
安徽科大讯飞信息科技股份有限公司

ANHUI USTC IFLYTEK CO.,LTD

iFlytek Speech Suite For Multi-platform

讯飞汽车语音套件开发手册

安徽科大讯飞信息科技股份有限公司
ANHUI USTC IFLYTEK CO., LTD.

重要声明

版权声明

版权所有 © 2013, 安徽科大讯飞信息科技股份有限公司, 保留所有权利。

商标声明

安徽科大讯飞信息科技股份有限公司的产品是安徽科大讯飞信息科技股份有限公司专有。在提及及其他公司及其产品时将使用各自公司所拥有的商标, 这种使用的目的仅限于引用。本文档可能涉及安徽科大讯飞信息科技股份有限公司的专利 (或正在申请的专利)、商标、版权或其他知识产权, 除非得到安徽科大讯飞信息科技股份有限公司的明确书面许可协议, 本文档不授予使用这些专利 (或正在申请的专利)、商标、版权或其他知识产权的任何许可协议。

不作保证声明

安徽科大讯飞信息科技股份有限公司不对此文档中的任何内容作任何明示或暗示的陈述或保证, 而且不对特定目的的适销性及适用性或者任何间接、特殊或连带的损失承担任何责任。本手册内容若有变动, 恕不另行通知。本手册例子中所用的公司、人名和数据若非特别声明, 均属虚构。未得到安徽科大讯飞信息科技股份有限公司明确的书面许可, 不得为任何目的、以任何形式或手段 (电子的或机械的) 复制或传播手册的任何部分。

保密声明

本文档 (包括任何附件) 包含的信息是保密信息。接收人了解其获得的本文档是保密的, 除用于规定的目的外不得用于任何目的, 也不得将本文档泄露给任何第三方。

本软件产品受最终用户许可协议 (EULA) 中所述条款和条件的约束, 该协议位于产品文档和/或软件产品的联机文档中, 使用本产品, 表明您已阅读并接受了 EULA 的条款。

版权所有 © 安徽科大讯飞信息科技股份有限公司

Copyright © 2013 ANHUI USTC iFLYTEK CO., LTD.

修订文档历史记录

日期	版本	说明	编者
2014-9-8	3.52	1, 删除单唤醒模块 2. 添加芯片授权接口	蒋亚冲
2014-2-24	3.42	1, TTS 进度回调已实现, 更改 ISSTTSGetAudioData 接口参数	孟操
2014-2-24	3.32	1, 增加了硬件加密检测消息, 个性化数据 上传云端支持热词 (hotwords)。	孟操
2014-2-15	3.22	1, 增加附录六, 对识别模式进行了说明。	孟操
2014-1-8	3.12	1, 识别增加参数设置和主动加载大资源 的接口 2, 修改了识别接口 ISSSRCreate, 加了一个 资源加载方式的参数	孟操
2013-12-7	2.12	1, 增加套件授权接口	孟操
2013-12-6	1.12	1, 编译语法接口 ISSSRUploadDict, 增加 bOnlyUploadToCloud (是否只上传到 网络) 参数	孟操
2013-12-2	1.02	1, 修改了合成回调的名称; 2, 去掉 ISSSRCreate 接口返回内存不足 返回值	孟操
2013-11-7	1.01	1, 语音套件接口基础版本	孟操

目录

前言	3
1. 使用说明.....	4
1.1. 名词和缩略语.....	4
1.2. 功能介绍.....	4
1.3. 开发说明.....	4
1.4. 支持平台.....	5
2. 语音识别.....	5
2.1 接口简介.....	5
2.1.1 函数列表.....	5
2.1.2 功能说明.....	5
2.2 识别参数说明.....	6
2.3 接口说明.....	6
2.3.1 创建识别句柄.....	6
2.3.2 销毁识别句柄.....	7
2.3.3 开始语音识别.....	7
2.3.4 输入录音数据.....	9
2.3.5 停止录音.....	9
2.3.6 取消识别.....	10
2.3.7 编译语法.....	10
2.3.8 参数设置.....	12
2.3.9 加载本地转写资源.....	13
2.4 回调接口声明.....	13
2.4.1 识别回调接口.....	13
2.5 接口调用流程.....	15
3. 语音合成.....	15
3.1 接口简介.....	15
3.1.1 函数列表.....	15
3.1.2 功能说明.....	16
3.2 合成参数说明.....	16
3.3 接口说明.....	16
3.3.1 创建语音合成资源句柄.....	16
3.3.2 销毁语音合成资源句柄.....	17
3.3.3 创建语音合成句柄.....	17
3.3.4 销毁语音合成句柄.....	18
3.3.5 参数设置.....	18
3.3.6 合成指定文本.....	19
3.3.7 获取合成音频.....	19
3.3.8 取消合成.....	20
3.4 回调接口声明.....	21
3.4.1 合成资源初始化成功.....	21

3.5	接口调用流程.....	21
4.	多唤醒.....	22
4.1	接口介绍.....	22
4.1.1	函数列表.....	22
4.1.2	功能说明.....	22
4.2	多唤醒参数说明.....	22
4.3	接口说明.....	22
4.3.1	创建多语音唤醒句柄.....	22
4.3.2	销毁多语音唤醒句柄.....	23
4.3.3	设置多语音唤醒门限.....	23
4.3.4	启动多语音唤醒.....	24
4.3.5	输入录音数据.....	24
4.3.6	停止多语音唤醒.....	25
4.4	回调接口声明.....	25
4.4.1	多唤醒成功.....	25
4.5	接口调用流程.....	26
5.	套件软件授权.....	27
5.1	接口简介.....	27
5.1.1	函数列表.....	27
5.1.2	功能说明.....	27
5.2	授权参数说明.....	27
5.3	接口说明.....	27
5.3.1	设置机器码.....	27
5.3.2	获取激活码.....	28
5.3.3	激活.....	28
6.	套件芯片授权.....	29
6.1	接口简介.....	29
6.1.1	函数列表.....	29
6.1.2	功能说明.....	29
6.2	接口说明.....	30
6.2.1	设置 获取授权芯片信息的回调.....	30
6.3	回调接口说明.....	30
6.3.1	获取授权芯片信息.....	30
6.4	接口调用流程.....	31
7.	接口使用示例.....	31
7.1	场景 1	31
7.2	场景 2	32
8.	技术支持.....	34
附录一:	个性发音人列表.....	35
附录二:	错误码列表.....	37
附录三:	识别结果说明.....	40
附录四:	个性化数据上传格式.....	41
附录五:	个性化命令词使用说明.....	42
附录六:	语音识别模式简要说明.....	44

前言

欢迎使用讯飞汽车语音套件！

讯飞汽车语音套件是基于讯飞公司已有的ISP和IMS产品，开发出的一套符合车载用户使用的语音应用开发SDK，提供语音识别、语音合成等服务，为车载语音应用开发者提供方便易用的开发接口，使得用户能够基于该开发接口进行多种语音应用开发。其主要功能有：

- 1) 实现基于HTTP协议的语音应用服务器，集成讯飞公司最新的语音引擎，支持语音合成、语音听写、语音识别等服务；
- 2) 提供基于移动平台和PC上的语音客户端子系统，内部集成音频处理和音频编解码模块，提供关于语音合成、语音听写、语音识别完善的API。

科大讯飞作为中国最大的智能语音技术提供商，在智能语音技术领域有着长期的研究积累，并在语音合成、语音识别、口语评测、自然语言处理等多项技术上拥有国际领先的成果。科大讯飞是我国唯一以语音技术为产业化方向的“国家863计划成果产业化基地”、“国家规划布局内重点软件企业”、“国家高技术产业化示范工程”，并被原信息产业部确定为中文语音交互技术标准工作组组长单位，牵头制定中文语音技术标准。

2003年、2011年，科大讯飞两次荣获“国家科技进步奖”。2005年、2011年两次获得中国信息产业自主创新最高荣誉“信息产业重大技术发明奖”。2006年至2012年，已连续七届英文语音合成国际大赛（Blizzard Challenge）荣获第一名。2008年至2011年，连续四年在国际说话人、语种识别评测大赛中名列前茅。2011年，“国家智能语音高新技术产业化基地”、“语音及语言信息处理国家工程实验室”相继落户合肥，有利于进一步汇聚产业资源，提升科大讯飞产业龙头地位。

基于拥有自主知识产权的世界领先智能语音技术，科大讯飞已推出从大型电信级应用到小型嵌入式应用，从电信、金融等行业到企业和消费者用户，从手机到车载，从家电到玩具，能够满足不同应用环境的多种产品。科大讯飞已占有中文语音技术市场70%以上市场份额，开发伙伴超过5000家，以讯飞为核心的中文语音产业链已初具规模。

1. 使用说明

1.1. 名词和缩略语

☐ ISS (Iflytek Speech Suite)

讯飞语音套件(Iflytek Speech Suite, ISS), 是讯飞面向车载领域开发的语音服务套件。

☐ SR (Speech Recognition)

语音识别(Speech Recognition, SR)是在语音听写基础上的一种识别技术, 着重于对某一领域或特定文法的识别, 且识别结果和上传的内容紧密相关, 如果用户想自定义识别的范围, 在使用前, 需上传相关的命令词列表或语法文件。

☐ NLP (Natural Language Processing)

自然语言理解(Natural Language Processing, NLP)是人工智能 (AI) 的一个子领域, 也是人工智能中最为困难的问题之一, 它包含对语义的判断, 使计算机不仅能够识别出语音内容, 还能了解用户的意图, 帮助用户准确地搜索出想要的结果, 最终目的是让计算机做到像人一样的思考。

☐ TTS (Text to Speech)

语音合成(Text To Speech, TTS)是一种能够将任意文字实时转换为连续的自然语音的技术, 能够在任何时间、任何地点, 向任何人提供语音信息服务的高效便捷手段, 非常符合信息时代海量数据、动态更新和个性化查询的需求。

☐ IVW (Iflytek voice wakeUp)

语音唤醒 (Iflytek voice wakeUp, IVW) 是使用语音或音频命令词唤醒待机中的设备的技术, 能够通过语音技术识别处预先设置的命令词唤醒待机中的设备, 实现设备的无手操作。

1.2. 功能介绍

讯飞汽车语音套件以云+端的方式提供语音识别、语义理解、语音合成、语音唤醒等服务。为各个平台提供统一的解决方案, 退出基础产品服务, 减少跨平台移植工作。

主要提供语音识别 (SR)、语义理解 (NLP)、语音合成 (TTS)、和语音唤醒 (IVW) 等服务, 语音识别支持语法识别、语音听写功能, 支持本地、云端、混合三种模式。

1.3. 开发说明

- 1、 本文档定义了科大讯飞的语音识别、语音合成和语音唤醒的使用说明、体系结构。

- 其适用的读者为使用语音 SDK 进行开发的产品设计师、软件工程师,通过阅读本文档,读者可以掌握如何集成和使用语音识别、语音合成和语音唤醒服务。
- 使用语音服务,需要经过我们的授权,请到“<http://open.voicecloud.cn>”注册成为语音云开发者,并为所开发的软件申请 appid;
- 如果开发者申请了多个 appid,不同应用需要使用与之相对应的 SDK libs,因为 SDK 中已包含该应用相关信息,导入错误的开发包将难以保证正常的语音服务;
- 已申请 appid 的开发者,将可以免费使用语音识别、语音合成服务,但是对于文档中描述的语义等业务不具备使用权限,需要和我们签订商务协议并获得授权方可使用;
- 具体业务,请参考各部分开发章节。

