>(5) 方向罗盘 {head_forward,north_forward}</p>

	<code>route_drive_sort</code>	string	{route_drive_sort_1,route_drive_sort_2,route_drive_sort_3,route_drive_sort_4}	路线方式切换的参数。 route_drive_sort_1: 时间最短 route_drive_sort_2: 距离最短 route_drive_sort_3: 不走高速 route_drive_sort_4: 避免拥堵
	<code>street</code>	string	路线或点	北四环、南五环
	<code>arrival</code>	string	终点或途经点	除了一般地点外，还有如下保留字： terminal:终点 first_waypoint: 第一个途经点 next_waypoint: 下一个途经点 resting_area: 下一个休息区 first_transit: 第一个途经点
	<code>waypoint_no</code>	int	{1,2,3,...}	途经点的序号
	<code>waypoint</code>	string	途经点	途经点
	<code>navi_action</code>	string	行驶动作	左拐、调头、直行等

示例

query: 缩小地图

```
{
  "raw_text": "缩小地图",
  "parsed_text": "缩小 地图",
  "results": [
    {
      "domain": "navigate_instruction",
      "intent": "zoom_out",
      "score": 1,
      "object": {}
    }
  ]
}
```

query: 关闭导航

```
{  
  "raw_text": "关闭导航",  
  "parsed_text": "关闭 导航",  
  "results": [  
    {  
      "domain": "navigate_instruction",  
      "intent": "quit",  
      "score": 1,  
      "object": {}  
    }  
  ]  
}
```

第2章

概念解释

对本文中将提到的名词约定如下：

语音识别：也被称为自动语音识别 Automatic Speech Recognition (ASR)，其目标是将人类的语音中的词汇内容转换为计算机可读的输入，例如按键、二进制编码或者字符序列。

Nbest 识别结果：指识别结果的 N 候选。

应用程序：在开发中调用了 API、具有语音识别功能的产品线产品。

开发者：想在应用程序中使用 API、正在阅读本文档的开发人员。

2.1 简介

百度语音识别通过 REST API 的方式给开发者提供一个通用的 HTTP 接口，基于该接口，开发者可以轻松的获取语音识别能力，本文档描述了使用语音识别服务 REST API 的方法。

2.1.1 功能介绍

REST API 支持整段录音文件的识别，对录音格式有一定的要求，支持语音识别控件：集成提示音、音量反馈动效整套交互的对话框控件，方便开发者快速集成。

本文档适用使用 HTTP 接口的开发人员。

2.1.2 支持的语音格式

原始 PCM 的录音参数必须符合 8k/16k 采样率、16bit 位深、单声道，支持的压缩格式有：pcm (不压缩)、wav、opus、amr、x-flac。

2.2 集成指南

2.2.1 获取 Access Token

使用语音识别 REST API 需要获取 Access Token。Access Token 是用户身份验证和授权的凭证，语音识别采用的是Client Credentials授权方式，即采用应用公钥、密钥获取Access Token，适用于任何带server类型应用，通过此授权方式获取Access Token仅可访问平台授权类的接口，详见百度 OAuth 授权 “[Client Credentials授权](#)” 部分。

使用Client Credentials获取Access Token需要应用在其服务端发送请求（推荐用POST方法）到百度OAuth2.0授权服务的“<https://openapi.baidu.com/oauth/2.0/token>”地址上，并带上以下参数：

- grant_type: 必须参数，固定为“client_credentials”；
- client_id: 必须参数，应用的 API Key；
- client_secret: 必须参数，应用的 Secret Key；

例如：

```
https://openapi.baidu.com/oauth/2.0/token?  
grant_type=client_credentials&  
client_id=Va5yQRH1*****LT0vuXV4&  
client_secret= 0rDSjzQ20XUj5i*****PQSzr5pVw2&
```

响应数据包如下所示，其中“access_token”字段即为请求 REST API 所需的令牌，默认情况下，Access Token 有效期为一个月，开发者需要对 Access Token的有效性进行判断，如果Access Token过期可以重新获取。

例如：

```
HTTP/1.1 200 OK  
Content-Type: application/json  
Cache-Control: no-store  
  
{  
  "access_token": "1.a6b7dbd428f731035f771b8d*****.86400.1292922000-2346678-124328",  
  "expires_in": 86400,  
  "refresh_token": "2.385d55f8615fdfd9edb7c4b*****.604800.1293440400-2346678-124328",  
  "scope": "public",  
  "session_key": "ANXxSNjwQDugf86150nqeik*****Cd1Lxn",  
  "session_secret": "248APvxvjCZ0VEC*****aK4oZEExMB",  
}
```

2.3 API请求方式基本说明

- 语音识别接口支持 POST 方式
- 目前 API 仅支持整段语音识别的模式，即需要上传整段语音进行识别
- 语音数据上传方式有两种：隐示发送和显示发送
- 原始语音的录音格式目前只支持评测 8k/16k 采样率 16bit 位深的单声道语音
- 压缩格式支持：pcm（不压缩）、wav、opus、speex、amr、x-flac
- 系统支持语言种类：中文（zh）、粤语（ct）、英文（en）
- 正式地址：http://vop.baidu.com/server_api

2.3.1 语音上传模式

语音数据和其他参数通过标准 JSON 格式串行化 POST 上传，JSON 里包括的参数：

字段名	数据类型	可需	描述
format	string	必填	语音压缩的格式，请填写上述格式之一，不区分大小写
rate	int	必填	采样率，支持 8000 或者 16000
channel	int	必填	声道数，仅支持单声道，请填写 1
cuid	string	必填	用户唯一标识，用来区分用户，填写机器 MAC 地址或 IMEI 码，长度为 60 以内
token	string	必填	开放平台获取到的开发者 access_token
ptc	int	选填	协议号，下行识别结果选择，默认 nbest 结果
lan	string	选填	语种选择，中文=zh、粤语=ct、英文=en，不区分大小写，默认中文
url	string	选填	语音下载地址
callback	string	选填	识别结果回调地址

字段名	数据类型	可需	描述
speech	string	选填	真实的语音数据，需要进行base64 编码
len	int	选填	原始语音长度，单位字节

其中，开发者可以把语音数据放在 JSON 序列的“speech”字段中，需要将语音先进行 base64 编码，并标明语音数据的原始长度，填写“len”字段；也可以直接提供语音下载地址放在“url”字段中，并且提供识别结果的回调地址，放在“callback”参数中。因此“speech”和“len”参数绑定，“url”和“callback”参数绑定，这两组参数二选一填写，如果都填，默认处理第一种。

- 表单类型在 HTTP-HEADER 里的 content-type 表明，例：

```
Content-Type:application/json
```

Content-length 请填写 JSON 串的长度。

- 上传实例：

```
{
    "format": "speex",
    "rate": 8000,
    "channel": 1,
    "token": "xxx",
    "cuid": "baidu_workshop",
    "len": 4096,
    "speech": "xxx",
}
```

注意事项：

- len 字段表示原始语音长度，不是 base64 编码之后的长度。
- speech 和 len 字段绑定验证，url 和 callback 绑定验证，两组参数二选一必填，如果都填，默认第一种方式。
- 如果采用 base64 编码语音数据，数据量会增大 1/3。

语音数据直接放在 HTTP-BODY 中，控制参数以及相关统计信息通过 REST 参数传递，REST 参数说明：

字段名	数据类型	可需	描述
cuid	string	必填	用户 ID, 推荐使用设备mac 地址/手机 IMEI 等设备唯一性参数
token	string	必填	开发者身份验证密钥
lan	string	选填	语种选择, 中文=zh、粤语=ct、英文=en, 不区分大小写, 默认中文
ptc	int	选填	协议号, 下行识别结果选择, 默认nbest 结果

- 语音数据的采样率和压缩格式在 HTTP-HEADER 里的 content-type 表明, 例:

```
Content-Type:audio/amr;rate=8000
```

Content-length 请填写原始语音长度。

- URL示例:

```
http://vop.baidu.com/server_api?lan=zh&cuid=***&token=***
```

2.3.2 下行接口定义

两种上传方式都返回统一的结果, 采用 JSON 格式封装, 如果识别成功, 识别结果放在 JSON 的“result”字段中, 统一采用 utf-8 方式编码。

字段名	数据类型	可需	描述
err_no	int	必填	错误码
err_msg	string	必填	错误码描述
sn	string	必填	语音数据唯一标识, 系统内部产生, 用于 debug
result	array ([string,string, ...])	选填	识别结果数组, 提供1-5个候选结果, string 类型为识别的字符串, utf-8 编码

- 识别成功返回 case

```
{\"err_no\":0,\"err_msg\":\"success.\",\"corpus_no\":\"15984125203285346378\",\"sn\":\"481D633F-73BA-726F-49EF-8659ACCC2F3D\",\"result\":[\"北京天气\"]}
```

- 识别错误返回 case

```
{"err_no":2000,"err_msg":"dataEmpty.","sn":null}
```

2.3.3 错误码解释

错误码	含义
3300	输入参数不正确
3301	识别错误
3302	验证失败
3303	语音服务器后端问题
3304	请求 GPS 过大，超过限额
3305	产品线当前日请求数超过限额

2.4 注意事项

1. 请严格按照文档里描述的参数进行开发，特别请关注原始录音参数以及语音压缩格式的建议，否则会影响识别率，进而影响到产品的用户体验。
2. 目前系统支持的语音时长上限为60s，请不要超过这个长度，否则会返回错误。

第3章

简介

3.0.1 概念解释

对本文中将提到的名词约定如下：

语音识别 (Automatic Speech Recognition, ASR)：也被称为自动语音识别，其目标是将人类的语言中的词汇内容转换为计算机可读的输入，例如按键、二进制编码或者字符序列。

自然语言理解 (Natural Language Understanding, NLU)：俗称人机对话，人工智能的分支学科。研究用电子计算机模拟人的语言交际过程，使计算机能理解和运用人类社会的自然语言如汉语、英语等，实现人机之间的自然语言通信，以代替人的部分脑力劳动，包括查询资料、解答问题、摘录文献、汇编资料以及一切有关自然语言信息的加工处理。

语音识别SDK离在线融合版 (Baidu Voice Recognition Client V2, BDASRv2)：文中简称为BDASRv2。BDASRv2是一个封装了语音采集、语音预处理、云端识别、离线识别等功能的语音识别解决方案。使用BDASRv2可以快速在应用程序中集成功能。

3.0.2 功能介绍

BDVRClient是运行在Android 平台的一体化语音识别解决方案，以JAR包 + SO库的形式发布，BDVRClient支持下列功能：

- 语音识别控件：集成提示音、音量反馈动效整套交互的对话框控件，方便开发者快速集成；
- 基本功能：录音，语音数据处理，端点检测、网络通讯和状态通知，返回识别结果；
- 播放提示音：在录音前后播放提示音，优化用户体验；
- 监听语音能量：实时反馈用户当前说话声音能量强度；
- 语义理解：将语音识别成领域相关的语义结果。

3.0.3 兼容性

系统：支持Android 2.3 (API Level 9) 及以上系统。需要开发者通过minSdkVersion来保证支持系统的检测。

机型：手机和平板均可。

构架：支持ARM平台（只提供armeabi架构的动态库，可以兼容其他架构）。

硬件要求：要求设备上有麦克风。

网络：支持WIFI及移动网络，支持2G、3G、4G移动网络。

3.0.4 开发包说明

文件/文件夹名	说明
/demo/	语音识别 SDK demo工程
/libs/	包括so库及jar包。
/res	语音识别对话框资源文件及音效文件，如果不使用对话框可以不集成
/DEMO-VoiceRecognition-2.0.1.apk	Android 示例应用

3.0.5 总体框图



图 3.1: 图

3.1 集成指南

本章将讲解如何快速地集成BDVRClient到现有应用中。一个完整的Demo请参考开发包中的示例程序VoiceRecognitionDemo。

3.1.1 创建应用

请参考《百度语音开放平台使用指南》创建应用，开通服务并完成个性化设置。

3.1.2 添加BDVRClient到工程

将开发包中的libs和res目录分别合并到工程目录的libs和res目录。

3.1.3 配置AndroidManifest.xml

AndroidManifest.xml配置主要内容为：增加权限，填写鉴权信息，注册语音服务，注册对话框。具体示例如下：

```
...
<uses-permission android:name=" android.permission.RECORD_AUDIO" ></uses>
<uses-permission android:name=" android.permission.INTERNET" ></uses>
<uses-permission android:name=" android.permission.ACCESS_NETWORK_STATE" ></uses>
<uses-permission android:name=" android.permission.ACCESS_WIFI_STATE" ></uses>
<uses-permission android:name=" android.permission.CHANGE_WIFI_STATE" ></uses>
<uses-permission android:name=" android.permission.READ_PHONE_STATE" ></uses>
<uses-permission android:name=" android.permission.WRITE_EXTERNAL_STORAGE" ></
    uses>
...
<application
    android:icon="@drawable/ic_launcher"
    android:label="@string/app_name" >

    <!-- begin: baidu speech sdk -->
        <!-- 请填写应用实际的APP_ID -->
        <meta-data android:name=" com.baidu.speech.APP_ID" android:value=" your_app_id
            " />
        <!-- 请填写应用实际的API_KEY -->
        <meta-data android:name=" com.baidu.speech.API_KEY" android:value=" your_api_key
            " />
        <!-- 请填写应用实际的SECRET_KEY -->
        <meta-data android:name=" com.baidu.speech.SECRET_KEY" android:value=" your_api
            _secret" />
    <service android:name=" com.baidu.speech.VoiceRecognitionService" android:exported
        = " false" ></service>
```

```
<activity
    android:name=" com.baidu.voicerecognition.android.ui.BaiduASRDigitalDialog"
    android:configChanges=" orientation|keyboardHidden|screenLayout"
    android:theme="@android:style/Theme.Dialog"
    android:exported=" false"
    android:screenOrientation=" portrait" >
<intent-filter>
    <action android:name=" com.baidu.action.RECOGNIZE_SPEECH" ></action>
    <category android:name=" android.intent.category.DEFAULT" ></category>
</intent-filter>
</ activity>
<!-- end : baidu speech sdk-->
...
</application>
...
```

各权限含义说明：

名称	用途
android.permission.RECORD_AUDIO	允许应用使用麦克风录音
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	获取当前的网络状态，优化录音及网络参数。
android.permission.ACCESS_WIFI_STATE	获取当前wifi状态，优化录音及网络参数
android.permission.READ_PHONE_STATE	获取用户手机的IMEI，用来唯一的标识用户。

3.2 重要接口说明

3.2.1 语音识别器

语音识别器SpeechRecognizer，使用谷歌系统框架，函数与使用android自有框架基本相同。

创建实例

- 方法

```
SpeechRecognizer createSpeechRecognizer(Context context, ComponentName serviceComponent);
```

- 参数

参数	含义
context	上下文环境
serviceComponent	用来提供语音识别的服务

- 返回

语音识别器。

- 说明

创建语音识别器。使用谷歌系统框架，使用百度语音识别只需将serviceComponent指定为百度语音提供的服务即可。

设置语音识别监听器

- 方法

```
void setRecognitionListener(RecognitionListener listener)
```

- 参数

参数	含义
listener	监听器

- 返回

无。

- 说明

设置语音识别监听器，语音识别监听器需要实现RecognitionListener中的接口对语音过程和结果做出处理。

开始识别

- 方法

```
void startListening(Intent recognizerIntent)
```

- 参数

参数	含义
recognizerIntent	识别参数

- 返回

无。

- 说明

所有识别的参数都需要在recognizerIntent中设置。

停止录音

- 方法

```
void stopListening();
```

- 参数

无。

- 返回

无。

- 说明

停止录音，但是识别将继续。

取消识别

- 方法

```
void cancel();
```

- 参数

无。

- 返回

无。

- 说明

取消本次识别，已有录音也将不再识别。

销毁

- 方法

```
void destroy();
```

- 参数

无。

- 返回

无。

- 说明

销毁语音识别器，释放资源。

3.2.2 语音识别监听器

语音识别监听器RecognitionListener，该接口包含如下方法回调。

准备就绪

- 方法

```
void onReadyForSpeech(Bundle params)
```

- 参数

参数	含义
params	识别参数

- 返回

无。

- 说明

只有当此方法回调之后才能开始说话，否则会影响识别结果。

开始说话

- 方法

```
void onBeginningOfSpeech()
```

- 参数

无。

- 返回

无。

- 说明

当用户开始说话，会回调此方法。

音量变化

- 方法

```
void onRmsChanged(float rmsdB)
```

- 参数

参数	含义
rmsdB	音量值

- 返回

无。

- 说明

引擎将对每一帧语音回调一次该方法返回音量值。

获取原始语音

- 方法

```
void onBufferReceived(byte[] buffer)
```

- 参数

参数	含义
buffer	pcm语音数据

- 返回

无。

- 说明

此方法会被回调多次，buffer是当前帧对应的PCM语音数据，拼接后可得到完整的录音数据。

说话结束

- 方法

```
void onEndOfSpeech()
```

- 参数

无

- 返回

无。

- 说明

当用户停止说话后，将会回调此方法。

识别出错

- 方法

```
void onError(int error)
```

- 参数

参数	含义
error	错误码

- 返回

无。

- 说明

识别出错，将会回调此方法，调用该方法之后将不再调用onResults方法。

识别最终结果

- 方法

```
void onResults(Bundle results)
```

- 参数

参数	含义
results	识别结果

- 返回

无。

- 说明

返回最终识别结果，将会回调此方法。

识别临时结果

- 方法

```
void onPartialResults(Bundle partialResults)
```

- 参数

参数	含义
partialResults	临时识别结果

- 返回

无。

- 说明

返回临时识别结果，将会回调此方法。

识别事件返回

- 方法

```
void onEvent(int eventType, Bundle params)
```

- 参数

参数	含义
eventType	事件类型
eventType	参数

- 返回

无。

- 说明

返回识别事件，将会回调此方法。

3.3 参数和错误码说明

3.3.1 识别参数

支持的识别参数

参数名	类型	值	描述
appid	int		开放平台创建应用后分配，设置后会覆盖 manifest 中配置的 APP_ID
key	String		开放平台创建应用后分配，设置后会覆盖 manifest 中配置的 API_KEY
secret	String		开放平台创建应用后分配，设置后会覆盖 manifest 中配置的SECRET_KEY
sample	int	-	采样率

参数名	类型	值	描述
		16000	
		8000	离线暂不支持
sound_start	int	资源ID	说话开始的提示音
sound_end	int	资源ID	说话结束的提示音
sound_success	int	资源ID	识别成功的提示音
sound_error	int	资源ID	识别出错的提示音
sound_cancel	int	资源ID	识别取消的提示音
infile	String	音频源	** 该参数支持设置为： **1. 文件系统路径，如：“/sdcard/test/test.pcm” 2. java 资源路径，如：“res:///com/baidu.test/16k_test.pcm” 3. 数据源方法全名，格式如：“#com.test.Factory.create8kInputStream()”（解释： Factory类中存在一个返回InputStream的方法create8kInputStream()）注意：必须以#号开始；方法原型必须为：public static InputStream your_method()，而且该方法和类一定不能混淆，否则无法获取音频源。
outfile	String	文件路径	保存识别过程产生的录音文件
language	String	-	语种
		cmn-Hans-CN	中文普通话
		sichuan-Hans-CN	中文四川话（离线暂不支持）

参数名	类型	值	描述
		yue-Hans-CN	粤语（离线暂不支持）
		en-GB	英语（离线暂不支持）
nlu	String	-	语义解析设置
		disable	禁用语义解析
		enable	开启语义解析
grammar	String	-	离线识别的语法路径（支持asset目录访问，如:assets:///mygram.bsg）
keyword	String	-	语义功能要“砍掉”的前缀词，配合唤醒功能可以实现唤醒+识别一体化。如某次会话中用户说了百度一下明天天气怎么样，将该参数设置为“百度一下”则语义功能将只解析“明天天气怎么样”
vad	String	-	语音活动检测
		search	搜索模式，适合短句输入
		input	输入模式，适合短信、微博内容等长句输入
asr-base-file-path	String	文件路径	**2.1 版本已被 grammar 参数取代 ** 支持离线语音识别的基础包，下载方式参见集成指南

参数名	类型	值	描述
lm-res-file-path	String	文件路径	**2.1 版本已被 grammar参数取代 ** 支持离线语音识别的附加包，包括输入法和地图资源，下载方式参见集成指南
license-file-path	String	文件路径	可选，支持离线语音识别的授权文件，下载方式参见集成指南，如果不设置，将在联网时尝试在线获取
slot-data	String	JSON字符串	用于自定义离线识别，具体格式参见 demo
prop	int	-	垂直领域， **2.1版本后离线功能请使用grammar参数**
		20000	输入（离线支持）
		10005	热词
		10060	地图（离线支持）
		10001	音乐（离线支持）
		10002	视频
		10003	应用（离线支持）
		10004	网页
		10006	购物
		10007	健康
		10008	电话（离线支持）
		10052	医疗
		10053	汽车
		10054	娱乐餐饮
		10055	财经
		10056	游戏
		10057	菜谱
		10058	助手

参数名	类型	值	描述
		100014	联系人（离线支持）
		100016	手机设置（离线支持）
		100018	电视指令（离线支持）
		100019	播放器指令（离线支持）
		100020	收音机指令（离线支持）
		100021	命令词（离线支持）

3.3.2 离线识别支持语法说明

本文档列举了离在线融合语音识别离线部分支持的命令说法，并给出示例。说法是由槽组成的，每种命令首先列举出槽，然后说明由这些槽组成的哪些语法。关于槽内内容，如果数量较少，本文档会一一列举出来，如果数量较多，本文档会举出一些例子。具有“包括”字样的就是将所有的列举出来。本文档为了便于理解，同时是为了说明支持的说法，命名和识别出的json结果可能有些出入，请开发者注意。

需要输入法资源文件支持，s_2_InputMethod[下载链接](#)，支持随便说。

需要地图资源文件支持，s_2_Navi[下载链接](#)。

槽名	说明
Cmd_to	表示寻找目的地的动词（包括导航，导航到，导航去，出发，走起，我要去，带我去，走，去，上，到，我想去，带我到，现在去，改去）
Arrival	任意地名
Nearby	表示附件的介词或者动词（包括附近的，周边的，旁边的，最近的，搜附近的，搜周边的，搜旁边的，搜最近的，我附近的，我周围的，导航去周边的，搜索周边的，我要去周边的，导航去附近的，搜索附近的，我要去附近的，导航去最近的，搜索最近的，我要去最近的）

槽名	说明
NearbyCore	表示一类事物的名词，例如加油站，港口等；

槽说明（导航）

支持说法	说明举例
cmd_to Arrival	导航到百度大厦
Arrival	用户只说地名，例如 百度大厦
Nearby NearbyCore	附近的 加油站

支持说法

槽名	说明
PlayCmd	播放命令（包括播放音乐，放音乐，请播放歌曲，请播放音乐，来一首，我要听，给我来一首，播放）
song	歌曲名称，**需要开发者传入**
artist	歌手名称，**需要开发者传入**
Tail	包括 的歌，的专辑

槽说明（音乐）

支持说法	说明举例
PlayCmd Song	播放《小苹果》
PlayCmd Artist	播放周杰伦的歌

支持说法

槽名	说明
app	应用名称，**需要开发者传入**
LaunchCmd	包括打开，开启，启动
UnInstallCmd	包括 删除，卸掉，卸载，删掉；

槽说明（应用）

支持说法	说明举例
App	直接说应用名称，例如，微信
LaunchCmd App	开启 手机百度
UnInstallCmd App	卸载 360手机助手

支持说法

槽名	说明
name	通讯录人名，需要开发者传入
PhoneNum	电话号码
Call	表示打电话的命令（包括打电话，打个电话，我想打电话，我要打电话，帮我打电话，请打电话，拨号，请拨号，帮我拨号，请帮我拨号，我要拨号）
CallTo	表示“打电话给XXX”的命令（例如，呼叫，拨打，拨号等）；
FirstCmd	一些前缀（包括给，我想给，我要给，帮我给，请给，请帮我给）；
LastCmd	包括打电话，打个电话；
PhoneType	号码类型（包括号码前缀，例如，132，188等，包括运营商，例如移动联通电信，包括号码类型，例如，座机，手机，手机号码等）
View	包括我要看，看一下，我想看，看看，查看；
ObjectCall	包括没接通电话，未接来电，已接电话，通话记录；
Sms	表示发短信的命令（例如，发短信，发个短信，等等）；
SmsTo	表示“发短信给XXX”的命令（发短信给，我想发给等）；
SmsLastCmd	包括 发短信，发个短信，发信息，发个信息；
ObjectSms	包括 未读短信，已读短信，全部短信；

槽说明（打电话/发短信）

支持说法	说明举例
Call	表示用户打电话的意图，比如，打电话
CallTo PhoneNum	例如 打电话给10086
CallTo Name	例如打电话给张三
CallTo Name PhoneType	例如，打电话给张三的移动号码，打电话给李四的132号码，等；
FirstCmd Name LastCmd	例如，请给张三打电话；
View ObjectCall	例如，查看未接来电；
Sms	表示用户打电话的意图，比如，发个短信
SmsTo PhoneNum	例如 发短信给10086
SmsTo Name	例如 发短信给张三
SmsTo Name PhoneType	例如发短信给张三的移动号码，发个短信给李四的132号码，等；
FirstCmd Name SmsLastCmd	例如，请给张三发个短信；
View ObjectSms	例如，查看未读短信；

