

# 科大讯飞 MSC 新手指南



科大讯飞股份有限公司  
(版权所有, 翻版必究)

## 目录

重要声明 .....	3
1. 概述 .....	4
2. 工程创建 .....	4
2.1. 添加头文件，库文件，资源文件 .....	4
2.2. 创建源文件 .....	4
2.3. 打印日志 .....	4
2.4. 编译 .....	5
2.5. 运行目标文件 .....	6
FAQ .....	7

# 重要声明

## 版权声明

版权所有 © 2015, 科大讯飞股份有限公司, 保留所有权利。

## 商标声明

科大讯飞股份有限公司的产品是科大讯飞股份有限公司专有。在提及其他公司及其产品时将使用各自公司所拥有的商标, 这种使用的目的仅限于引用。本文档可能涉及科大讯飞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, 除非得到科大讯飞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

科大讯飞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。未得到科大讯飞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 © 科大讯飞股份有限公司

Copyright © 2015 iFLYTEK CO., LTD.

# 1.概述

本文档是开发科大讯飞 linux 语音程序的用户指南，提供工程创建、配置、编译、运行等相关信息的说明。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库文件。

## 2.工程创建

### 2.1.添加头文件，库文件，资源文件

- 在当前目录建立一个新目录 demo:\$ mkdir demo。见图 1。

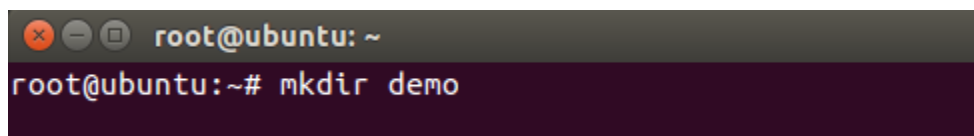


图 1

- 将 SDK 下的 bin, include, libs 目录拷贝到 demo 目录下。

### 2.2.创建源文件

- 在 demo 目录新建文件 demo.c:\$ touch demo.c。见图 2

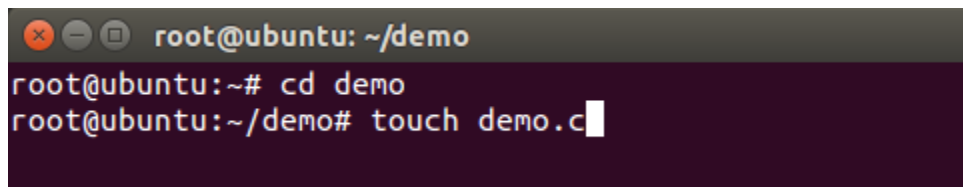


图 2

- 编辑程序源代码：代码可参考 samples 中的语音示例。

### 2.3.打印日志

- 运行程序后，bin/msc 目录下会生成日志。见图 3。

（注意：msc 文件夹下需有 msc.cfg 文件）

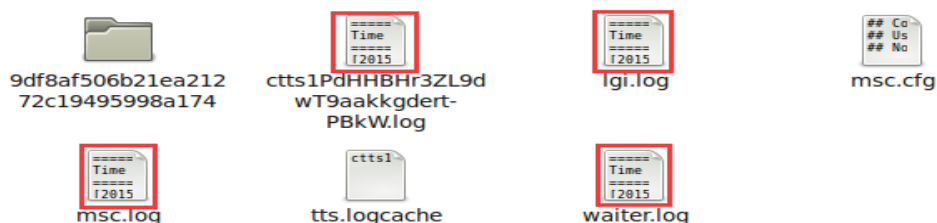


图 3

## 2.4.编译

- 在 demo 目录下，创建 Makefile 文件，具体参见 samples 下的 Makefile，修改下路径和目标文件即可。见图 4。

#common makefile header

```
DIR_INC = ./include
DIR_BIN = ./bin
DIR_LIB = ./libs

TARGET = demo
BIN_TARGET = $(DIR_BIN)/$(TARGET)

CROSS_COMPILE =
CFLAGS = -g -Wall -I$(DIR_INC)

ifdef LINUX64
LDFLAGS := -L$(DIR_LIB)/x64
else
LDFLAGS := -L$(DIR_LIB)/x86
endif
LDFLAGS += -lmsc -ldl -lpthread -lrt

OBJECTS := $(patsubst %.c,%.o,$(wildcard *.c))

$(BIN_TARGET) : $(OBJECTS)
    $(CROSS_COMPILE)gcc $(CFLAGS) $^ -o $@ $(LDFLAGS)

%.o : %.c
    $(CROSS_COMPILE)gcc -c $(CFLAGS) $< -o $@

clean:
    @rm -f *.o $(BIN_TARGET)

.PHONY:clean
```

图 4

- 将 samples 目录下“32bit\_make.sh”文件或者“64bit\_make.sh”文件拷到 demo 目录下，修改 libmsc.so 库搜索路径。见图 5，图 6。

```
#编译32位可执行文件
make clean;make
#设置libmsc.so库搜索路径
export LD_LIBRARY_PATH=$(pwd)/libs/x86/
```

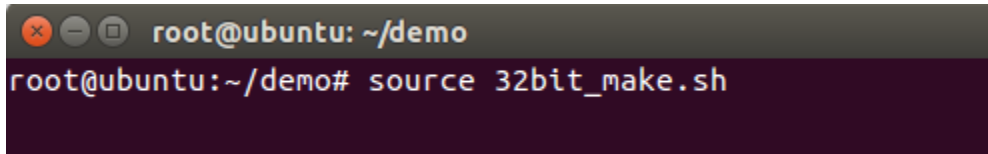
图 5

```
#编译64位可执行文件
make clean;make LINUX64=1
#设置libmsc.so库搜索路径
export LD_LIBRARY_PATH=$(pwd)/libs/x64/
```