1.4. 支持平台

- 支持 wince、Linux 平台。

2. 语音识别

2.1 接口简介

2.1.1 函数列表

□ 提供如下接口:

函数名称	功能简介
ISSSRCreate	创建识别句柄
ISSSRCreateW	创建识别句柄
ISSSRDestroy	销毁语音识别句柄
ISSSRStart	开始一次语音识别
ISSSRAppendAudioData	输入录音数据
ISSSREndAudioData	停止传入录音
ISSSRStop	停止此次识别
ISSSRUpLoadDict	编译语法
ISSSRSetParam	参数设置
ISSSRLoadBigSrRes	加载本地转写资源

2.1.2 功能说明

- 识别包括语法识别、语音听写;
- 识别支持本地、网络及混合(网络+本地)三种模式;
- 识别目前为单路会话,不支持同时启动多路;

- 4、 识别结果为 XML 格式，可根据业务进行解析。

2.2 识别参数说明

无。

2.3 接口说明

2.3.1 创建识别句柄

□ 函数原型

ISSErrID [ISSAPI](#) ISSSRCreate(HISSR* phISSRObj, const char* szResourceDir, Proc_OnMsgProc pfnOnMsgProc, void* pUsrArg, int bLoadBigSrResDelay);

ISSErrID [ISSAPI](#) ISSSRCreateW (HISSR* phISSRObj, const wchar_t* szResourceDir, Proc_OnMsgProc pfnOnMsgProc, void* pUsrArg, int bLoadBigSrResDelay);

□ 功能说明

- 1、 创建识别句柄，在调用 ISSTTCreate 创建识别句柄前，必须已经调用 ISSSetMachineCode 设置机器码；

□ 参数说明

参数名	参数解释
phISSRObj [out]	识别句柄；
szResourceDir [in]	资源目录，一般情况下：ISSSRCreate 版本，在 windows 中文环境下为 GBK 编码，在 linux 环境下为 UTF-8 编码；ISSSRCreateW 版本，在 windows vc 环境下为 UNICODE-16 编码，在 linux gcc 环境下为 UNICODE-32 编码。
pfnOnMsgProc [in]	识别消息回调接口，请参考 识别回调接口 声明，不可以传 NULL；
pUsrArg [in]	用户自定义参数
bLoadBigSrResDelay[in]	是否延迟加载本地转写大资源。

□ 返回值

- 1、ISS_SUCCESS：成功创建语音识别句柄；
- 2、ISSErrID：语音识别引擎返回的错误码，错误代码参见[附录二](#)；
- 3、ISS_ERROR_FILE_NOT_FOUND：没有找到相应的资源文件；
- 4、ISS_ERROR_INVALID_PARA：无效的参数值；
- 5、ISS_ERROR_MACHINE_CODE_NOT_SET：还未设置机器码；

2.3.2 销毁识别句柄

□ 函数原型

ISSErrID *ISSAPI* ISSSRDestroy(HISSSR hISSSRObj);

□ 功能说明

- 1、销毁语音识别句柄；

□ 参数说明

参数名	参数解释
hISSSRObj [in]	识别句柄。

□ 返回值

- 1、ISS_SUCCESS: 成功销毁语音识别句柄并释放资源；
- 2、ISSErrID: 语音识别引擎返回的错误码，错误代码参见[附录二](#)；
- 3、ISS_ERROR_INVALID_HANDLE: 无效语音识别句柄。

2.3.3 开始语音识别

□ 函数原型

ISSErrID *ISSAPI* ISSSRStart(HISSSR hISSSRObj, const char* szScene, int iMode, const char* szCmd);

□ 功能说明

- 1、开始一次语音识别；
- 2、如果已经处于识别状态,会停止之前的识别,重新开始这次新的识别；
- 3、当 i_Mode = ISS_SR_MODE_LOCAL_CMDLIST 时, 编译命令词表的动作是在套件维护的线程中完成的, 不会阻塞 ISSSRStart 接口。
- 4、在 Proc_OnInitStatus 返回成功后才能调用 ISSSRStart 开启一次识别。
- 5、ISSSRStart 返回 ISS_SUCCESS 后, 就可以调用 ISSSRAppendAudioData 送录音了。
- 6、szCmd 的示例:

```
i_Mode = ISS_SR_MODE_LOCAL_CMDLIST 时:
{
    "grm": [
        {
            "dictname": "cmd",
            "dictcontent": [
                { "name": "第一个", "id": 0 },
                { "name": "第二个", "id": 1 },
                { "name": "第三个", "id": 2 },
                { "name": "最后一个", "id": 3 }
            ]
        }
    ]
}
```

```
i_Mode = ISS_SR_MODE_LOCAL_NLP 时:
{
    "nlptext": "今天天气怎么样? "
}
```

□ 参数说明

参数名	参数解释
hISSRObj [in]	识别句柄
szScene [in]	指定场景，限定识别范围 支持： ISS_SR_SCENE_ALL、 ISS_SR_SCENE_POI、 ISS_SR_SCENE_CONTACTS、 ISS_SR_SCENE_SELECT、 ISS_SR_SCENE_CONFIRM、 ISS_SR_SCENE_CMDLIST_WITHALL (iMode 为 ISS_SR_MODE_LOCAL_CMDLIST 时，本地使用全场景，并开 启云端识别)、 ISS_SR_SCENE_SELECT_MVW(此时忽略 iMode 参数)、 ISS_SR_SCENE_CONFIRM_MVW(此时忽略 iMode 参数)、 ISS_SR_SCENE_ANSWER_CALL_MVW(此时忽略 iMode 参 数)
iMode [in]	i_Mode = ISS_SR_MODE_CLOUD_REC,纯网络端识别; i_Mode = ISS_SR_MODE_LOCAL_REC,纯本地端识别; i_Mode = ISS_SR_MODE_MIX_REC,云加端识别; i_Mode = ISS_SR_MODE_LOCAL_CMDLIST,本地命令词 识别 ,此时忽略 szScene 参数; i_Mode = ISS_SR_MODE_LOCAL_NLP,纯本地语义,,此时 忽略 szScene 参数.
szCmd[in]	i_Mode = ISS_SR_MODE_LOCAL_CMDLIST 时,为本地命 令词识别所用命令词列表,json 格式; i_Mode = ISS_SR_MODE_LOCAL_NLP 时,为送给语义解 析的字符串 UTF8 编码,json 格式;

□ 返回值

- 1、ISS_SUCCESS: 成功启动一次语音识别;
- 2、ISSErrID: 语音识别引擎返回的错误码，错误代码参见[附录二](#)。
- 3、ISS_ERROR_INVALID_HANDLE: 无效语音识别句柄;
- 4、ISS_ERROR_INVALID PARA: 无效的数值，输入了错误的场景或识别模式;
- 5、ISS_ERROR_INVALID_JSON_FMT: 输入的 Json 格式有问题;
- 6、ISS_ERROR_INVALID_JSON_INFO: 没有从 Json 输入中提取到必要的语法信息。
- 7、ISS_ERROR_BUILDING_GRM:

配置文件 CouldStartLocalRecogniseWhenBuildLocalGrm 设为 0 后，如启动本地识
别时 正在构建本地语法，则返回这一错误码

2.3.4 输入录音数据

□ 函数原型

ISSErrID [ISSAPI](#) ISSSRAppendAudioData(HISSSR hISSSRObj, short* pSamples, unsigned int nNumberOfToWrite, unsigned int* pNumberOfWritten);

□ 功能说明

- 1、输入录音数据,支持输入 16K 采样率、S16-LE、单声道的 PCM 录音;
- 2、必须处于识别启动状态下才能输入音频数据。
为了避免语音识别引擎内部缓冲区溢出,增加了 pNumberOfWritten 参数.用户程序在发生 *pNumberOfWritten != nNumberOfToWrite 时,可以等待一会再写入或者丢弃这个缓冲区数据(一般智能手机,车机处理器不会出现语音识别处理跟不上效率的情况,有可能一次写入的数据太大,或者是写测试程序时).
请注意有可能在写入数据期间,语音识别引擎回调识别函数.

□ 参数说明

名称	参数取值范围
hISSSRObj [in]	识别句柄
pSamples[in]	输入的音频 buffer 地址
nNumberOfToWrite[in]	要写入的采样点的个数,非字节数
pNumberOfWritten[out]	实际写入的采样点的个数

□ 返回值

- 1、ISS_SUCCESS: 成功写入录音,但写入的录音的大小可能为 0,需要判断 *pNumberOfWritten != nNumberOfToWrite 来判断是否全部写入;
- 2、ISSErrID: 语音识别引擎返回的错误码,错误代码参见[附录二](#);
- 3、ISS_ERROR_INVALID_HANDLE: 无效语音识别句柄;
- 4、ISS_ERROR_INVALID_PARA: 无效的参数值;
- 5、ISS_ERROR_INVALID_CALL: 错误调用(还未调用 ISSSRStart,识别还没有开始);

2.3.5 停止录音

□ 函数原型

ISSErrID [ISSAPI](#) ISSREndAudioData(HISSSR hISSSRObj);

□ 功能说明

- 1、用户点击了界面上的“我说完了”,表示主动停止传送录音,不需要再写入数据。

□ 参数说明

参数名	参数解释
hISSSRObj [in]	识别句柄。

□ 返回值

- 1、ISS_SUCCESS: 成功;
- 2、ISSErrID: 语音识别引擎返回的错误码, 错误代码参见[附录二](#);
- 3、ISS_ERROR_INVALID_HANDLE: 无效语音识别句柄;
- 4、ISS_ERROR_INVALID_CALL: 错误调用(还未调用 ISSSRStart, 识别还没有开始);
- 5、ISS_ERROR_NO_SPEECH: 没有检测到使用者语音, 识别引擎已经停止, 不需要再写入数据。

2.3.6 取消识别

□ 函数原型

ISSErrID **ISSAPI** ISSSRStop(HISSSR hISSSRObj);

□ 功能说明

- 1、 停止此次语音识别（取消此次识别操作）。

□ 参数说明

参数名	参数解释
hISSSRObj [in]	识别句柄。

□ 返回值

- 1、ISS_SUCCESS: 成功停止了此次识别;
- 2、ISSErrID: 语音识别引擎返回的错误码, 错误代码参见[附录二](#);
- 3、ISS_ERROR_INVALID_HANDLE: 无效语音识别句柄;
- 4、ISS_ERROR_INVALID_CALL: 错误调用(还未调用 ISSSRStart, 识别还没有开始);

2.3.7 编译语法

□ 函数原型

ISSErrID **ISSAPI** ISSSRUploadDict(HISSSR hISSSR, const char *szList, int

bOnlyUploadToCloud);

□ 功能说明

- 1、 异步上传个性化词典。
- 2、 本地个性化词典支持 songs、singer、contact、apps,
- 3、 云端个性化词典支持 contact、hotwords.
- 4、 个性化词典上传是否成功, 会通过回调返回消息:
- 5、 ISS_SR_MSG_UploadDictToLocalStatus 表示上传到本地是否成功;
- 6、 ISS_SR_MSG_UploadDictToCloudStatus 表示上传到云端是否成功;
- 7、 异步接口。在创建好句柄后, 不用等 ISS_SR_MSG_InitStatus 的消息, 就可以调用 ISSSRUploadDict 上传词典了。
如果最后一个参数 bOnlyUploadToCloud 为 0, 则:
每次调用 ISSSRUploadDict 接口后,
返回消息 ISS_SR_MSG_UploadDictToLocalStatus, 标识本地个性化数据上传是否成功,
返回消息 ISS_SR_MSG_UploadDictToCloudStatus, 表示云端个性化数据上传是否成功

功;

如果最后一个参数 `bOnlyUploadToCloud` 为非 0, 则:

每次调用 `ISSSRUploadDict` 接口后,

返回消息 `ISS_SR_MSG_UploadDictToCloudStatus`, 表示云端个性化数据上传是否成功.