支持说法

槽名	说明
View	包括我要看，看一下，我想看，看看，查看；
name	通讯录人名，**需要开发者传入**
PhoneType	号码类型（包括号码前缀，例如，132，188等，包括运营商，例如移动联通电信，包括号码类型，例如，座机，手机，手机号码等）
Create	包括新建，添加，增加，加入，加上
Edit	包括编辑，更改，更新，修改；
Del	删除，删掉，去掉；
Contactor	联系人

槽说明（联系人）

支持说法	说明举例
View Name PhoneType	查看张三的移动号码
Create (Contactor)	新建(联系人)
Edit Name PhoneType	更改张三的座机
Del Name PhoneType	删除张三的132号码

支持说法

槽名	说明
PhoneSetting	电话设置，包括常用电话设置，包括设定时间，打开wifi等常用设置命令

槽说明 (手机设置)

支持说法	说明举例
PhoneSetting	

支持说法

槽名	说明
Change	换台命令，包括调台，切换，换台，我要跳台，我要切换，我要换台，我要看
Number	数字，范围0~999
TvChannel	电视台，包括北京电视台等常用电视台；
Channel	包括，频道，台；

槽说明 (电视指令)

支持说法	说明举例
Change Number Channel	我要看 74 频道
Change TvChannel	切换河北卫视

支持说法

槽名	说明
Setplayer	常用设置命令，包括退出播放，退出，继续播放，继续，暂停播放，暂停，上一首，下一首，设置
Play	包括播放，听，观看
Time	表示时间的槽
From	从

槽说明 (播放器指令)

支持说法	说明举例
Setplayer	继续播放
From Time Play	从43分32秒开始播放

支持说法

槽名	说明
OpenCmd	常见打开命令，例如，收听电台，听FM，打开AM等
ChangeMode	用于切换AM/FM的指令，例如，打开FM，切换到AM等
CloseCmd	关闭收音机，关机
Number	数字
Channel	频道，电台

槽说明 (收音机指令)

支持说法	说明举例
OpenCmd	打开电台
CloseCmd	关闭收音机
OpenCmd Number (Channel)	收听FM八七点五频道
ChangeMode	切换到FM

支持说法

槽名	说明
usercommand	**需要开发者传入**

槽说明（用户命令）

支持说法	说明举例
usercommand	

支持说法

3.3.3 错误码说明

错误码	常量名	描述
1	ERROR_NETWORK_TIMEOUT	网络超时
2	ERROR_NETWORK	网络错误
3	ERROR_AUDIO	录音错误
4	ERROR_SERVER	服务端错误
5	ERROR_CLIENT	客户端调用错误
6	ERROR_SPEECH_TIMEOUT	超时
7	ERROR_NO_MATCH	没有识别结果
8	ERROR_RECOGNIZER_BUSY	引擎忙
9	ERROR_INSUFFICIENT_PERMISSION	缺少权限

通用错误码

详细错误信息 当语音识别发生错误时，将通过[RecognitionListener#onError\(int error\)](#)方法返回错误码，并且通过[RecognitionListener#onEvent\(int eventType, Bundle params\)](#)方法返回详细的错误信息。获取详细的错误信息示例如下：

```

@Override
public void onEvent(int eventType, Bundle params) {
    switch(eventType) {
        case 11:// eventType == 11 表是返回详细错误信息
            String reason = params.get("reason") + " ";
            print("EVENT_ERROR," + reason);
            break;
    }
}

```

```
    }  
}
```

借助该方法返回的错误信息，开发者可以更快的发现和调试问题。

详细错误码	描述
3002	服务器后端错误
3003	没有识别结果
3004	apikey/secretkey无语音识别权限
3005	语音质量过低
7051	没有在指定路径找到s_1 (离线识别基础资源包)
7052	没有在指定路径找到s_2_xxx (prop=20000 或10060时对应的附加包)
7055	命令词超过数量限制 (10个)
7056	没有指定prop参数
7057	不支持的采样率 (仅支持16000采样率)
7058	不支持的垂类 (参见 “ 识别参数 ”)
7059	设置的离线语义数据无效
7060	不支持的语言类型 (参见 “ 识别参数 ”)
7998	离线识别没有得到结果
9005/9101/9xxx	下载授权失败 (没有申请离线授权或者提交的信息不正确，参见 DEMO工程无法使用离线识别？)

业务错误码

3.4 完整示例

3.4.1 API方式识别示例

完整的示例代码如下：

```
import java.util.ArrayList;  
import java.util.Arrays;  
  
import org.json.JSONObject;
```

```
import com.baidu.speech.VoiceRecognitionService;
import com.baidu.speech.recognizerdemo.R;

import android.content.ComponentName;
import android.content.Intent;
import android.os.Bundle;
import android.sax.StartElementListener;
import android.speech.SpeechRecognizer;

class DemoActivity extends Activity implements RecognitionListener {
    private SpeechRecognizer speechRecognizer;

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main_activity);
        // 创建识别器
        speechRecognizer = SpeechRecognizer.createSpeechRecognizer(this, new
            ComponentName(this, VoiceRecognitionService.class));
        // 注册监听器
        speechRecognizer.setRecognitionListener(this);
        startASR();
    }

    // 开始识别
    void startASR() {
        Intent intent = new Intent();
        bindParams(intent);
        speechRecognizer.startListening(intent);
    }

    void bindParams(Intent intent) {
        // 设置识别参数
    }

    public void onReadyForSpeech(Bundle params) {
        // 准备就绪
    }

    @Override
    public void onBeginningOfSpeech() {
        // 开始说话处理
    }

    public void onRmsChanged(float rmsdB) {
        // 音量变化处理
    }

    public void onBufferReceived(byte[] buffer) {
```

```
// 录音数据传出处理
}

public void onEndOfSpeech() {
    // 说话结束处理
}

public void onError(int error) {
    // 出错处理
}

public void onResults(Bundle results) {
    // 最终结果处理
}

public void onPartialResults(Bundle partialResults) {
    // 临时结果处理
}

public void onEvent(int eventType, Bundle params) {
    // 处理事件回调
}
}
```

3.4.2 语音识别控件示例

语音识别控件提供了整套语音交互、提示音、音量反馈、动效反馈。开发者可以像调起一个普通的Activity一样简单的使用语音识别。

```
public class DemoActivity extends Activity {

    private static final int REQUEST_UI = 1;

    private void start() {
        Intent recognizerIntent = new Intent();
        // recognizerIntent.put(" ... ", " ... ") TODO 为设置参数，支持的参数见本文档的“识别参数”一节
        recognizerIntent
        // 为了支持离线识别能力，请参考“离线语音识别参数设置”一节
        startActivityForResult(recognizerIntent, REQUEST_UI);
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        if (resultCode == RESULT_OK) {
            ArrayList<String> results = data.getStringArrayList(SpeechRecognizer.
            RESULTS_RECOGNITION);
        }
    }
}
```

```
// data.get ... TODO 识别结果包含的信息见本文档的“结果解析”一节  
}  
}  
}  
}
```

3.4.3 离线识别参数设置示例

注意事项 由于离线识别仅支持部分识别参数，如果需要支持离线识别能力，需要确保当前的参数设置被离线所支持：

```
intent.putExtra(" sample" , 16000); // 离线仅支持采样率16000  
intent.putExtra(" language" , " cmn-Hans-CN" ); // 离线仅支持中文普通话  
intent.putExtra(" prop" , 20000); // 输入  
// intent.putExtra(" prop" , 10060); // 地图  
// intent.putExtra(" prop" , 10001); // 音乐  
// intent.putExtra(" prop" , 10003); // 应用  
// intent.putExtra(" prop" , 10008); // 电话  
// intent.putExtra(" prop" , 100014); // 联系人  
// intent.putExtra(" prop" , 100016); // 手机设置  
// intent.putExtra(" prop" , 100018); // 电视指令  
// intent.putExtra(" prop" , 100019); // 播放器指令  
// intent.putExtra(" prop" , 100020); // 收音机指令  
// intent.putExtra(" prop" , 100021); // 命令词
```

```
// 替换为资源文件实际路径value  
intent.putExtra(" asr-base-file-path" , " /path/to/s_1" );  
// 替换为文件实际路径，仅在使用临时文件时需要进行设置，如果在valueLICENSE应用管  
理中开通了离线授权，不需要设置该参数[]  
intent.putExtra(" license-file-path" , " /path/to/license-tmp-20150530.txt" );  
if (prop == 10060) {  
    // 地图类附加资源，替换为资源文件实际路径value  
    intent.putExtra(" lm-res-file-path" , " /path/to/s_2_Navi" );  
} else if (prop == 20000) {  
    // 语音输入附加资源，替换为资源文件实际路径value  
    intent.putExtra(" lm-res-file-path" , " /path/to/s_2_InputMethod" );  
}
```

```
JSONObject slotData = new JSONObject();  
JSONArray name = new JSONArray().put(" 张三" ).put(" 李四" );  
JSONArray song = new JSONArray().put(" 七里香" ).put(" 冰雨" );  
JSONArray artist = new JSONArray().put(" 周杰伦" ).put(" 刘德华" );
```

```
JSONArray app = new JSONArray().put(" 手机百度").put(" 百度地图");
JSONArray usercommand = new JSONArray().put(" 关灯").put(" 开门");
try {
    slotData.put(" name" , name);
    slotData.put(" song" , song);
    slotData.put(" artist" , artist);
    slotData.put(" app" , app);
    slotData.put(" usercommand" , usercommand);
} catch (JSONException e) {

}
intent.putExtra(" slot-data" , slotData.toString());
```

3.5 语音唤醒

语音唤醒是指说出指定的语音指令（自定义的唤醒词），使程序激活某个功能的能力。百度语音唤醒支持自定义唤醒词。

3.5.1 接入唤醒功能

唤醒功能属于语音识别离在线融合SDK（依赖2.1及以上版本）的一部分，开发者根据集成指南接入之后即可使用。

3.5.2 自定义唤醒词

唤醒词是指用于激活程序某个功能的指令。开发者可以使用唤醒词评估工具评估和导出唤醒资源。[唤醒词评估工具](#)

3.5.3 接入示例

1. 创建唤醒事件管理器

```
EventManager wakeup = EventManagerFactory.create(ActivityWakeUp.this, "wp");
```

2. 注册唤醒事件监听器

```
// 2) 注册唤醒事件监听器
mWpEventManager.registerListener(new EventListener() {
    @Override
    public void onEvent(String name, String params, byte[] data, int offset, int length) {
    }
});
```

3. 启动唤醒功能

```
// 3) 通知唤醒管理器，启动唤醒功能
HashMap params = new HashMap();
params.put("kws-file", "assets:///WakeUp.bin"); // 设置唤醒资源，唤醒资源
请到 http://yuyin.baidu.com/wake#m4 来评估和导出
mWpEventManager.send("wp.start", new JSONObject(params).toString(), null, 0, 0);
```

4. 停止唤醒监听

```
mWpEventManager.send("wp.stop", null, null, 0, 0);
```

3.5.4 完整示例

```
public class ActivityWakeUp extends Activity {

    private EventManager mWpEventManager;

    @Override
    protected void onResume() {
        super.onResume();

        // 唤醒功能打开步骤
        // 1) 创建唤醒事件管理器
        mWpEventManager = EventManagerFactory.create(ActivityWakeUp.this, "wp");

        // 2) 注册唤醒事件监听器
        mWpEventManager.registerListener(new EventListener() {
            @Override
            public void onEvent(String name, String params, byte[] data, int offset, int length) {
                try {
                    JSONObject json = new JSONObject(params);
                    if ("wp.data".equals(name)) { // 每次唤醒成功，将会回调
                        name=wp.data的时间，被激活的唤醒词在params的word字段
                        String word = json.getString("word"); // 唤醒词
                    } else if ("wp.exit".equals(name)) {
                        // 唤醒已经停止
                    }
                } catch (JSONException e) {
                    throw new AndroidRuntimeException(e);
                }
            }
        });
    }

    // 3) 通知唤醒管理器，启动唤醒功能
    HashMap params = new HashMap();
    params.put("kws-file", "assets:///WakeUp.bin"); // 设置唤醒资源，唤醒资源
    请到 http://yuyin.baidu.com/wake#m4 来评估和导出
    mWpEventManager.send("wp.start", new JSONObject(params).toString(), null, 0, 0);
}

@Override
protected void onPause() {
    super.onPause();
    // 停止唤醒监听
    mWpEventManager.send("wp.stop", null, null, 0, 0);
}
```

```
}
```

3.6 其他说明

3.6.1 结果解析

最终识别结果将通过onResults(Bundle results)方法返回，其中results支持以下字段：

参数名	类型	描述
results_recognition	ArrayList	识别结果
origin_result	String(JSONObject)	字符串形式的JSON结构体，其中包含原始的识别结果信息

3.6.2 常见问题

DEMO工程无法使用离线识别？ DEMO工程要使用离线识别，需要注意以下三点： 1. 离线授权 2. 离线数据包 3. 识别参数是否支持离线

离线授权

- 方法一（推荐）：在[应用管理](#)中申请离线授权：请参考“[集成指南](#)”一节申请离线授权，并且确认填写的包名或者bundle ID是否正确。
- 方法二：下载30天有效期的临时license文件，存放到设备的任意可读路径，并通过如下方法设置（此处示例为存放于sdcard根目录）：

```
intent.putExtra("license_file_path", "/sdcard/temp_license_2015-05-19");
```

离线数据包

在语音识别的“[相关下载](#)”里下载离线数据包，存放到设备的任意可读路径，并通过如下方法设置（此示例为存放于sdcard根目录）：

```
intent.putExtra("asr-base-file-path", "/sdcard/s_1");
if (prop == 10060) { // 地图
    intent.putExtra("lm-res-file-path", "/sdcard/s_2_Navi");
} else if (prop == 20000) { // 输入法
    intent.putExtra("lm-res-file-path", "/sdcard/s_2_InputMethod");
}
```

[识别参数是否支持离线](#)

请参考[“离线识别参数设置”](#) - [“注意事项”](#)一节，确认当前设置的参数是否支持离线。

如何获取录音数据 设置outfile参数可以指定语音数据的保存路径，设置方式如：

```
intent.putExtra("outfile", "/sdcard/your_audio.pcm");
```

语音数据的保存格式为PCM，播放和压缩请自行查找相关类库。

第4章

简介

4.0.3 概念解释

对本文中将提到的名词约定如下：

语音识别 (Automatic Speech Recognition, ASR)：也被称为自动语音识别，其目标是将人类的语言中的词汇内容转换为计算机可读的输入，例如按键、二进制编码或者字符序列。

自然语言理解 (Natural Language Understanding, NLU)：俗称人机对话，人工智能的分支学科。研究用电子计算机模拟人的语言交际过程，使计算机能理解和运用人类社会的自然语言如汉语、英语等，实现人机之间的自然语言通信，以代替人的部分脑力劳动，包括查询资料、解答问题、摘录文献、汇编资料以及一切有关自然语言信息的加工处理。

语音识别SDK离在线融合版 (Baidu Voice Recognition Client, BDVRCClient)：文中简称为BDVRCClient。BDVRCClient是一个封装了语音采集、语音预处理、云端识别、离线识别等功能的语音识别解决方案。使用BDVRCClient可以快速在应用程序中集成语音识别功能。

4.0.4 功能介绍

BDVRCClient支持下列功能：

- 基本功能：录音、语音数据处理、端点检测、网络通讯、状态通知、返回文字结果；
- 语音识别控件：集成提示音、音量反馈动效整套交互的对话框控件，方便开发者快速集成；
- 播放提示音：在录音前后播放提示音，优化用户体验；
- 监听语音音量：实时反馈用户当前说话声音能量强度；
- 语义理解：将语音识别成领域相关的语义结果。

4.0.5 兼容性

系统：支持iOS 5.0及以上系统。

架构： armv7、 armv7s、 arm64、 i386、 x86_64。

机型： iPhone 4+， iPad 2+和iPod 5+。

硬件要求：需要有麦克风，用于支持语音录入。

网络：支持NET、 WiFi网络环境。

4.0.6 开发包说明

一级目录	二级目录	说明
Headers	BDVoiceRecognitionClient.h	BDVRClient无UI头文件
	BDRecognizerViewController.h	BDVRClient UI头文件
	BDRecognizerViewDelegate.h	BDVRClient UI结果回调接口头文件
	BDRecognizerViewParamsObject.h	BDVRClient UI中启动参数头文件
	BDTheme.h	BDVRClient UI主题头文件
	BDVRRawDataRecognizer.h	BDVRClient音频数据识别头文件
	BDVRFileRecognizer.h	BDVRClient音频文件识别头文件
BDVRClientSample	SDK Demo源代码	开发示例
Third-party	各种第三方库	需要添加到项目中的第三方库
libBDVoiceRecognitionClient	libBDVoiceRecognitionClient.a	通用库，合并了真机armv7、 armv7s、 arm64和模拟器版的库
BDVoiceRecognitionClientResources	Tone	提示音资源文件
	Theme	识别控件主题
Data	数据文件	离线语音识别所需要的数据文件

一级目录	二级目录	说明
License	授权文件	离线语音识别所需要的授权文件
Doc	百度语音识别iOS版开发手册	开发者使用指南

4.1 集成指南

本章将讲解如何快速地集成BDVRClient到现有应用中。一个完整的Demo请参考开发包中的示例程序VoiceRecognitionDemo。

4.1.1 创建应用

请参考《百度语音开放平台使用指南》创建应用，开通服务并完成个性化设置。

4.1.2 引入编译需要的Framework

BDVRClient使用了录音和播放功能，因此需要在Xcode工程中引入AudioToolbox.framework和AVFoundation.framework；BDVRClient还使用到了网络状态检测功能，因此还需要引入SystemConfiguration.framework；为了生成设备UDID，需要引入Security.framework；为了支持gzip压缩，需要引入libz.1.dylib；网络模块需要引入CFNetwork.framework；某些场景需要获取设备地理位置以提高识别准确度，需引入CoreLocation.framework。

为了支持识别控件，需要引入OpenGLES.framework, QuartzCore.framework, GLKit.framework, CoreGraphics.framework和CoreText.framework。

添加方式：右键点击Xcode中的工程文件，在出现的界面中，选中TARGETS中应用，在出现的界面中选中Build Phase->Link Binary With Libraries，点击界面中的“+”图标，在弹出的界面中选择此7个Framework即可，添加完成效果图如图所示（libBDVoiceRecognition-Client.a将在随后添加）。

Link Binary With Libraries (15 items)		Status
	libBDVoiceRecognitionClientWithUI.a	Required ▼
	CoreGraphics.framework	Required ▼
	CFSocket.framework	Required ▼
	CoreLocation.framework	Required ▼
	OpenGLES.framework	Required ▼
	GLKit.framework	Required ▼
	CoreText.framework	Required ▼
	QuartzCore.framework	Required ▼
	Security.framework	Required ▼
	libz.1.dylib	Required ▼
	Foundation.framework	Required ▼
	UIKit.framework	Required ▼
	AudioToolbox.framework	Required ▼
	SystemConfiguration.framework	Required ▼
	AVFoundation.framework	Required ▼

图 4.1：图

4.1.3 引入BDVRClient的头文件

首先将BDVRClient提供的头文件拷贝到工程目录下，在XCode中添加此文件，引入BDVRClient提供的头文件。

如果使用识别UI，请添加如下头文件：

```
## import "BDRecognizerViewController.h"  
  
## import "BDRecognizerViewDelegate.h"
```

如果只使用识别接口，添加如下头文件：

```
## import "BDVoiceRecognitionClient.h"
```

如果要对音频数据或音频文件直接进行识别，请分别添加如下头文件：

```
## import "BDVRRawDataRecognizer.h"  
  
## import "BDVRFileRecognizer.h"
```

4.1.4 引入静态库文件

BDVRCClient提供了模拟器5.0及更新版本，真机armv7、armv7s和arm64四种环境所使用的静态库文件，分别存放在开发包的libBDVoiceRecognitionClient文件夹下，详细见“开发包说明”部分。引入前准备：静态库中采用Objective C++实现，因此需要保证工程中引用静态库头文件的实现文件的扩展名必须为.mm。

引入静态库文件的具体方式是：将libBDVoiceRecognitionClient文件夹下的libBDVoiceRecognitionClient.a采用添加文件方式添加到工程的Framework目录下，添加完成效果如图所示。

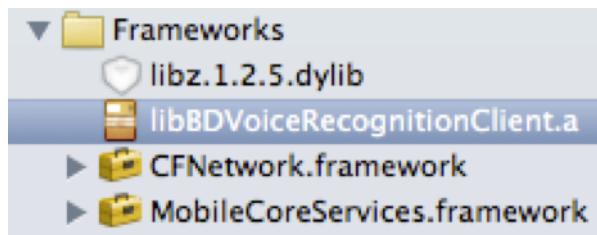


图 4.2：图

说明：libBDVoiceRecognitionClient.a是采用lipo命令将armv7, armv7s, arm64和模拟器Debug版的.a合并成的一个通用的.a文件，避免开发者在build不同target时频繁替换.a文件的问题。

4.1.5 添加第三方开源库

BDVRCClient中使用了第三方开源库，包括TTTAttributedLabel和苹果非官方的录音API，如果产品项目中已经包含其中的部分库，请勿重复添加，否则，请添加这三种第三方开源库到项目中，第三方库文件可以在SDK开发包下的Third-party目录下找到。由于SDK中使用了类别扩展，请在Build Setting中的Other Linker Flags中添加`-ObjC`。

注意：其中第三方库TTTAttributedLabel需要设置为ARC方式编译。

4.1.6 引入库所需的资源文件

将开发包中的BDVoiceRecognitionClientResources文件夹加入到工程中，具体的添加办法，将文件夹拷入工程文件中，以“create folder references for any adds”方式添加到工程的资源Group中，添加方式如下图所示。

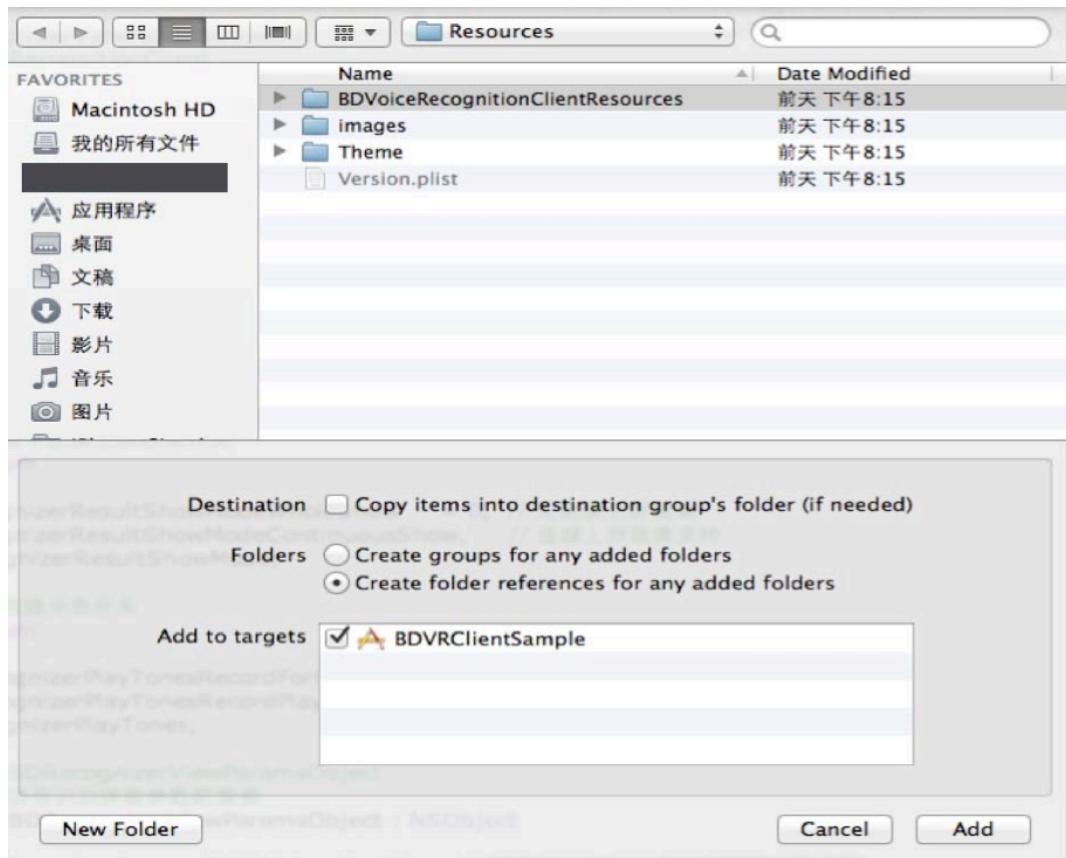


图 4.3：图

4.1.7 引入授权文件和数据文件

在识别过程中，需要将离线识别用到的授权文件和数据文件的访问路径传递给BDVRClient，因此，需要将开发包中License目录下的授权文件和Data目录下的数据文件加入到工程，或者放入程序的documents目录下，以便运行时访问。

