云端节点语音识别框架

开发文档

**中国科学院软件所**

目录

前言

欢迎使用Harebot机器人平台（Harebot Robot Platform，HRP）！

Harebot机器人平台是基于中科院软件所自主研发的Harebot机器人，开发的一款机器人应用开发平台，提供语音识别、人脸识别、定位导航等服务，为智能机器人应用开发爱好者提供方便易用的开发平台，使得用户能够基于该平台进行多种智能应用开发。

本文档主要就Harebot机器人平台的语音识别框架的开发过程进行详细阐述。文档主要讲述如何利用讯飞移动语音平台客户端子系统（Mobile Speech Client 5.0, MSC5.0）进行语音命令词识别、语音合成等方面的应用开发。

本文档基本内容：

1. 概述

简要概括云端节点语音识别框架的原理、功能，以及一些名词、缩略语介绍等。

1. 通用接口

介绍本语音识别框架中通用性较强的接口功能与应用，并详细列出开发步骤与示例。

1. ASR接口

介绍ASR接口函数功能与应用，并详细列出开发步骤与示例。

1. TTS接口

介绍TTS接口函数功能与应用，并详细列出开发步骤与示例。

1. Ros下语音模块代码

给出集成到ROS系统中的语音识别、语音合成的代码说明

1. 常见问题解答

通过阅读该文档，读者可以：

1. 了解Harebot机器人平台云端节点语音识别框架的基本功能与特色。
2. 掌握语音识别、语音合成系统接口规范
3. 了解语音合成系统、语音识别应用开发的基本思想和方法
4. 利用语音应用系统接口规范进行二次开发。
5. 概述

语音模块用于获取用户语音指令，给机器人装上耳朵，实现类人“听力”；语音合成，让机器人实现开口说话。模块集成科大讯飞“命令词识别”“语音合成”两组语音服务开发包，实现识别用户指令，并以语音形式将识别结果以及执行动作反馈给用户。

（一）**模块具体功能如下**：

1. 录制用户语音指令，并保存成.wav格式音频。
2. 对用户指令音频进行内容识别。
3. 将识别结果映射成相应的执行动作，并将其对外发布出去。
4. 相关client订阅发布出来的指令，并执行相应的动作；在当前版本中，client接收到发布的执行后，实现将动作内容以语音形式反馈给用户。
5. **相关说明**：
6. 音频录制格式：

采样率16k/8k；

采用精度16bit；

单声道；

wav或者pcm格式；

1. 命令词识别语法可定制：

离线识别的命令词是开发者自己定义，命令词最大长度为31个字，个数为215个。需要先构建语法，然后指定使用的语法。其中，简单的语法示例如下：

<commands>:(找一下|打电话给) <name>;

<name>: 张磊|李超;

. . . . . . .

该语法使识别引擎可以支持以下说法：找一下张磊 、打电话给张磊、找一下李超、打电话给李。凡是用户说出这个范围中的任意一句话，均可以被识别系统识别。如果用户说的话不在上述范围中，识别系统可能拒绝识别。

1. **名词和缩略语**

* ASR（Automatic Speech Recognition）

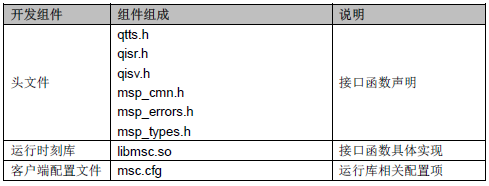
语音识别技术语音识别技术是一种使计算机能够识别人通过麦克风或者电话输入的词语或语句的技术，简单的说就是能够让计算机听懂人说话。它的最终目标是使得计算机不受词汇量限制，在各种噪声环境、语音信道下，能够实时、准确地识别不同方言、口音等特点的说话人的语句。

* TTS（Text to Speech）

语音合成（Text To Speech，TTS）技术能够自动将任意文字实时转换为连续的自然语音，是一种能够在任何时间、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。