语法示例:

```
{
  "grm": [
    {
      "dictname": "contact",
      "dictcontent": [
        { "name": "科大讯飞", "id": 0 },
        { "name": "白雪", "id": 1 },
        { "name": "刘俊峰", "id": 2 },
        { "name": "白亚伟", "id": 3 }
      ]
    },
    {
      "dictname": "singers",
      "dictcontent": [
        { "name": "凤凰传奇", "id": 0 },
        { "name": "那英", "id": 1 },
        { "name": "杨坤", "id": 2 },
        { "name": "S.H.E", "id": 3 }
      ]
    },
    {
      "dictname": "songs",
      "dictcontent": [
        { "name": "恭喜发财", "id": 0 },
        { "name": "偏爱", "id": 1 },
        { "name": "铿锵玫瑰", "id": 2 },
        { "name": "孤枕难眠", "id": 3 }
      ]
    },
    {
      "dictname": "apps",
      "dictcontent": [
        { "name": "讯飞语点", "id": 0 },
        { "name": "愤怒的小鸟", "id": 1 },
        { "name": "浏览器", "id": 2 }
      ]
    }
  ]
}
```

```

"dictname": "hotwords",
"dictcontent": [
    { "name": "导航到沸腾渔乡", "id": 0 },
    { "name": "我想去沸腾渔乡", "id": 1 },
    { "name": "导航到锦明大厦", "id": 2 },
    { "name": "我想去锦明大厦", "id": 3 }
]
}
]
}

```

□ 参数说明

参数名	参数解释
hISSR [in]	识别句柄
szList [in]	语法信息, Json 格式, 格式可参见 附录四 , UTF8 编码
bOnlyUploadToCloud [in]	是否只上传词典到云端

□ 返回值

- 1、ISS_SUCCESS: 成功添加了一个异步编语法的任务;
- 2、ISSErrID: 语音识别引擎返回的错误码, 错误代码参见[附录二](#);
- 3、ISS_ERROR_INVALID_HANDLE: 无效语音识别句柄;
- 4、ISS_ERROR_INVALID_PARAM: 无效的参数值;

2.3.8 参数设置

□ 函数原型

ISSErrID [ISSAPI](#) ISSRSetParam(HISSR hISSRObj, const char* szParam, const char* szParamValue);

□ 功能说明

- 1、参数设置。

□ 参数说明

参数名	参数解释
hISSRObj [in]	识别句柄。
szParam [in]	参数名称, 支持 ISS_SR_PARAM_XXXX
szParamValue [in]	参数值。

□ 返回值

- 1、ISS_SUCCESS: 成功设置了参数;
- 2、ISSErrID: 语音识别引擎返回的错误码, 错误代码参见[附录二](#);
- 3、ISS_ERROR_INVALID_HANDLE: 无效语音识别句柄;
- 4、ISS_ERROR_INVALID_PARAM: 无效的参数值;

2.3.9 加载本地转写资源

□ 函数原型

ISSErrID [ISSAPI](#) ISSRLoadBigSrRes(HISSR hISSRObj);

□ 功能说明

- 1、主动加载本地转写较大资源，加载是否成功会通过消息 ISS_SR_MSG_LoadBigSrResStatus 返回。

□ 参数说明

参数名	参数解释
hISSRObj [in]	识别句柄。

□ 返回值

- 1、ISS_SUCCESS: 成功设置了参数;
- 2、ISSErrID: 语音识别引擎返回的错误码，错误代码参见[附录二](#);
- 3、ISS_ERROR_INVALID_CALL: 错误调用;

2.4 回调接口声明

2.4.1 识别回调接口

□ 接口声明

[typedef void \(ISSCALLBACK *Proc_OnMsgProc\)\(void* pUsrArg,unsigned int uMsg,unsigned int wParam,void *lParam\);](#)

□ 接口说明

- 1、识别回调接口，引擎通过 Proc_OnMsgProc 回调发出各种消息;

□ 参数说明

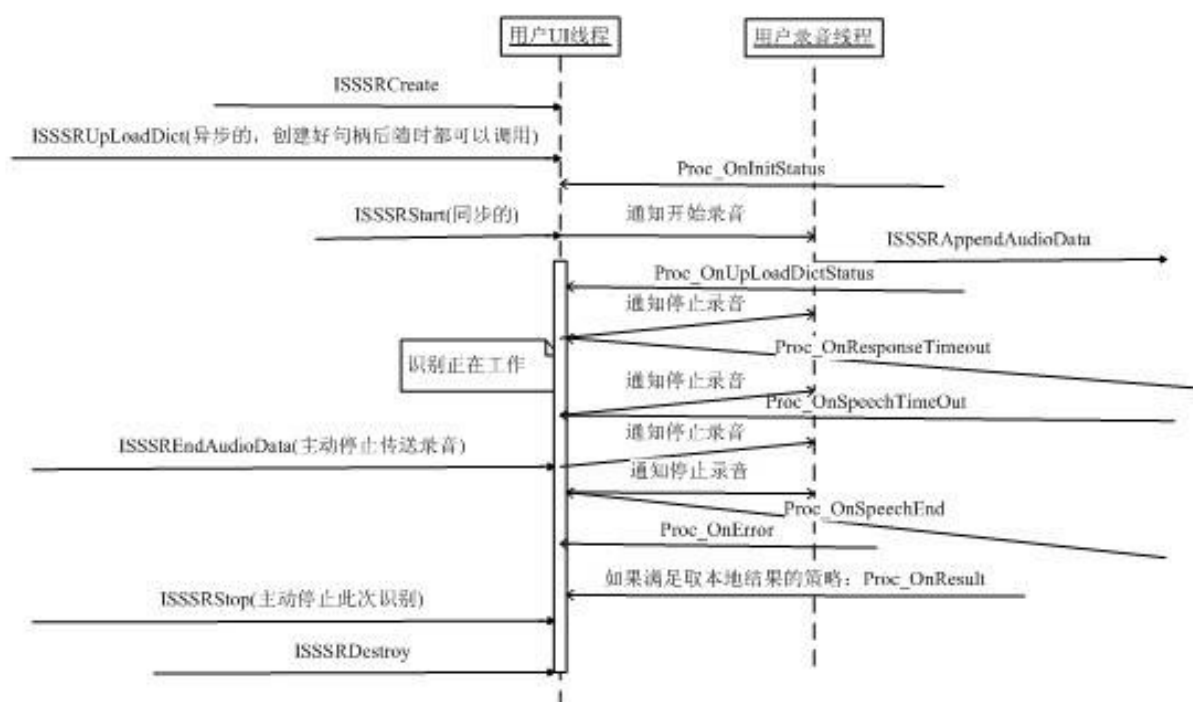
参数名	参数解释
pUsrArg	用户自定义参数

uMsg	消息类型，说明如下	
	event	说明
	ISS_SR_MSG_InitStatus	wParam: ISS_SUCCESS:初始化成功 ISS_ERROR_FAIL:初始化失败 ISS_ERROR_OUT_OF_MEMORY:内存不足 lParam: NULL
	ISS_SR_MSG_UplLoadDictToLocalStatus	wParam: ISS_SUCCESS:本地个性化数据上传成功; ISS_ERROR_INVALID_JSON_FMT:编语法时,输入的 Json 格式有问题 ISS_ERROR_INVALID_JSON_INFO:编语法时,没有从 Json 输入中提取到必要的语法信息 lParam: NULL
	ISS_SR_MSG_UplLoadDictToCloudStatus	wParam: ISS_SUCCESS:云端个性化数据上传成功; ISS_ERROR_INVALID_JSON_FMT:编语法时,输入的 Json 格式有问题 ISS_ERROR_INVALID_JSON_INFO:编语法时,没有从 Json 输入中提取到必要的语法信息 lParam: NULL
	ISS_SR_MSG_VolumeLevel	wParam:音量大小，最小为 0，最大为 931 lParam: NULL
	ISS_SR_MSG_ResponseTimeout	wParam: NULL lParam: NULL
	ISS_SR_MSG_SpeechStart	wParam: NULL lParam: NULL
	ISS_SR_MSG_SpeechTimeOut	wParam: NULL lParam: NULL
	ISS_SR_MSG_SpeechEnd	wParam: NULL lParam: NULL
	ISS_SR_MSG_Error	wParam: 错误码，错误代码参见 附录二 lParam: const char*类型，错误信息
	ISS_SR_MSG_Result	wParam: NULL lParam: const char*类型，识别结果
	ISS_SR_MSG_LoadBigSrResStatus	wParam: ISS_ERROR_OUT_OF_MEMORY:系统内存不足;ISS_ERROR_FAIL:加载失败 ISS_SUCCESS:成功加载; lParam: NULL
	ISS_SR_MSG_ErrorDecodingAudio	针对硬件授权的检测机制，即使用了硬件加密的引擎，在送入录音时会对音频进行检测，如果是非法音频时会发出此消息。

□ 返回值

1、无

2.5 接口调用流程



3. 语音合成

3.1 接口简介

3.1.1 函数列表

□ 提供如下接口：

函数名称	功能简介
ISSTTSInitRes	创建语音合成资源句柄
ISSTTSInitResW	创建语音合成资源句柄
ISSTTSUnInitRes	销毁语音合成资源句柄
ISSTTSCreate	创建语音合成句柄
ISSTTSDestroy	销毁语音合成句柄
ISSTTSSetParam	设置合成参数
ISSTTSStart	合成指定文本

ISSTTSGetAudioData	取音频数据
ISSTTSStop	停止合成

3.1.2 功能说明

- 语音合成支持播放、合成音频流；

3.2 合成参数说明

- 可选参数如下表所示，如不设置，使用默认值：

名称	参数	取值范围
发音人	ISS_TTS_PARAM_SPEAKER	请参考附件发音人列表
语速	ISS_TTS_PARAM_VOICE_SPEED	<pre>#define ISS_TTS_SPEED_MIN (-32768) /* 最慢语速 */ #define ISS_TTS_SPEED_NORMAL_DEFAULT (0) /* 正常语速 默认值 */ #define ISS_TTS_SPEED_MAX (+32767) /* 最快语速 */</pre>
声调	ISS_TTS_PARAM_VOICE_PITCH	<pre>#define ISS_TTS_PITCH_MIN (-32768) /* 最低声调 */ #define ISS_TTS_PITCH_NORMAL_DEFAULT (0) /* 正常声调 默认值 */ #define ISS_TTS_PITCH_MAX (+32767) /* 最高声调 */</pre>
音量	ISS_TTS_PARAM_VOLUME	<pre>#define ISS_TTS_VOLUME_MIN (-32768) /* 最小音量 */ #define ISS_TTS_VOLUME_NORMAL (0) /* 正常音量 */ #define ISS_TTS_VOLUME_MAX_DEFAULT (+32767) /* 最大音量 默认值 */</pre>