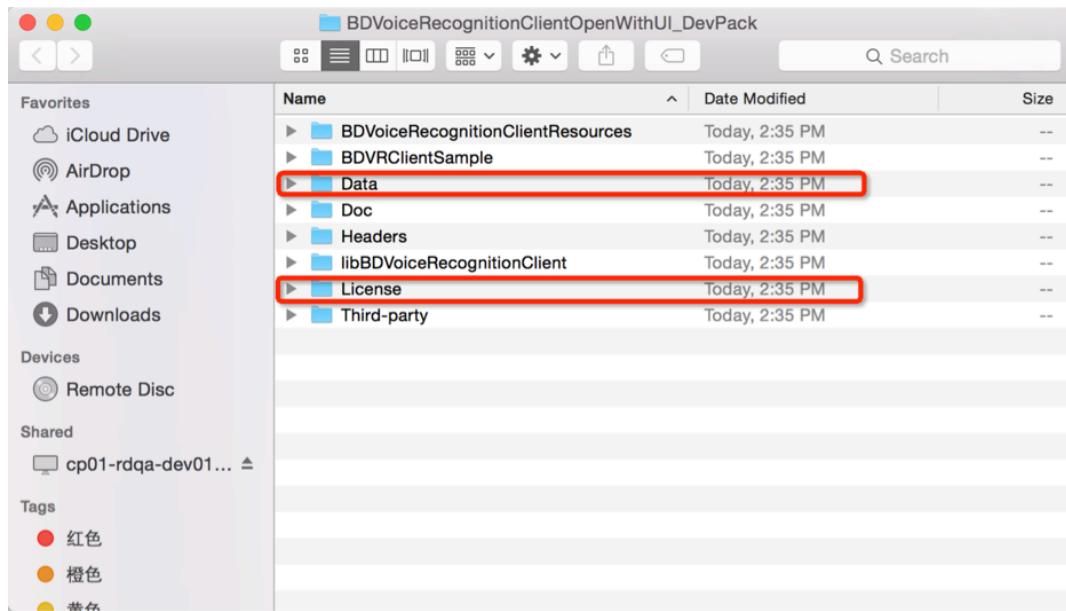


图 4.4：件

4.2 重要接口说明

百度语音SDK提供中文普通话、中文粤语、中文四川话、英语识别，支持输入、热词、地图、音乐、应用、web、购物、健康、打电话和视频等不同的应用场景。其中搜索、地图、音乐等模式适合短Query的输入场景，在识别过程中会将语气词、标点等过滤，尾点检测会更灵敏。输入模式适合短信输入等长句场景，尾点检测会迟钝一些，可以进行连续多句识别。开发者可以通过调用语音识别控件快速集成，也可以使用底层API接口自行设计语音交互。

SDK还支持语义理解能力，可以将用户的语音直接转换成需求意图。语义具有领域性特征，不属于任何领域的语义是不存在的。同样的语言，在不同的领域中所代表的含义可能截然不同。语义理解就是把语言在特定领域所代表语义通过计算机可处理的表示方式理解出来。具体支持的领域及数据格式请参考《百度语义理解协议》。

如果开启了语义解析能力，结果将做为JSON字符串放至在候选数组的首个元素。JSON对象的结构如下：

字段名	数据类型	描述
item	JSONArray	为 String 类型的 JSONArray，存储语音识别结果
json_res	String	语义解析结果，为Json格式的String

json_res转换成Json对象结构如下，详细参考《百度语义理解协议》。

字段名	数据类型	描述
raw_text	string	原始文本
parsed_text	string	分词结果
results	JSONArray	语义理解意图数组，可能为空

4.2.1 语音识别控件

BDVRCClient提供了语音识别控件BDRecognizerDialogController，并提供了更换主题功能，方便开发者快速引入语音识别功能。

创建识别控件对象

- 方法

```
(id)initWithOrigin:(CGPoint)origin withTheme:(BDTheme *)theme;
```

- 参数

参数	含义
origin	控件左上角的坐标
theme	控件的主题，如果为nil，则为默认主题

- 返回

弹窗实例

- 说明

第二个参数为识别控件的主题，如果参数为空，则为默认主题，主题可以从下面接口中任选其一，当选择为某一个主题，则将开发包中BDVoiceRecognitionClientResources/Theme中对应的主题Bundle添加到项目资源目录中。生成一个主题的方式如下：

```
“ ‘objective-c BDTheme *theme=[BDTheme defaultTheme]; // 获取默认皮肤
“ “
```

目前BDVRCClient的主题分为亮和暗两种系列，各分为红，橙，蓝，绿四种，开发者可以自行选用，具体选择的接口如下：

```
(instancetype) defaultTheme;  
(instancetype) lightBlueTheme;  
(instancetype) darkBlueTheme  
(instancetype) lightGreenTheme  
(instancetype) darkGreenTheme  
(instancetype) lightOrangeTheme  
(instancetype) darkOrangeTheme  
(instancetype) lightRedTheme  
(instancetype) darkRedTheme
```

因为程序一般只需要一套皮肤，因此开发者在使用中不用将Theme文件中的所有Bundle都拷入工程。

启动识别

- 方法

```
(BOOL)startWithParams:(BDRecognizerViewParamsObject *)params;
```

- 参数

参数	含义
params	识别过程的参数

- 返回

开始识别是否成功，成功为YES，否则为NO

- 说明

识别过程的参数，具体项目参考BDRecognizerViewParamsObject类声明，注意参数不能为空，必须要设置apiKey和secretKey。

取消本次识别

- 方法

```
(void)cancel;
```

- 参数

无。

- 返回

无。

- 说明

取消本次识别，并移除View。

调整控件坐标

- 方法

(void)changeFrameAfterOriented:(CGPoint)origin;

- 参数

参数	含义
origin	控件左上角的坐标

- 返回

无。

- 说明

设置识别弹出窗位置，一般在屏幕旋转后调用。

4.2.2 语音识别控件回调对象

BDVRCClient提供了语音识别控件回调对象BDRecognizerViewDelegate，通过实现该对象的接口，可以接收识别结果。

语音识别最终结果返回回调

- 方法

(void)onEndWithViews:(BDRecognizerViewController *)aBDRecognizerView withResults:(NSArray *)aResults;

- 参数

参数	含义
aBDRecognizerView	弹窗UI
aResults	返回结果

- 返回

无。

- 说明

语音识别完成会调用此函数。返回结果均为数组，搜索结果数组元素为整体识别结果，输入结果数组元素为字典。

录音数据返回

- 方法

```
(void)onRecordDataArrived:(NSData *)recordData sampleRate:(int)sampleRate;
```

- 参数

参数	含义
recordData	录音数据
sampleRate	采样率

- 返回

无。

- 说明

返回为原始录音数据，格式为pcm，累积拼接即可得到完整录音数据。

返回中间识别结果

- 方法

```
(void)onPartialResults:(NSString *)results;
```

- 参数

参数	含义
results	中间识别结果

- 返回

无。

- 说明

返回中间识别结果。

发生错误

- 方法

```
(void)onError:(int)errorCode;
```

- 参数

参数	含义
errorCode	错误码

- 返回

无。

- 说明

发生错误。

4.2.3 语音识别客户端

BDVRClient除了提供语音识别空间之外，还提供了API方式的调用接口：BDVoiceRecognitionClient。

创建语音识别客户对象

- 方法

```
(BDVoiceRecognitionClient *)sharedInstance;
```

- 参数

无。

- 返回

语音识别客户对象。

- 说明

创建的语音识别客户对象是个单例，不支持并发开启多次语音识别。

释放语音识别客户端对象

- 方法

```
(void)releaseInstance;
```

- 参数

无。

- 返回

无。

- 说明

释放语音识别客户端对象，完成资源释放。

判断是否可以录音

- 方法

```
(BOOL)isCanRecorder;
```

- 参数

无。

- 返回

可以录音返回YES，不可以录音返回NO。

- 说明

判断是否可以录音。

开始语音识别

- 方法

```
(int)startVoiceRecognition:(id<MVoiceRecognitionClientDelegate>)aDelegate;
```

- 参数

参数	含义
aDelegate	语音识别客户端回调对象

- 返回

错误码。

- 说明

开始语音识别。

主动说话结束

- 方法

```
(void)speakFinish;
```

- 参数

无。

- 返回

无。

- 说明

主动说话结束，调用该接口后将不再录音或者接收传入音频数据，但是已有音频数据将会继续识别完成。

结束语音识别

- 方法

```
(void)stopVoiceRecognition;
```

- 参数

无。

- 返回

无。

- 说明

结束语音识别，调用该接口后将不再录音或者接收传入音频数据，已经传入BDVRClient的音频数据也将被丢弃不再识别，返回当前已经识别的内容。

获取当前识别的采样率

- 方法

```
(int)getCurrentSampleRate;
```

- 参数

无。

- 返回

采样率。

- 说明

返回的采样率支持两种：16000和8000。

指定语音采样率

- 方法

```
(void)getCurrentSampleRate:(int)aSampleRate;
```

- 参数

参数	含义
aSampleRate	采样率

- 返回

无。

- 说明

指定本次语音识别的采样率，支持16000和8000。

设置识别类型列表

- 方法

```
(void)setPropertyList: (NSArray*)prop_list;
```

- 参数

参数	含义
prop_list	识别类型

- 返回

无。

- 说明

设置识别类型列表, 除EVoiceRecognitionPropertyInput和EVoiceRecognitionPropertySong外，其余类型可以复合。

获取当前识别类型列表

- 方法

```
(NSArray*)getRecognitionPropertyList;
```

- 参数

无。

- 返回

当前识别类型列表。

- 说明

获取当前识别类型列表。

设置播放提示音

- 方法

```
(BOOL)setPlayTone:(int)aTone isPlay:(BOOL)aIsPlay;
```

- 参数

参数	含义
aTone	提示音资源文件
aIsPlay	是否播放

- 返回

如果没有找到需要播放的资源文件，返回NO。

- 说明

声音资源需要加到项目工程里，用户可替换资源文件，文件名不可以变，建音提示音不宜过长，0。5秒左右。资源文件放置在BDVoiceRecognitionClientResources/Tone，录音开始声音文件名为record_start.caf，录音结束声音文件名为record_end.caf。

监听当前音量级别

- 方法

```
(BOOL)listenCurrentDBLevelMeter;
```

- 参数

无。

- 返回

如果再工作状态设定，返回NO。

- 说明

需要在开始识别前调用，如果在工作状态设定，返回结果为NO，且本次调用无效。

获得当前音量级别

- 方法

```
(int)getCurrentDBLevelMeter;
```

- 参数

无。

- 返回

当前音量值。

- 说明

获取当前音量级别，取值需要考虑全平台。

取消监听音量级别

- 方法

```
(void)cancelListenCurrentDBLevelMeter;
```

- 参数

无。

- 返回

无。

- 说明

取消监听音量级别。

设置识别语言

- 方法

```
(void)setLanguage:(int)language;
```

- 参数

识别语言。

- 返回

无。

- 说明

识别语言有效值参见枚举类型TVoiceRecognitionLanguage。

设置开发者申请的api key和secret key

- 方法

```
(void)setApiKey:(NSString *)apiKey withSecretKey:(NSString *)secretKey;
```

- 参数

参数	含义
apiKey	开发者在语音官网申请的apiKey
secretKey	开发者在语音官网申请的secretKey

- 返回

无。

- 说明

本接口仅为设置鉴权所需的keys，真正鉴权验证在首次识别时发起，故首次识别时间可能较长。

验证apiKey和secretKey

- 方法

```
(int)verifyApiKey:(NSString *)apiKey withSecretKey:(NSString *)secretKey;
```

- 参数

参数	含义
apiKey	开发者在语音官网申请的apiKey
secretKey	开发者在语音官网申请的secretKey

- 返回

返回值0代表验证通过，EVoiceRecognitionStartWorkGetAccessTokenFailed代表验证失败。

- 说明

调用此方法后，不再需要调用

```
(void)setApiKey:(NSString *)apiKey withSecretKey:(NSString *)secretKey;
```

此方法可以提前检查apikey、secretkey并获取验证数据，降低第一次识别需要等待的时间。

设置参数

- 方法

```
(void)setParamForKey:(NSString *)key withValue:(NSString *)value;
```

- 参数

参数	含义
key	参数类型
value	参数值

- 返回

无。

- 说明

支持参数列表详见。

关闭标点

- 方法

```
(void)disablePuncs:(BOOL)flag;
```

- 参数

参数	含义
flag	是否关闭标点

- 返回

无。

- 说明

关闭标点。

设置本地VAD开关

- 方法

```
(void)setLocalVAD:(BOOL)enable;
```

- 参数

参数	含义
enable	YES: 打开; NO: 不打开

- 返回

无。

- 说明

设置本地VAD开关。

设置服务器VAD开关

- 方法

```
(void)setServerVAD:(BOOL)enable;
```

- 参数

参数	含义
enable	YES: 打开; NO: 不打开

- 返回

无。

- 说明

设置服务器VAD开关。

设置是否使用压缩开关

- 方法

```
(void)setNeedCompressFlag: (BOOL)flag;
```

- 参数

参数	含义
flag	YES: 打开; NO: 不打开

- 返回

无。

- 说明

设置是否使用压缩开关。

设置是否使用AMR压缩开关

- 方法

```
(void)setUseAmrWhenWwanFlag: (BOOL)flag;
```

- 参数

参数	含义
flag	YES: 打开; NO: 不打开

- 返回

无。

- 说明

设置是否使用AMR压缩开关。在wwan网络下，如果使用16K采样率，可以设置此标志使用amr压缩。

是否进行车载环境下的噪声消除

- 方法

```
(void)setEnableDRCFlag:(BOOL)flag;
```

- 参数

参数	含义
flag	YES: 打开; NO: 不打开

- 返回

无。

- 说明

是否进行车载环境下的噪声消除。

设置是否获取原始json结果

- 方法

```
(void)setNeedOriginJsonResultFlag: (BOOL)flag;
```

- 参数

参数	含义
flag	YES: 打开; NO: 不打开

- 返回

无。

- 说明

设置是否获取原始json结果， 默认不开启， 设置后SDK不再进行解析（暂时只支持中间结果）。

判断是否开启获取原始json结果

- 方法

```
(BOOL)getNeedOriginJsonResultFlag;
```

- 参数

无。

- 返回

YES：开启获取原始JSON结果；NO：不开启获取原始JSON结果。

- 说明

判断是否开启获取原始json结果。

加载离线识别引擎

- 方法

```
(int)loadOfflineEngine: (NSString*)appCode  
                  license: (NSString*)licenseFile  
                 datFilePath: (NSString*)datFilePath  
                LMDatFilePath: (NSString*)LMDatFilePath  
                grammSlot: (NSDictionary*)dictSlot;
```

- 参数

参数	含义
appCode	用户获取的appCode
licenseFile	用户授权文件路径
datFilePath	识别模型文件路径
LMDatFilePath	导航使用的识别模型文件，没有可以设置为nil
dictSlot	垂类识别时，设置语法槽

- 返回

无。

- 说明

加载离线识别引擎。

设置在线识别时等待超时时间

- 方法

```
(void)setOnlineWaitTime:(NSTimeInterval)time;
```

- 参数

参数	含义
time	等待服务器反馈时间

- 返回

无。

- 说明

设置在线识别时等待超时时间。

是否正在进行离线识别

- 方法

```
(BOOL)isOfflineRecognition;
```

- 参数

无。

- 返回

正在使用离线引擎返回YES，否则返回NO。

- 说明

是否正在进行离线识别。

4.2.4 语音识别客户端回调对象

BDVRCClient提供了语音识别客户端回调对象MVoiceRecognitionClientDelegate，通过实现该对象的接口，可以接收识别结果。

识别状态变化回调

- 方法

```
(void)VoiceRecognitionClientWorkStatus:(int) aStatus obj:(id)aObj;
```

- 参数

参数	含义
aStatus	状态码
aObj	该状态码对应的对象

- 返回

无。

- 说明

在该接口中处理各种识别状态。

识别出错回调

- 方法

```
(void)VoiceRecognitionClientErrorStatus:(int) aStatus subStatus:(int)aSubStatus;
```

- 参数

参数	含义
aStatus	错误码类型
aSubStatus	具体错误码

- 返回

无。

- 说明

识别出错回调。

识别网络状态变化回调

- 方法 “ ‘objective-c (void)VoiceRecognitionClientNetWorkStatus:(int) aStatus;

“ ”

- 参数

参数	含义
aStatus	网络状态类型

- 返回

无。

- 说明

识别网络状态变化回调，目前该接口处于预留状态，并未实际使用。

4.3 参数和错误码说明

4.3.1 语音识别控件参数

语音识别控件参数列表详见类BDRecognizerViewParamsObject，此参数不能为空，否则将返回NO，即启动识别失败。其中apiKey和secretKey不能为非法值，否则将无法识别。

参数名称	参数类型	默认值	描述
apiKey	NSString		语音官网认证 API_key
secretKey	NSString		语音官网认证 Secret_key

参数名称	参数类型	默认值	描述
language	TBVoiceRecognitionLanguage	LANGUAGE_CHINESE	语种, 见 BDRecognizerViewParamsObject.h中定义的枚举
isNeedNLU	BOOL	false	是否回传语义理解结果, 仅在搜索模式下有效
recogPropList	NSArray		领域参数, 取值参见 BDVoiceRecognitionClient.h 中对应枚举类型 TBVoiceRecognitionResourceType
disablePuncs	BOOL	false	是否禁用标点
resultShowMode	TBDRecognizerResultShowMode	BDRecognizerResultShowModeContinuousShow	设置识别结果显示方式, 分为连续上屏和无连续上屏效果, 取值参见 BDRecognizerViewParamsObject.h 中定义的枚举
recordPlayTones	TBDRecognizerPlayTones	EVDRecognizerPlayTonesRecordPlay	提示音开关, 取值参见 BDRecognizerViewParamsObject.h 中定义的枚举
tipsTitle	NSString		提示语列表界面标题
tipsList	NSArray		提示语列表, 用于提示用户说出的话
isShowTipsOnStart	BOOL	YES	是否在第一次启动时打开提示语列表
isShowTipAfter3sSilence	BOOL	YES	是否在用户3秒未说话时浮出随机提示语
isShowHelpButtonWhenSilence	BOOL	YES	静音超时后将“取消”按钮替换为“帮助”
isNeedVad	BOOL	YES	是否需要进行语音端点检测

参数名称	参数类型	默认值	描述
isNeedCompress	BOOL	YES	是否对上传音频进行压缩

4.3.2 语音识别器参数

播放提示音 类型TVoiceRecognitionPlayTones，支持的参数如下：

参数	含义
EVoiceRecogniti onPlayTonesRe cStart	录音开始提示音
EVoiceRecogniti onPlayTonesRe cEnd	录音结束提示音
EVoiceRecogniti onPlayTonesAll	录音开始和结束提示音

语音识别类型 语音识别类型用TBDVoiceRecognitionProperty类型表示，支持参数如下：

参数	含义
EVoiceRecogniti onPropertyMus ic	音乐
EVoiceRecogniti onPropertyVid eo	视频
EVoiceRecogniti onPropertyApp	应用
EVoiceRecogniti onPropertyWeb	网页
EVoiceRecogniti onPropertySea rch	热词
EVoiceRecogniti onPropertyESh opping	购物
EVoiceRecogniti onPropertyHea lth	健康
EVoiceRecogniti onPropertyCall	打电话
EVoiceRecogniti onPropertySong	录歌识别
EVoiceRecogniti onPropertySha ke	摇一摇识台
EVoiceRecogniti onPropertyMed icalCare	医疗
EVoiceRecogniti onPropertyCar	汽车
EVoiceRecogniti onPropertyCat ering	餐饮娱乐
EVoiceRecogniti onPropertyFin anceAndE-conom ics	财经
EVoiceRecogniti onPropertyGame	游戏
EVoiceRecogniti onPropertyCoo kbook	菜谱
EVoiceRecogniti onPropertyAss istant	助手

参数	含义
EVoiceRecogniti onPropertyRec harge	话费充值
EVoiceRecogniti onPropertyMap	地图
EVoiceRecogniti onPropertyInp ut	输入
EVoiceRecogniti onPropertyCon tacts	100014
EVoiceRecogniti onPropertySet ting	100016
EVoiceRecogniti onPropertyTVI nstruction	100018
EVoiceRecogniti onPropertyPla yerInstruc- tion	100019
EVoiceRecogniti onPropertyRad io	100020

采样率 采样率用TVoiceRecognitionRecordSampleRateFlags类型表示，支持参数如下：

参数	含义
EVoiceRecogniti onRecordSampl eR- ateAuto	默认值，交给BDVRClient自己选择
EVoiceRecogniti onRecordSampl eRate8K	8000
EVoiceRecogniti onRecordSampl eRate16K	16000

语言 采样率用TVoiceRecognitionLanguage类型表示，支持参数如下：

参数	含义
EVoiceRecogniti onLanguageChi nese	中文普通话
EVoiceRecogniti onLanguageCan tonese	粤语
EVoiceRecogniti onLanguageEng lish	英语
EVoiceRecogniti onLanguageSic huanDi- ialect	四川话

4.3.3 语音识别状态

语音识别状态用TVoiceRecognitionClientWorkStatus表示，主要用来在

```
(void)VoiceRecognitionClientWorkStatus:(int) aStatus obj:(id)aObj;
```

中检测，支持的类型如下：

参数	含义
EVoiceRecogniti onClientWorkS tatusNone	空闲
EVoiceRecogniti onClientWorkP layStart- Tone	播放开始提示音
EVoiceRecogniti onClientWorkP layStart- ToneFinish	播放开始提示音完成
EVoiceRecogniti onClientWorkS tatusStart- Wor kIng	识别工作开始，开始采集及处理数据
EVoiceRecogniti onClientWorkS tatusStart	检测到用户开始说话
EVoiceRecogniti onClientWorkS tatusSen- tence End	输入模式下检测到语音说话完成
EVoiceRecogniti onClientWorkS tatusEnd	本地声音采集结束结束，等待识别结果返 回并结束录音
EVoiceRecogniti onClientWorkP layEnd- Tone	播放结束提示音
EVoiceRecogniti onClientWorkP layEnd- ToneFinish	播放结束提示音完成
EVoiceRecogniti onClientWorkS tatus- NewRecor dData	录音数据回调
EVoiceRecogniti onClientWorkS tatus- FlushData	连续上屏
EVoiceRecogniti onClientWorkS tatusRe- ceiveD ata	输入模式下有识别结果返回
EVoiceRecogniti onClientWorkS tatusFin- ish	语音识别功能完成，服务器返回正确结果
EVoiceRecogniti onClientWorkS tatusCan- cel	用户取消
EVoiceRecogniti onClientWorkS tatusError	发生错误

4.3.4 语音识别错误码

语音识别错误码用TVoiceRecognitionClientErrorStatus表示，支持的错误码如下：

参数	含义
EVoiceRecogniti onClientError_StatusUn- Know	未知错误(异常)
EVoiceRecogniti onClientError_Status- NoSpeech	用户未说话

参数	含义
EVoiceRecogniti onClientError StatusShort	用户说话声音太短
EVoiceRecogniti onClientError StatusEx-ception	语音前端库检测异常
EVoiceRecogniti onClientError Sta-tusChangeN otAvailable	录音设备不可用
EVoiceRecogniti onClientError StatusIntr-err uption	录音中断
EVoiceRecogniti onClientError NetWork-Status Unusable	网络不可用
EVoiceRecogniti onClientError NetWork-Status Error	网络发生错误
EVoiceRecogniti onClientError NetWork-Status TimeOut	网络本次请求超时
EVoiceRecogniti onClientError NetWork-Status ParseError	解析失败
EVoiceRecogniti onClientError NetWork-Status ServerParamEr ror	协议参数错误
EVoiceRecogniti onClientError NetWork-Status ServerRecognE rrор	识别过程出错
EVoiceRecogniti onClientError NetWork-Status ServerNoFindR esult	没有找到匹配结果
EVoiceRecogniti onClientError NetWork-Status ServerAppName UnknownError	鉴权失败
EVoiceRecogniti onClientError NetWork-Status ServerSpeechQ ualityProblem	声音不符合识别要求
EVoiceRecogniti onClientError NetWork-Status ServerSpeechT ooLong	语音过长
EVoiceRecogniti onClientError NetWork-Status ServerUnknown Error	未知错误
EVoiceRecogniti onClientError Offli-neEngine GetLicenseFai led	离线获取license失败
EVoiceRecogniti onClientError Offli-neEngine VerifyLicense Faild	离线验证license失败
EVoiceRecogniti onClientError Offli-neEngine DatFileNotExi st	指定的模型文件不存在

参数	含义
EVoiceRecogniti onClientError OfflineEngine SetSlotFailed	设置离线识别引擎槽失败
EVoiceRecogniti onClientError OfflineEngine InitializeFai led	初始化失败
EVoiceRecogniti onClientError OfflineEngine SetParamFailed	设置参数错误
EVoiceRecogniti onClientError OfflineEngine LMDataFileNot Exist	导航模型文件不存在
EVoiceRecogniti onClientError OfflineEngine SetPropertyFa iled	设置识别垂类失败
EVoiceRecogniti onClientError OfflineEngine FeedAudioData Failed	传入数据失败
EVoiceRecogniti onClientError OfflineEngine StopRecogniti onFailed	停止识别失败
EVoiceRecogniti onClientError OfflineEngine RecognizeFail ed	识别失败