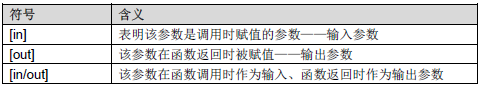
1. **开发包组件**

Linux平台



1. **文档说明**

本文针对的读者是具有linux平台开发经验的C/C++程序员。文档中使用的符号约定：



1. 通用接口

**2.1 通用接口简介**

**2.1.1 通用接口函数列表**

提供如下通用接口：



**2.1.2 返回值说明**

对于开发接口，如果调用成功，返回值为int 型的接口都会返回MSP\_SUCCESS，否则返回错误代码，错误代码参见msp\_errors.h；返回值为指针类型的接口，函数执行失败返回空指针0，错误代码可以通过相关的回传参数查看，函数执行成功但无数据时返回空指

针0，有数据时返回非0 指针。

**2.2 函数调用**

**2.2.1 MSPLogin**

**函数原型：**

int MSPAPI MSPLogin( const char\* usr, const char\* pwd, const char\* params )

**功能：**

用户登录，每一项语音业务必须先调用MSPLogin，如果登录失败则无法获取任何语音服务。

**参数：**

♦ usr [in]

用户名，用户可以从网

http://dev.voicecloud.cn/index.php?vt=1 申请。

♦ pwd [in]

密码。

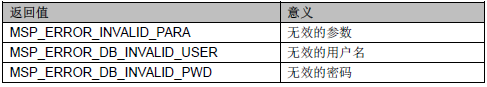
♦ params [in]

登录时传入的参数， 可以设置的参数及其取值范围请参考《

设置参数列表\_MSC50.xls》。

**返回值：**

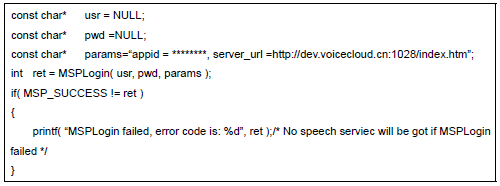
如果函数调用成功返回MSP\_SUCCESS ， 否则返回错误代码， 错误代码参见msp\_errors.h，几个主要的返回值如下：



**说明：**

本接口为通用登录接口。MSP 系统所有功能都需要用户在登录的状态下使用。本接口应当在应用程序中仅调用一次，多次调用本函数（中间没有调MSPLogout）时只有第一次调用此函数会进行实际的登录操作，后面的调用会返回成功，但不会完成实际的登录工作。同样MSPLogout 也应当在应用程序退出时仅调用一次。

**用法示例：**



**2.2.2 MSPUploadData**

**函数原型：**

const char\* MSPAPI MSPUploadData(const char\* dataName, void\* data, unsigned int dataLen, const char\* params, int\* errorCode)

**功能：**

用户数据上传。

**参数：**

♦ dataName [in]

数据名称。

♦ data [in]

数据缓冲区起始地址。

♦ dataLen [in]

数据长度。

♦ params [in]

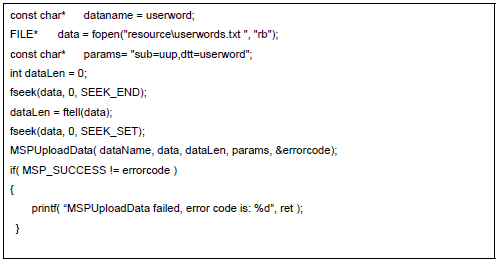
数据描述参数。

♦ errorCode [out]

如果函数调用成功返回0，否则返回错误代码，错误代码参见msp\_errors.h，几个主要的返回值如下：



**用法示例：**



**2.2.3 MSPDownloadData**

**函数原型：**

const void\* MSPAPI MSPDownloadData(const char\* params, unsigned int\* dataLen,int\* errorCode)

**功能：**

用户数据下载。

**参数：**

♦ params [in]

可以设置的参数及其取值范围请参考《可设置参数列表\_MSC50.xls》。

♦ dataLen [out]

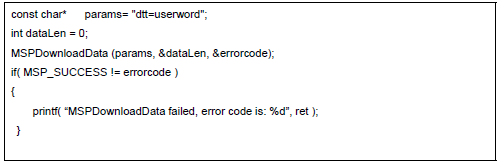
下载数据长度。

♦ errorCode [out]

如果函数调用成功返回0，否则返回错误代码，错误代码参见msp\_errors.h，几个主要的返回值如下：



**用法示例：**



**2.2.4 MSPSearch**

**函数原型：**

const void\* MSPAPI MSPSearch(const char\* params, const char\* text, unsigned int\*contentLen, int\* errorCode)