3.3 接口说明

3.3.1 创建语音合成资源句柄

□ 函数原型

ISSErrID [ISSAPI](#) ISSTTInitRes(HISSTTSRES*phISSTTSRES, const char*szResourceDir,int

iResMode);

ISSErrID [ISSAPI](#) ISSTTSInitResW(HISSTTSRES* phISSTTSRES,const wchar_t* szResourceDir, int

iResMode);

☐ 功能说明

- 1、创建语音合成资源句柄;

☐ 参数说明

参数名	参数解释
phISSTTSRES [out]	TTS 资源句柄
szResourceDir [in]	资源目录,一般情况下: ISSTTSInitRes 版本, 在 windows 中文环境下为 GBK 编码,在 linux 环境下为 UTF-8 编码; ISSTTSInitResW 版本, 在 windows vc 环境下为 UNICODE-16 编码,在 linux gcc 环境下为 UNICODE-32 编码。
iResMode [in]	载入资源的方式: iResMode = 0 打开文件句柄; iResMode = 1 直接载入内存。

☐ 返回值

- 1、ISS_SUCCESS: 成功创建语音合成资源句柄;
- 2、ISSErrID: 语音合成引擎返回的错误码, 错误代码参见[附录二](#);
- 3、ISS_ERROR_FILE_NOT_FOUND: 没有找到相应的资源文件;
- 4、ISS_ERROR_INVALID_PARA: 无效的参数值;
- 5、ISS_ERROR_OUT_OF_MEMORY: 内存不够。

3.3.2 销毁语音合成资源句柄

☐ 函数原型

ISSErrID [ISSAPI](#) ISSTTSUninitRes (HISSTTSRES hISSTTSRES);

☐ 功能说明

- 1、销毁语音合成资源句柄;

☐ 参数说明

参数名	参数解释
hISSTTSRES [in]	TTS 资源句柄。

☐ 返回值

- 1、ISS_SUCCESS: 成功销毁语音合成资源句柄;
- 2、ISSErrID: 语音合成引擎返回的错误码, 错误代码参见[附录二](#);
- 3、ISS_ERROR_INVALID_PARA: 无效的参数值。

3.3.3 创建语音合成句柄

☐ 函数原型

ISSErrID *ISSAPI* ISSTTSCreate(HISSTTS* phISSTTS, HISSTTSRES hISSTTSRES,
Proc_OnTTSDDataReady pcbOnTTSDDataReady, void* pUsrArg);

☐ 功能说明

- 1、创建语音合成句柄；

☐ 参数说明

参数名	参数解释
phISSTTS [out]	TTS 句柄
hISSTTSRES[in]	TTS 资源句柄
pcbOnTTSDDataReady[in]	用户 OnTTSDDataReady 回调函数指针，详细说明见 3.4
pUsrArg[in]	用户自定义参数，将在 Callback 中提供给用户

☐ 返回值

- 1、ISS_SUCCESS: 成功创建语音合成句柄；
- 2、ISSErrID: 语音合成引擎返回的错误码，错误代码参见[附录二](#)；
- 3、ISS_ERROR_INVALID_PARA: 无效的参数值；
- 4、ISS_ERROR_OUT_OF_MEMORY: 内存不够。

注：可以使用同一 TTS 资源句柄创建多个 TTS 示例。

3.3.4 销毁语音合成句柄

☐ 函数原型

ISSErrID *ISSAPI* ISSTTSDestroy(HISSTTS hISSTTS);

☐ 功能说明

- 1、通过此接口销毁语音合成句柄；

☐ 参数说明

名称	参数取值范围
hISSTTS [in]	TTS 句柄

☐ 返回值

- 1、ISS_SUCCESS: 成功销毁语音合成句柄并释放资源；
- 2、ISSErrID: 语音合成引擎返回的错误码，错误代码参见[附录二](#)；
- 3、ISS_ERROR_INVALID_HANDLE: 无效语音合成句柄。

3.3.5 参数设置

☐ 函数原型

ISSErrID *ISSAPI* ISSTTSSetParam(HISSTTS hISSTTS,int iParamID,int iParamValue);

☐ 功能说明

- 1、设置参数；

☐ 参数说明

参数名	参数解释
hISSTTS [in]	TTS句柄。
iParamID [in]	TTS参数索引。
iParamValue [in]	相应参数值。

□ 返回值

- 1、ISS_SUCCESS: 成功设置了参数;
- 2、ISSErrID: 语音合成引擎返回的错误码, 错误代码参见[附录二](#);
- 3、ISS_ERROR_INVALID_HANDLE: 无效语音合成句柄;
- 4、ISS_ERROR_INVALID_PARA: 无效的参数类型;
- 5、ISS_ERROR_INVALID_PARA_VALUE: 无效的参数值。

3.3.6 合成指定文本

□ 函数原型

ISSErrID **ISSAPI** ISSTTSStart(HISSTTS hISSTTS,const void* pText,int iSize,ISSTTSCodePage iTTsCodePage);

□ 功能说明

- 1、合成指定文本,如果引擎已经在合成一段文本时,又调用 ISSTTSStart 去合成另一段文本,之前一次的合成任务会直接丢掉。
- 2、传入文本可以使用文本标注来是朗读更加符合预期,具体使用方式,请参考《Aisound5 简单文本标注用户手册.pdf》。

□ 参数说明

参数名	参数解释
hISSTTS [in]	TTS 句柄
pText [in]	合成文本 Buffer
iSize [in]	合成文本 Buffer 大小
iTTsCodePage [in]	合成文本的代码页

□ 返回值

- 1、ISS_SUCCESS: 成功开始一次语音合成;
- 2、ISSErrID: 语音合成引擎返回的错误码, 错误代码参见[附录二](#);
- 3、ISS_ERROR_INVALID_HANDLE: 无效语音合成句柄;
- 4、ISS_ERROR_INVALID_PARA: 无效的参数值。

3.3.7 获取合成音频

□ 函数原型

ISSErrID **ISSAPI** ISSTTSGetAudioData(HISSTTS hISSTTS,void *pOutBuffer,unsigned int iBufferFrames,unsigned int *piBufferFramesGot,unsigned int *piProgBegin,unsigned int *piProgLen);

□ 功能说明

- 1、取音频数据，支持获取 16000 采样率、S16-LE、单声道的音频，一个采样点的大小为 2Bytes；
- 2、同步阻塞函数。

□ 参数说明

参数名	参数解释
hISSTTS [in]	TTS 句柄
pOutBuffer [in]	音频输出 Buff 的地址
iBufferFrames [in]	期望获取的采样点数,非字节数
piBufferFramesGot [out]	实际输出的采样点数
piProgBegin[out]	当前进度的偏移量
piProgLen[out]	当前进度长度

□ 返回值

- 1、ISS_SUCCESS: 成功获取了合成的语音；
- 2、ISSErrID: 语音合成引擎返回的错误码，错误代码参见[附录二](#)；
- 3、ISS_ERROR_INVALID_HANDLE: 无效语音合成句柄；
- 4、ISS_ERROR_INVALID PARA: 无效的参数值；
- 5、ISS_ERROR_TTS_STOPPED: 错误调用，合成已经结束，却调用 ISSTTSGetAudioData 取录音。
- 6、ISS_ERROR_TTS_COMPLETED: 取完了所有合成的音频，此次合成结束。

3.3.8 取消合成

□ 函数原型

ISSErrID [ISSAPI](#) ISSTTSStop(HISSTTS hISSTTS);

□ 功能说明

- 1、停止合成。

□ 参数说明

参数名	参数解释
hISSTTS [in]	TTS 句柄

□ 返回值

- 1、ISS_SUCCESS: 成功取消了此次合成；
- 2、ISSErrID: 语音合成引擎返回的错误码，错误代码参见[附录二](#)；
- 3、ISS_ERROR_INVALID_HANDLE: 无效语音合成句柄；
- 4、ISS_ERROR_TTS_STOPPED: 错误调用，合成已经结束。

3.4 回调接口声明

3.4.1 合成资源初始化成功

□ 接口声明

```
typedef void (ISSCALLBACK* Proc_OnTTSDataReady) (void* pUsrArg);
```

□ 接口说明

1、合成开始后抛出此回调通知，可以调用 ISSTTSGetAudioData 取音频数据了；

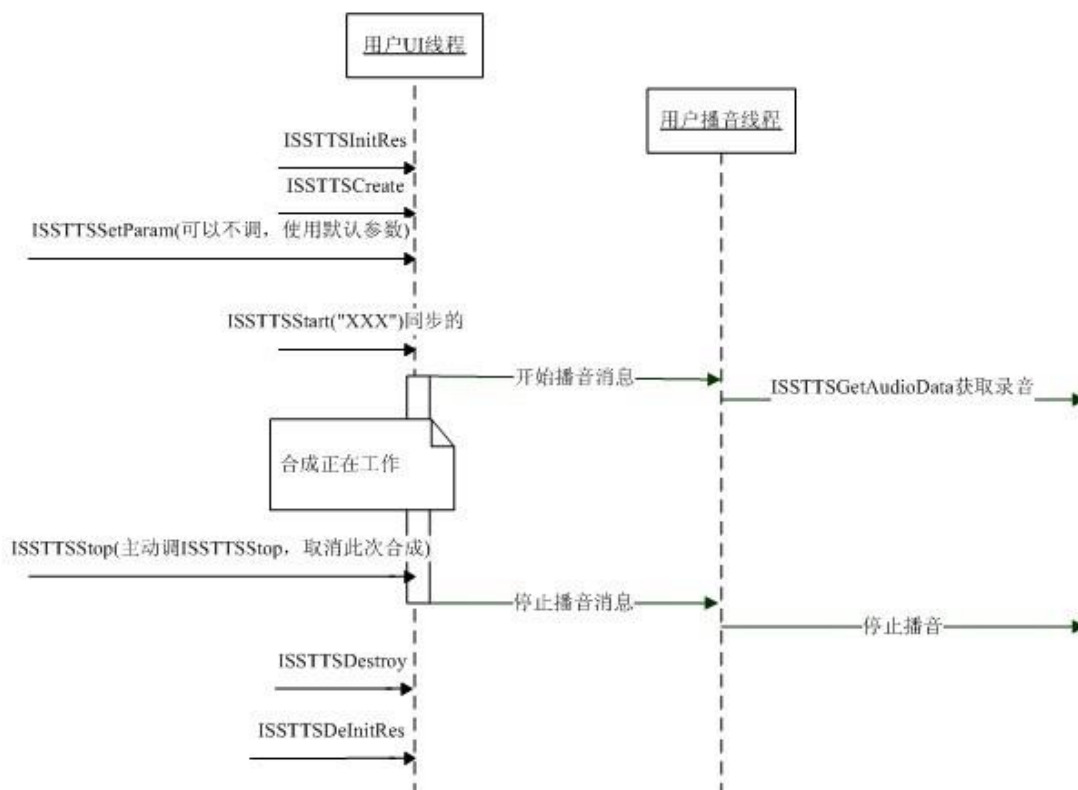
□ 参数说明

参数名	参数解释
pUsrArg [in]	用户自定义参数

□ 返回值

1、无；

3.5 接口调用流程



4. 多唤醒

4.1 接口介绍

4.1.1 函数列表

□ 提供如下接口：

函数名称	功能简介
ISSMVWCreate	创建多语音唤醒句柄
ISSMVWCreateW	创建多语音唤醒句柄
ISSMVWDestroy	销毁多语音唤醒句柄
ISSMVWSetThreshold	设置多语音唤醒门限
ISSMVWStart	启动多语音唤醒,如果多唤醒已经开始了,依然会返回ISS_SUCCESS
ISSMVWAppendAudioData	输入录音数据,支持输入16K采样率、S16-LE、单声道的PCM录音
ISSMVWStop	停止多语音唤醒,如果多唤醒已经停止了,依然会返回ISS_SUCCESS