4.4 完整示例

4.4.1 语音识别控件完整示例

语音识别控件完整示例如下：

```
- (IBAction)startRecognitionAction
{
    // 创建识别控件
    BDRecognizerViewController *tmpRecognizerViewController = [[BDRecognizerViewController alloc] initWithFrame:[BDVRSConfig sharedInstance].theme];

    // 全屏UI
    tmpRecognizerViewController.enableFullScreenMode = YES;

    // 设置委托
    tmpRecognizerViewController.delegate = self;
    self.recognizerViewController = tmpRecognizerViewController;
    [tmpRecognizerViewController release];

    // 设置识别参数
}
```

```
BDRecognizerViewParamsObject *paramsObject = [[BDRecognizerViewParamsObject alloc] init];

// 开发者信息，必须修改API_KEY和SECRET_KEY为在百度开发者平台申请得到的值，否则示例不能工作
paramsObject.apiKey = API_KEY;
paramsObject.secretKey = SECRET_KEY;

// 加载离线识别引擎
NSString* appCode = APPID;
NSString* datFilePath = [[NSBundle mainBundle] pathForResource:@"s_1" ofType:@""];
NSString* LMDatFilePath = nil;
if ([[BDVRSConfig sharedInstance].recognitionProperty intValue] == EVoiceRecognitionPropertyMap)
    LMDatFilePath = [[NSBundle mainBundle] pathForResource:@"s_2_Navi" ofType:@""];
} else if ([[BDVRSConfig sharedInstance].recognitionProperty intValue] == EVoiceRecognitionPropertyInput)
    LMDatFilePath = [[NSBundle mainBundle] pathForResource:@"s_2_InputMethod" ofType:@""];
}

NSDictionary* recogGrammSlot = @{@"$name_CORE" : @"张三\\n李四\\n",
                                 @"$song_CORE" : @"小苹果\\n朋友\\n",
                                 @"$app_CORE" : @"QQ\\n百度\\n微信\\n百度地图\\n",
                                 @"$artist_CORE" : @"刘德华\\n周华健\\n"};
int ret = [[BDVoiceRecognitionClient sharedInstance] loadOfflineEngine:appCode
                                                               license:nil
                                                               datFile:datFilePath
                                                               LMDatFile:LMDatFilePath
                                                               grammSlot:recogGrammSlot];

// 开始识别
[_recognizerViewController startWithParams:paramsObject];
[paramsObject release];
}

// 实现委托函数
- (void)onEndWithViews:(BDRecognizerViewController *)aBDRecognizerView withResults:(NSArray *)aResults
{
    _resultView.text = nil;

    if ([[BDVoiceRecognitionClient sharedInstance] getRecognitionProperty] !=
        EVoiceRecognitionPropertyInput)
    {
        // 搜索模式下的结果为数组，示例为
        // ["公园", "公元"]
    }
}
```

```
NSMutableArray *audioResultData = (NSMutableArray *)aResults;
NSMutableString *tmpString = [[NSMutableString alloc] initWithString:@""];
for (int i=0; i < [audioResultData count]; i++)
{
    [tmpString appendFormat:@"%@\\r\\n",[audioResultData objectAtIndex:i]];
}
_resultView.text = [_resultView.text stringByAppendingString:tmpString];
_resultView.text = [_resultView.text stringByAppendingString:@"\\n"];
[tmpString release];
}
else
{
    NSString *tmpString = [[BDVRSConfig sharedInstance] composeInputModeResult:aResults];
    _resultView.text = [_resultView.text stringByAppendingString:tmpString];
    _resultView.text = [_resultView.text stringByAppendingString:@"\\n"];
}
}
```

4.4.2 语音识别器完整示例

语音识别器完整示例如下：

```
// 开始语音识别
- (IBAction)startVoiceRecognitionAction
{
    // 设置鉴权信息
    [[BDVoiceRecognitionClient sharedInstance] setApiKey:API_KEY withSecretKey:SECRET_KEY];

    // 启动语音识别
    int startStatus = -1;
    startStatus = [[BDVoiceRecognitionClient sharedInstance] startVoiceRecognition:self];
    if (startStatus != EVoiceRecognitionStartWorking)
    {
        // 处理启动出错
    }
}

// 实现委托
- (void)VoiceRecognitionClientErrorStatus:(int) aStatus subStatus:(int)aSubStatus
{
```

```
// 处理出错
}

- (void)VoiceRecognitionClientWorkStatus:(int)aStatus obj:(id)aObj
{
    //处理网络状态变化
}

- (void)VoiceRecognitionClientNetWorkStatus:(int) aStatus
{
    // 处理网络状态变化
}
```


第5章

简介

5.0.3 概念解释

对本文中将提到的名词约定如下：

语音识别 (Automatic Speech Recognition, ASR)：也被称为自动语音识别，其目标是将人类的语言中的词汇内容转换为计算机可读的输入，例如按键、二进制编码或者字符序列。

自然语言理解 (Natural Language Understanding, NLU)：俗称人机对话，人工智能的分支学科。研究用电子计算机模拟人的语言交际过程，使计算机能理解和运用人类社会的自然语言如汉语、英语等，实现人机之间的自然语言通信，以代替人的部分脑力劳动，包括查询资料、解答问题、摘录文献、汇编资料以及一切有关自然语言信息的加工处理。

语音SDK (Baidu Voice Recognition Client, BDVRCClient)：即百度语音在线识别SDKAndroid版开发包，BDVRCClient是一个封装了语音采集、处理、网络收发等功能的语音识别解决方案。使用BDVRCClient可以在应用程序中快速集成语音识别功能。

5.0.4 功能介绍

BDVRCClient是运行在Android 平台的一体化语音识别解决方案，以JAR包 + SO库的形式发布，BDVRCClient支持下列功能：

- 语音识别控件：集成提示音、音量反馈动效整套交互的对话框控件，方便开发者快速集成；
- 基本功能：录音，语音数据处理，端点检测、网络通讯和状态通知，返回识别结果；
- 播放提示音：在录音前后播放提示音，优化用户体验；
- 监听语音能量：实时反馈用户当前说话声音能量强度；
- 语义理解：将语音识别成领域相关的语义结果。

5.0.5 兼容性

- **系统：**支持Android 2.3 (API Level 9) 及以上系统。需要开发者通过minSdkVersion来保证支持系统的检测。
- **机型：**手机和平板均可。
- **构架：**支持ARM平台、x86平台。
- **硬件要求：**要求设备上有麦克风。
- **网络：**支持WIFI及移动网络，支持2G、3G、4G移动网络。

5.0.6 开发包说明

文件/文件夹名	说明
/libs/	语音识别SDK lib库，包括各平台的SO库及Jar，SO库开发者可以按需集成
/res/raw/	语音识别对话框音效文件，如果不使用对话框可以不集成
/docs/	JAVADOC
百度语音在线识别SDK Android版开发手册	本手册
百度语义解析协议	语义理解意图表示说明
/VoiceRecognitionDemo/	Demo工程

5.1 集成指南

本章将讲解如何快速地集成BDVRCClient到现有应用中。一个完整的Demo请参考开发包中的示例程序VoiceRecognitionDemo。

5.1.1 创建应用

请参考《百度语音开放平台使用指南》创建应用，开通服务并完成个性化设置。

5.1.2 添加BDVRCClient到工程

将开发包中的libs目录整体拷贝到工程目录，libs目录包含了各平台的so库，开发者视应用需要可以进行删减。galaxy.jar是百度Android公共基础库，如果项目中还集成了其它百度SDK，如百度地图SDK，在打包过程中出现类似如下的错误信息：

```
[2015-10-22 11:02:57 - Dex Loader] Unable to execute dex: Multiple dex files define Lcom/baidu/android/common/logging/Configuration;
```

```
[2015-10-22 11:02:57 - VoiceRecognitionDemo] Conversion to Dalvik format failed: Unable to execute dex: Multiple dex files define Lcom/baidu/android/common/logging/Configuration;
```

请将此Jar包移除。

将开发包中的res目录整体拷贝到工程目录，此文件夹中包含语音识别对话框的音效文件，如果不使用可以忽略。

5.1.3 添加权限

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

各权限含义说明：

名称	用途
android.permission.RECORD_AUDIO	允许应用使用麦克风录音
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	获取当前的网络状态，优化录音及网络参数。
android.permission.ACCESS_WIFI_STATE	获取当前wifi状态，优化录音及网络参数
android.permission.READ_PHONE_STATE	获取用户手机的IMEI，用来唯一的标识用户。

5.1.4 Proguard配置

如果应用配置了代码混淆，需要在Proguard配置文件增加以下参数：

```
-keep class com.baidu.android.**{*;}
-keep class com.baidu.vipecognition.android.**{*;}
```

5.2 重要接口说明

5.2.1 识别器

识别器，类名VoiceRecognitionClient，主要用来控制识别进程：设置参数，开始，结束，取消等。

创建对象

- 说明

识别器仅支持单例模式，同一时刻只有开启一次识别，已经添加同步机制，线程安全。

- 方法

```
public VoiceRecognitionClient getInstance(Context context)
```

- 参数

名称	描述
context	上下文环境

- 返回

识别器单例对象。

设置授权

- 说明

传入创建应用信息时获得的key值，用于注册使用语音服务。

- 方法

```
public void setTokenApis (String apiKey, String secretKey)
```

- 参数

名称	描述
apiKey	开发者在开放平台上注册的apiKey
secretKey	开发者在开放平台上注册的secretKey

- 返回

无。

开始识别

- 说明

传入发起本次识别时的配置参数，用于确定该次识别的采样率、识别垂类等。

- 方法

```
public int startVoiceRecognition(final VoiceClientStatusChangeListener listener,  
                                VoiceRecognitionConfig config)
```

- 参数

名称	描述
listener	监听器
config	配置文件

- 返回

错误码。

说话结束

- 说明

说话结束，调用此接口后，将停止录音，但是已有录音数据将持续识别，直到结果返回。

- 方法

```
public void speakFinish ()
```

- 参数

无。

- 返回

无。

停止识别

- 说明

停止本次识别，调用此接口后，讲停止录音并停止识别，如果有录音数据正在识别，也将被丢弃。

- 方法

```
public void stopVoiceRecognition ()
```

- 参数

无。

- 返回

无。

释放对象

- 说明

释放语音识别器对象及其所占用的资源。

- 方法

```
public void releaseInstance ()
```

- 参数

无。

- 返回

无。

5.2.2 识别参数配置器

识别参数配置器，类名VoiceRecognitionConfig，主要用来设置各种识别参数。

创建识别参数配置器

- 说明

创建识别参数配置器，该配置器将用于传入识别参数。

- 方法

```
public VoiceRecognitionConfig()
```

- 参数

无。

- 返回

识别参数配置器对象。

设置采样率

- 说明

设置音频数据的采样率，支持8000和16000两种。

- 方法

```
public void setSampleRate(int rate)
```

- 参数

名称	描述
rate	采样率

- 返回

无。

启用语音识别开始音效

- 说明

启用语音识别开始音效，参数为音效文件的资源ID。

- 方法

```
public void enableBeginSoundEffect(int soundResourceId)
```

- 参数

名称	描述
soundResourceId	音效文件的资源ID

- 返回

无。

禁用语音识别开始音效

- 说明

禁用语音识别开始音效。

- 方法

```
public void disableBeginSoundEffect()
```

- 参数

无。

- 返回

无。

启用语音识别结束音效

- 说明

禁用语音识别开始音效。

- 方法

```
public void enableEndSoundEffect(int soundResourceId)
```

- 参数

名称	描述
soundResourceId	音效文件的资源ID

- 返回

无。

禁用语音识别结束音效

- 说明

禁用语音识别结束音效。

- 方法

```
public void disableEndSoundEffect()
```

- 参数

无。

- 返回

无。

禁用标点符号

- 说明

禁用标点符号。

- 方法

```
public void disablePunctuation()
```

- 参数

无。

- 返回

无。

启用语义理解

- 说明

启用语义理解功能。

- 方法

```
public void enableNLU()
```

- 参数

无。

- 返回

无。

启用通讯录识别

- 说明

启用通讯录识别。注意：在启用通讯录识别前要求先上传通讯录。

- 方法

```
public void enableContacts()
```

- 参数

无。

- 返回

无。

设置垂直领域

- 说明

• 设置识别的垂直领域，正确设置该领域有助于提高识别率，支持同时设置多个垂直领域，也支持只设置一个垂直领域，可以根据实际情况自行选择。

- 返回

无。

- 方法1

```
public void setProp(int[] arrayProp)
```

- 参数

名称	描述
arrayProp	识别垂类数组

- 方法2

```
public void setProp(int prop)
```

- 参数

名称	描述
prop	识别垂类

设置资源类型

- 说明

设置资源类型，主要用于区别后处理方式。

- 方法

```
public void setResourceType(int type)
```

- 参数

名称	描述
type	资源类型

- 返回

无。

设置是否使用缺省的录音

- 说明

设置是否使用缺省的录音。

- 方法

```
public void setUseDefault AudioSource(boolean useDefaultSource)
```

- 参数

名称	描述
useDefaultSource	true: 使用; false: 不使用

- 返回

无。

设置语种

- 说明

设置识别的语种。

- 方法

```
public void setLanguage(String language)
```

- 参数

名称	描述
language	语种

- 返回

无。

5.2.3 识别状态监听器

识别状态监听器，类名VoiceClientStatusChangeListener，是识别状态监听器接口，开发者可以通过实现此接口对识别状态做监听。

更新识别状态变化

- 说明

识别状态变化的回调函数。

- 方法

```
void onClientStatusChange(int status, Object obj)
```

- 参数

名称	描述
status	识别状态
obj	识别状态下的返回对象

- 返回

无。

更新网络状态变化

- 说明

网络状态变化的回调函数，该函数为保留接口，暂未使用。

- 方法

```
void onNetworkStatusChange(int status, Object obj)
```

- 参数

名称	描述
status	网络状态
obj	网络状态下的返回对象

- 返回

无。

出错

- 说明

出错时的回调函数。

- 方法

```
void onError(int type, Object errorCode)
```

- 参数

名称	描述
type	错误类型
errorCode	错误码

- 返回

无。

5.2.4 语音识别对话框

使用百度语音在线识别SDK中默认的对话框，可以方便快捷地使用百度语音识别功能。开发者只需要在创建对话框中传入合适的参数，之后注册对话框监听器来接收识别结果即可。

创建对话框

- 说明

创建一个语音识别的对话框，创建对话框前需要对要传给对话框的参数做设置。

- 方法

```
public BaiduASRDialog(Context context, Bundle params)
```

- 参数

名称	描述
context	应用上下文环境
params	参数

- 返回

百度语音识别对话框实例。

设置对话框监听器

- 说明

设置百度语音识别对话框的监听器。

- 方法

```
public void setDialogRecognitionListener(DialogRecognitionListener listener)
```

- 参数

名称	描述
listener	监听器

- 返回

无。

获取识别结果

- 说明

百度语音识别对话框监听器用来获取识别结果的回调函数。

- 方法

```
public void onResults(Bundle results)
```

- 参数

名称	描述
results	识别结果

- 返回

无。

5.3 参数和错误码说明

5.3.1 识别器参数说明

采样率 采样率是在录制音频时需要用到的参数，目前支持的类型如下：

参数	含义
SAMPLE_RATE_8K	8000采样率
SAMPLE_RATE_16K	16000采样率

语种或方言 可以通过设置语种或方言，支持不同地域的人做语音识别，支持类型如下：

参数	含义
LANGUAGE_CHINESE	中文普通话
LANGUAGE_CANTONESE	中文粤语
LANGUAGE_SICHUAN	中文四川话
LANGUAGE_ENGLISH	英语

语言模型垂类 在不同的应用场景下使用对应的语言模型垂类，可以有效提升识别率。目前支持的语言模型垂类如下：

参数	含义
PROP_MUSIC	音乐
PROP_VIDEO	视频
PROP_APP	应用
PROP_WEB	网页
PROP_SEARCH	热词
PROP_SHOPPING	购物
PROP_HEALTH	健康
PROP_PHONE	电话
PROP_SONG	歌曲
PROP_MEDICAL	医疗
PROP_CAR	汽车

参数	含义
PROP_CATERING	餐饮
PROP_FINANCE	金融
PROP_GAME	游戏
PROP_RECIPES	菜谱
PROP_ASSISTANT	助手
PROP_MAP	地图
PROP_INPUT	输入

5.3.2 开始识别返回错误码说明

参数	含义
START_WORK_RESULT_WORKING	正常启动就绪，可以识别
START_WORK_RESULT_NET_UNUSABLE	网络不可用，无法识别
START_WORK_RESULT_NULL_LISTENER	监听器位空，无法识别
START_WORK_RESULT_RECOGNITING	正在识别，不能开始新任务
START_WORK_RESULT_RELEASED	单例已被释放，不能识别

5.3.3 出错回调返回错误码说明

参数	含义
ERROR_CLIENT_UNKNOWN	未知错误
ERROR_CLIENT_NO_SPEECH	没有检测到语音
ERROR_CLIENT_TOO_SHORT	语音太短
ERROR_CLIENT_WHOLE_PROCESS_TIMEOUT	识别超时
ERROR_RECORDER_UNAVAILABLE	录音设备不可用
ERROR_RECORDER_INTERCEPTED	录音被中断
ERROR_NETWORK_UNUSABLE	网络不可用
ERROR_NETWORK_CONNECT_ERROR	网络连接失败
ERROR_NETWORK_PARSE_ERROR	解析错误
ERROR_SERVER_PARAMETER_ERROR	参数错误
ERROR_SERVER_BACKEND ER-ROR 后 端 错 误 ER-ROR SERVER_RECOGNITION_ERROR	识别错误

参数	含义
ERROR_SERVER_INVALID_APP_NAME_ERROR	鉴权失败
ERROR_SERVER_SPEECH_QUALITY_ERROR	语音质量未达标，无法识别
ERROR_SERVER_SPEECH_TOO_LONG_ERROR	语音过长

5.3.4 语音识别对话框参数说明

参数	说明	取值
PARAM_API_KEY	用户申请的api_key	用户从语音官网申请
PARAM_SECRET_KEY	用户申请的secret_key	用户从语音官网申请
PARAM_PROP	语言模型垂类选择（单个）	参见4.1.3
PARAM_PROP_ARRAY	语言模型垂类选择（数组）	参见4.1.3
PARAM_VAD_LIB	vad类型选择	MFE或LWVAD
PARAM_LANGUAGE	语种选择	参见4.1.2
PARAM_PROPMT_TEXT	提示语	用户可自定义
PARAM_TIPS_TONE_ENABLE	是否启用过程提示音	1/0：使用/弃用
PARAM_START_TONE_ENABLE	是否启用开始提示音	1/0：使用/弃用
PARAM_END_TONE_ENABLE	是否启用结束提示音	1/0：使用/弃用
PARAM_CONTACTS_ENABLE	是否使用通讯录	1/0：使用/弃用
PARAM_PUNCTUATION_ENABLE	是否使用标点符号	1/0：使用/弃用
PARAM_NLU_ENABLE	是否使用语义解析	1/0：使用/弃用
PARAM_IS_VAD	是否使用vad	1/0：使用/弃用
PARAM_IS_COMPRESS	是否使用压缩	1/0：使用/弃用

5.4 完整示例

5.4.1 自调用接口方式示例

```
// 重写监听器
class MyVoiceRecogListener implements VoiceClientStatusChangeListener {
    public void onClientStatusChange(int status, Object obj) {
        // 处理识别状态变化
    }

    public void onError(int errorType, int errorCode) {
```

```
// 处理出错异常
}

public void onNetworkStatusChange(int status, Object obj) {
    // 处理网络状态变化
}
}

// 获取识别器对象
VoiceRecognitionClient mASREngine = VoiceRecognitionClient.getInstance(this);
// 获取授权
mASREngine.setTokenApis(Constants.API_KEY, Constants.SECRET_KEY);
// 获得参数配置对象
VoiceRecognitionConfig config = new VoiceRecognitionConfig();
// 为识别器绑定监听器
MyVoiceRecogListener mListener = new MyVoiceRecogListener();
// 开始识别
int code = mASREngine.startVoiceRecognition(mListener, config);
// 判断是否正常开始
if (code != VoiceRecognitionClient.START_WORK_RESULT_WORKING) {
    // 处理出错异常
}
```

5.4.2 语音识别对话框模式示例

```
// 识别结果
final String recognition_result = " ";
// 参数，其中和为必须配置参数，其他根据实际需要配置apiKeysecretKey
Bundle params = new Bundle();
// 配置apkKey
params.putString(BaiduASRDigitalDialog.PARAM_API_KEY, your_api_key);
// 配置secretKey
params.putString(BaiduASRDigitalDialog.PARAM_SECRET_KEY, your_secret_key);
// 创建百度语音识别对话框
BaiduASRDigitalDialog mDialog = new BaiduASRDigitalDialog(this, params);
// 设置对话框回调监听器
mDialog.setDialogRecognitionListener(new DialogRecognitionListener(){
    // 识别结果处理函数
    public void onResults(Bundle arg0) {
        ArrayList<String> rs = results != null ? results
            .getStringArrayList(RESULTS_RECOGNITION) : null;
        if (rs != null && rs.size() > 0) {
            recognition_result = rs.get(0);
        }
    }
});

// 显示对话框
mDialog.show();
```


第6章

简介

6.0.3 概念解释

对本文中将提到的概念约定如下：

语音识别 (Automatic Speech Recognition, ASR)：也被称为自动语音识别，其目标是将人类的语音中的词汇内容转换为计算机可读的输入，例如按键、二进制编码或者字符序列。

自然语言理解Natural Language Understanding (NLU)：俗称人机对话，人工智能的分支学科。研究用电子计算机模拟人的语言交际过程，使计算机能理解和运用人类社会的自然语言如汉语、英语等，实现人机之间的自然语言通信，以代替人的部分脑力劳动，包括查询资料、解答问题、摘录文献、汇编资料以及一切有关自然语言信息的加工处理。

语音识别SDK (BDVRClient)：即本开发包，文中简称为BDVRClient。BDVRClient是一个封装了语音采集、处理、网络收发等功能的语音识别解决方案。借助BDVRClient可以快速地在应用程序中集成语音识别功能。

应用程序：在开发中使用了BDVRClient，具有语音识别功能的产品线产品。

开发者：在应用程序中使用BDVRClient的开发人员。

6.0.4 功能介绍

百度语音识别IOS版本SDK (以下简称BDVRClient) 是一种面向IOS移动设备的语音识别解决方案，以静态库方式提供。通过该方案，开发者可以轻松构建出功能丰富、交互性强的语音识别应用程序。BDVRClient支持下列功能：

- 基本功能：录音、语音数据处理、端点检测、网络通讯、状态通知、返回文字结果；
- 语音识别控件：集成提示音、音量反馈动效整套交互的对话框控件，方便开发者快速集成；
- 播放提示音：在录音前后播放提示音，优化用户体验；

- 监听语音音量：实时反馈用户当前说话声音能量强度；
- 语义理解：将语音识别成领域相关的语义结果。
- 本文档适用于对IOS应用开发有基本了解的开发人员。

6.0.5 兼容性

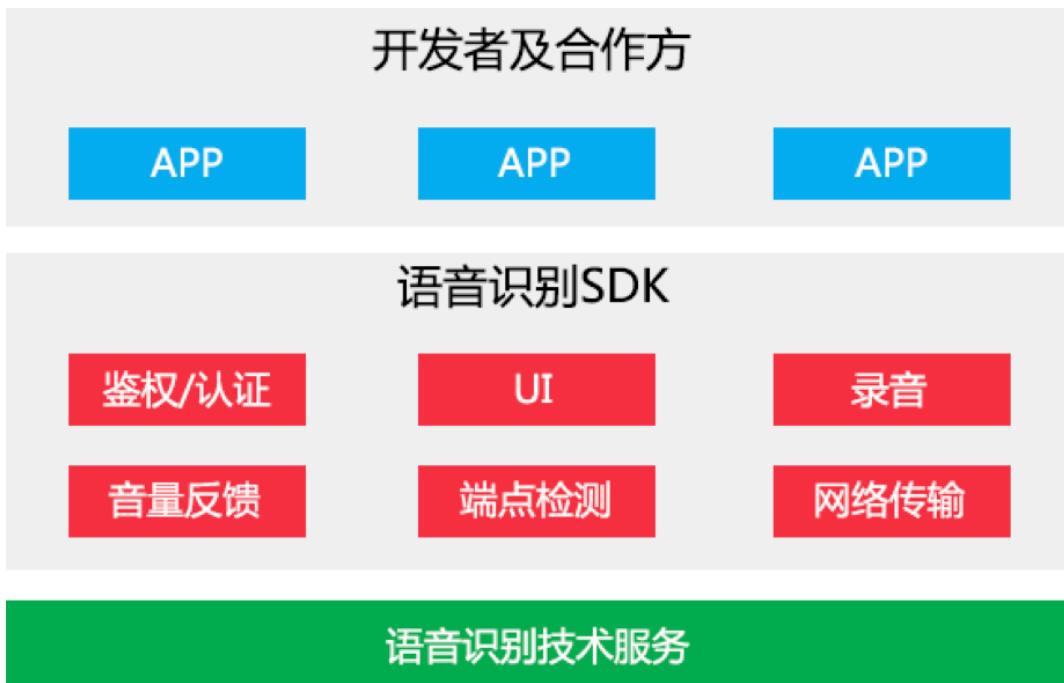
- 系统：支持iOS 6.0及以上系统，支持Bitcode功能。
- 架构：armv7、armv7s、arm64、i386、x86_64。
- 机型：iPhone 4+，iPad 2+和iPod 5+。
- 硬件要求：需要有麦克风，用于支持语音录入。
- 网络：支持NET、Wifi网络环境。