**功能：**

语义。

**参数：**

♦ params [in]

可以设置的参数及其取值范围请参考《可设置参数列表\_MSC50.xls》。

♦ text [in]

上传文本

♦ contentLen [in]

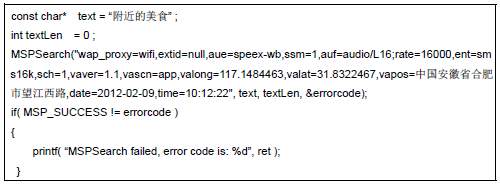
上传文本长度

♦ errorCode [out]

如果函数调用成功返回0，否则返回错误代码，错误代码参见msp\_errors.h，几个主要的返回值如下：



**用法示例：**



**2.2.5 MSPLogout**

**函数原型：**

int MSPAPI MSPLogout()

**功能：**

退出登录。

**参数：**

无

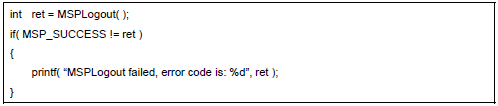
**返回值：**

如果函数调用成功返回MSP\_SUCCESS ， 否则返回错误代码， 错误代码参见msp\_errors.h。

**说明：**

本接口对应于MSPLogin 接口，在使用通用接口相关功能时最后一个被调用，用来对MSC 的通用接口部分进行逆初始化。没有调用MSPLogin 直接调用本接口，则不会有任何效果，调用了MSPLogin 不调用本接口而直接结束线程，则MSC 会产生资源泄漏。

**用法示例：**



1. ASR接口

**3.1 ASR接口简介**

**3.1.1 ASR接口函数列表**

**关于ASR提供如下函数调用：**



本组接口不是线程安全的，用户需要自行保证某些数据的线程安全性，如sessionID。

**3.1.2 返回值说明**

对于开发接口，如果调用成功，返回值为int 型的接口都会返回MSP\_SUCCESS，否则返回错误代码，错误代码参见msp\_errors.h；返回值为指针类型的接口，函数执行失败返回空指针0，错误代码可以通过相关的回传参数查看，函数执行成功但无数据时返回空指

针0，有数据时返回非0 指针。

**3.2 函数调用**

**3.2.1 QISRSessionBegin**

**函数原型：**

const char\* MSPAPI QISRSessionBegin( const char\* grammarList, const char\*params, int\* errorCode )

**功能：**

本接口用来开始一路ASR 会话，并在参数中指定本路ISR 会话用到的语法列表，本次会话所用的参数等。

**参数：**

♦ grammarList [in]

uri-list 格式的语法，可以是一个语法文件的URL 或者一个引擎内置语法列表。可以同时指定多个语法，不同的语法之间以“,”隔开。进行语音听写时不需要语法，此参数设定为NULL 或空串即可；进行语音识别时则需要语法，语法可以在此参数中指定，也可以随后调用QISRGrammarActivate 指定识别所用的语法。

♦ params [in]

本路ISR 会话使用的参数，可设置的参数及其取值范围请参考《可设置参数列表\_MSC50.xls》。

♦ errorCode [out]

如果函数调用成功则其值为0，否则返回错误代码，错误代码参见msp\_errors.h。几个主要的返回值如下：



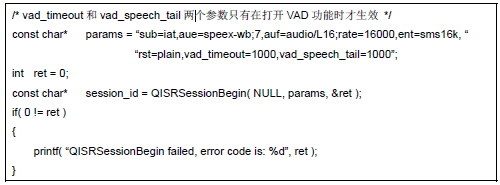
**返回值：**

MSC 为本路会话建立的ID，用来唯一的标识本路会话，供以后调用其他函数时使用。函数调用失败则会返回NULL。

说明：

此处设定的参数在本路会话中一直有效。此函数需要和接口QISRSessionEnd 配对使用，在这两个接口之间可以调用实际完成识别功能的函数如激活语法、写入音频和获取结果等。没有调用此函数则MSC 不会建立和当前线程有关的会话实例，后续的函数调用会因为没有合法的会话ID 而无法完成对应的功能。