4.1.2 功能说明

- 1、多语音唤醒，采用本地引擎的方式进行；

4.2 多唤醒参数说明

无

4.3 接口说明

4.3.1 创建多语音唤醒句柄

□ 函数原型

```
ISSErrID ISSAPI ISSMVWCreate (HISSMVW *phISSMVW, const wchar_t * szResourceDir,
Proc_ISSMVWOnWakeup pfnOnWakeup, void* pUsrArg);
ISSErrID ISSAPI ISSMVWCreateW (HISSMVW *phISSMVW, const wchar_t * szResourceDir,
Proc_ISSMVWOnWakeup pfnOnWakeup, void* pUsrArg);
```

□ 功能说明

- 1、创建多语音唤醒句柄；
- 2、不能创建多实例。

□ 参数说明

参数名	参数解释
phISSMVW [out]	多唤醒句柄
szResourceDir[in]	资源目录,一般情况下: 在windows中文环境下为GBK编码, 在linux环境下为UTF-8编码
Proc_ISSMVWOnWakeUp [in]	多唤醒回调接口, 请参考 多唤醒回调接口 声明, 不可以传NULL;
pUsrArg [in]	用户自定义参数

□ 返回值

- 1、ISS_SUCCESS: 成功创建多语音唤醒句柄；
- 2、ISSErrID: 多语音唤醒引擎返回的错误码, 错误代码参见[附录二](#);
- 3、ISS_ERROR_FILE_NOT_FOUND: 没有找到相应的资源文件;
- 4、ISS_ERROR_INVALID_PARA: 无效的参数值;
- 5、ISS_ERROR_OUT_OF_MEMORY: 内存不够;
- 6、ISS_ERROR_INVALID_DATA: 资源数据出错。

4.3.2 销毁多语音唤醒句柄

□ 函数原型

ISSErrID [ISSAPI](#) ISSMVWDestroy(HISSMVW hISSMVW)

□ 功能说明

- 1、销毁多唤醒句柄；

□ 参数说明

参数名	参数解释
hISSMVW [in]	多唤醒句柄

□ 返回值

- 1、ISS_SUCCESS: 成功销毁多语音唤醒句柄并释放资源;
- 2、ISSErrID: 多语音唤醒引擎返回的错误码, 错误代码参见[附录二](#);
- 3、ISS_ERROR_INVALID_HANDLE: 无效多语音唤醒句柄。

4.3.3 设置多语音唤醒门限

□ 函数原型

ISSErrID [ISSAPI](#) ISSMVWSetThreshold(HISSMVW hISSMVW, unsigned int nMvwScene, unsigned int nMvwId, int nThreshold);

□ 功能说明

- 1、设置多语音唤醒门限;

□ 参数说明

参数名	参数解释
hISSMVW [in]	多唤醒句柄
nThreshold [in]	门限值
nMvwScene[in]	多唤醒场景
nMvwId[in]	唤醒词 ID

□ 返回值

- 1、ISS_SUCCESS: 成功设置门限;
- 2、ISSErrID: 多语音唤醒引擎返回的错误码, 错误代码参见[附录二](#);
- 3、ISS_ERROR_INVALID_HANDLE: 无效多语音唤醒句柄;

4.3.4 启动多语音唤醒

□ 函数原型

ISSErrID [ISSAPI](#) ISSMVWStart(HISSMVW hISSMVW, unsigned int nMvwScene)

□ 功能说明

- 1、启动多语音唤醒;
- 2、接口是同步的;

□ 参数说明

参数名	参数解释
hISSMVW [in]	多唤醒句柄
nMvwScene [in]	多唤醒场景

□ 返回值

- 1、ISS_SUCCESS: 成功启动多语音唤醒;
- 2、ISSErrID: 多语音唤醒引擎返回的错误码, 错误代码参见[附录二](#);
- 3、ISS_ERROR_INVALID_HANDLE: 无效多语音唤醒句柄。

4.3.5 输入录音数据

□ 函数原型

ISSErrID [ISSAPI](#) ISSMVWAppendAudioData(HISSMVW hISSMVW, short* pSamples, unsigned int nNumberOfToWrite, unsigned int* pNumberOfWritten);

□ 功能说明

- 1、输入录音数据;
- 2、支持输入 16K 采样率、S16-LE、单声道的 PCM 录音;
- 3、只能在一个线程中调用此接口

□ 参数说明

参数名	参数解释
hISSMVW [in]	多唤醒句柄
pSamples [in]	输入的音频 buffer 地址;
nNumberOfToWrite [in]	要写入的采样点的个数,非字节数
pNumberOfWritten [out]	实际写入的采样点的个数

□ 返回值

- 1、ISS_SUCCESS: 成功写入录音, 但写入的录音的大小可能为 0, 需要判断 *pNumberOfWritten != nNumberOfToWrite 来判断是否全部写入;;
- 2、ISSErrID: 多语音唤醒引擎返回的错误码, 错误代码参见[附录二](#);
- 3、ISS_ERROR_INVALID_HANDLE: 无效多语音唤醒句柄;
- 4、ISS_ERROR_INVALID_PARA: 无效的数值;
- 5、ISS_ERROR_INVALID_CALL: 错误调用(还未调用 ISSMVWStart,多唤醒还没有开始)。

4.3.6 停止多语音唤醒

□ 函数原型

ISSErrID [ISSAPI](#) ISSMVWStop(HISSMVW hISSMVW);

□ 功能说明

- 1、停止多语音唤醒;
- 2、用户程序可以在 Proc_ISSMVWOnWakeup 回调中,调用 ISSMVWStop 进行终止唤醒引擎.
- 3、接口是同步的

□ 参数说明

参数名	参数解释
hISSMVW [in]	多唤醒句柄

□ 返回值

- 1、ISS_SUCCESS: 成功停止多语音唤醒;
- 2、ISSErrID: 多语音唤醒引擎返回的错误码, 错误代码参见[附录二](#);
- 3、ISS_ERROR_INVALID_HANDLE: 无效多语音唤醒句柄。

4.4 回调接口声明

4.4.1 多唤醒成功

□ 接口声明

```
typedef void (ISSCALLBACK *Proc_ISSMVWOnWakeup) (void* pUsrArg, unsigned int
nMvwScene, unsigned int nMvwId, signed int nMvwScore);
```

□ 接口说明

- 1、 在用户说出唤醒词时,多唤醒引擎会通过此回调函数通知调用者有一次唤醒.
如果用户多次说出唤醒词时,会多次调用此回调函数,但不会同时调用,用户程序可以在此回调函数中,调用 `ISSMVWStop` 进行终止多唤醒引擎.

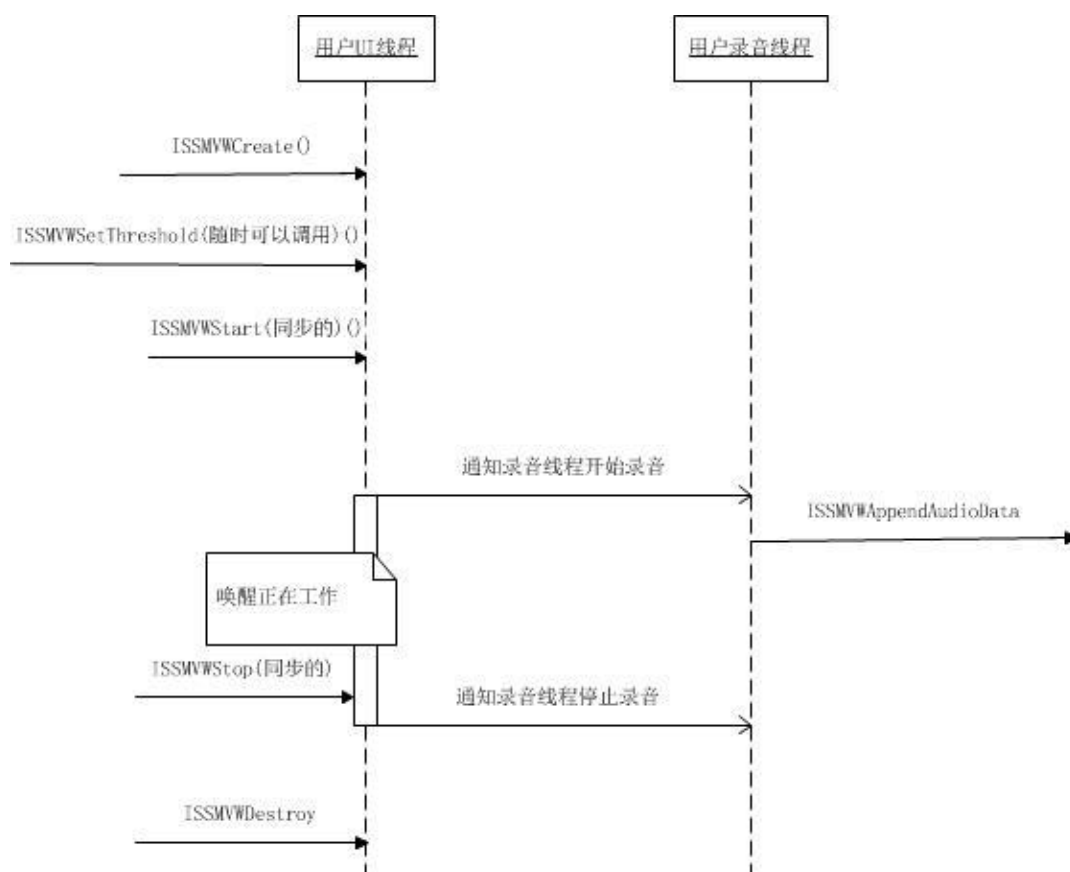
□ 参数说明

参数名	参数解释
<code>pUsrArg [in]</code>	用户自定义参数
<code>nMvwScene[in]</code>	多唤醒场景
<code>nMvwId[in]</code>	唤醒词 ID
<code>nMvwScore</code>	唤醒的置信度得分

□ 返回值

- 2、 无;

4.5 接口调用流程



5. 套件软件授权

5.1 接口简介

5.1.1 函数列表

□ 提供如下接口：

函数名称	功能简介
ISSetMachineCode	输入机器码
ISSGetActiveKey	获取激活码
ISSActivate	使用机器码和激活码,激活套件

5.1.2 功能说明

- 1、 套件授权管理接口，可以有三种激活套件方法：软件加密、硬件加密和芯片加密。
- 2、 软加密使用 `ISSetMachineCode+ISSGetActiveKey + ISSActivate` 接口，获取激活码，只需要在联网状态下调用一次。

5.2 授权参数说明

无。

5.3 接口说明

5.3.1 设置机器码

□ 函数原型

```
ISSErrID ISSetMachineCode(const char* szMachineCode);
```

□ 功能说明

- 1、 输入机器码；
- 2、 在调用 `ISSGetActiveKey` 获取激活码前、在调用 `ISSActivate` 激活前、在调用 `ISSRCreate` 创建识别句柄前，都需要至少调用一次此接口。
每个客户端必须输入不同的机器码,机器码的长度不能超过 1024。
每个机器码每天最多只能访问云端进行 1000 次识别。
机器码必须唯一，用于标识个性化资源

□ 参数说明

参数名	参数解释
szMachineCode [in]	机器码

□ 返回值

- 1、ISS_SUCCESS: 输入机器码成功;
- 2、ISS_ErrID: 语音套件授权激活的错误码, 错误代码参见[附录二](#);
- 3、ISS_ERROR_INVALID_PARA: 无效的数值;
- 4、ISS_ERROR_FAIL: 输入机器码失败;

5.3.2 获取激活码

□ 函数原型

ISS_ErrID ISSGetActiveKey(const char* szResourceDir);

□ 功能说明

- 1、获取激活码, 只需要在联网状态下成功调用一次.
- 2、在调用 ISSGetActiveKey 获取激活码前, 必须已经调用 ISSSetMachineCode 设置机器码.