图 6

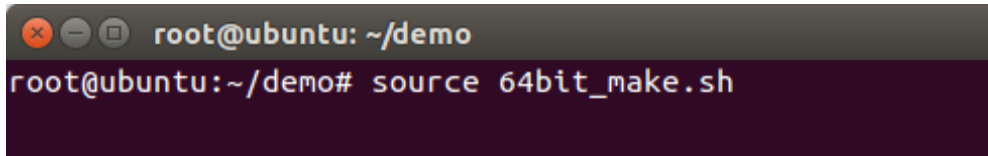
- cd 到 demo 目录下，执行“source 32bit\_make.sh”或者“source 64bit\_make.sh”

命令。见图 7，图 8。



```
root@ubuntu: ~/demo
root@ubuntu:~/demo# source 32bit_make.sh
```

图 7

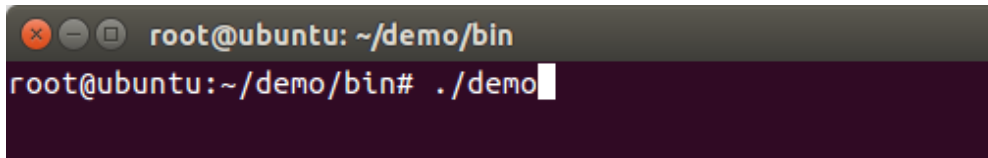


```
root@ubuntu: ~/demo
root@ubuntu:~/demo# source 64bit_make.sh
```

图 8

## 2.5.运行目标文件

- cd 到 bin 目录下运行目标文件。见图 9。



```
root@ubuntu: ~/demo/bin
root@ubuntu:~/demo/bin# ./demo
```

图 9

## FAQ

### 1. 如何联系我们获得技术支持？

科大讯飞提供以下方式的技术支持：

#### 1) 电话支持

请于周一～周五，北京时间 9：00～17：00 间，拨打电话： 0551—65309063 获得技术支持信息。

#### 2) 电子邮件支持

请将问题的详细描述发至：[msp\\_support@iflytek.com](mailto:msp_support@iflytek.com)。

#### 3) 在线支持

请登录我们的论坛 <http://club.voicecloud.cn>。

联系时对问题的描述请尽量包含以下内容：

1. 系统配置（包括 CPU、内存、硬盘、操作系统及产品版本等信息）
2. 问题细节（包括问题的重现过程及合成的文本内容、识别音频等）
3. 问题重现（包括详细的操作过程和运行日志等）

### 2. 拿到了合成音频但不知道如何来播放

答：合成拿到的音频是没有音频头的，音频头中含有音频格式、采样率、音频长度等播放音频所需信息。拿到合成音频后，用户可以添加音频头，可参考例子 `tts_sample` 中的代码，然后使用常规播放器来播放；也可以使用 Cool Edit 等软件手动选择音频参数来播放。

### 3. 如何进行大文本的合成

答：语音云一次语音合成允许的合成文本大小不超过 8192 个字节，所以对于长度超过此值的大合成文本，用户可以采用“分段合成”的方式，即先将大文本按照标点符号如句号进行切分，然后对每一段文本分别进行合成。进行分段合成时，用户既可以在一路会话中循环使用 `QTTSTextPut+QTTSAudioGet` 组合完成合成，也可以为每一段文本使用一路独立的会话完成合成。

### 4. 获取不到识别/听写结果。

答：原因可能是：

- 1) `QISRSessionBegin` 的参数设置不正确，如没有设置好正确的引擎类型等。
- 2) 音频格式不对，客户端支持的音频编解码算法只支持 16 位 Intel PCM 格式的音频。

### 5. 能获取到语音听写结果但是不全。

答：此问题主要是在调用 QISRAudioWrite 时没有正确设置参数 audioStatus 所致，此参数在写入非最后一个音频数据块时需要设置为 2，写入最后一个数据块时需要设置为 4，以告诉 MSC 音频写入完毕。如果只有一个音频数据块，audioStatus 也需要设置为 4。

6. 可以拿到识别或转写结果但是响应很慢。

答：此问题可以尝试如下方法来解决：

- 1) 调用 QISRAudioWrite 接口写音频数据时，尽量做到“匀速发送”——周期性的发送定长数据，做到边录边发，避免一次发送数据量过大的音频。
- 2) 采用 QISRAudioWrite 接口和 QISRGetResult 接口混调的方式。在调用 QISRAudioWrite 接口时，可以检查 out 型参数 recogStatus，如果其值为 0，表明已经有（部分）识别结果缓存在 MSC 中了，此时可以调用 QISRGetResult 来获取结果。

7. 运行 demo 时报错：error while loading shared libraries: libmsc.so: cannot open shared object file: No such file or directory

答：出现这类错误表示，系统不知道 libmsc.so 放在哪个目录下，可通过设置变量 LD\_LIBRARY\_PATH 解决，具体解决方法：export LD\_LIBRARY\_PATH=XXX（XXX 为 libmsc.so 所在目录路径），一般来说这只是一种临时的解决方案，在没有权限或临时需要时使用。其它解决方案可在网上查阅相关资料获得。