6.0.6 开发包说明

一级目录	二级目录	说明
Headers	BDVoiceRecognit ion-Client.h	BDVRClient无UI头文件
Headers	BDRecognizerVie wController.h	BDVRClient UI头文件
Headers	BDRecognizerVie wDele-gate.h	BDVRClient UI结果回调接口头文件
Headers	BDRecognizerVie wParam-sObject .h	BDVRClient UI中启动参数头文件
Headers	BDTheme.h	BDVRClient UI主题头文件
Headers	BDVRRawDataReco gnizer.h	BDVRClient音频数据识别头文件
Headers	BDVRFileRecogni zer.h	BDVRClient音频文件识别头文件
BDVRClientSample	SDK Demo源代码	开发示例
Third-party	各种第三方库	需要添加到项目中的第三方库
libBDVoiceRecog nition-Client (语音识别库静态库)	libBDVoiceRecog nition-Client.a	通用库，合并了真机armv7、armv7s、arm64和模拟器版的库
BDVoiceRecognit ionClientReso urces (语音识别库资源文件)	Tone	提示音资源文件
	Theme	识别控件主题

一级目录	二级目录	说明
Doc (使用文档)	百度语音识别iOS版开发手册	开发者使用指南

6.0.7 总体框图



6.1 集成指南

从本章开始将进行Step-By-Step的讲解，介绍如何在应用工程中集成BDVRClient。一个完整的Demo请参考开发包中的示例程序BDVRClientSample。

6.1.1 创建应用

请参考《百度语音开放平台使用指南》创建应用，开通服务并完成个性化设置。

6.1.2 引入编译需要的Framework

BDVRClient包含的功能以及需要的Framework有：

功能	framework名称
录音和播放	AudioToolbox.framework
	AVFoundation.framework

功能	framework名称
网络状态检测	SystemConfiguration.framework
判断当前网络连接类型 (2G/3G/4G)	CoreTelephony.framework
生成设备UDID	Security.framework
支持gzip压缩	libz.1.dylib
网络模块	CFNetwork.framework
获取设备地理位置以提高识别准确度	CoreLocation.framework
支持识别控件	OpenGL ES.framework
	QuartzCore.framework
	GLKit.framework
	CoreGraphics.framework
	CoreText.framework

添加方式：右键点击Xcode中的工程文件，在出现的界面中，选中TARGETS中应用，在出现的界面中选中Build Phase->Link Binary With Libraries，点击界面中的“+”图标，在弹出的界面中选择此7个Framework即可（libBDVoiceRecognitionClient.a将在随后添加），添加完成效果图如下所示。

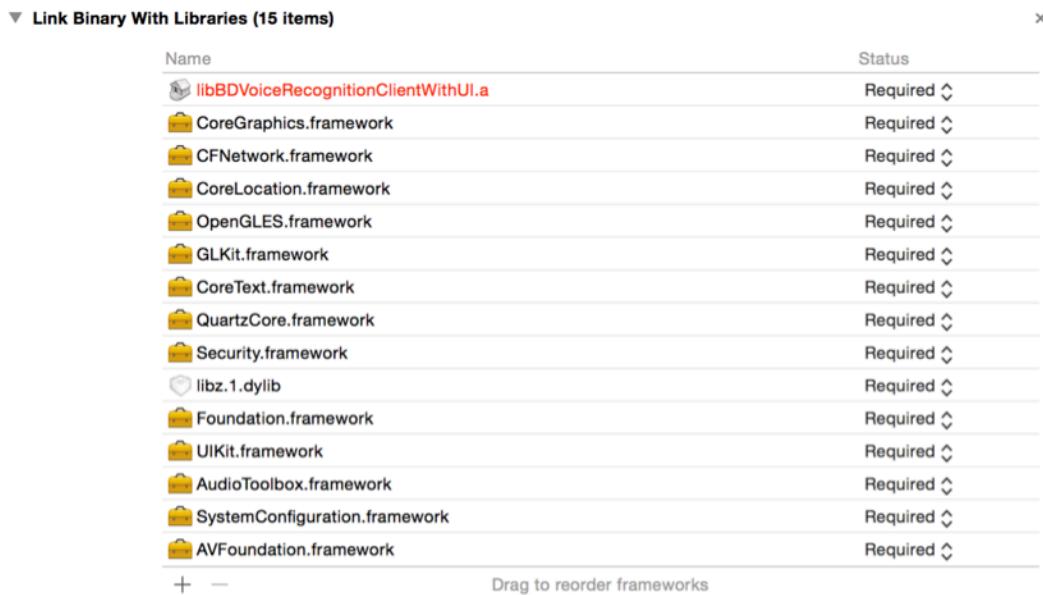


图 6.1：图

6.1.3 引入BDVRCClient的头文件

首先将BDVRCClient提供的头文件拷贝到工程目录下，在XCode中添加此文件，引入BDVRCClient提供的头文件。

如果使用识别UI，请添加如下头文件：

```
## import "BDRecognizerViewController.h"
```

```
## import "BDRecognizerViewDelegate.h"
```

如果只使用识别接口，添加如下头文件：

```
## import "BDVoiceRecognitionClient.h"
```

如果要对音频数据或音频文件直接进行识别，请分别添加如下头文件：

```
## import "BDVRRawDataRecognizer.h"
```

```
## import "BDVRFileRecognizer.h"
```

6.1.4 引入静态库文件

BDVRClient提供了模拟器6.0及更新版本，真机armv7、armv7s和arm64四种环境所使用的静态库文件，分别存放在开发包的libBDVoiceRecognitionClient文件夹下，详细见“开发包说明”部分。

引入前准备：静态库中采用Objective C++实现，因此需要保证工程中引用静态库头文件的实现文件的扩展名必须为.mm。

引入静态库文件的具体方式是：将libBDVoiceRecognitionClient文件夹下的libBDVoiceRecognitionClient.a采用添加文件方式添加到工程的Framework目录下，添加完成效果如图所示。

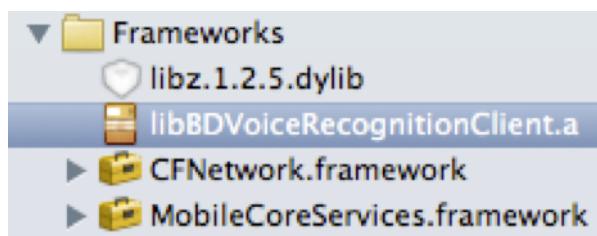


图 6.2: 图

说明：libBDVoiceRecognitionClient.a是采用lipo命令将armv7, armv7s, arm64和模拟器Debug版的.a合并成的一个通用的.a文件，避免开发者在build不同target时频繁替换.a文件的问题。

6.1.5 添加第三方开源库

BDVRCClient中使用了第三方开源库，包括TTTAttributedLabel和苹果非官方的录音API，如果产品项目中已经包含其中的部分库，请勿重复添加，否则，请添加这两种第三方开源库到项目中，第三方库文件可以在SDK开发包下的Third-party目录下找到。注意：由于SDK中使用了类别，应用需要在Build Settings的Other Linker Flags中添加-ObjC；第三方库TTTAttributedLabel需要设置为ARC方式编译。

6.1.6 引入库所需的资源文件

将开发包中的BDVoiceRecognitionClientResources文件夹加入到工程中，具体的添加办法，将文件夹拷入工程文件中，以“create folder references for any adds”方式添加到工程的资源Group中，添加方式如下图所示。

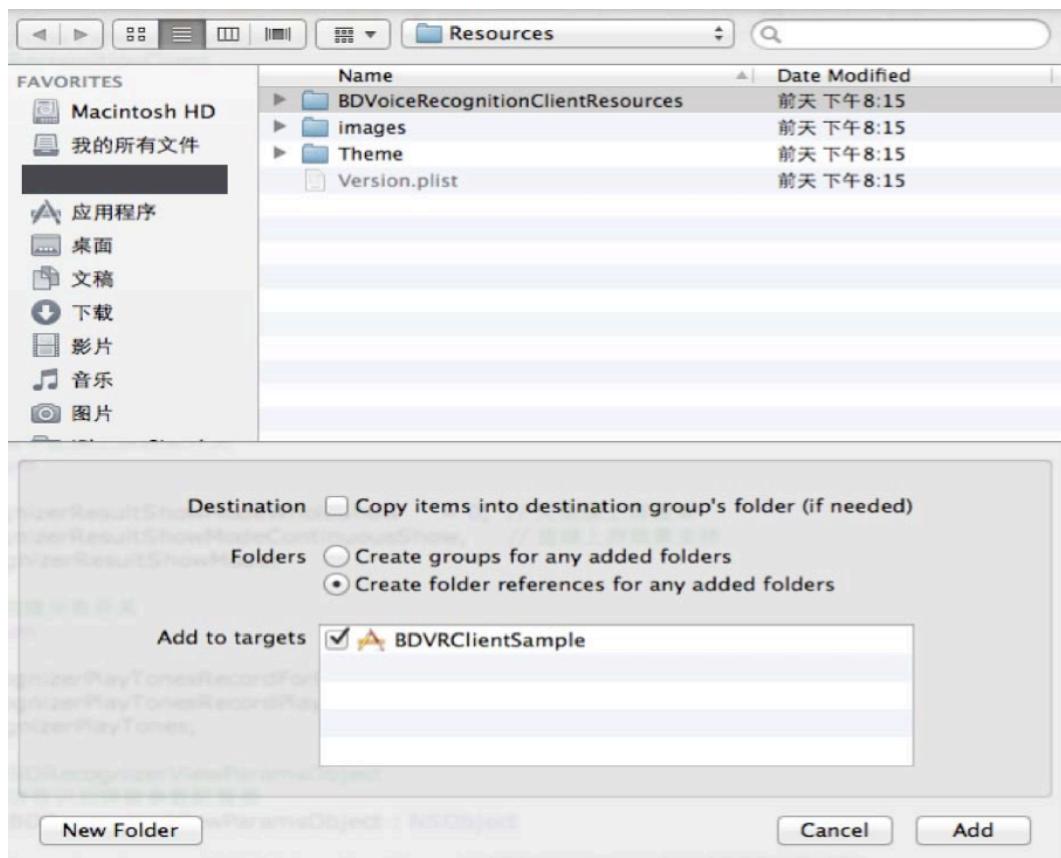


图 6.3：式

6.2 重要接口说明

百度语音SDK提供中文普通话、中文粤语、中文四川话、英语识别，支持输入、热词、地图、音乐、应用、web、购物、健康、打电话和视频等不同的应用场景。其中搜索、地图、音乐等模式适合短Query的输入场景，在识别过程中会将语气词、标点等过滤，尾点检测会更灵敏。输入模式适合短信输入等长句场景，尾点检测会迟钝一些，可以进行连续多句识别。开发者可以通过调用语音识别控件快速集成，也可以使用底层API接口自行设计语音交互。

SDK还支持语义理解能力，可以将用户的语音直接转换成需求意图。语义具有领域性特征，不属于任何领域的语义是不存在的。同样的语言，在不同的领域中所代表的含义可能截然不同。语义理解就是把语言在特定领域所代表语义通过计算机可处理的表示方式理解出来。具体支持的领域及数据格式请参考《百度语义理解协议》。

如果开启了语义解析能力，结果将做为JSON字符串放至在候选数组的首个元素。JSON对象的结构如下：

字段名	数据类型	描述
item	JSONArray	为 String 类型的 JSONArray，存储语音识别结果
json_res	String	语义解析结果，为Json格式的String

json_res转换成Json对象结构如下，详细参考《百度语义理解协议》。

字段名	数据类型	描述
raw_text	string	原始文本
parsed_text	string	分词结果
results	JSONArray	语义理解意图数组，可能为空

6.2.1 语音识别控件

BDVRClient提供了语音识别控件BDRecognizerDialogController，并提供了更换主题功能，方便开发者快速引入语音识别功能。

创建识别控件对象

- 方法

```
(id)initWithOrigin:(CGPoint)origin withTheme:(BDTheme *)theme;
```

- 参数

参数	含义
origin	控件左上角的坐标
theme	控件的主题，如果为nil，则为默认主题

- 返回

弹窗实例

- 说明

第二个参数为识别控件的主题，如果参数为空，则为默认主题，主题可以从下面接口中任选其一，当选择为某一个主题，则将开发包中BDVoiceRecognitionClientResources/Theme中对应的主题Bundle添加到项目资源目录中。生成一个主题的方式如下：

```
BDTheme *theme=[BDTheme defaultTheme]; // 获取默认皮肤
```

目前BDVRClient的主题分为亮和暗两种系列，各分为红，橙，蓝，绿四种，开发者可以自行选用，具体选择的接口如下：

```
(instancetype) defaultTheme;  
(instancetype) lightBlueTheme;  
(instancetype) darkBlueTheme  
(instancetype) lightGreenTheme  
(instancetype) darkGreenTheme  
(instancetype) lightOrangeTheme  
(instancetype) darkOrangeTheme  
(instancetype) lightRedTheme  
(instancetype) darkRedTheme
```

因为程序一般只需要一套皮肤，因此开发者在使用中不用将Theme文件中的所有Bundle都拷入工程。

启动识别

- 方法

```
(BOOL)startWithParams:(BDRecognizerViewParamsObject *)params;
```

- 参数

参数	含义
params	识别过程的参数

- 返回

开始识别是否成功，成功为YES，否则为NO

- 说明

识别过程的参数，具体项目参考BDRecognizerViewParamsObject类声明，注意参数不能为空，必须要设置apiKey和secretKey。

取消本次识别

- 方法

```
(void)cancel;
```

- 参数

无。

- 返回

无。

- 说明

取消本次识别，并移除View。

调整控件坐标

- 方法

```
(void)changeFrameAfterOriented:(CGPoint)origin;
```

- 参数

参数	含义
origin	控件左上角的坐标

- 返回

无。

- 说明

设置识别弹出窗位置，一般在屏幕旋转后调用。

6.2.2 语音识别控件回调对象

BDVRCClient提供了语音识别控件回调对象BDRecognizerViewDelegate，通过实现该对象的接口，可以接收识别结果。

语音识别最终结果返回回调

- 方法

```
(void)onEndWithViews:(BDRecognizerViewController *)aBDRecognizerView withResults:(NSArray *)aResults;
```

- 参数

参数	含义
aBDRecognizerView	弹窗UI
aResults	返回结果

- 返回

无。

- 说明

语音识别完成会调用此函数。返回结果均为数组，搜索结果数组元素为整体识别结果，输入结果数组元素为字典。

录音数据返回

- 方法

```
(void)onRecordDataArrived:(NSData *)recordData sampleRate:(int)sampleRate;
```

- 参数

参数	含义
recordData	录音数据
sampleRate	采样率

- 返回

无。

- 说明

返回为原始录音数据，格式为pcm，累积拼接即可得到完整录音数据。

返回中间识别结果

- 方法

```
(void)onPartialResults:(NSString *)results;
```

- 参数

参数	含义
results	中间识别结果

- 返回

无。

- 说明

返回中间识别结果。

发生错误

- 方法

```
(void)onError:(int)errorCode;
```

- 参数

参数	含义
errorCode	错误码

- 返回

无。

- 说明

发生错误。

6.2.3 语音识别客户端

BDVRClient除了提供语音识别空间之外，还提供了API方式的调用接口：BDVoiceRecognitionClient。

创建语音识别客户对象

- 方法

```
(BDVoiceRecognitionClient *)sharedInstance;
```

- 参数

无。

- 返回

语音识别客户对象。

- 说明

创建的语音识别客户对象是个单例，不支持并发开启多次语音识别。

释放语音识别客户端对象

- 方法

```
(BDVoiceRecognitionClient *)sharedInstance;
```

- 参数

无。

- 返回

无。

- 说明

释放语音识别客户端对象，完成资源释放。

判断是否可以录音

- 方法

```
(BOOL)isCanRecorder;
```

- 参数

无。

- 返回

可以录音返回YES，不可以录音返回NO。

- 说明

判断是否可以录音。

开始语音识别

- 方法

```
(int)startVoiceRecognition:(id<MVoiceRecognitionClientDelegate>)aDelegate;
```

- 参数

参数	含义
aDelegate	语音识别客户端回调对象

- 返回

错误码。

- 说明

开始语音识别。

主动说话结束

- 方法

```
(void)speakFinish;
```

- 参数

无。

- 返回

无。

- 说明

主动说话结束，调用该接口后将不再录音或者接收传入音频数据，但是已有音频数据将会继续识别完成。

结束语音识别

- 方法

```
(void)stopVoiceRecognition;
```

- 参数

无。

- 返回

无。

- 说明

结束语音识别，调用该接口后将不再录音或者接收传入音频数据，已经传入BDVRClient的音频数据也将被丢弃不再识别，返回当前已经识别的内容。

获取当前识别的采样率

- 方法

```
(int)getCurrentSampleRate;
```

- 参数

无。

- 返回

采样率。

- 说明

返回的采样率支持两种：16000和8000。

指定语音采样率

- 方法

```
(void)setRecordSampleRate:(int)aSampleRate;
```

- 参数

参数	含义
aSampleRate	采样率

- 返回

无。

- 说明

指定本次语音识别的采样率，支持16000和8000。

设置识别类型列表

- 方法

```
(void)setPropertyList: (NSArray*)prop_list;
```

- 参数

参数	含义
prop_list	识别类型

- 返回

无。

- 说明

设置识别类型列表，除EVoiceRecognitionPropertyInput和EVoiceRecognitionPropertySong外，其余类型可以复合。

获取当前识别类型列表

- 方法

```
(NSArray*)getRecognitionPropertyList;
```

- 参数

无。

- 返回

当前识别类型列表。

- 说明

获取当前识别类型列表。

设置播放提示音

- 方法

```
(BOOL)setPlayTone:(int)aTone isPlay:(BOOL)aIsPlay;
```

- 参数

参数	含义
aTone	提示音资源文件
aIsPlay	是否播放

- 返回

如果没有找到需要播放的资源文件，返回NO。

- 说明

声音资源需要加到项目工程里，用户可替换资源文件，文件名不可以变，建音提示音不宜过长，0。5秒左右。资源文件放置在BDVoiceRecognitionClientResources/Tone，录音开始声音文件名为record_start.caf，录音结束声音文件名为record_end.caf。

监听当前音量级别

- 方法

```
(BOOL)listenCurrentDBLevelMeter;
```

- 参数

无。

- 返回

如果再工作状态设定，返回NO。

- 说明

需要在开始识别前调用，如果在工作状态设定，返回结果为NO，且本次调用无效。

获得当前音量级别

- 方法

```
(int)getCurrentDBLevelMeter;
```

- 参数

无。

- 返回

当前音量值。

- 说明

获取当前音量级别，取值需要考虑全平台。

取消监听音量级别

- 方法

```
(void)cancelListenCurrentDBLevelMeter;
```

- 参数

无。

- 返回

无。

- 说明

取消监听音量级别。

设置识别语言

- 方法

```
(void)setLanguage:(int)language;
```

- 参数

识别语言。

- 返回

无。

- 说明

识别语言有效值参见枚举类型TVoiceRecognitionLanguage。

设置开发者申请的api key和secret key

- 方法

```
(void)setApiKey:(NSString *)apiKey withSecretKey:(NSString *)secretKey;
```

- 参数

参数	含义
apiKey	开发者在语音官网申请的apiKey
secretKey	开发者在语音官网申请的secretKey

- 返回

无。

- 说明

本接口仅为设置鉴权所需的keys，真正鉴权验证在首次识别时发起，故首次识别时间可能较长。

验证apiKey和secretKey

- 方法

```
(int)verifyApiKey:(NSString *)apiKey withSecretKey:(NSString *)secretKey;
```

- 参数

参数	含义
apiKey	开发者在语音官网申请的apiKey
secretKey	开发者在语音官网申请的secretKey

- 返回

返回值0代表验证通过，EVoiceRecognitionStartWorkGetAccessTokenFailed代表验证失败。

- 说明

调用此方法后，不再需要调用

```
(void)setApiKey:(NSString *)apiKey withSecretKey:(NSString *)secretKey;
```

此方法可以提前检查apikey、secretkey并获取验证数据，降低第一次识别需要等待的时间。

设置参数

- 方法

```
(void)setParamForKey:(NSString *)key withValue:(NSString *)value;
```

- 参数

参数	含义
key	参数类型
value	参数值

- 返回

无。

- 说明

支持参数列表详见。

关闭标点

- 方法

```
(void)disablePuncs:(BOOL)flag;
```

- 参数

参数	含义
flag	是否关闭标点

- 返回

无。

- 说明

关闭标点。

设置本地VAD开关

- 方法

```
(void)setLocalVAD:(BOOL)enable;
```

- 参数

参数	含义
enable	YES: 打开; NO: 不打开

- 返回

无。

- 说明

设置本地VAD开关。

设置服务器VAD开关

- 方法

```
(void)setServerVAD:(BOOL)enable;
```

- 参数

参数	含义
enable	YES: 打开; NO: 不打开

- 返回

无。

- 说明

设置服务器VAD开关。

设置是否使用压缩开关

- 方法

```
(void)setNeedCompressFlag: (BOOL)flag;
```

- 参数

参数	含义
flag	YES: 打开; NO: 不打开

- 返回

无。

- 说明

设置是否使用压缩开关。

设置是否使用AMR压缩开关

- 方法

```
(void)setUseAmrWhenWwanFlag: (BOOL)flag;
```

- 参数

参数	含义
flag	YES: 打开; NO: 不打开

- 返回

无。

- 说明

设置是否使用AMR压缩开关。在wwan网络下，如果使用16K采样率，可以设置此标志使用amr压缩。

是否进行车载环境下的噪声消除

- 方法

```
(void)setEnableDRCFlag:(BOOL)flag;
```

- 参数

参数	含义
flag	YES: 打开; NO: 不打开

- 返回

无。

- 说明

是否进行车载环境下的噪声消除。

设置是否获取原始json结果

- 方法

```
(void)setNeedOriginJsonResultFlag: (BOOL)flag;
```

- 参数

参数	含义
flag	YES: 打开; NO: 不打开

- 返回

无。

- 说明

设置是否获取原始json结果， 默认不开启， 设置后SDK不再进行解析（暂时只支持中间结果）。

判断是否开启获取原始json结果

- 方法

```
(BOOL)getNeedOriginJsonResultFlag;
```

- 参数

无。

- 返回

YES: 开启获取原始JSON结果； NO: 不开启获取原始JSON结果。

- 说明

判断是否开启获取原始json结果。

6.2.4 语音识别客户端回调对象

BDVRClient提供了语音识别客户端回调对象MVoiceRecognitionClientDelegate，通过实现该对象的接口，可以接收识别结果。

识别状态变化回调

- 方法

```
(void)VoiceRecognitionClientWorkStatus:(int) aStatus obj:(id)aObj;
```

- 参数

参数	含义
aStatus	状态码
aObj	该状态码对应的对象

- 返回

无。

- 说明

在该接口中处理各种识别状态。

识别出错回调

- 方法

```
(void)VoiceRecognitionClientErrorStatus:(int) aStatus subStatus:(int)aSubStatus;
```

- 参数

参数	含义
aStatus	错误码类型
aSubStatus	具体错误码

- 返回

无。

- 说明

识别出错回调。

识别网络状态变化回调

- 方法

```
(void)VoiceRecognitionClientNetworkStatus:(int) aStatus;
```

- 参数

参数	含义
aStatus	网络状态类型

- 返回

无。

- 说明

识别网络状态变化回调，目前该接口处于预留状态，并未实际使用。

6.3 参数和错误码说明

6.3.1 语音识别控件参数

语音识别控件参数列表详见类BDRecognizerViewParamsObject，此参数不能为空，否则将返回NO，即启动识别失败。其中apiKey和secretKey不能为非法值，否则将无法识别。

参数名称	参数类型	默认值	描述
apiKey	NSString		语音官网认证 API_key
secretKey	NSString		语音官网认证 Secret_key
language	TBVoiceRecognitionLanguage	LANGUAGE_CHINESE	语种, 见 BDRecognizerViewParamsObject.h 中定义的枚举
isNeedNLU	BOOL	false	是否回传语义理解结果, 仅在搜索模式下有效
recogPropList	NSArray		领域参数, 取值参见 BDVoiceRecognitionClient.h 中对应枚举类型 TBVoiceRecognitionResourceType
disablePuncs	BOOL	false	是否禁用标点
resultShowMode	TBDRecognizerResultShowMode	BDRecognizerResultShowModeContinuousShow	设置识别结果显示方式, 分为连续上屏和无连续上屏效果, 取值参见 BDRecognizerViewParamsObject.h 中定义的枚举
recordPlayTones	TBDRecognizerPlayTones	EBDRecognizerPlayTonesRecordPlay	提示音开关, 取值参见 BDRecognizerViewParamsObject.h 中定义的枚举
tipsTitle	NSString		提示语列表界面标题
tipsList	NSArray		提示语列表, 用于提示用户说出的话
isShowTipsOnStart	BOOL	YES	是否在第一次启动时打开提示语列表
isShowTipAfter3sSilence	BOOL	YES	是否在用户3秒未说话时浮出随机提示语