□ 参数说明

参数名	参数解释
szResourceDir [in]	资源目录, 一般情况下: 在 windows 中文环境下为 GBK 编码, 在 linux 环境下为 UTF-8 编码;

□ 返回值

- 1、ISS_SUCCESS: 获取激活码成功;
- 2、ISS_ErrID: 语音套件授权激活的错误码, 错误代码参见[附录二](#);
- 3、ISS_ERROR_FAIL: 获取激活码失败;
- 4、ISS_ERROR_MACHINE_CODE_NOT_SET: 还未设置机器码;
- 5、ISS_ERROR_INVALID_MACHINE_CODE: 此机器码未经授权;
- 6、ISS_ERROR_ACTIVATE_TO_MANY_TIMES: 次机器码授权超出数量;
- 7、ISS_ERROR_NET_XXXX: 网络错误

5.3.3 激活

□ 函数原型

ISS_ErrID ISSActivate(const char* szResourceDir);

□ 功能说明

- 1、输入激活码目录, 激活;
- 2、在调用 ISSSRCreate 创建识别句柄前、在调用 ISSTTSInitRes 创建语音合成资源句柄前, 都需要至少调用一次此接口。在调用 ISSActivate 激活前, 必须已经调用 ISSSetMachineCode 设置机器码。

□ 参数说明

参数名	参数解释
szResourceDir [in]	资源目录,一般情况下: 在 windows 中文环境下为 GBK 编码, 在 linux 环境下为 UTF-8 编码;

□ 返回值

- 1、ISS_SUCCESS: 授权激活成功;
- 2、ISS_ErrID: 语音套件授权激活的错误码, 错误代码参见[附录二](#);
- 3、ISS_ERROR_INVALID_PARA: 无效的数值;
- 4、ISS_ERROR_INVALID_ACTIVE_KEY: 无效的激活码;
- 5、ISS_ERROR_MACHINE_CODE_NOT_SET: 还未设置机器码;

6. 套件芯片授权

6.1 接口简介

6.1.1 函数列表

□ 提供如下接口:

函数名称	功能简介
ISSSetChipLoadinfoCb	设置 获取授权芯片信息的回调

□ 提供如下回调:

回调名称	功能简介
Proc_ISSChipLoadinfo	获取授权芯片信息的回调
Proc_OnISSChipCryptError	芯片授权失败的回调

6.1.2 功能说明

1. 套件授权芯片授权需要集成讯飞的硬件授权芯片, 套件在工作时, 会通过 I2C 访问授权芯片, 如果授权成功, 套件才能正确工作。
2. 在调用套件其他接口前, 需要先调用 ISSSetChipLoadinfoCb 设置 获取授权芯片信息的回调。授权芯片的信息一般是通过 I2C 访问的, 不同的平台接口不一定一致, 通过回调由用户实现。如果授权失败, 会回调 Proc_OnISSChipCryptError 函数 或 发出 ISS_ERROR_CHIP_LOAD_INFO、ISS_ERROR_CHIP_CRYPT_FAIL 的错误码消息。
3. MTK3360 平台, 设置获取授权芯片信息的回调、通过 I2C 访问授权芯片的套件 sample 示例程序 参见 test_libissmvw、test_libissr 或 TTSPay_SimpleSample。

6.2 接口说明

6.2.1 设置 获取授权芯片信息的回调

□ 函数原型

```
ISSErrID ISSAPI ISSSetChipLoadinfoCb(Proc_ISSChipLoadinfo pfnChipLoadinfo);
```

□ 功能说明

1. 设置 获取授权芯片信息的回调
2. 采用芯片加密版本的套件，在调用套件其他接口前，必须先调用这一接口

□ 参数说明

参数名	参数解释
pfnChipLoadinfo	获取授权芯片信息的回调

□ 返回值

- 1、ISS_SUCCESS: 回调函数设置成功;
- 2、ISSErrID: 语音套件授权激活的错误码，错误代码参见[附录二](#);
- 3、ISS_ERROR_INVALID_PARA: 无效的参数值，如回调函数指针为空;

6.3 回调接口说明

6.3.1 获取授权芯片信息

□ 接口声明

```
typedef ISSErrID (ISSCALLBACK *Proc_ISSChipLoadinfo) (  
    const unsigned char* pHint,  
    unsigned char* pChipinfo);
```

□ 接口说明

在不同的客户平台，访问硬件授权芯片的接口不完全一致，需要客户在回调中获取授权芯片信息。

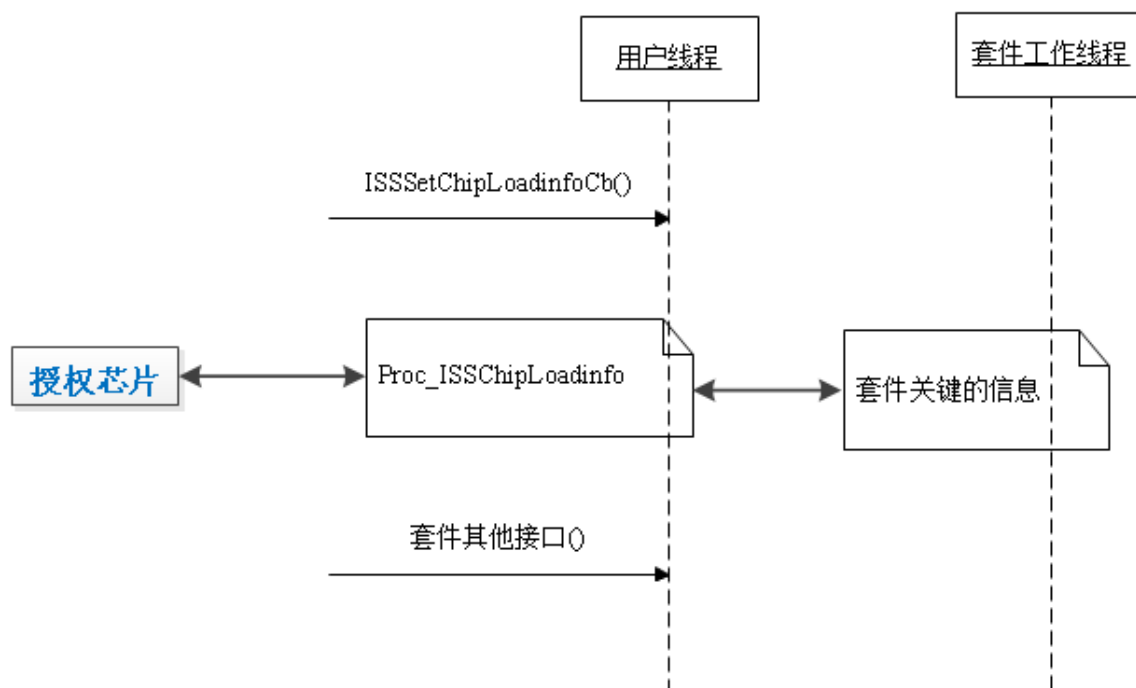
□ 参数说明

参数名	参数解释
const unsigned char* pHint [in]	8 字节的解密提示信息
unsigned char* pChipinfo[Out]	从授权芯片读取的 32 字节解密提示信息

□ 返回值

ISSErrID: 获取芯片信息成功还是失败
ISS_SUCCESS: 获取芯片信息成功;

6.4 接口调用流程



7. 接口使用示例

7.1 场景 1

使用语音识别录音前播放提示音，用于提示用户开始输入语音，可以将识别和合成结合起来使用：

```
#define PCM_16K_BUFFER_SIZE (320)
//返回值
int res;

//唤醒句柄
HISSVW *phISSVW = NULL;

//输入唤醒的数据块
Char *data_buff = new char[PCM_16K_BUFFER_SIZE];

//唤醒回调
static void OnWakeUp(void *);
```

```
//创建唤醒句柄
res = ISSVWCreate(phISSVW, "\\storage card\\", OnWakeUp, NULL);

//设置唤醒门限
res= ISSVWSetThreshold(phISSVW,0);

// 启动唤醒
Res = ISSVWStart(phISSVW);

// 输入录音模块
Res = IvwAppendAudioData(phISSVW, data_buff, PCM_16K_BUFFER_SIZE/2);
... ..
//语音唤醒成功
Static void OnWakeUp(void *)
{
    //已创建和初始化识别句柄
    //已创建和初始化合成句柄
    ... ..
    While (1)
    {
        If(ISS_ERROR_TTS_COMPLETED != ISSTTSGetAudioData(...))
        {
            //播放获取到的音频数据(提示音)
        }
        Else
        {
            //提示音播放完成, 识别开始录音
            Res = ISSRStart(...);
            Break;
        }
        Sleep(100);
    }
}
```

7.2 场景 2

在交互中, 启动选择场景, 例如在打电话业务中, 打电话给张三, 当联系人有多个时, 会提供一次让用户选择哪个联系人的交互, 此时要用到选择场景识别:

```
//TTS 状态枚举
typedef enum _tagTTSSState
{
    TTS_State_Init,                //初始状态
    TTS_State_Record,              //播报之后要录音
    TTS_State_Record_Contacts,     //播报之后要录音,联系人场景
    TTS_State_Record_SMS,          //播报之后要录音,输入短信场景
    TTS_State_Select,              //选择场景
    TTS_State_Record_Confirm,      //确认场景
}TTSSState;

//返回值
Int Res;

//当前 TTS 状态
TTSSState m_ttsstate;

//电话联系人有多个, 要提示用户选择哪一个联系人时
m_ttsstate = TTS_State_Select;
Res = ISSTTSSStart("请选择第几个联系人...");
...

//语音播报完成后, 开始启动选择场景识别
void OnTTSSCompleted()
{
    switch(m_ttsstate)
    {
        ...
        Case TTS_State_Select:
        {
            Res = ISSSRStart(hISSSRObj,"select",2,NULL);
        }
        ...
    }
}
```

