

百度语音开放平台-iOS SDK文档

1. 概述

本文档是百度语音开放平台iOS SDK的用户指南，描述了语音识别、语音唤醒、语音合成等相关接口的使用说明。

资源占用描述

静态库占用:

SDK类型	静态库大小	二进制增量	__TEXT增量
识别+唤醒	62.5M	1.9M	1.0M~1.2M

资源占用:

资源类型	资源大小
唤醒语言模型	2.3M
离线命令词语言模型	2.3M
离线输入法语言模型	56.8M

由于 **BITCODE** 开启会导致二进制文件体积增大，这部分会在 **APPStore** 发布时进行进一步编译优化，并不会引起最终文件的体积变化，故此处计算的是关闭 **BITCODE** 下的二进制增量。

2. 集成（略）

3. 语音识别接口

语音识别包含数据上传接口和离在线识别接口，接口概述如下：

1. 创建相关接口对象（createEventManagerWithName:）
2. 设置代理对象（setDelegate:）
3. 配置参数（setParameter:forKey:）
4. 发送预定义指令（sendCommand:）
5. 参数列表见相关parameters头文件，预定义值见相关defines头文件

3.1 数据上传接口

3.1.1 代码示例

```
// 创建数据上传对象
self.uploaderEventManager = [BDSEventManager createEventManagerWithName
:BDS_UPLOADER_NAME];
// 设置数据上传代理
[self.uploaderEventManager setDelegate:self];
// 参数配置：词条名
[self.uploaderEventManager setParameter:@"songs" forKey:BDS_UPLOADER_SLOT_NAME];
// 参数配置：词条
[self.uploaderEventManager setParameter:@[@"百度", @"语音"] forKey:BDS_UPLOADER_SLOT_WORDS];
// 启动上传
[self.uploaderEventManager sendCommand:BDS_UP_CMD_START];
```

3.1.2 词条说明

词条名	功能描述
songs	歌曲词条
contacts	通讯录词条

3.1.3 上传功能代理：

```
@protocol BDSClientUploaderDelegate <NSObject>
- (void)UploadCompleteWithError:(NSError *)error;
@end
```

3.2 识别接口

3.2.1 在线识别代码示例

```

// 创建语音识别对象
self.asrEventManager = [BDSEventManager createEventManagerWithName:BDS_
ASR_NAME];
// 设置语音识别代理
[self.asrEventManager setDelegate:self];
// 参数配置：在线身份验证
[self.asrEventManager setParameter:@[API_KEY, SECRET_KEY] forKey:BDS_AS
R_API_SECRET_KEYS];
// 参数配置：垂类设置
[self.asrEventManager setParameter:@[(EVoiceRecognitionPropertySearch)
] forKey:BDS_ASR_PROPERTY_LIST];
// 发送指令：启动识别
[self.asrEventManager sendCommand:BDS_ASR_CMD_START];

```

3.2.2 离线识别代码示例

```

// 创建语音识别对象
self.asrEventManager = [BDSEventManager createEventManagerWithName:BDS_
ASR_NAME];
// 设置语音识别代理
[self.asrEventManager setDelegate:self];
// 参数设置：识别策略
[self.asrEventManager setParameter:@(EVR_STRATEGY_OFFLINE) forKey:BDS_A
SR_STRATEGY];
// 参数设置：离线识别引擎类型
[self.asrEventManager setParameter:@(EVR_OFFLINE_ENGINE_GRAMMER) forKey
:BDS_ASR_OFFLINE_ENGINE_TYPE];
// 参数配置：离线授权APPID
[self.asrEventManager setParameter:@"Your APPID" forKey:BDS_ASR_OFFLINE
_APP_CODE];
// 参数配置：命令行引擎语法文件路径
[self.asrEventManager setParameter:@"命令行引擎语法文件路径" forKey:BDS_ASR
_OFFLINE_ENGINE_GRAMMER_FILE_PATH];
// 参数配置：命令行引擎语言模型文件路径
[self.asrEventManager setParameter:@"命令行引擎语言模型文件路径" forKey:BDS_
ASR_OFFLINE_ENGINE_DAT_FILE_PATH];
// 发送指令：加载离线引擎
[self.asrEventManager sendCommand:BDS_ASR_CMD_LOAD_ENGINE];
// 发送指令：启动识别
[self.asrEventManager sendCommand:BDS_ASR_CMD_START];