**用法示例：**

**3.2.2 QISRGrammarActivate**

**函数原型：**

int MSPAPI QISRGrammarActivate( const char\* sessionID, const char\* grammar,const char\* type, int weight )

**功能：**

本函数用来激活一个指定的语法，语法类型可以是任何一种合法的语法。

参数：

♦ sessionID [in]

由QISRSessionBegin 返回过来的会话ID。

♦ grammar [in]

语法字符串。

♦ type [in]

语法类型，可以是uri-list、abnf 等。

♦ weight [in]

本次传入语法的权重，本参数在MSC5.0 中会被忽略。

**返回值：**

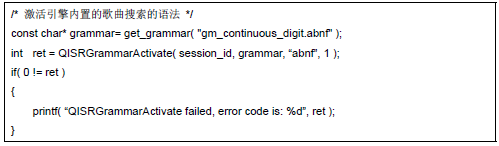
如果函数调用成功返回0，否则返回错误代码，错误代码参见msp\_errors.h。几个主要的返回值如下：



**说明：**

本函数只用在语音识别会话中。在一路会话中，用户可以调用此接口激活一个或多个语法，多个语法之间以“,”隔开，此接口定义的语法和QISRSessionBegin 指定的语法彼此之间相互独立，地位平等，互不干扰，用户可以通过查看识别结果中的grammar 字段来了解识别结果对应的语法。激活的语法从激活成功时刻到会话结束时一直有效。具体的识别语法编写规范可以参阅：《离线命令词识别语法开发指南.pdf》

**用法示例：**



**3.2.3 QISRAudioWrite**

**函数原型：**

int MSPAPI QISRAudioWrite( const char\* sessionID, const void\* waveData, unsigned int waveLen, int audioStatus, int \*epStatus, int \*rsltStatus )

**功能：**

写入本次识别的音频，本接口需要反复调用直到音频写完为止。

**参数：**

♦ sessionID [in]

由QISRSessionBegin 返回过来的会话ID。

♦ waveData [in]

音频数据缓冲区起始地址。

♦ waveLen [in]

音频数据长度，单位字节，其大小不能超过设定的max\_audio\_size。

♦ audioStatus [in]

用来指明用户本次识别的音频是否发送完毕，可能的值如下：



在MSC5.0 中，MSP\_AUDIO\_SAMPLE\_LAST = 4（0x04）用来指明当前的音频已经发送完毕，除此之外的任何值都将被MSC 视为还有后继的音频。

♦ epStatus [out]

端点检测（End-point detected）器所处的状态，可能的值如下：



当epStatus 大于等于3 时，用户应当停止写入音频的操作，否则写入MSC 的音频会被忽略。

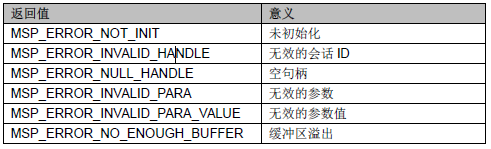
♦ rsltStatus [out]

识别器所处的状态，可能的值如下：



**返回值：**

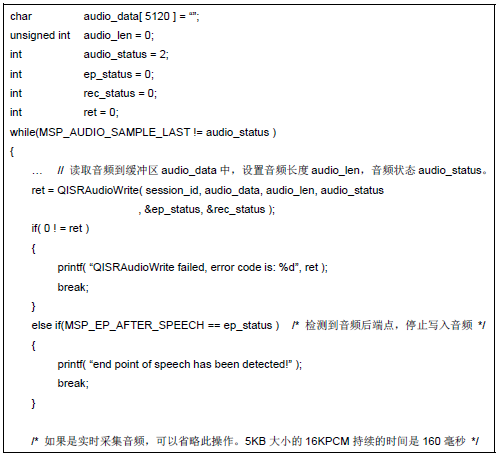
如果函数调用成功返回0，否则返回错误代码，错误代码参见msp\_errors.h。几个主要的返回值如下：