参数名称	参数类型	默认值	描述
isShowHelpButtonWhenSilence	BOOL	YES	静音超时后将“取消”按钮替换为“帮助”
isNeedVad	BOOL	YES	是否需要进行语音端点检测
isNeedCompress	BOOL	YES	是否对上传音频进行压缩

6.3.2 语音识别器参数

播放提示音 类型TVoiceRecognitionPlayTones，支持的参数如下：

参数	含义
EVoiceRecognitionPlayTonesRecStart	录音开始提示音
EVoiceRecognitionPlayTonesRecEnd	录音结束提示音
EVoiceRecognitionPlayTonesAll	录音开始和结束提示音

语音识别类型 语音识别类型用TBDVoiceRecognitionProperty类型表示，支持参数如下：

参数	含义
EVoiceRecognitionPropertyMusic	音乐
EVoiceRecognitionPropertyVideo	视频
EVoiceRecognitionPropertyApp	应用
EVoiceRecognitionPropertyWeb	网页
EVoiceRecognitionPropertySearch	热词
EVoiceRecognitionPropertyShopping	购物
EVoiceRecognitionPropertyHealth	健康
EVoiceRecognitionPropertyCall	打电话
EVoiceRecognitionPropertySong	录歌识别
EVoiceRecognitionPropertyShake	摇一摇识台
EVoiceRecognitionPropertyMedicalCare	医疗
EVoiceRecognitionPropertyCar	汽车
EVoiceRecognitionPropertyCatering	餐饮娱乐

参数	含义
EVoiceRecogniti onPropertyFin anceAndEconom ics	财经
EVoiceRecogniti onPropertyGame	游戏
EVoiceRecogniti onPropertyCoo kbook	菜谱
EVoiceRecogniti onPropertyAss istant	助手
EVoiceRecogniti onPropertyRec harge	话费充值
EVoiceRecogniti onPropertyMap	地图
EVoiceRecogniti onPropertyInp ut	输入

采样率 采样率用TVoiceRecognitionRecordSampleRateFlags类型表示，支持参数如下：

参数	含义
EVoiceRecogniti onRecordSampl eR ateAuto	默认值，交给BDVRClient自己选择
EVoiceRecogniti onRecordSampl eRate8K	8000
EVoiceRecogniti onRecordSampl eRate16K	16000

语言 采样率用TVoiceRecognitionLanguage类型表示，支持参数如下：

参数	含义
EVoiceRecogniti onLanguageChi nese	中文普通话
EVoiceRecogniti onLanguageCan tonese	粤语
EVoiceRecogniti onLanguageEng lish	英语
EVoiceRecogniti onLanguageSic huanDi alect	四川话

6.3.3 语音识别状态

语音识别状态用TVoiceRecognitionClientWorkStatus表示，主要用来在

```
(void)VoiceRecognitionClientWorkStatus:(int) aStatus obj:(id)aObj;
```

中检测，支持的类型如下：

参数	含义
EVoiceRecogniti onClientWorkS tatusNone	空闲
EVoiceRecogniti onClientWorkP layStart- Tone	播放开始提示音
EVoiceRecogniti onClientWorkP layStart- ToneFinish	播放开始提示音完成
EVoiceRecogniti onClientWorkS tatusStart- Wor kIng	识别工作开始，开始采集及处理数据
EVoiceRecogniti onClientWorkS tatusStart	检测到用户开始说话
EVoiceRecogniti onClientWorkS tatusSen- tence End	输入模式下检测到语音说话完成
EVoiceRecogniti onClientWorkS tatusEnd	本地声音采集结束结束，等待识别结果返 回并结束录音
EVoiceRecogniti onClientWorkP layEnd- Tone	播放结束提示音
EVoiceRecogniti onClientWorkP layEnd- ToneFinish	播放结束提示音完成
EVoiceRecogniti onClientWorkS tatus- NewRecor dData	录音数据回调
EVoiceRecogniti onClientWorkS tatus- FlushData	连续上屏
EVoiceRecogniti onClientWorkS tatusRe- ceiveD ata	输入模式下有识别结果返回
EVoiceRecogniti onClientWorkS tatusFin- ish	语音识别功能完成，服务器返回正确结果
EVoiceRecogniti onClientWorkS tatusCan- cel	用户取消
EVoiceRecogniti onClientWorkS tatusError	发生错误

6.3.4 语音识别错误码

语音识别错误码用TVoiceRecognitionClientErrorStatus表示，支持的错误码如下：

参数	含义
EVoiceRecogniti onClientError_StatusUn- Know	未知错误(异常)
EVoiceRecogniti onClientError_Status- NoSpeech	用户未说话

参数	含义
EVoiceRecogniti onClientError StatusShort	用户说话声音太短
EVoiceRecogniti onClientError StatusEx-ception	语音前端库检测异常
EVoiceRecogniti onClientError Sta-tusChangeNotAvailable	录音设备不可用
EVoiceRecogniti onClientError StatusIntr-err uption	录音中断
EVoiceRecogniti onClientError NetWork-Status Unusable	网络不可用
EVoiceRecogniti onClientError NetWork-Status Error	网络发生错误
EVoiceRecogniti onClientError NetWork-Status TimeOut	网络本次请求超时
EVoiceRecogniti onClientError NetWork-Status ParseError	解析失败
EVoiceRecogniti onClientError NetWork-Status ServerParamEr ror	协议参数错误
EVoiceRecogniti onClientError NetWork-Status ServerRecognE rrор	识别过程出错
EVoiceRecogniti onClientError NetWork-Status ServerNoFindR esult	没有找到匹配结果
EVoiceRecogniti onClientError NetWork-Status ServerAppName UnknownError	鉴权失败
EVoiceRecogniti onClientError NetWork-Status ServerSpeechQ ualityProblem	声音不符合识别要求
EVoiceRecogniti onClientError NetWork-Status ServerSpeechT ooLong	语音过长
EVoiceRecogniti onClientError NetWork-Status ServerUnknown Error	未知错误

6.4 完整示例

6.4.1 语音识别控件完整示例

语音识别控件完整示例如下：

```
- (IBAction)startRecognitionAction
{
    // 创建识别控件
    BDRecognizerViewController *tmpRecognizerViewController = [[BDRecognizerViewController alloc] initWithFrame:[BDVRSConfig sharedInstance].theme];

    // 全屏UI
    tmpRecognizerViewController.enableFullScreenMode = YES;

    // 设置委托
    tmpRecognizerViewController.delegate = self;
    self.recognizerViewController = tmpRecognizerViewController;
    [tmpRecognizerViewController release];

    // 设置识别参数
    BDRecognizerViewParamsObject *paramsObject = [[BDRecognizerViewParamsObject alloc] init];

    // 开发者信息，必须修改API_KEY和SECRET_KEY为在百度开发者平台申请得到的值，否则示例不能工作
    paramsObject.apiKey = API_KEY;
    paramsObject.secretKey = SECRET_KEY;

    // 开始识别
    [_recognizerViewController startWithParams:paramsObject];
    [paramsObject release];
}

// 实现委托函数
- (void)onEndWithViews:(BDRecognizerViewController *)aBDRecognizerView withResults:(NSArray *)aResults
{
    _resultView.text = nil;

    if ([[BDVoiceRecognitionClient sharedInstance] getRecognitionProperty] != EVoiceRecognitionPropertyInput)
    {
        // 搜索模式下的结果为数组，示例为
        // ["公园", "公元"]
        NSMutableArray *audioResultData = (NSMutableArray *)aResults;
        NSMutableString *tmpString = [[NSMutableString alloc] initWithString:@""];
        for (int i=0; i < [audioResultData count]; i++)
        {
            [tmpString appendFormat:@"%@", [audioResultData objectAtIndex:i]];
        }
    }
}
```

```
_resultView.text = [_resultView.text stringByAppendingString:tmpString];
_resultView.text = [_resultView.text stringByAppendingString:@"\\n"];

[tmpString release];
}
else
{
    NSString *tmpString = [[BDVRSConfig sharedInstance] composeInputModeResult:aResults];

    _resultView.text = [_resultView.text stringByAppendingString:tmpString];
    _resultView.text = [_resultView.text stringByAppendingString:@"\\n"];
}
}
```

6.4.2 语音识别器完整示例

语音识别器完整示例如下：

```
// 开始语音识别
- (IBAction)startVoiceRecognitionAction
{
    // 设置鉴权信息
    [[BDVoiceRecognitionClient sharedInstance] setApiKey:API_KEY withSecretKey:SECRET_KEY];

    // 启动语音识别
    int startStatus = -1;
    startStatus = [[BDVoiceRecognitionClient sharedInstance] startVoiceRecognition:self];
    if (startStatus != EVoiceRecognitionStartWorking)
    {
        // 处理启动出错
    }
}

// 实现委托
- (void)VoiceRecognitionClientErrorStatus:(int) aStatus subStatus:(int)aSubStatus
{
    // 处理出错
}

- (void)VoiceRecognitionClientWorkStatus:(int)aStatus obj:(id)aObj
{
    // 处理网络状态变化
}
```

```
- (void)VoiceRecognitionClientNetWorkStatus:(int) aStatus
{
    // 处理网络状态变化
}
```

第7章

常见问题

更多常见问题已经迁移至论坛，见：<http://yuyin.baidu.com/bbs/question/stick>

Q：语音识别REST API支持的音频格式、采样率有哪些？A：支持的压缩格式有：pcm（不压缩）、wav、opus、speex、amr、x-flac。原始PCM的录音参数必须符合8k/16k采样率、16bit位深、单声道。

Q：语音识别 REST API 最长支持多长的录音？A：最长支持60s的录音文件。对文件大小没有限制，只对时长有限制。

Q：语音识别SDK支持的音频格式、采样率有哪些？A：Android SDK：支持Android 2.2及以上系统，支持ARM\X86架构。安装包最小增加200k，支持8k/16k采样率，支持pcm格式。

iOS SDK：支持iOS 5.0及以上系统，支持ARM-v7\ARM-v7s\ARM64\i386\x86_64架构。安装包最小增加400k，支持8k/16k采样率，支持pcm格式。

Q：什么是语音识别REST API？有什么注意事项？A：语音识别全平台REST API，采用http方式请求，可适用于任何平台的语音识别。使用REST API，录音、压缩及上传模块需要自行开发。且REST API语音识别暂时不支持语义解析。

Q：百度语音是否单独提供录音功能，以及语音通讯功能？A：不单独提供独立录音的功能，目前识别SDK包含语音输入和识别功能。语音通讯功能需开发者自行搭建。

Q：语音识别REST API和SDK的区别是什么？A：REST API：开发者上传录音——百度语音进行识别——识别结果返回开发者SDK：百度语音提供从录音到识别结果返回的整体解决方案

Q：语音识别通过哪个接口获取音频信息？A：Android SDK：当VoiceClientStatusChangeListener.onClientStatusChange(int status, Object obj)的status是VoiceRecognitionClient.CLIENT_STATUS_AUDIO_DATA时，obj为byte[]音频数据；

iOS SDK：MVoiceRecognitionClientDelegate的(void)VoiceRecognitionClientWorkStatus:(int)aStatus obj:(id)aObj;的aStatus是EVoiceRecognitionClientWorkStatusNewRecordData时，aObj为NSData音频数据。数据格式均为pcm，采样率可通过VoiceRecognitionConfig.getSampleRate()/

[[BDVoiceRecognitionClient sharedInstance] getCurrentSampleRate]获取，获取的音频格式是位深16bit，单声道。

Q：如何才能提高语音识别的准确率？A：自定义语音识别设置：打开百度开放云平台，在目前创建的应用下进行自定义语音识别设置。上传识别关键词文本，保存并生效。设置对应的语音识别垂类：在代码中填写识别的垂类领域，识别结果将优先指向已设置的垂类，可以使得识别结果更准确

Q：百度语音SDK与其他百度SDK，或其他第三方SDK冲突时，如何解决？A：与其他百度SDK冲突一般是由于使用了相同的基础库galaxy.jar，请检查是否重复导入了该jar包；与其他第三方SDK冲突一般是由于so库的架构不统一，请保证工程libs目录下armeabi/armeabi-v7a/x86/mips目录的so库均一致，如果不能保证一致，则一般只能所有SDK仅使用armeabi架构的so库

Q：首次开启语音识别功能的延迟时间较长，需要如何调控？A：首次延迟时间较长一般是由于权限验证造成，可以通过预先调用接口：(int)verifyApiKey:(NSString)apiKey withSecretKey:(NSString)secretKey; 来进行验证。首次开启语音时就不需要再发送验证请求，从而降低语音识别启动的延迟。

Q：如何在语音识别时获取音频文件？A：Api：cp服务器自己备存用户录音信息 Sdk：可以通过CLIENT_STATUS_AUDIO_DATA回调，将回调对象强转成byte[]顺序写入到文件中即可得到原始的音频文件。

Q：语音识别垂类是什么意思？A：语音识别垂类就是特指某个领域的意思，开发者可以根据用户使用场景设定特定领域使得识别结果更准确。比如垂直领域设定为”音乐“，则在语音识别时会优先识别为音乐相关的歌曲名歌手名或者歌词。

Q：为什么语音识别Demo的API_Key和Secret_Key是空的还能进行识别呢？我开发的应用的API_Key和Secret_Key是不是必须得填写呢？A：SDK自带demo用的特殊的方式验证的，故可以不使用AK和SK。开发者自行开发的应用AK和SK是必须填写的，否则无法调用语音识别。

Q：【Android纯在线】如何从识别SDK中获取音频？A：通过监听onClientStatusChange函数的CLIENT_STATUS_AUDIO_DATA消息（在一次会话中会回调多次）拼接音频，如：

```
class MyVoiceRecogListener implements VoiceClientStatusChangeListener{
    OutputStream outFile;
    @Override
    public void onClientStatusChange(int status, Object obj) {
        switch(status) {
        ... // 省略其它消息
        case VoiceRecognitionClient.CLIENT_STATUS_AUDIO_DATA:
            // 有音频数据输出
            if (obj!= null && obj instanceof byte[]) {
                // 演示如何保持PCM音频，细节请各位工程师更加实际情况优化
                byte[] buf = (byte[])obj;
                FileOutputStream out = new FileOutputStream("sdcard/your_file.pcm", true);

```

```
    out.write(buf);
    out.close();
}
break;
default:
break;
}
}
```

Q: 【Android离在线】如何从识别SDK中获取音频？ A: 方式1：设置outfile参数可以指定语音数据的保存路径，设置方式如：

```
intent.putExtra("outfile", "/sdcard/your_audio.pcm");
```

语音数据的保存格式为PCM，播放和压缩请自行查找相关类库。

方式2：通过监听onBufferReceived(byte[] buf)回调，拼接音频实现

第8章

概念解释

对本文中将提到的名词约定如下：

语音合成：也被称为文本转换技术（TTS），它是将计算机自己产生的、或外部输入的文字信息转变为可以听得懂的、流利的口语输出的技术。

应用程序：在开发中调用了API、具有语音合成功能的产品线产品。

开发者：想在应用程序中使用API、正在阅读本文档的开发人员。

8.1 简介

百度语音合成通过REST API的方式给开发者提供一个通用的HTTP接口，基于该接口，开发者可以轻松的获取语音合成能力，本文档描述了使用语音合成服务REST API的方法。

8.1.1 功能介绍

REST API支持对一段文本的合成，其中对于文本格式以及参数有一定的要求。请按照文档中描述进行相应请求操作。

本文档适用使用HTTP接口的开发人员。

8.2 集成指南

8.2.1 获取 Access Token

使用语音识别 REST API 需要获取 Access Token。Access Token 是用户身份验证和授权的凭证，语音识别采用的是Client Credentials授权方式，即采用应用公钥、密钥获取Access

Token，适用于任何带server类型应用，通过此授权方式获取Access Token仅可访问平台授权类的接口，详见百度 OAuth 授权 “[Client Credentials授权](#)” 部分。

使用Client Credentials获取Access Token需要应用在其服务端发送请求（推荐用POST方法）到百度OAuth2.0授权服务的“<https://openapi.baidu.com/oauth/2.0/token>”地址上，并带上以下参数：

- grant_type: 必须参数，固定为“client_credentials”；
- client_id: 必须参数，应用的 API Key；
- client_secret: 必须参数，应用的 Secret Key；

例如：

```
https://openapi.baidu.com/oauth/2.0/token?  
grant_type=client_credentials&  
client_id=Va5yQRHl*****LT0vuXV4&  
client_secret= 0rDSjzQ20XUj5i*****PQSzr5pVw2&
```

响应数据包如下所示，其中“access_token”字段即为请求REST API所需的令牌，默认情况下，Access Token有效期为一个月，开发者需要对Access Token的有效性进行判断，如果Access Token过期可以重新获取。

例如：

```
HTTP/1.1 200 OK  
Content-Type: application/json  
Cache-Control: no-store  
  
{  
    "access_token": "1.a6b7dbd428f731035f771b8d*****.86400.1292922000-2346678-124328",  
    "expires_in": 86400,  
    "refresh_token": "2.385d55f8615fdfd9edb7c4b*****.604800.1293440400-2346678-124328",  
    "scope": "public",  
    "session_key": "ANXxSNjwQDugf86150nqeik*****Cd1Lxn",  
    "session_secret": "248APvxvjCZ0VEC*****aK4oZExMB",  
}
```

8.3 API请求方式基本说明

语音合成接口支持 POST 和 GET两种方式 正式地址：<http://tsn.baidu.com/text2audio>

8.3.1 文本上传模式

- 上传参数

参数	可需	描述
tex	必填	合成的文本，使用UTF-8编码，请注意文本长度必须小于1024字节
lan	必填	语言选择,填写zh
tok	必填	开放平台获取到的开发者 access_token
ctp	必填	客户端类型选择，web端填写1
cuid	必填	用户唯一标识，用来区分用户，填写机器MAC地址或IMEI码，长度为60以内
spd	选填	语速，取值0-9，默认为5中语速
pit	选填	音调，取值0-9，默认为5中语调
vol	选填	音量，取值0-9，默认为5中音量
per	选填	发音人选择，取值0-1, 0为女声，1为男声，默认为女声

8.3.2 GET调用方式

将所有的参数都填写到URL地址中，可以通过浏览器可以播放合成的语音结果。

http://tsn.baidu.com/text2audio?tex=***&lan=zh&cuid=***&ctp=1&tok=***

8.3.3 POST调用方式

将文本以及其他参数写入到body里面，利用表单的方式将参数传递到服务端。调用地址为<http://tsn.baidu.com/text2audio>，所有的参数都在body中。body里面的数据为：

tex=***&lan=zh&cuid=***&ctp=1&tok=***

注意：不论是将参数放到URL地址中，还是利用表单的方式放入到http body中，参数都需要经过两次 URLENCODE 操作。如果是直接在浏览器地址栏输入则不需要。

8.3.4 下行接口定义

如果合成成功，下行数据为二进制语音文件，具体header信息 Content-Type: audio/mp3；如果合成出现错误，则会返回json结果，具体header信息为：Content-Type:application/json。其中sn数据主要用于DEBUG追查问题，如果出现问题，可以提供sn帮助确认问题。

错误实例为：

```
{"err_no":500,"err_msg":"notsupport.","sn":"abcdefgh","idx":1}
```

8.3.5 错误码解释

错误码	含义
500	不支持输入
501	输入参数不正确
502	token验证失败
503	合成后端错误

8.4 注意事项

请严格按照文档里描述的参数进行开发。请注意以下几个问题：

1. 合成文本长度必须小于1024字节，如果本文长度较长，可以采用多次请求的方式。切忌不可文本长度超过限制。
2. 语音合成 rest api初次申请默认请求数配额 200000次/天，如果默认配额不能满足需求，请申请提高配额。
3. 必填字段中，严格按照文档描述中内容填写。

第9章

简介

9.0.1 概念解释

语音合成是实现人机语音交互，建立一个有听和讲能力的交互系统所必需的关键技术。随着语音技术的发展，百度自主研发了语音合成系统（TTS），功能是接收用户发送的文本，生成语音发送给用户。对本文中将提到的名词约定如下：

语音合成（Text To Speech，TTS）：将文本合成为语音，即声音文件。

合成引擎：将文本合成为语音的核心模块。

百度语音合成SDK（BDTTSClient）：本开发包。BDTTSClient是一个封装了网络首发、音频播放功能的语音合成解决方案。借助BDTTSClient可以在应用程序中快速集成语音合成功能。

9.0.2 功能介绍

百度语音合成客户端Android版SDK是一种面向Android移动设备的语音合成解决方案，以JAR包 + SO库的形式发布。目前版本已支持SDK内部直接播放合成语音和从SDK获取语音数据，并支持男女声、语速、音调、音量、音频码率设置，后续版本将支持更多参数设置（最新信息请参见百度语音官网）。

9.0.3 兼容性

系统：支持Android 2.3（API Level 9）及以上系统。需要开发者通过minSdkVersion来保证支持系统的检测。

机型：手机和平板均可。

构架：支持arm64-v8a, armeabi, armeabi-v7a, x86, x86_64。

网络：支持WIFI及移动网络，支持2G、3G、4G移动网络。

9.0.4 开发包说明

文件(夹)名	说明
doc/百度语音离在线合成Android版开发手册.pdf	本文档
doc/javaDoc	Javadoc
libs	语音合成SDK Lib库，包括各平台的so库及Jar包
BaiduTtsSample	开发示例(Eclipse project)
data	语音合成资源文件
license	授权文件

9.1 集成指南

本章将讲解如何快速地集成BDTTSClient到现有应用中。

9.1.1 创建应用

请参考《百度语音开放平台使用指南》创建应用，开通服务并完成个性化设置。

9.1.2 添加BDTTSClient到工程

将开发包中的libs目录整体拷贝到工程目录，libs目录包含了各平台的so库，开发者视应用需要可以进行删减。galaxy_lite.jar是百度Android公共基础库，如果项目中还集成了其它百度SDK，如Push.jar，在打包过程中出现类似如下的错误信息：

```
[2015-10-22 11:02:57 - Dex Loader] Unable to execute dex: Multiple dex files define Lcom/baidu/android/common/logging/Configuration;  
[2015-10-22 11:02:57 - BaiduTtsSample] Conversion to Dalvik format failed: Unable to execute dex: Multiple dex files define Lcom/baidu/android/common/logging/Configuration;
```

请将此Jar包移除。

如果Eclipse ADT版本插件低于17，需要手工添加依赖库，添加方法为：Project => Properties => Java Build Path => Libraries => Add JAR…。

9.1.3 添加语音合成资源文件

将开发包中的data目录下的dat文件放到某一可读路径下，以便设置资源文件参数时使用。

9.1.4 添加权限

在工程AndroidManifest.xml文件中添加如下权限：

```
<uses-permission android:name=" android.permission.INTERNET" />
<uses-permission android:name=" android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name=" android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission android:name=" android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name=" android.permission.WRITE_SETTINGS" />
<uses-permission android:name=" android.permission.READ_PHONE_STATE" />
<uses-permission android:name=" android.permission.MODIFY_AUDIO_SETTINGS" />
<uses-permission android:name=" android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name=" android.permission.CHANGE_WIFI_STATE" />
```

各权限含义说明：

名称	用途
android.permission.INTERNET	允许应用联网
android.permission.ACCESS_NETWORK_STATE	获取当前网络状态
android.permission.READ_EXTERNAL_STORAGE	读外部存储
android.permission.WRITE_EXTERNAL_STORAGE	写外部存储
android.permission.WRITE_SETTINGS	允许程序读取或写入系统设置
android.permission.READ_PHONE_STATE	允许程序读取手机状态
android.permission.MODIFY_AUDIO_SETTINGS	允许程序修改全局声音设置权限
android.permission.ACCESS_WIFI_STATE	允许应用程序获取WIFI状态
android.permission.CHANGE_WIFI_STATE	允许应用程序改变WI-FI网络链接状态的权限

9.1.5 Proguard配置

请不要混淆sdk的jar包。

9.2 重要接口说明

9.2.1 合成器

合成器，类名SpeechSynthesizer，主要用来控制合成进程：设置参数，开始，结束，取消等。

创建对象

- 说明

获取语音合成引擎实例。

- 方法

```
public SpeechSynthesizer getInstance()
```

- 参数

无。

- 返回

语音合成引擎实例。

设置Context

- 说明

设置语音合成实例的上下文环境。

- 方法

```
public void setContext(Context context)
```

- 参数

名称	描述
context	上下文信息

- 返回

无。

设置监听器

- 说明 设置语音合成监听器，回调合成和播放进度以及错误信息。
- 方法

```
public void setSpeechSynthesizerListener(SpeechSynthesizerListener speechSynthesizerListener  
)
```