```

3.2.3 识别功能代理：

```

@protocol BDSClientASRDelegate<NSObject>
- (void)VoiceRecognitionClientWorkStatus:(int)workStatus obj:(id)aObj;
@end

```


4. 语音唤醒接口

语音唤醒为离线功能，需配置离线授权信息（临时授权文件或开放平台申请的 APPID），加载唤醒所需语言模型文件，接口与语音识别接口相同。

4.1 代码示例

```
// 创建语音识别对象
self.wakeupEventManager = [BDSEventManager createEventManagerWithName:BDS_WAKEUP_NAME];
// 设置语音唤醒代理
[self.wakeupEventManager setDelegate:self];
// 参数配置：离线授权APPID
[self.wakeupEventManager setParameter:@"Your APPID" forKey:BDS_WAKEUP_APP_CODE];
// 参数配置：唤醒语言模型文件路径
[self.wakeupEventManager setParameter:@"唤醒语言模型文件路径" forKey:BDS_WAKEUP_DATA_FILE_PATH];
// 发送指令：加载语音唤醒引擎
[self.wakeupEventManager sendCommand:BDS_WP_CMD_LOAD_ENGINE];
// 发送指令：启动唤醒
[self.wakeupEventManager sendCommand:BDS_WP_CMD_START];
```

4.2 唤醒功能回调接口：

```
@protocol BDSClientWakeupDelegate<NSObject>
- (void)WakeupClientWorkStatus:(int)workStatus obj:(id)aObj;
@end
```

4.3 唤醒辅助识别说明

如需要唤醒后立刻进行识别，为保证不丢音，启动语音识别前请添加如下配置，获取录音缓存：

```
[self.asrEventManager setParameter:@(YES) forKey:BDS_ASR_NEED_CACHE_AUDIO];
```

5. 语音合成接口（略）

6. 常见问题

7. 附录

7.1 音频数据源配置

目前SDK支持三种音频数据源。说明如下：

音频数据源	优先级	说明
文件输入	高	8K或16K采样率、单声道PCM格式音频文件
输入流	中	用以支持外接音源，如车载录音机
本地录音机	低	手机设备内置录音功能

默认使用本地录音机，如设置过文件或输入流相关参数，将其置空即可恢复为默认设置；

录音模块运行时，切换数据源不生效；

多模块共享录音模块，重复设置会产生覆盖；

7.2 语音识别策略

目前SDK支持五种识别策略。说明如下：

识别策略	说明
在线识别	识别请求发至语音服务器进行解析
离线识别	本地完成语音识别解析工作
在线优先	在线识别失败后，自动切换至离线识别
离线优先	离线识别失败后，自动切换至在线识别
并行模式	离在线识别同时进行，取第一个返回的识别结果

7.3 引擎验证方法

在线识别、离线识别与唤醒都需要进行相关验证后方可使用：

引擎类型	验证方法
在线识别	开放平台使用API/SECRET KEY进行验证
离线识别	测试可以使用临时授权文件，正式版本请移除临时授权文件，使用APPID进行验证
唤醒引擎	与离线识别验证方法一致

7.4 离线识别引擎类型

离线识别引擎有两种类型，对应不同工作模式：

引擎类型	描述
语法模式	该模式需要配置资源文件和语法文件， 离线引擎将根据传入的语法文件进行相关的匹配识别
输入法模式	该模式需要配置资源文件，进行普通模式的识别， 即对输入的语音进行识别，类似与在线识别