**说明：**

MSC 处理音频的策略是边接收、边压缩（如果音频编码格式不为raw）、边发送。由于音频压缩速度和网络速度的限制，如果音频写入太快太急（如20 倍于音频码率），可能会造成原始音频或压缩音频在MSC 中积累过多，从而造成缓冲区无法再容纳更多的数据而产生MSP\_ERROR\_NO\_ENOUGH\_BUFFER（10117，缓冲区溢出）的错误。调用本接口时，推荐用户在写入音频时采取“边录边发”的方式，即每隔一小段时间将采集到的音频通过本接口写入MSC。这种“边录边发”的方式可以加快结果返回的速度：发送靠后的音频时，前面的音频或许已经被服务器处理过并将部分结果返回了。调用接口时请设置好audioStatus 的值，并检查返回型参数epStatus 的值，以便及时了解音频的前后端点等信息。如果当前写入的不是最后一块音频，需要将audioStatus的值设为2（ISR\_AUDIO\_SAMPLE\_CONTINUE）。如果在音频写入过程中检测到epStatus 的值为3（ISR\_EP\_AFTER\_SPEECH），说明系统已经检测到音频的后端点，则应该立即结束写入音频；如果用户要在检测到音频后端点之前结束音频的写入，需要将最后一块音频数据的audioStatus 设为4（ISR\_AUDIO\_SAMPLE\_LAST）。如果识别状态recogStatus 值为0（ISR\_REC\_STATUS\_SUCCESS）表示已经有部分或全部识别结果缓存在MSC 中了，用户可以调用QISRGetResult 获取这部分结果，再继续调用QISRAudioWrite 以写入后续的音频数据，这种两个接口混调的方式，可以很快的获得（部分）识别结果，特别是使用较大音频进行语音听写时。

**用法示例：**





**3.2.4 QISRGetResult**

**函数原型：**

const char \* MSPAPI QISRGetResult( const char\* sessionID, int\* rsltStatus, int waitTime, int \*errorCode )

**功能：**

获取识别结果。

**参数：**

♦ sessionID [in]

由QISRSessionBegin 返回过来的会话ID。

♦ rsltStatus [out]

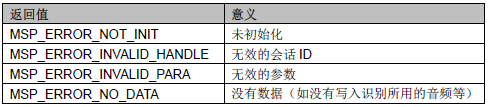
识别结果的状态，其取值范围和含义请参考QISRAudioWrite 的参数recogStatus。

♦ waitTime [in]

与服务器交互的间隔时间，MSC5.0 中，此参数做保留用。

♦ errorCode [out]

如果函数调用成功返回0，否则返回错误代码，错误代码参见msp\_errors.h。几个主要的返回值如下：



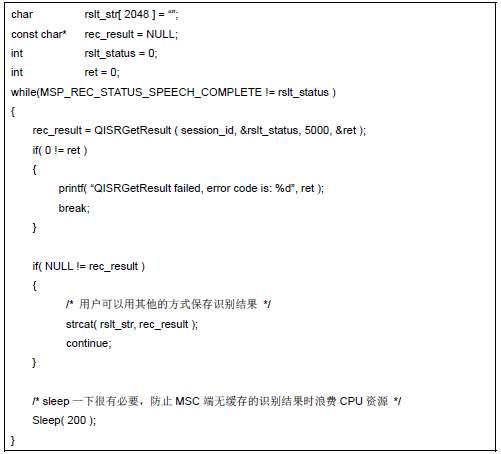
**返回值**：

函数执行失败返回NULL。函数执行成功并且获取到识别结果时返回识别结果，函数执行成功没有获取到识别结果时返回NULL。

**说明：**

本接口一般是在音频写入完毕后调用，以获取语音识别结果。实际应用中，为了及时获取（部分）结果，本接口也可以在写入音频过程中调用，写入音频过程中的调用方法请参考QISRAudioWrite 接口的说明部分。在音频写入完毕后，用户需要反复调用此接口，直到识别结果获取完毕（rlstStatus 值为5）或返回错误码。使用此接口时请注意，如果某次成功调用后没有获得识别结果，请将当前线程sleep 一段时间后再次调用，以防止频繁调用浪费CPU 资源。

**用法示例：**



**3.2.5 QISRSessionEnd**

**函数原型：**

int MSPAPI QISRSessionEnd( const char\* sessionID, const char\* hints )

**功能：**

结束一路ISR 会话。

**参数：**

♦ sessionID [in]

由QISRSessionBegin 返回过来的会话ID。

♦ hints [in]