- 参数

名称	描述
SpeechSynthesizerListener	语音合成监听器

- 返回

无。

初始化TTS引擎

- 说明

初始化合成引擎，可以指定使用online在线，或者mix离在线混合引擎。mix混合引擎会在online在线不能用的情况下自动使用offline离线引擎。

- 方法

```
public int initTts(TtsMode ttsMode)
```

- 参数

名称	描述
ttsMode	引擎选择 (online是纯在线合成引擎, Mix 是离在线混合引擎)

- 返回

[错误码。](#)

获取版本信息

- 说明

获取当前的SDK版本信息。

- 方法

```
public String libVersion()
```

- 参数

无。

- 返回

版本信息。

设置在线授权

- 说明

设置在线授权时需要用到的apiKey和secretKey，关系到在线合成是否可用。

- 方法

```
public int setApiKey(String apiKey, String secretKey)
```

- 参数

名称	描述
apiKey	开放平台注册app的apiKey
secretKey	开放平台注册app的secretKey

- 返回

错误码。

设置离线授权

- 说明

设置离线授权所需要的AppId，关系到离线合成是否可用。

- 方法

```
public int setAppId(String appId)
```

- 参数

名称	描述
appId	开放平台注册app的appId

- 返回

错误码。

设置参数

- 说明

设置TTS参数。

- 方法

```
public int setParam(String key, String value)
```

- 参数

名称	描述
key	参数的键
value	参数的值

- 返回

错误码。

设置播放器的音频流类型

- 说明

设置播放器的音频流类型。

- 方法

```
public int setAudioStreamType(int streamType)
```

- 参数

名称	描述
streamType	音频流类型

- 返回

错误码。

暂停任务

- 说明

暂停当前的TTS任务。

- 方法

```
public int pause()
```

- 参数

无

- 返回

错误码。

恢复任务

- 说明

恢复暂停的TTS任务。

- 方法

```
public int resume()
```

- 参数

无

- 返回

错误码。

停止任务

- 说明

停止所有的TTS任务。

- 方法

```
public int stop()
```

- 参数

无

- 返回

错误码。

释放TTS资源

- 说明

释放TTS的资源。

- 方法

```
public int release()
```

- 参数

无

- 返回

错误码。

加载离线自定义模型文件

- 说明

加载离线自定义模型文件，仅离线引擎可用。

- 方法

```
public int loadCustomResource(String customModelPath)
```

- 参数

名称	描述
customModelPath	自定义模型文件路径

- 返回

错误码。

释放离线自定义模型文件

- 说明

释放离线自定义模型文件，仅离线引擎可用。

- 方法

```
public int freeCustomResource()
```

- 参数

无。

- 返回

错误码。

切换离线发音人接口

- 说明

该接口用来切换离线的发音人（发音人模型文件）。传入对应的要切换的资源文件的绝对路径即可，如果speechModelPath不同，则传入对应的路径；如果textModelPath不同，则传入对应的路径，如果相同，则传null即可。该方法必须在initTTS成功后才可调用。如果在调用initTTS之后，离线引擎没有被初始化，该接口会重新初始化离线引擎，并加载模型文件。

- 方法

```
public int loadModel(String speechModelPath, String textModelPath)
```

- 参数

名称	描述
speechModelPath	后端模型文件路径
textModelPath	前端模型文件路径

- 返回

错误码。

加载英文模型文件

- 说明

加载英文模型文件，仅离线引擎可用。

- 方法

```
public int loadEnglishModel(String englishTextModelPath, String englishSpeechModelPath)
```

- 参数

名称	描述
englishTextMode lPath	英文前端模型文件路径
englishSpeechMo delPath	英文后端模型文件路径

- 返回

错误码。

合成并播放文本

- 说明

合成并播放给定文本，合成并播放功能有多个重载函数，其参数不同，但是返回值相同。

- 返回

错误码。

- 方法1：合成并播放文本

```
public int speak(String text)
```

- 参数

名称	描述
text	要合成的文本（需要合成的文本text的长度不能超过1024个GBK字节）

- 方法2：可排队的合成并播放

```
public int speak(SpeechSynthesizeBag speechSynthesizeBag)
```

- 参数

名称	描述
speechSynthesizeBag	语音合成包

- 方法3：可排队的合成并播放

```
public int speak(String text, String utteranceId)
```

- 参数

名称	描述
text	要合成的文本（需要合成的文本text的长度不能超过1024个GBK字节）
utteranceId	合成文本的ID

- 方法4：可排队的合成并播放，尚未使用

```
public int speak(String text, String utteranceId, Bundle params)
```

- 参数

名称	描述
text	要合成的文本（需要合成的文本text的长度不能超过1024个GBK字节）
utteranceId	合成文本的ID
params	预留参数

合成文本

- 说明

合成给定文本，合成文本功能有多个重载函数，其参数不同，但是返回值相同。

- 返回

错误码。

- 方法1：合成文本

```
public int synthesize(String text)
```

- 参数

名称	描述
text	要合成的文本（需要合成的文本text的长度不能超过1024个GBK字节）

- 方法2：可排队的合成

```
public int synthesize (SpeechSynthesizeBag speechSynthesizeBag)
```

- 参数

名称	描述
speechSynthesiz eBag	语音合成包

- 方法3：可排队的合成

```
public int synthesize(String text, String utteranceId)
```

- 参数

名称	描述
text	要合成的文本（需要合成的文本text的长度不能超过1024个GBK字节）
utteranceId	合成文本的ID

- 方法4：可排队的合成，尚未使用

```
public int synthesize(String text, String utteranceId, Bundle params)
```

- 参数

名称	描述
text	要合成的文本（注：需要合成的文本text的长度不能超过1024个GBK字节）
utteranceId	合成文本的ID
params	预留参数

批量合成并播放

- 说明

批量合成并播放文本文件，一次可传入一个list，这个list会按顺序合成播放。

- 方法

```
public int batchSpeak(List<SpeechSynthesizeBag> speechSynthesizeBags)
```

- 参数

名称	描述
speechSynthesizeBags	语音合成包队列

- 返回

错误码。

授权验证

- 说明

授权验证接口。

- 方法

```
public int auth(TtsMode ttsMode)
```

- 参数

名称	描述
ttsMode	语音合成模式

- 返回

错误码。

9.2.2 合成状态监听器

SpeechSynthesizerListener是合成状态监听器接口，开发者可以通过实现此接口对识别状态做监听。

合成开始

- 说明

合成开始时的回调方法。

- 方法

```
void onSynthesizeStart(String utteranceId)
```

- 参数

名称	描述
utteranceId	该次表达的标识

- 返回 无。

获取合成数据

- 说明

当获取合成数据时调用的方法，合成数据会分多次返回，需自行保存。

- 方法

```
void onSynthesizeDataArrived(String utteranceId, byte[] audioData, int progress)
```

- 参数

名称	描述
utteranceId	该次表达的标识
audioData	返回的合成数据
progress	播放文字的进度

- 返回

无。

合成完成状态

- 说明

当合成完成时调用的方法。

- 方法

```
void onSynthesizeFinish(String utteranceId)
```

- 参数

名称	描述
utteranceId	该次表达的标识

- 返回

无。

播放开始状态

- 说明

当开始播放时调用的方法。

- 方法

```
void onSpeechStart(String utteranceId)
```

- 参数

名称	描述
utteranceId	该次表达的标识

- 返回

无。

播放进度状态

- 说明

当进度进度改变时调用的方法。

- 方法

```
void onSpeechProgressChanged(String utteranceId, int progress)
```

- 参数

名称	描述
utteranceId	该次表达的标识
progress	播放文字的进度

- 返回

无。

播放完成

- 说明

播放完成时调用的方法。

- 方法

```
void onSpeechFinish(String utteranceId)
```

- 参数

名称	描述
utteranceId	该次表达的标识

- 返回

无。

出错

- 说明

出错时的回调函数。

- 方法

```
void onError(String utteranceId, SpeechError error)
```

- 参数

名称	描述
utteranceId	该次表达的标识
error	错误码

- 返回 无。

9.3 参数和错误码说明

9.3.1 合成器参数设置说明

在调用speak方法开始合成并朗读之前，可以对参数进行配置（未设置的参数将使用默认值），目前支持的类型如下：

参数名	默认值	备注
PARAM_VOLUME	“5”	中级音量，范围[0-9]
PARAM_SPEED	“5”	中速，范围[0-9]
PARAM_PITCH	“5”	中调，范围[0-9]
PARAM_SPEAKER	“0”	0 (普通女声) 1 (普通男声) 2 (特别男声) 3 (情感男声) 4 (情感儿童声)
PARAM_MIX_MODE	MIX_MODE_DEFAULT	MIX_MODE_DEFAULT(mix 模式下， wifi 使用在线合成，非wifi使用离线合成) MIX_MODE_HIGH_SPEED_NETWORK(mix 模式下， wifi,4G,3G使用在线合成，其他使用离线合成) MIX_MODE_HIGH_SPEED_SYNTHESIZE(mix 模式下，在线返回速度如果慢（超时，一般为1.2秒）直接切换离线，适用于网络环境较差的情况)

参数名	默认值	备注
		MIX_MODE_HIGH_SPEED_SYNTHESIZE_WIFI(mix模式下，仅 wifi 使用在线合成,返回速度如果慢(超时，一般为1.2秒)直接切换离线，适用于仅WIFI网络环境较差的情况)
PARAM_AUDIO_ENCODE	AUDIO_ENCODE_AMR	AUDIO_ENCODE_BV
		AUDIO_ENCODE_AMR
		AUDIO_ENCODE_OPUS
PARAM_AUDIO_RATE	AUDIO_BITRATE_AMR_15K85	AUDIO_BITRATE_BV_16K
		AUDIO_BITRATE_AMR_6K6
		AUDIO_BITRATE_AMR_8K85
		AUDIO_BITRATE_AMR_12K65
		AUDIO_BITRATE_AMR_14K25
		AUDIO_BITRATE_AMR_15K85
		AUDIO_BITRATE_AMR_18K25
		AUDIO_BITRATE_AMR_19K85
		AUDIO_BITRATE_AMR_23K05
		AUDIO_BITRATE_AMR_23K85
		AUDIO_BITRATE_OPUS_8K
		AUDIO_BITRATE_OPUS_16K
		AUDIO_BITRATE_OPUS_18K
		AUDIO_BITRATE_OPUS_20K
		AUDIO_BITRATE_OPUS_24K
		AUDIO_BITRATE_OPUS_32K
PARAM_VOCODER_OPTIM_LEVEL		合成引擎速度优化等级，取值范围[0, 2]，值越大速度越快(离线引擎)

9.3.2 错误码说明

错误码值	错误码描述
-1	在线引擎授权失败
-2	在线合成请求失败

错误码值	错误码描述
-3	在线合成停止失败
-4	在线授权中断异常
-5	在线授权执行时异常
-6	在线授权时间超时
-7	在线合成返回错误信息
-8	在线授权token为空
-9	在线引擎没有初始化
-10	在线引擎合成时异常
-100	离线引擎授权失败
-101	离线合成停止失败
-102	离线授权下载License失败
-103	离线授权信息为空
-104	离线授权类型未知
-105	离线授权中断异常
-106	离线授权执行时异常
-107	离线授权执行时间超时
-108	离线合成引擎初始化失败
-109	离线引擎未初始化
-110	离线合成时异常
-111	离线合成返回值非0
-112	离线授权已过期
-113	离线授权包名不匹配
-114	离线授权签名不匹配
-115	离线授权设备信息不匹配
-116	离线授权平台不匹配
-117	离线授权的license文件不存在
-200	混合引擎离线在线都授权失败
-201	混合引擎授权中断异常
-202	混合引擎授权执行时异常
-203	混合引擎授权执行时间超时
-300	合成文本为空
-301	合成文本长度过长 (不要超过GBK1024个字节)

错误码值	错误码描述
-302	合成文本无法获取GBK字节
-400	TTS未初始化
-401	TTS模式无效
-402	TTS合成队列已满（最大限度为1000）
-403	TTS批量合成文本过多（最多为100）
-404	TTS停止失败
-405	TTS APP ID无效
-406	TTS被调用方法参数无效
-500	Context被释放或为空
-600	播放器为空
-9999	未知错误

9.4 完整示例

一个简单的语音合成示例如下所示，请替换相关参数和操作：

```
public class DemoActivity extends Activity implements SpeechSynthesizerListener {
    // 语音合成客户端
    private SpeechSynthesizer mSpeechSynthesizer;

    // 的函数ActivityonCreate
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        startTTS();
    }

    // 初始化语音合成客户端并启动
    private void startTTS() {
        // 获取语音合成对象实例
        mSpeechSynthesizer = SpeechSynthesizer.getInstance();
        // 设置context
        mSpeechSynthesizer.setContext(this);
        // 设置语音合成状态监听器
        mSpeechSynthesizer.setSpeechSynthesizerListener(this);
        // 设置在线语音合成授权，需要填入从百度语音官网申请的和api_keysecret_key
        mSpeechSynthesizer.setApiKey(" your_api_key" , " your_secret_key" );
        // 设置离线语音合成授权，需要填入从百度语音官网申请的app_id
        mSpeechSynthesizer.setAppId(" your_app_id" );
        // 设置语音合成文本模型文件
        mSpeechSynthesizer.setParam(SpeechSynthesizer.PARAM_TTS_TEXT_MODEL_FILE,
            " your_txt_file_path" );
```

```
// 设置语音合成声音模型文件
mSpeechSynthesizer.setParam(SpeechSynthesizer.
    PARAM_TTS_SPEECH_MODEL_FILE, " your_speech_file_path" );
// 设置语音合成声音授权文件
mSpeechSynthesizer.setParam(SpeechSynthesizer.PARAM_TTS_LICENCE_FILE, "
    your_licence_path" );
// 获取语音合成授权信息
AuthInfo authInfo = mSpeechSynthesizer.auth(TtsMode.MIX);
// 判断授权信息是否正确，如果正确则初始化语音合成器并开始语音合成，如果失
败则做错误处理
if (authInfo.isSuccess()) {
    mSpeechSynthesizer.initTts(TtsMode.MIX);
    mSpeechSynthesizer.speak("百度语音合成示例程序正在运行" );
} else {
    // 授权失败
}

}

public void onError(String arg0, SpeechError arg1) {
    // 监听到出错，在此添加相关操作
}

public void onSpeechFinish(String arg0) {
    // 监听到播放结束，在此添加相关操作
}

public void onSpeechProgressChanged(String arg0, int arg1) {
    // 监听到播放进度有变化，在此添加相关操作
}

public void onSpeechStart(String arg0) {
    // 监听到合成并播放开始，在此添加相关操作
}

public void onSynthesizeDataArrived(String arg0, byte[] arg1, int arg2) {
    // 监听到有合成数据到达，在此添加相关操作
}

public void onSynthesizeFinish(String arg0) {
    // 监听到合成结束，在此添加相关操作
}

public void onSynthesizeStart(String arg0) {
    // 监听到合成开始，在此添加相关操作
}
```

第10章

简介

10.0.1 概念解释

语音合成是实现人机语音交互，建立一个有听和讲能力的交互系统所必需的关键技术。随着语音技术的发展，百度自主研发了语音合成系统（TTS），功能是接收用户发送的文本，生成语音发送给用户。

对本文中将提到的名词约定如下：

语音合成（Text To Speech, TTS）：将文本合成为语音，即声音文件。

合成引擎：将文本合成为语音的核心模块。

百度语音合成SDK（BDSSpeechSynthesizer）：本开发包。BDSSpeechSynthesizer是一个封装了网络首发、音频播放功能的语音合成解决方案。借助BDSSpeechSynthesizer可以在应用程序中快速集成语音合成功能。

10.0.2 功能介绍

百度语音合成客户端iOS离在线融合版SDK（以下简称BDSSpeechSynthesizer）是一种面向iOS移动设备的语音合成解决方案，以Cocoa Touch Static Library形式发布。支持离线语音合成，在线语音合成，在线优先等合成模式。目前版本已支持SDK内部直接播放合成语音。离线合成支持语速、音调、音量、引擎优化级别设置，后续版本将支持如男女声等更多的合成参数；在线合成支持男女声、语速、音调、音量、音频码率设置，后续版本将支持从SDK获取语音数据、合成进度提示以及更多参数设置。本版本优先使用在线语音合成服务合成，以获得更好的合成效果。如在线合成服务不可用，如网络连接异常，蜂窝信号差等，将会使用离线合成引擎合成本文，保证功能可用，并按规则定期侦测在线语音合成服务，如在线服务可用，下次语音合成将使用在线服务合成。

10.0.3 兼容性

- 系统：支持iOS 6.0及以上。
- 机型：iPhone和iPad皆可。
- 架构：支持i386、x86_64、armv7、arm64。（离线合成不支持i386和x86_64架构。）

10.0.4 开发包说明

文件(夹)名	说明
Doc/Baidu_Combined_TTS_SDK_iOS_Manual.pdf	本文档
BDSSpeechSynthesizer_SDK	语音合成SDK Lib库，支持simulator和iOS设备
BDSSpeechSynthesizerSample	开发示例(xcode project)
OfflineTTSDatFiles/Chinese_Speech_Female.dat	语音合成资源文件(speech data file, 中文, 女声)
OfflineTTSDatFiles/Chinese_Speech_Male.dat	语音合成资源文件(speech data file, 中文, 男声)
OfflineTTSDatFiles/Chinese_Text.dat	语音合成资源文件(text data file, 中文)
OfflineTTSDatFiles/English_Speech_Female.dat	语音合成资源文件(speech data file, 英文, 女声)
OfflineTTSDatFiles/English_Speech_Male.dat	语音合成资源文件(speech data file, 英文, 男声)
OfflineTTSDatFiles/English_Text.dat	语音合成资源文件(text data file, 英文)
OfflineTTSDatFiles/offline_engine_tmp_license.dat	授权文件

10.1 集成指南

10.1.1 创建应用

请参考《百度语音开放平台使用指南》创建应用，开通服务并完成个性化设置。

10.1.2 添加BDSSpeechSynthesizer到工程

BDSSpeechSynthesizer使用了一些系统的framework，需要添加到工程里面。添加方式：右键点击 Xcode 中的工程文件，在出现的界面中，选中 TARGETS 下的应用，在出现的界面中选中 Build Phases->Link Binary With Libraries，点击界面中的“+”图标，在弹出的界面中选择上下

图中的一个frame即可,逐个添加后,添加完成效果图如图所示(libetts_device_simulator.a将在随后添加)。

Name	Status
libstdc++-6.0.9.tbd	Required ▼
libc++.tbd	Required ▼
libz.1.2.5.tbd	Required ▼
libetts_device_simulator.a	Required ▼
AudioToolbox.framework	Required ▼
UIKit.framework	Required ▼

图 10.1: k

10.1.3 添加授权文件

将开发包中的license目录下的bdtts_licence.dat文件添加到工程,或者安装app后部署到app对应的Documents目录下。

10.1.4 添加语音合成资源文件

将开发包中的data目录下的bd_tts.dat文件添加到工程或者安装app后部署到指定目录(代码中启动合成引擎时需要指定该资源文件的访问路径)。

10.1.5 引入BDSSpeechSynthesizer的头文件

首先将 BDSSpeechSynthesizer 提供的头文件拷贝到工程目录下,在 XCode 中添加此文件,引入 BDSSpeechSynthesizer 提供的头文件。

添加如下头文件:

```
## import "BDSSpeechSynthesizer.h"
```

10.1.6 引入静态库文件

BDSSpeechSynthesizer提供了支持模拟器 5.0 及更新版本、真机 armv7, armv7s, arm64 及更新架构所使用的静态库文件,存放在开发包lib目录下。

引入静态库文件的具体方式为:将libetts_device_simulator.a 采用添加文件方式添加到工程的 Framework 目录下。

说明：libetts_device_simulator.a 是一个通用的库文件，支持armv7、arm64、i386、x86_64, 避免开发者在 build 不同 target 时频繁替换.a 文件的问题

10.2 重要接口说明

10.2.1 语音合成器

合成器，类名BDSSpeechSynthesizer，主要用来控制合成进程：设置参数，开始，结束，取消等。

获取合成器唯一实例

- 方法

```
(BDSSpeechSynthesizer*)sharedInstance;
```

- 参数

无。

- 返回

语音合成引擎实例。

- 说明

获取语音合成引擎实例，该实例为单例对象。

释放合成器唯一实例

- 方法

```
(void)releaseInstance;
```

- 参数

无。

- 返回

无。

- 说明

释放语音合成器实例。

设置合成器代理

- 方法

```
(void)setSynthesizerDelegate: (id<BDSSpeechSynthesizerDelegate>)delegate;
```

- 参数

参数	含义
delegate	代理对象

- 返回

无。

- 说明

设置合成器代理，代理对象负责处理合成器各类事件。

设置合成参数

- 方法

```
(NSError*)setSynthParam:(id)param forKey:(BDSSynthesizerParamKey)key;
```

- 参数

参数	含义
param	参数值
key	参数键

- 返回

nil或错误信息。

- 说明

设置合成参数。

获取合成参数

- 方法

```
(id)getSynthParamForKey:(BDSSynthesizerParamKey)key withError:(NSError**)err;
```

- 参数

参数	含义
key	参数键
err	如果失败, 返回错误信息

- 返回

参数值。

- 说明

获取合成参数。

设置认证信息

- 方法

```
(void)setApiKey:(NSString *)apiKey withSecretKey:(NSString *)secretKey;
```

- 参数

参数	含义
apiKey	用户从语音官网申请的apiKey
secretKey	用户从语音官网申请的secretKey

- 返回

无。

- 说明

设置认证信息。

设置回调队列

- 方法

```
(NSError*)setSDKCallbackQueue:(dispatch_queue_t)callbackQueue;
```

- 参数

参数	含义
callbackQueue	回调队列

- 返回

nil或错误信息。

- 说明

设置回调队列。

获得当前回调队列

- 方法

```
(dispatch_queue_t)getCurrentCallbackQueue;
```

- 参数

无。

- 返回

回调队列。

- 说明

获得当前回调队列。

设置合成线程优先级

- 方法

```
(void)setThreadPriority:(double)priority;
```

- 参数

参数	含义
priority	优先级

- 返回

无。

- 说明

设置合成线程优先级。

启动合成引擎

- 方法

```
(NSError*)loadOfflineEngine: (NSString*)textDatFilePath  
                      speechDataPath: (NSString*)speechDatFilePath  
                     licenseFilePath: (NSString*)licenseFilePath  
                        withAppCode: (NSString*)appCode;
```

- 参数

参数	含义
textDatFilePath	中文文本分析数据文件路径
speechDataPath	中文声学模型数据文件路径
licenseFilePath	授权文件路径，如果没有本地授权可传入nil
appCode	用户持有的授权app code

- 返回

nil或错误信息。

- 说明

启动合成引擎。

加载英文合成数据文件及模型文件

- 方法

```
- (NSError*)loadEnglishDataForOfflineEngine: (NSString*)textDataPath  
                           speechData: (NSString*)speechDataPath;
```

- 参数

参数	含义
textDataPath	英文文本分析数据文件路径
speechDataPath	英文声学模型数据文件路径

- 返回

nil或错误信息。

- 说明

加载英文合成数据文件及模型文件。

重新加载文本分析数据文件或者声学模型数据文件

- 方法

```
(NSError*)reinitOfflineEngineData: (NSString*)datFilePath;
```

- 参数

参数	含义
datFilePath	数据文件路径

- 返回

nil或错误信息。

- 说明

重新加载文本分析数据文件或者声学模型数据文件。

加载定制库

- 方法

```
(NSError*)loadDomainDataForOfflineEngine:(NSString*)datFilePath;
```

- 参数

参数	含义
datFilePath	数据文件路径

- 返回

nil或错误信息。

- 说明

加载定制库。

卸载定制库

- 方法

```
(NSError*)unloadDomainDataFromOfflineEngine;
```

- 参数

无。

- 返回

nil或错误信息。

- 说明

卸载定制库。

验证音库文件的有效性

- 方法

```
(BOOL)verifyDataFile: (NSString*) datFilePath error:(NSError**)err;
```

- 参数

参数	含义
datFilePath	数据文件路径
err	如果验证失败，返回错误信息

- 返回

验证成功返回YES，验证失败返回NO。

- 说明

验证音库文件的有效性。

获取音库文件相关参数

- 方法

```
(BOOL)getDataFileParam: (NSString*)datFilePath  
                    type: (TTSDataparam)paramType  
                   value: (NSString**)paramValue  
                  error: (NSError**)err;
```

- 参数

参数	含义
datFilePath	数据文件路径
paramType	参数键
paramValue	传出对应参数的值
err	如果失败, 返回错误信息

- 返回

成功返回YES, 失败返回NO。

- 说明

获取音库文件相关参数。

批量开始文本合成但不朗读或添加文本至当前合成过程

- 方法

```
(NSInteger)synthesizeSentence:(NSString*)sentence withError:(NSError**)err;
```