8. 技术支持

如果您在使用中遇到任何问题或者有任何建议，请与我们联系！

联系时对问题的描述请尽量包含以下内容：

- * 系统配置（包括CPU、内存、硬盘、操作系统及产品版本等信息）
- * 问题细节（包括问题的重现过程及表现内容等）
- * 问题重现（包括详细的操作过程）

电话支持

请于周一～周五，北京时间9：00～17：00 间，

拨打电话：18805691018 获得技术支持信息。

电子邮件支持

请将问题的详细描述发至：autoflySupport@iflytek.com。

信件支持

请将问题的详细描述发至：

中国安徽省合肥市望江西路666 号科大讯飞语音信息化产业基地，邮编：230088

或传真至：0551—65331801 / 65331802

附录一：个性发音人列表

- 1、语言为中英文的发音人可以支持中英文的混合朗读。
- 2、英文发音人只能朗读英文，中文无法朗读。
- 3、汉语发音人只能朗读中文，遇到英文会以单个字母的方式进行朗读。

发音人名称	属性	语言	参数名称	备注
Jiajia	童年女声	中英文（普通话）	ivTTS_ROLE_JIAJIA	默认
Tianchang	青年女声	中英文（普通话）	ivTTS_ROLE_TIANCHANG	
Wenjing	青年女声	中英文（普通话）	ivTTS_ROLE_WENJING	
Xiaoyan	青年女声	中英文（普通话）	ivTTS_ROLE_XIAOYAN	
YanPing	青年女声	中英文（普通话）	ivTTS_ROLE_YANPING	
XiaoFeng	青年男声	中英文（普通话）	ivTTS_ROLE_XIAOFENG	
YuFeng	青年男声	中英文（普通话）	ivTTS_ROLE_YUFENG	
Sherri	青年女声	US English	ivTTS_ROLE_SHERRI	
Xiaojin	青年女声	中英文（普通话）	ivTTS_ROLE_XIAOJIN	
Nannan	童年男声	中英文（普通话）	ivTTS_ROLE_NANNAN	
Jinger	青年女声	中英文（普通话）	ivTTS_ROLE_JINGER	
Yuer	青年女声	中英文（普通话）	ivTTS_ROLE_YUER	
Xiaoqian	青年女声	汉语（东北话）	ivTTS_ROLE_XIAOQIAN	
Laoma	青年男声	中英文（普通话）	ivTTS_ROLE_LAOMA	
Bush	青年男声	US English	ivTTS_ROLE_BUSH	
Xiaorong	青年女声	汉语（四川话）	ivTTS_ROLE_XIAORONG	
Xiaomei	青年女声	汉语（广东话）	ivTTS_ROLE_XIAOMEI	
Anni	青年女声	中英文（普通话）	ivTTS_ROLE_ANNI	

			NNI	
John	青年男声	US English	ivTTS_ROLE_J OHN	
Anita	青年女声	British English	ivTTS_ROLE_A NITA	
Terry	青年女声	US English	ivTTS_ROLE_T ERRY	
Catherine	青年女声	US English	ivTTS_ROLE_C ATHERINE	
Terry	青年女声	US English Word	ivTTS_ROLE_ TERRYW	
Xiaolin	青年女声	中英文（普通话）	ivTTS_ROLE_X IAOLIN	
Xiaomeng	青年女声	中英文（普通话）	ivTTS_ROLE_ XIAOMENG	
Xiaoqiang	青年男声	中英文（普通话）	ivTTS_ROLE_X IAOQIANG	
XiaoKun	青年男声	中英文（普通话）	ivTTS_ROLE_X IAOKUN	
Jiu Xu	青年男声	中英文（普通话）	ivTTS_ROLE_ JIUXU	
Duo Xu	青年男声	中英文（普通话）	ivTTS_ROLE_ DUOXU	
Xiaoping	青年女声	中英文（普通话）	ivTTS_ROLE_X IAOPING	
Donald Duck	青年男声	中英文（普通话）	ivTTS_ROLE_ DONALDDUCK	
Baby Xu	童年男声	中英文（普通话）	ivTTS_ROLE_B ABYXU	
Dalong	青年男声	汉语（广东话）	ivTTS_ROLE_D ALONG	
Tom	青年男声	US English	ivTTS_ROLE_T OM	

附录二：错误码列表

1、错误码参见下表：

错误码	错误值	意义
ISS_SUCCESS	0	成功状态
ISS_ERROR_FAIL	-1	失败
ISS_ERROR_EXCEPTION	-2	异常
ISS_ERROR_INVALID_CALL	10000	非法调用
ISS_ERROR_INVALID_JSON_FMT	10001	无效的 JSON 格式
ISS_ERROR_INVALID_JSON_INFO	10002	无效的 JSON 内容信息
ISS_ERROR_TTS_STOPPED	10003	错误调用，合成已经结束
ISS_ERROR_TTS_COMPLETED	10004	取完了所有合成的音频，此次合成结束
ISS_ERROR_CREATE_THREAD_FAIL	10005	创建线程失败
ISS_ERROR_NO_SPEECH	10006	没有检测到语音
ISS_ERROR_GET_RESULT_TIMEOUT	10007	获取结果超时
ISS_ERROR_BUILDGRM_INIT_ERROR	10008	构建词典初始化错误
ISS_ERROR_INVALID_ACTIVE_KEY	10009	非法的激活码
ISS_ERROR_INVALID_MACHINE_CODE	10010	非法的机器码
ISS_ERROR_ACTIVATE_TO_MANY_TIMES	10011	激活次数已超过限制
ISS_ERROR_ACCESS_TO_MANY_TIMES	10012	访问次数已超过限制
ISS_ERROR_MACHINE_CODE_NOT_SET	10013	机器码还未设置
ISS_ERROR_GET_ACTIVE_KEY_TIMEOUT	10014	获取激活码超时
ISS_ERROR_NO_RESULT	10015	识别结果拒识
ISS_ERROR_BUILDING_GRM	10016	配置文件
ISS_ERROR_GENERAL	10100	
ISS_ERROR_OUT_OF_MEMORY	10101	内存不足
ISS_ERROR_FILE_NOT_FOUND	10102	没有找到文件
ISS_ERROR_INVALID_PARA	10106	无效的参数类型
ISS_ERROR_INVALID_PARA_VALUE	10107	无效的参数值
ISS_ERROR_INVALID_HANDLE	10108	无效的句柄
ISS_ERROR_INVALID_DATA	10109	无效
ISS_ERROR_NO_LICENSE	10110	没有授权许可
ISS_ERROR_NOT_INIT	10111	没有初始化
ISS_ERROR_NULL_HANDLE	10112	空句柄
ISS_ERROR_OVERFLOW	10113	溢出
ISS_ERROR_TIME_OUT	10114	超时
ISS_ERROR_OPEN_FILE	10115	打开文件出错
ISS_ERROR_NOT_FOUND	10116	没有发现

ISS_ERROR_NO_ENOUGH_BUFFER	10117	没有足够的内存
ISS_ERROR_NO_DATA	10118	没有数据
ISS_ERROR_NO_MORE_DATA	10119	没有更多的数据
ISS_ERROR_NO_RESPONSE_DATA	10120	没有反馈数据
ISS_ERROR_ALREADY_EXIST	10121	已经存在
ISS_ERROR_LOAD_MODULE	10122	加载模块错误
ISS_ERROR_BUSY	10123	忙碌
ISS_ERROR_INVALID_CONFIG	10124	无效的配置项
ISS_ERROR_VERSION_CHECK	10125	版本错误
ISS_ERROR_CANCELED	10126	取消
ISS_ERROR_INVALID_MEDIA_TYPE	10127	无效的媒体类型
ISS_ERROR_CONFIG_INITIALIZE	10128	初始化 Config 实例
ISS_ERROR_CREATE_HANDLE	10129	建立句柄
ISS_ERROR_CODING_LIB_NOT_LOAD	10130	库未加载
ISS_ERROR_NET_GENERAL	10200	网络一般错误
ISS_ERROR_NET_OPENSOCK	10201	打开套接字
ISS_ERROR_NET_CONNECTSOCK	10202	套接字连接
ISS_ERROR_NET_ACCEPTSOCK	10203	套接字接收
ISS_ERROR_NET_SENDSOCK	10204	发送错误
ISS_ERROR_NET_RECVSOCK	10205	接收错误
ISS_ERROR_NET_INVALIDSOCK	10206	无效的套接字
ISS_ERROR_NET_BADADDRESS	10207	无效的地址
ISS_ERROR_NET_BINDSEQUENCE	10208	绑定次序
ISS_ERROR_NET_NOTOPENSOCK	10209	套接字没有打开
ISS_ERROR_NET_NOTBIND	10210	没有绑定
ISS_ERROR_NET_NOTLISTEN	10211	没有监听
ISS_ERROR_NET_CONNECTCLOSE	10212	连接关闭
ISS_ERROR_NET_NOTDGRAMSOCK	10213	非数据报套接字
ISS_ERROR_NET_DNS	10214	DNS 解析错误
ISS_ERROR_MSG_GENERAL	10300	消息一般错误
ISS_ERROR_MSG_PARSE_ERROR	10301	解析出错
ISS_ERROR_MSG_BUILD_ERROR	10302	构建出错
ISS_ERROR_MSG_PARAM_ERROR	10303	参数出错
ISS_ERROR_MSG_CONTENT_EMPTY	10304	Content 为空
ISS_ERROR_MSG_INVALID_CONTENT_TYPE	10305	Content 类型无效
ISS_ERROR_MSG_INVALID_CONTENT_LENGTH	10306	Content 长度无效
ISS_ERROR_MSG_INVALID_CONTENT_ENCODE	10307	Content 编码无效
ISS_ERROR_MSG_INVALID_KEY	10308	Key 无效
ISS_ERROR_MSG_KEY_EMPTY	10309	Key 为空
ISS_ERROR_MSG_SESSION_ID_EMPTY	10310	会话 ID 为空
ISS_ERROR_MSG_LOGIN_ID_EMPTY	10311	登录 ID 为空
ISS_ERROR_MSG_SYNC_ID_EMPTY	10312	同步 ID 为空
ISS_ERROR_MSG_APP_ID_EMPTY	10313	应用 ID 为空