结束本次会话的原因描述，用于记录日志，便于用户查阅或者跟踪某些问题。

**返回值：**

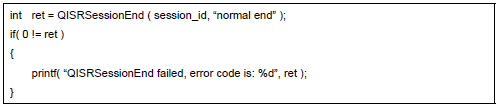
如果函数调用成功返回0， 否则返回错误代码， 错误代码参见msp\_errors.h 和msp\_error.h。几个主要的返回值如下：



**说明：**

本接口需要和QISRSessionBegin 配合使用，用来结束一路ISR 会话。调用本函数后，关于当前会话的所有资源（参数，语法，音频，会话实例等）都会被释放，所以用户不应该再针对该实例做任何操作（比如使用其SessionID 等）。

**用法示例：**



**3.2.6 QISRGetParam**

**函数原型：**

int MSPAPI QISRGetParam( const char\* sessionID, const char\* paramName, char\* paramValue, unsigned int\* valueLen )

**功能**：

查询MSC 记录下来的一些信息如上行流量和下行流量，输入语音的音量等。

**参数：**

♦ sessionID [in]

由QISRSessionBegin 返回过来的会话ID。

♦ paramName [in]

要获取的参数名称；支持同时查询多个参数，查询多个参数时，参数名称按“,” 或“\n” 分隔开来，MSC5.0 支持的参数如下：



♦ paraValue [out]

获取的参数值，以字符串形式返回；查询多个参数时，参数值之间以“;”分开，不支持的参数将返回空的值。

♦ valueLen [out]

参数值的长度。

**返回值：**

如果函数调用成功返回0，否则返回错误代码，错误代码参见msp\_errors.h。几个主要的返回值如下：



**说明：**

流量信息的获取方法和含义请参考3.2.6 节QTTSGetParam 接口的说明部分，由于音量的获取只能在一路会话中进行，所以使用本接口获取音量信息时参数sessionID 不能为NULL。

1. TTS接口

**4.1 TTS接口简介**

**4.1.1 TTS接口函数列表**

关于TTS 提供如下函数调用：



**4.1.2 返回值说明**

对于开发接口，如果调用成功，返回值为int 型的接口都会返回MSP\_SUCCESS，否则返回错误代码，错误代码参见msp\_errors.h；返回值为指针类型的接口，函数执行失败返回空指针0，错误代码可以通过相关的回传参数查看，函数执行成功但无数据时返回空指针0，有数据时返回非0 指针。

**4.2 函数调用**

**4.2.1 QTTSSessionBegin**

**函数原型：**

const char\* MSPAPI QTTSSessionBegin( const char\* params, int\* errorCode )

**功能：**

MSPLogin 成功后开始一路TTS 会话，并在参数中指明本路会话用到的参数。

**参数：**

♦ params [in]

本路TTS 会话使用的参数，可以设置的参数及其取值范围请参考《可设置参数列表\_MSC50.xls》。

♦ errorCode [out]

如果函数调用成功返回0，否则返回错误代码，错误代码参见msp\_errors.h，几个主要的返回值如下：



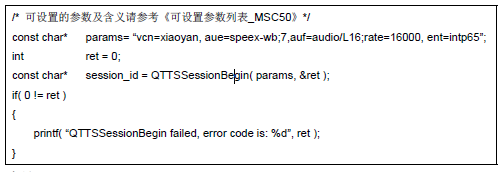
**返回值：**

MSC 为本路会话建立的ID，用来唯一的标识本路会话，供以后调用其他函数时使用。函数调用失败则会返回NULL。

**说明：**

此处设定的参数在本路会话中一直有效。此函数需要和接口QTTSSessionEnd 配对使用，在这两个接口之间可以调用实际完成合成功能的函数如写入合成文本和取音频数据等。没有调用此函数则后续的函数调用会因为没有合法的会话ID 而无法完成对应的功能。

**用法示例：**



**4.2.2 QTTSTextPut**

**函数原型**：

int MSPAPI QTTSTextPut( const char\* sessionID, const char\* textString, unsigned int textLen, const char\* params )

**功能：**

写入要合成的文本。

**参数：**

♦ sessionID [in]

由QTTSSessionBegin 返回过来的会话ID。

♦ textString [in]

将要进行合成的文本。

♦ textLen [in]