- 参数

参数	含义
sentence	需要语音合成的文本
err	如果失败, 返回错误信息

- 返回

SDK生成的文本ID, -1代表合成失败, 错误信息详见err。

- 说明

批量开始文本合成但不朗读或添加文本至当前合成过程。

批量开始文本合成并朗读或添加文本至当前合成过程

- 方法

```
(NSInteger) speakSentence:(NSString*)sentence withError:(NSError**)err;
```

- 参数

参数	含义
sentence	需要语音合成的文本
err	如果失败, 返回错误信息

- 返回

SDK生成的文本ID, -1代表合成失败, 错误信息详见err。

- 说明

批量开始文本合成并朗读或添加文本至当前合成过程。

取消本次合成并停止朗读

- 方法

```
(void)cancel;
```

- 参数

无。

- 返回

无。

- 说明

取消本次合成并停止朗读。

暂停文本合成并朗读

- 方法

```
(BDSSynthesizerStatus)pause;
```

- 参数

无。

- 返回

合成状态。

- 说明

暂停文本合成并朗读。

继续文本合成并朗读

- 方法

(BDSSynthesizerStatus)resume;

- 参数

无。

- 返回

合成状态。

- 说明

继续文本合成并朗读。

获取合成器状态

- 方法

(BDSSynthesizerStatus)synthesizerStatus;

- 参数 无。

- 返回

合成状态。

- 说明

获取合成器状态。

设置播放器音量

- 方法

```
(void)setPlayerVolume:(float)volume;
```

- 参数

参数	含义
volume	音量值

- 返回

无。

- 说明

设置播放器音量。

设置AudioSessionCategory类型

- 方法

```
(void)setAudioSessionCategory:(NSString *)category;
```

- 参数

参数	含义
category	AudioSessionCategory类型

- 返回

无。

- 说明

设置AudioSessionCategory类型。

10.2.2 语音合成器委托对象

语音合成器委托对象BDSSpeechSynthesizerDelegate，用来处理语音合成器的各种回调。

开始合成

- 方法

```
(void)synthesizerStartWorkingSentence:(NSInteger)SynthesizeSentence;
```

- 参数

参数	含义
SynthesizeSentence	句子序号

- 返回

无。

- 说明

开始合成。

结束合成

- 方法

```
(void)synthesizerFinishWorkingSentence:(NSInteger)SynthesizeSentence;
```

- 参数

参数	含义
SynthesizeSentence	句子序号

- 返回

无。

- 说明

结束合成。

开始朗读

- 方法

```
(void)synthesizerSpeechStartSentence:(NSInteger)SpeakSentence;
```

- 参数

参数	含义
SpeakSentence	句子序号

- 返回

无。

- 说明

开始朗读。

结束朗读

- 方法

```
(void)synthesizerSpeechEndSentence:(NSInteger)SpeakSentence;
```

- 参数

参数	含义
SpeakSentence	句子序号

- 返回

无。

- 说明

结束朗读。

新的语音数据已经合成

- 方法

```
(void)synthesizerNewDataArrived:(NSData *)newData  
    DataFormat:(BDSAudioFormat)fmt  
    characterCount:(int)newLength  
    sentenceNumber:(NSInteger)SynthesizeSentence;
```

- 参数

参数	含义
newData	语音数据
DataFormat	音频格式
newLength	语音数据长度
SynthesizeSentence	句子序号

- 返回

无。

- 说明

新的语音数据已经合成。

播放进度变更

- 方法

```
(void)synthesizerTextSpeakLengthChanged:(int)newLength  
    sentenceNumber:(NSInteger)SpeakSentence;
```

- 参数

参数	含义
newLength	语音数据长度
SpeakSentence	句子序号

- 返回

无。

- 说明

播放进度变更。

合成器发生错误

- 方法

```
(void)synthesizerErrorOccurred:(NSError *)error  
                      speaking:(NSInteger)SpeakSentence  
                     synthesizing:(NSInteger)SynthesizeSentence;
```

- 参数

参数	含义
error	错误码
SpeakSentence	正在播放的句子序号
SynthesizeSentence	正在合成的句子序号

- 返回

无。

- 说明

合成器发生错误。

10.3 参数和错误说明

10.3.1 语音合成器参数

合成参数 百度语音合成支持的参数用BDSSynthesizerParamKey类型表示，具体如下：

参数	含义
BDS_SYNTHESIZER_PARAM_SPEED	语速
BDS_SYNTHESIZER_PARAM_PITCH	音调
BDS_SYNTHESIZER_PARAM_VOLUME	音量
BDS_SYNTHESIZER_PARAM_PID	产品号
BDS_SYNTHESIZER_PARAM_LANGUAGE	语言
BDS_SYNTHESIZER_PARAM_TEXT_ENCODE	文本编码类型
BDS_SYNTHESIZER_PARAM_AUDIO_ENCODE	音频编码类型
BDS_SYNTHESIZER_PARAM_SPEAKER	发音人
BDS_SYNTHESIZER_PARAM_USER_AGENT	UA
BDS_SYNTHESIZER_PARAM_ONLINE_REQUEST_TIMEOUT	超时时间
BDS_SYNTHESIZER_PARAM_ETTS_AUDIO_FORMAT	音频格式
BDS_SYNTHESIZER_PARAM_ETTS_VOCODER	合成引擎速度优化等级
BDS_SYNTHESIZER_PARAM_SYNTH_STRATEGY	合成策略

语音合成本文语言 语音合成本文语言用BDSSynthesizerLanguages类型表示，具体如下：

参数	含义
BDS_SYNTHESIZER_LANGUAGE_ZH	中文
BDS_SYNTHESIZER_LANGUAGE_EN	英文

语音合成本文编码格式 语音合成本文编码格式用BDSSynthesizerTextEncodings类型表示，具体如下：

参数	含义
BDS_SYNTHESIZER_TEXT_ENCODE_GBK	GBK编码
BDS_SYNTHESIZER_TEXT_ENCODE_BIG5	大五码编码
BDS_SYNTHESIZER_TEXT_ENCODE_UTF8	UTF8编码

语音合成音频编码格式 语音合成音频编码格式用BDSSynthesizerAudioEncoding类型表示，具体如下：

参数	含义
BDS_SYNTHESIZER_AUDIO_ENCODE_BV_16K	bv 16k比特率

参数	含义
BDS_SYNTHESIZER_AUDIO_ENCODE_AMR_6K6mr	6.6k比特率
BDS_SYNTHESIZER_AUDIO_ENCODE_AMR_8K85mr	8.85k比特率
BDS_SYNTHESIZER_AUDIO_ENCODE_AMR_12K65mr	12.65k比特率
BDS_SYNTHESIZER_AUDIO_ENCODE_AMR_14K25mr	14.25k比特率
BDS_SYNTHESIZER_AUDIO_ENCODE_AMR_15K85mr	15.85k比特率
BDS_SYNTHESIZER_AUDIO_ENCODE_AMR_18K25mr	18.25k比特率
BDS_SYNTHESIZER_AUDIO_ENCODE_AMR_19K85mr	19.85k比特率
BDS_SYNTHESIZER_AUDIO_ENCODE_AMR_23K05mr	23.05k比特率
BDS_SYNTHESIZER_AUDIO_ENCODE_AMR_23K85mr	23.85k比特率
BDS_SYNTHESIZER_AUDIO_ENCODE_OPUS_8Kopus	8k比特率
BDS_SYNTHESIZER_AUDIO_ENCODE_OPUS_16Kopus	16k比特率
BDS_SYNTHESIZER_AUDIO_ENCODE_OPUS_18Kopus	18k比特率
BDS_SYNTHESIZER_AUDIO_ENCODE_OPUS_20Kopus	20k比特率
BDS_SYNTHESIZER_AUDIO_ENCODE_OPUS_24Kopus	24k比特率
BDS_SYNTHESIZER_AUDIO_ENCODE_OPUS_32Kopus	32k比特率
BDS_SYNTHESIZER_AUDIO_ENCODE_MP3_8Kmp3	8k比特率
BDS_SYNTHESIZER_AUDIO_ENCODE_MP3_11Kmp3	11k比特率
BDS_SYNTHESIZER_AUDIO_ENCODE_MP3_16Kmp3	16k比特率
BDS_SYNTHESIZER_AUDIO_ENCODE_MP3_24Kmp3	24k比特率
BDS_SYNTHESIZER_AUDIO_ENCODE_MP3_32Kmp3	32k比特率

10.3.2 语音合成错误码

开始合成错误 开始合成错误用BDSStartSynthesisError类型表示，具体如下：

参数	含义
BDS_START_SYNTHESIS_OK	启动成功
BDS_START_SYNTHESIS_SYNTHESIZER_UNINITIALIZED	合成器未初始化
BDS_START_SYNTHESIS_TEXT_EMPTY	合成文本为空
BDS_START_SYNTHESIS_TEXT_TOO_LONG	合成文本过长
BDS_START_SYNTHESIS_ENGINE_BUSY	合成引擎繁忙
BDS_START_SYNTHESIS_MALLOC_ERROR	获取资源失败

参数	含义
BDS_START_SYNTHESIS_NO_NETWORK	无网络连接
BDS_START_SYNTHESIS_NO_VERIFY_INFO	无授权信息
BDS_START_SYNTHESIS_OFFLINE_ENGINE_N	离线合成引擎未加载

合成错误 合成错误用BDSSynthesisError类型表示，具体如下：

参数	含义
BDS_UNKNOWN_ERROR	未知错误
BDS_PLAYER_FAILED_GET_STREAM_PROPERTY	获取流属性失败
BDS_PLAYER_FAILED_OPEN_DEVICE	打开设备失败
BDS_PLAYER_FAILED_OPEN_STREAM	打开流失败
BDS_PLAYER_ALLOC_FAIL	资源申请失败
BDS_PLAYER_BAD_STREAM	音频流错误
BDS_ONLINE_TTS_CONNECT_ERROR	在线连接错误
BDS_ONLINE_TTS_RESPONSE_PARSE_ERROR	在线解析错误
BDS_ONLINE_TTS_PARAM_ERROR	在线参数错误
BDS_ONLINE_TTS_TEXT_ENCODE_NOT_SUPPORTED	文本编码格式不支持
BDS_ONLINE_TTS_VERIFY_ERROR	在线鉴权错误
BDS_ONLINE_TTS_GET_ACCESS_TOKEN_FAILED	获取token失败
BDS_ETTS_ERR_PARTIAL_SYNTH	离线部分合成错误
BDS_ETTS_ERR_CONFIG	离线配置错误
BDS_ETTS_ERR_RESOURCE	离线资源错误
BDS_ETTS_ERR_HANDLE	离线句柄错误
BDS_ETTS_ERR_PARMAM	离线参数错误
BDS_ETTS_ERR_MEMORY	离线内存错误
BDS_ETTS_ERR_TOO_MANY_TEXT	离线文本过长
BDS_ETTS_ERR_RUN_TIME	离线运行时错误
BDS_ETTS_ERR_NO_TEXT	离线空文本错误
BDS_ETTS_ERR_LICENSE	离线授权错误

10.4 完整示例

10.4.1 语音合成完整示例

```
// 设置apiKey和secretKey
[[BDSSpeechSynthesizer sharedInstance] setApiKey:@"your API key" withSecretKey:@"your secret key"];

// 设置离线引擎
NSString *ChineseSpeechData = [[NSBundle mainBundle] pathForResource:@"Chinese_Speech_Female" ofType:@"dat"];
NSString *ChineseTextData = [[NSBundle mainBundle] pathForResource:@"Chinese_text" ofType:@"dat"];
NSString *EnglishSpeechData = [[NSBundle mainBundle] pathForResource:@"English_Speech_Female" ofType:@"dat"];
NSString *EnglishTextData = [[NSBundle mainBundle] pathForResource:@"English_text" ofType:@"dat"];
NSString *LicenseData = [[NSBundle mainBundle] pathForResource:@"bd tts_license" ofType:@"dat"];
NSError* loadErr = [[BDSSpeechSynthesizer sharedInstance] startTTSEngine:ChineseTextData
licenseFilePath:LicenseData withAppCode:@""];

if(loadErr != nil)
{
    // 处理出错状况
}

// 加载英文资源
loadErr = [[BDSSpeechSynthesizer sharedInstance] loadEnglishData:EnglishTextData speechData:EnglishSpeechData];
if(loadErr != nil)
{
    // 处理出错状况
}

// 获得合成器实例
[[BDSSpeechSynthesizer sharedInstance] setSynthesizerDelegate:self];

// 设置委托对象
[[BDSSpeechSynthesizer sharedInstance] setSynthesizerDelegate:self];

// 开始合成并播放
NSError* speakError = nil;
if([[BDSSpeechSynthesizer sharedInstance] speakSentence:@"您好" withError:&speakError] == -1)
{
    // 错误
    NSLog("错误: %ld, %@", (long)speakError.code, speakError.localizedDescription);
}
if([[BDSSpeechSynthesizer sharedInstance] speakSentence:@"今天天气真不错" withError:&speakError] == -1)
{
    // 错误
    NSLog("错误: %ld, %@", (long)speakError.code, speakError.localizedDescription);
```

```
}

if([[BDSSpeechSynthesizer sharedInstance] speakSentence:@"Today's weather is really good!" withError:&error])
{
    // 错误
    NSLog(@"错误: %ld, %@", (long)speakError.code, speakError.localizedDescription);
}

- (void)synthesizerStartWorkingSentence:(NSInteger)SynthesizeSentence
{
    NSLog(@"Began synthesizing sentence %ld", (long)SynthesizeSentence);
}

- (void)synthesizerFinishWorkingSentence:(NSInteger)SynthesizeSentence
{
    NSLog(@"Finished synthesizing sentence %ld", (long)SynthesizeSentence);
}

- (void)synthesizerSpeechStartSentence:(NSInteger)SpeakSentence
{
    NSLog(@"Began playing sentence %ld", (long)SpeakSentence);
}

- (void)synthesizerSpeechEndSentence:(NSInteger)SpeakSentence
{
    NSLog(@"Finished playing sentence %ld", (long)SpeakSentence);
}
```

第11章

概念解释

语音合成是实现人机语音交互，建立一个有听和讲能力的交互系统所必需的关键技术。随着语音技术的发展，百度自主研发了语音合成系统（TTS），功能是接受用户发送的文本，生成语音发送给用户。

对本文中将提到的概念约定如下：

- **语音合成**：将文本合成为语音，即声音文件。
- **合成引擎**：将文本合成为语音的核心模块。
- **TTS**：Text To Speech，即“从文本到语音”。
- **BDTTSClient**：语音合成 SDK 简称，详见下条。
- **语音合成 SDK**：即本开发包，文中简称为 BDTTSClient。BDTTSClient 是一个封装了网络收发、音频播放功能的语音合成解决方案。借助 BDTTSClient 可以快速地在应用程序中集成语音合成功能。

11.1 简介

百度语音合成客户端 iOS 版 SDK（以下简称 BDTTSClient）是一种面向 Android 移动设备的语音合成解决方案，以静态库(.a)的形式发布。目前版本已支持 SDK 内部直接播放合成语音，并支持男女声、语速、音调、音量、音频码率设置，后续版本将支持从 SDK 获取语音数据、合成进度提示以及更多参数设置。

11.1.1 兼容性

系统：支持 iOS 4.3 及以上版本

机型：iPhone 3GS+，iPad 1+和 iPod 3+

构架：armv7/armv7s/arm64

网络：支持 WIFI 及移动网络,移动网络支持使用 NET 网关及 WAP 网关(CMWAP、CTWAP、UNIWAP、3GWAP)

11.1.2 开发包说明

开发包说明表

文件(夹)名	说明
libBDSSpeechSynthesizer.a	语音合成 SDK Lib 库，支持模拟器/armv7/armv7s/arm64
BDSSpeechSynthesizerSample	开发示例(Xcode project)
Headers	接口头文件

11.2 集成指南

11.2.1 引入编译需要的 Framework

BDTTSClient 使用了播放功能，因此需要在 Xcode 工程中引入 AVFoundation.framework；BDTTSClient 还使用到了第三方网络库 AFNetworking，因此还需要引入 SystemConfiguration.framework、MobileCoreServices.framework、CoreGraphics.framework、Security.framework。

如果需要使用 SDK 提供的控件，需要引入 OpenGL ES.framework、QuartzCore.framework、GLKit.framework，和 CoreText.framework。

添加方式：右键点击 Xcode 中的工程文件，在出现的界面中，选中 TARGETS 下的应用，在出现的界面中选中 Build Phases->Link Binary With Libraries，点击界面中的“+”图标，在弹出的界面中选择上述几个 Framework 即可，添加完成效果图如图 8 所示（libBDSSpeechSynthesizer.a 将在随后添加）。

▼ Link Binary With Libraries (8 items)

Name
MobileCoreServices.framework
Security.framework
SystemConfiguration.framework
AVFoundation.framework
libBDSSpeechSynthesizer.a
CoreGraphics.framework
UIKit.framework
Foundation.framework

+

-

11.2.2 引入BDVRClient的头文件

首先将 BDTTSClient 提供的头文件拷贝到工程目录下，在 XCode 中添加此文件，引入 BDTTSClient 提供的头文件。添加如下头文件：

```
## import "BDSSpeechSynthesizer.h"  
## import "BDSSpeechSynthesizerDelegate.h"
```

如果需要播放缓存后的语音数据，可以使用内置的播放器，引入如下头文件：

```
## import "BDSBuiltInPlayer.h"
```

11.2.3 引入静态库文件

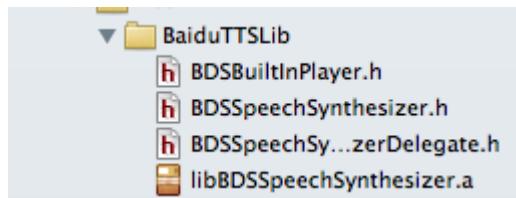
BDTTSClient 提供了支持模拟器 5.0 及更新版本、真机 armv7 及更新架构所使用的静态库文件，存放在开发包根目录下。

注意：静态库中采用 Objective C++实现，因此需要保证工程中引用静态库头文件实现文件的扩展名必须为.mm。

引入静态库文件的具体方式为：将 libBDSSpeechSynthesizer.a 采用添加文件方式添加到工程的Framework 目录下。

说明：libBDVoiceRecognitionClient.a 是采用 lipo 命令将真机 armv7，真机 armv7s、真机 arm64 和模拟器版的.a 合并成的一个通用的.a 文件，避免开发者在 build 不同 target 时频繁替换.a 文件的问题。

所有文件添加完成后如图 9 所示：



11.3 语音合成

11.3.1 合成同时播放

```
// 注：第 1 个参数当前请传入任意非空字符串即可
BDSSpeechSynthesizer synthesizer = [[BDSSpeechSynthesizer alloc] initSynthesizer:@"holder" delegate:nil];
// 注：your-apiKey 和 your-secretKey 需要换成在百度开发者中心注册应用得到的对应值
[synthesizer setApiKey:@"your-apiKey" withSecretKey:@"your-secretKey"];
[synthesizer speak:@"百度一下"];
```

11.3.2 参数设置

在调用 speak 方法开始合成并朗读之前，可以对参数进行配置，代码如下（完整参数列表详见头文件 BDSSpeechSynthesizer.h）

```
[self.synthesizer setParamForKey:BDS_PARAM_SPEAKER value: BDS_SPEAKER_FEMALE];
[self.synthesizer setParamForKey:BDS_PARAM_VOLUME value:@"5"];
[self.synthesizer setParamForKey:BDS_PARAM_SPEED value:@"5"];
[self.synthesizer setParamForKey:BDS_PARAM_PITCH value:@"5"];
[self.synthesizer setParamForKey:BDS_PARAM_AUDIO_ENCODE
    value: BDS_AUDIO_ENCODE_AMR];
[self.synthesizer setParamForKey:BDS_PARAM_AUDIO_RATE
    value: BDS_AUDIO_BITRATE_AMR_15K85];
```

注意：目前暂不支持 BDS_PARAM_NUM_PRON, BDS_PARAM_ENG_PRON, BDS_PARAM_PUNC, BDS_PARAM_BACKGROUND, BDS_PARAM_STYLE, BDS_PARAM_TERRITORY 等参数。

参数默认值列表如下：

参数名	默认值
BDS_PARAM_SPEAKER	BDS_SPEAKER_FEMALE
BDS_PARAM_VOLUME	“5”
BDS_PARAM_SPEED	“5”
BDS_PARAM_PITCH	“5”
BDS_PARAM_AUDIO_ENCODE	BDS_AUDIO_ENCODE_AMR
BDS_PARAM_AUDIO_RATE	BDS_AUDIO_BITRATE_AMR_15K85amr

11.3.3 音频流类型设置

如果需要对音频播报的音频流进行设置，可以调用如下接口：

```
[self.synthesizer setAudioStreamType: AVAudioSessionCategoryPlayback];
```

传入的参数值复用 iOS SDK 的 AVAudioSession 类中声明的相关常量，开发者可直接查阅 AVAudioSession Class Reference。

11.3.4 播放控制

通过 `- (void)cancel;`、`- (int)pause;`、`- (int)resume;` 方法可以对播放流程进行控制。

11.3.5 状态监听

为了更好地实现用于界面，BDTTSClient 提供了 `BDSSpeechSynthesizerDelegate` 代理对象用于对合成器的状态进行通知，监听器的回调方法参见头文件 `BDSSpeechSynthesizerDelegate.h`。完整的开发示例请参见开发包所附示例 xcode project——`BDSSpeechSynthesizerSample`。

11.3.6 获取合成数据自行播放

BDTTSClient 还支持仅获取合成数据，开发者可以在获取数据后先本地缓存，在合适的时候再进行播放，以避免较差网络情况下播放出现卡顿。参数设置同音频流类型设置。

```
// 注：第 1 个参数当前请传入任意非空字符串即可
BDSSpeechSynthesizer synthesizer = [[BDSSpeechSynthesizer alloc] initSynthesizer:@"holder" delegate:nil];
// 注：your-apiKey 和 your-secretKey 需要换成在百度开发者中心注册应用得到的对应值
[synthesizer setApiKey:@"your-apiKey" withSecretKey:@"your-secretKey"];
[synthesizer synthesize:@"百度一下"];
```

11.3.7 获取合成数据

通过 BDSSpeechSynthesizerDelegate 的 synthesizerNewDataArrived 方法获取数据，代码如下：

```
- (void)synthesizerNewDataArrived:(BDSSpeechSynthesizer *)speechSynthesizer data:(NSData *)newData isLastData:(BOOL)lastDataFlag
{
    [self logDebug:[NSString stringWithFormat:@"新的音频数据: %ld%@", (long)[newData length], (lastDataFlag ? @"(end)" : "")]];
}
```

合成数据分多次返回，请根据 isLastDataFlag 的值判断是否已返回所有数据。

11.3.8 播放音频数据

BDTTSClient 内置了用于播放合成数据的播放器，通过如下方法可以播放本地缓存的合成音频数据。

```
BDSBuiltInPlayer *tempPlayer = [[BDSBuiltInPlayer alloc] init];
[tempPlayer playPcmData:_ttsPcmAudio error:nil];
```

11.3.9 播放控制

通过 `- (void)cancel;`、`- (int)pause;`、`- (int)resume;` 方法可以对播放流程进行控制。

11.3.10 播放器状态监听

通过 `BDSBuiltInPlayerDelegate` 监听接口对播放器状态进行监听，参见 `BDSBuiltInPlayer.h` 头文件。

11.4 日志

为了方便调试，BDTTSClient 提供了对日志级别进行设置的接口，通过如下代码可以设置日志级别。

```
[BDSSpeechSynthesizer setLogLevel:BDS_LOG_DEBUG];
```

提供 6 个级别的日志，从低到高依次为 `VERBOSE`、`DEBUG`、`INFO`、`WARN`、`ERROR`、`OFF`，其中 `VERBOSE` 将输出最多日志，`OFF` 将关闭所有日志，默认日志级别为 `DEBUG`。

第12章

常见问题

更多常见问题已经迁移至论坛，见：<http://yuyin.baidu.com/bbs/question/stick>

Q：语音合成提供 REST API 吗？A：语音合成REST API已经开放，请去语音合成相关下载页面下载使用。

Q：语音合成支持那些语言和音色的播报？A：语音合成目前支持中文普通话播报、中英文混读播报，音色支持男声和女声。

Q：有离线语音合成可以提供吗？A：目前没有开放纯离线模式下的语音合成SDK，但是我们开放了离在线融合语音合成SDK，会自动判断当前网络环境，自动匹配使用离线还是在线合成引擎。

Q：语音合成的语音，我想获取音频数据该怎么办？A：可以通过 SpeechSynthesizerListener 的 onNewDataArrive 方法获取音频数据。

Q：提示“Unable to execute dex: Multiple dex files define Lcom/baidu/android/common/logging/Log”错误？A：应用同时集成了百度其它 SDK，造成公共库冲突，请删除语音SDK中的galaxy_lite.jar。

Q：提示“Caused by: java.lang.UnsatisfiedLinkError: Couldn't load BDSpeechDecoder_V1: findLibrary returned null”错误？A：百度语音合成SDK提供了armeabi标准库及armeabi-v7a库。如项目只包含其中一个目录，请只将TTS SDK中的同名目录集成，否则会造成其它库无法正常加载的错误。如项目只有armeabi，请只集成armeabi，反之如果只有armeabi-v7a，则只集成armeabi-v7a。