ISS_ERROR_MSG_EXTERN_ID_EMPTY	10314	扩展 ID 为空
ISS_ERROR_MSG_INVALID_CMD	10315	无效的命令
ISS_ERROR_MSG_INVALID_SUBJECT	10316	无效的主题
ISS_ERROR_MSG_INVALID_VERSION	10317	无效的版本
ISS_ERROR_MSG_NO_CMD	10318	没有命令
ISS_ERROR_MSG_NO_SUBJECT	10319	没有主题
ISS_ERROR_MSG_NO_VERSION	10320	没有版本号
ISS_ERROR_MSG_MSSP_EMPTY	10321	消息为空
ISS_ERROR_MSG_NEW_RESPONSE	10322	新建响应消息失败
ISS_ERROR_MSG_NEW_CONTENT	10323	新建 Content 失败
ISS_ERROR_MSG_INVALID_SESSION_ID	10324	无效的会话 ID
ISS_ERROR_DB_GENERAL	10400	数据库一般错误
ISS_ERROR_DB_EXCEPTION	10401	异常
ISS_ERROR_DB_NO_RESULT	10402	没有结果
ISS_ERROR_DB_INVALID_USER	10403	无效的用户
ISS_ERROR_DB_INVALID_PWD	10404	无效的密码
ISS_ERROR_DB_CONNECT	10405	连接出错
ISS_ERROR_DB_INVALID_SQL	10406	无效的 SQL
ISS_ERROR_DB_INVALID_APPID	10407	无效的应用 ID
ISS_ERROR_RES_GENERAL	10500	资源一般错误
ISS_ERROR_RES_LOAD	10501	没有加载
ISS_ERROR_RES_FREE	10502	空闲
ISS_ERROR_RES_MISSING	10503	缺失
ISS_ERROR_RES_INVALID_NAME	10504	无效的名称
ISS_ERROR_RES_INVALID_ID	10505	无效的 ID
ISS_ERROR_RES_INVALID_IMG	10506	无效的映像
ISS_ERROR_RES_WRITE	10507	写操作错误
ISS_ERROR_RES_LEAK	10508	泄露
ISS_ERROR_RES_HEAD	10509	资源头部错误
ISS_ERROR_RES_DATA	10510	数据出错
ISS_ERROR_RES_SKIP	10511	跳过
ISS_ERROR_TTS_GENERAL	10600	合成一般错误
ISS_ERROR_TTS_TEXTEND	10601	文本结束
ISS_ERROR_TTS_TEXT_EMPTY	10602	文本为空
ISS_ERROR_REC_GENERAL	10700	一般错误
ISS_ERROR_REC_INACTIVE	10701	处于不活跃状态
ISS_ERROR_REC_GRAMMAR_ERROR	10702	语法错误
ISS_ERROR_REC_NO_ACTIVE_GRAMMARS	10703	没有活跃的语法
ISS_ERROR_REC_DUPLICATE_GRAMMAR	10704	语法重复
ISS_ERROR_REC_INVALID_MEDIA_TYPE	10705	无效的媒体类型
ISS_ERROR_REC_INVALID_LANGUAGE	10706	无效的语言
ISS_ERROR_REC_URI_NOT_FOUND	10707	没有对应的 URI
ISS_ERROR_REC_URI_TIMEOUT	10708	获取 URI 内容超时

ISS_ERROR_REC_URI_FETCH_ERROR	10709	获取 URI 内容时出错
ISS_ERROR_EP_GENERAL	10800	(EP) 一般错误
ISS_ERROR_EP_NO_SESSION_NAME	10801	(EP) 链接没有名字
ISS_ERROR_EP_INACTIVE	10802	(EP) 不活跃
ISS_ERROR_EP_INITIALIZED	10803	(EP) 初始化出错
ISS_ERROR_TUV_GENERAL	10900	
ISS_ERROR_TUV_GETHIDPARAM	10901	Get Busin Param huanid
ISS_ERROR_TUV_TOKEN	10902	Get Token
ISS_ERROR_TUV_CFGFILE	10903	Open cfg file
ISS_ERROR_TUV_RECV_CONTENT	10904	received content is error
ISS_ERROR_TUV_VERFAIL	10905	Verify failure
ISS_ERROR_LOGIN_SUCCESS	11000	登录成功
ISS_ERROR_LOGIN_NO_LICENSE	11001	无授权
ISS_ERROR_LOGIN_SESSIONID_INVALID	11002	无效的 SessionID
ISS_ERROR_LOGIN_SESSIONID_ERROR	11003	错误的 SessionID
ISS_ERROR_LOGIN_UNLOGIN	11004	未登录
ISS_ERROR_LOGIN_INVALID_USER	11005	无效的用户
ISS_ERROR_LOGIN_INVALID_PWD	11006	无效的密码
ISS_ERROR_LOGIN_SYSTEM_ERROR	11099	系统错误
ISS_ERROR_HCR_GENERAL	11100	
ISS_ERROR_HCR_RESOURCE_NOT_EXIST	11101	
ISS_ERROR_HCR_CREATE	11102	
ISS_ERROR_HCR_DESTROY	11103	
ISS_ERROR_HCR_START	11104	
ISS_ERROR_HCR_APPEND_STROKES	11105	
ISS_ERROR_HCR_GET_RESULT	11106	
ISS_ERROR_HCR_SET_PREDICT_DATA	11107	
ISS_ERROR_HCR_GET_PREDICT_RESULT	11108	
ISS_ERROR_HTTP_BASE	12000	HTTP 错误基码
ISS_ERROR_ISV_NO_USER	13000	用户不存在
ISS_ERROR_LUA_BASE	14000	
ISS_ERROR_LUA_YIELD	14001	
ISS_ERROR_LUA_ERRRUN	14002	
ISS_ERROR_LUA_ERRSYNTAX	14003	
ISS_ERROR_LUA_ERRMEM	14004	
ISS_ERROR_LUA_ERRERR	14005	

附录三：识别结果说明

1、语音识别结果为 XML 格式，示例如下：

示例 1：打开愤怒的小鸟

```
<rawtext>打开愤怒的小鸟</rawtext>
<version>1.0</version>
<result>
  <focus> app </focus>
  <action>
    <operation> launch</operation>
  </action>
  <object>
    <name>愤怒的小鸟</name>
  </object>
</result>
```

示例 2：打电话给张三

```
<rawtext>打电话给张三</rawtext>
<version>1.0</version>
<result>
  <focus>telephone</focus>
  <action>
    <operation> call</operation>
  </action>
  <object>
    <name>张三</name>
  </object>
</result>
```

示例 2 中，通过 XML 解析，可以获取到如下信息：业务是打电话；操作是拨打；联系人是张三。如果在知道张三的号码及系统提供拨打电话的接口的话，系统将很容易响应用户的语音指令：打电话给张三。

2、建议将解析过程封装成一个类实现，解析接口、业务操作等可以作为类的成员函数及成员变量，详细的语义解析格式可参见 语音套件语义处理协议 文档。

附录四：个性化数据上传格式

1、个性化数据格式如下：

```
{
  "grm": [
    {
      "dictname": "contact",
      "dictcontant": [
        { "name": "科大讯飞", "id": 0 },
        { "name": "白雪", "id": 1 },
        { "name": "刘俊峰", "id": 2 },
        { "name": "白亚伟", "id": 3 }
      ]
    }
  ],
}
```

```
{
  "dictname": "singers",
  "dictcontent": [
    { "name": "凤凰传奇", "id": 0 },
    { "name": "那英", "id": 1 },
    { "name": "杨坤", "id": 2 },
    { "name": "S.H.E", "id": 3 }
  ]
},
{
  "dictname": "songs",
  "dictcontent": [
    { "name": "恭喜发财", "id": 0 },
    { "name": "偏爱", "id": 1 },
    { "name": "铿锵玫瑰", "id": 2 },
    { "name": "孤枕难眠", "id": 3 }
  ]
},
{
  "dictname": "apps",
  "dictcontent": [
    { "name": "讯飞语点", "id": 0 },
    { "name": "愤怒的小鸟", "id": 1 },
    { "name": "浏览器", "id": 2 }
  ]
}
}
```

2、数据格式为 Json 格式，UTF8 编码。其中，dictname 标签表示上传个性化词典的类型，目前支持联系人（contact），歌手（singers），歌曲（songs）和应用名称（apps）四种类型。dictcontent 表示上传的数据项，name 子标签代表数据名称，id 表示数据的 id 号，例如，联系人数据上传成功后，即可使用上传的联系人，打电话给白雪、呼叫白亚伟等。

3、个性化数据上传接口 `ISSSRUploadDict(HISSSR hISSSR, const char *szList, int bOnlyUploadToCloud)`，其中参数 `bOnlyUploadToCloud` 表示是否只上传到云端；在上传个性化数据的过程中，如果当时未连接网络将导致网络数据上传失败，将不会返回 `ISS_SR_MSG_UploadDictToCloudStatus` 消息，开发者可以将此次上传失败记录下来，待网络连接后将 `bOnlyUploadToCloud` 设置为 `TRUE`，个性化数据上传至网络服务器，以避免网络上传失败后，识别模式切换到网络进行识别时，个性化数据不能识别的问题。

附录五：个性化命令词使用说明

1、在 `wince.arm\bin\SRRes\` 下可以找到 `issr.cfg`，示例如下：

示例:

```
{
  "Appid" : "526664b0",
  "Cmds" : [
    { "name": "正在播放的是什么歌",    "id": 1 },
    { "name": "这首歌叫什么",          "id": 1 },
    { "name": "到目的地还有多远",      "id": 2 },
    { "name": "双画面显示",             "id": 3 }
  ]
}
```

其中, Cmds 标签即为上传的个性化命令词。name 代表命令名称, id 代表的是命令对应的 id 号, 相同 id 代表命令所代表的意思是一致的, 示例中, 正在播放的是什么歌和这首歌叫什么, 它们的命令 id 都是 1, 表示它们的语义都是一样的, 代表相同的意思。语义解析的格式如下:

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<nlp>
```

```
  <version>1.0.0.2600</version>
```

```
  <rawtext>正在播放的是什么歌</rawtext>
```

```
  <parsedtext>正在 播放 的 是 什么 歌</parsedtext>
```

```
  <result>
```

```
    <focus>cmd</focus>
```

```
    <object>
```

```
      <name>正在播放的是什么歌</name>
```

```
      <id>1</id>
```

```
    </object>
```

```
  </result>
```

```
</nlp>
```

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<nlp>
```

```
  <version>1.0.0.2600</version>
```

```
  <rawtext>这首歌叫什么</rawtext>
```

```
  <parsedtext>这 首 歌 叫 什么</parsedtext>
```

```
  <result>
```

```
    <focus>cmd</focus>
```

```
    <object>
```

```
      <name>这首歌叫什么</name>
```

```
      <id>1</id>
```

```
    </object>
```

```
  </result>
```

```
</nlp>
```

这些命令词的解析格式除<id>*</id>外, 其余标签都是一样的, 解析时以 id 为标准, 相同 id 表示相同意思, 执行相同应用命令。

附录六：语音识别模式简要说明

识别模式共有网络、本地、网络+本地、命令词模式及本地语义模式，下面就命令词模式和语义解析模式作简要说明：

1、命令词模式：

当调用识别接口 `ISSSRStart` 并将 `iMode` 设置成 3 时，识别就进入了命令词模式的识别；此种模式的识别应用在特殊的情况下，比如要识别的候选结果有限（一般小于 5 个），对识别的准确率要求很高时，可使用这种模式（这种模式一般使用文本的结果即转写的结果），示例如下：

示例：在打电话时，如果联系人的号码有三个：电信、移动和联通的号码各一个，在使用语音选择哪一个号码时，例如：张三的号码有三个，电信、移动和联通，请问拨打哪一个？这时启动识别时就可以启动这种模式 `iMode = 3`，`szCmd =`

```
{
    "grm": [
        {
            "dictname": "cmd",
            "dictcontent": [
                {"name": "移动", "id": 0 },
                {"name": "电信", "id": 1 },
                {"name": "联通", "id": 2 }
            ]
        }
    ]
};
```

这种情况使用选择场景也同样可以完成号码选择，可以在应用界面上显示出三个号码，让用户进行选择，将 `szScene = ISS_SR_SCENE_SELECT` 即可。

2、语义解析模式：

语义解析模式是对文本的内容调用我们的本地语义引擎进行解析，提供了一种测试验证的方式，具体功能参见本地识别单元测试 Demo（`test_libisssr`）Local Nlp。