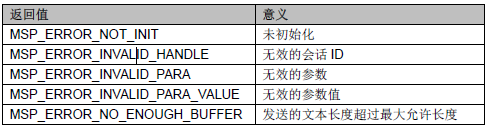
合成文本长度，单位字节。

♦ params [in]

本次合成所用的参数，只对本次合成的文本有效。

**返回值：**

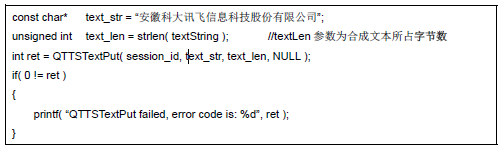
如果函数调用成功返回0，否则返回错误代码，错误代码参见msp\_errors.h。几个主要的值如下：



**说明：**

本接口不支持连续被调用。调用本接口写入合成文本后， 用户需要反复调用QTTSAudioGet 接口来获取音频。

**用法示例：**



**4.2.3 QTTSAudioGet**

**函数原型**：

const void\* MSPAPI QTTSAudioGet( const char\* sessionID, unsigned int\* audioLen,int\* synthStatus, int\* errorCode )

**功能**：

获取合成音频。

**参数：**

♦ sessionID [in]

由QTTSSessionBegin 返回过来的会话ID。

♦ audioLen [out]

合成音频长度，单位字节。

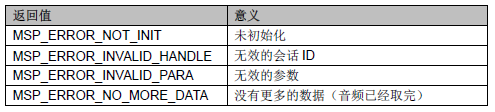
♦ synthStatus [out]

合成音频状态，可能的值如下：



♦ errorCode [out]

如果函数调用成功返回0，否则返回错误代码，错误代码参见msp\_errors.h，几个主要的值如下：



**返回值：**

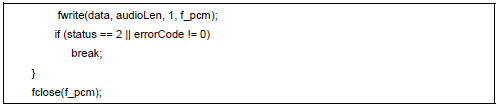
如果函数调用失败返回0 指针，调用成功且有音频数据时返回非0 指针，无音频数据时返回0 指针。

**说明：**

用户需要反复调用此接口，在errorCode 返回值为0 的情况下直到synthStatus 返回MSP\_TTS\_FLAG\_DATA\_END（音频已经取完）为止。在使用中，如果函数调用成功但没有获取到音频数据，用户需要将当前线程sleep 一段时间后再次调用，以防止频繁调用浪费CPU 资源。

**用法示例：**





**4.2.4 QTTSAudioInfo**

**函数原型**：

const char\* MSPAPI QTTSAudioInfo( const char\* sessionID, int\* errorCode )

**功能：**

获取关于合成音频的某些信息如合成音频对应的当前文本位置等。

**参数：**

♦ sessionID [in]

由QTTSSessionBegin 返回过来的会话ID。

♦ errorCode [out]

如果函数调用成功返回0，否则返回错误代码，错误代码参见msp\_errors.h，几个主要的返回值如下：



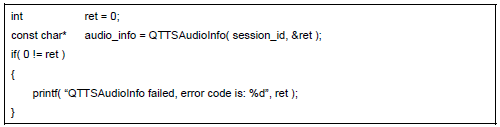
**返回值：**

如果函数调用成功返回字符串指针，否则返回NULL。

**说明：**

本接口用在调用QTTSAudioGet 后，获取关于此段音频的描述信息，目前支持的信息有：此次合成音频对应文本结束位置，用户可以利用此信息来做实时高亮度显示合成的文本等应用，格式为“ced=xxx”。

**用法示例：**



**4.2.5 QTTSSessionEnd**

**函数原型**：

int MSPAPI QTTSSessionEnd( const char\* sessionID, const char\* hints )

**功能：**

结束一路TTS 会话。

**参数：**

♦ sessionID [in]

由QTTSSessionBegin 返回过来的会话ID。

♦ hints [in]

结束本次会话的原因描述，用于记录日志，便于用户查阅或者跟踪某些问题。

**返回值：**

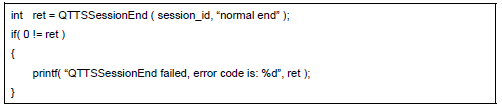
如果函数调用成功返回0，否则返回错误代码，错误代码参见msp\_errors.h。几个主要的返回值如下：



**说明：**

本接口对应于QTTSSessionBegin 接口，用来结束一路TTS 会话。调用本函数后，关于当前会话的所有资源（参数，合成文本，会话实例等）都会被释放，所以用户不应该再针对该实例做任何操作（比如使用其会话ID 等）。

**用法示例：**



**4.2.6 QTTSGetParam**

**函数原型：**

int MSPAPI QTTSGetParam( const char\* sessionID, const char\* paramName, char\*paramValue, unsigned int\* valueLen)

**功能：**

查询MSC 记录下来的一些信息如上行流量和下行流量等。

**参数：**

♦ sessionID [in]

由QISRSessionBegin 返回过来的会话ID，可以为NULL。

♦ paramName [in]

要获取的参数名称。支持同时查询多个参数，查询多个参数时，参数名称以“,” 或“\n”分开，MSC5.0 支持的参数如下：



♦ paraValue [out]

获取到的参数值，以字符串形式返回；查询多个参数时，参数值之间以“;”分开，不支持的参数将返回空的值。

♦ valueLen [in/out]

输入是存放参数值字符串paraValue 缓冲区的长度，输出为参数值字符串的长度。

**返回值：**

如果函数调用成功返回0，否则返回错误代码，错误代码参见msp\_errors.h。几个主要的返回值如下：

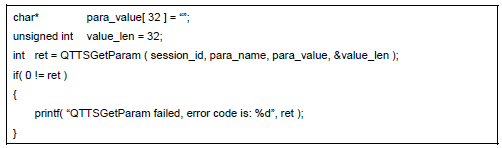


**说明：**

调用本接口时，以获取上行流量为例，如果用户将参数sessionID 设为NULL，则获取到的是用户进程启动后到本路会话开始之前产生的总上行流量；如果用户在一路会话的中间调用本接口并将参数sessionID 设为本路会话的sessionID，则获取到的是本路会话到此时为止产生的上行流量。下行流量与此相似。

**用法示例：**





1. Ros下语音模块代码

本模块代码主要包含以下：

1. user\_command\_client.cpp
2. user\_command\_server.cpp
3. command\_subscriber.cpp

1、user\_command\_server.cpp

在该文件中，建立“user\_command\_server”节点，提供“UserCommand.srv“服务，具体如下：

1）启动录音，并将录制的音频保存成.wav文件；

2）对音频内容进行识别；

3）发布识别的结果；

这里重点说明关于语音识别部分的代码。在该部分代码中需要调用一些通用接口进行基本的登陆登出、数据传输；自动语音识别（Automatic Speech Recognition,ASR）接口，用于开始一路会话，根据传入的语法，进行语音识别，获取识别结果，结束该路回话。

ASR接口函数列表



其返回值说明：

对于开发接口，如果调用成功，返回值为int 型的接口都会返

回MSP\_SUCCESS，否则返回错误代码，错误代码参见msp\_errors.h；返回值为指针类型的接口，函数执行失败返回空指针0，错误代码可以通过相关的回传参数查看，函数执行成功但无数据时返回空指针0，有数据时返回非0 指针。

另，“user\_command\_server”节点在成功进行语音识别之后，将识别结果通过“Command.msg“发布给各个订阅者。

2、user\_command\_client.cpp

在该文件中，建立“user\_command\_client”节点，通过调用

“UserCommand.srv “服务触发“user\_command\_server”节点，提供相应的服务内容。

3、command\_subscriber.cpp

在该文件中，建立“subscriber “节点，用于订阅由“user\_command\_client”节点发布的“Command.msg “消息。在当前版本中，我们加入了一个占位功能，即：接收到识别结果后，执行相应的内容动作，这里，我们将识别到的语音文本内容，再次合成为音频文件，并播放给用户。

因此，这里涉及到了语音合成（Text To Speech，tts）接口。

**其中TTS提供如下函数调用**：

**其返回值说明**：

对于开发接口，如果调用成功，返回值为int 型的接口都会返回MSP\_SUCCESS，否则返回错误代码，错误代码参见msp\_errors.h；返回值为指针类型的接口，函数执行失败返回空指针0，错误代码可以通过相关的回传参数查看，函数执行成功但无数据时返回空指针0，有数据时返回非0 指针。