# Fonts

## 添加字库到系统的完整步骤

添加字库到系统的完整步骤

这里我说一下添加字库到系统的完整步骤：

1：Copy 自己添加的字体库文件（后缀是.ttf）到目录 frameworks/base/data/fonts

2：Modify framworks/base/data/fonts/Android.mk

3：Modify frameworks/base/data/fonts/fonts.mk

4：Modify frameworks/base/data/fonts/system\_fonts.xml

5：Modify frameworks/base/data/fonts/fallback\_fonts.xml

6：Modify frameworks/base/data/fonts/fonts.xml

步骤一：

拷贝文件这个很简单，不需要细说，记住需要到的目录就好；

步骤二：

Modify framworks/base/data/fonts/Android.mk

在font\_src\_files 中添加需要编译的字体库名字：

font\_src\_files := \

Roboto-Regular.ttf \

Roboto-Bold.ttf \

...

simsun.ttc \

simkai.ttf \

msyh.ttf \

impact-Regular.ttf \

AvenirNextCondensed-Bold.ttf \

PosSong-8.ttf \

PosSong-16.ttf \

PosSong-24.ttf \

simfang.ttf

步骤三：

Modify frameworks/base/data/fonts/fonts.mk

在PRODUCT\_PACKAGES下添加需要编译的字体库名字：

PRODUCT\_PACKAGES := \

DroidSansFallback.ttf \

Roboto-Regular.ttf \

......

simsun.ttc \

simkai.ttf \

msyh.ttf \

impact-Regular.ttf \

AvenirNextCondensed-Bold.ttf \

PosSong-8.ttf \

PosSong-16.ttf \

PosSong-24.ttf \

simfang.ttf

步骤四：

Modify frameworks/base/data/fonts/system\_fonts.xml

下面对应用使用应用内部Assets资源与调用系统的字体库方式进行比较：

textView1.setTypeface(Typeface.create("simkai",Typeface.NORMAL));//调用系统

textView1.setTypeface(Typeface.createFromAsset(this.getAssets(),"simkai.ttf"));//调用应用内部Assets资源

从上面代码差异我们就知道，调用系统的时候不需要使用具体执行一个具体的ttf(simkai.ttf)库文件，但是需要指定一个具体的名字("simkai")，其实这个名字就是("simkai")就是指向了simkai.ttf 文件，那这个名字和具体的字体库文件是如何关联的吗(看着就像把字体库的全名(simkai.ttf)去掉.ttf的后缀得到的(simkai))？那具体是不是这样呢？答案是否定的，只是我们习惯的配置方式是这样配置而已，至于具体的配置在那里呢？

frameworks/base/data/fonts/system\_fonts.xml

在该文件中添加字体库的nameset和fileset对应关系；

<family>

<nameset>

<name>simkai</name>

</nameset>

<fileset>

<file>simkai.ttf</file>

</fileset>

</family>

<family>

<nameset>

<name>AvenirNextCondensed</name>

</nameset>

<fileset>

<file>AvenirNextCondensed-Bold.ttf</file>

</fileset>

</family>

看到没，这里simkai的nameset指向了simkai.ttf的fileset。你也可以根据你自己的名字来命名；

步骤5：

Modify frameworks/base/data/fonts/fallback\_fonts.xml

在该文件中添加字体库文件的fileset

<family>

<fileset>

<file>simkai.ttf</file>

</fileset>

</family>

<!-- add for msyh-->

<family>

<fileset>

<file>msyh.ttf</file>

</fileset>

</family>

<family>

<fileset>

<file>impact-Regular.ttf</file>

</fileset>

</family>

<family>

<fileset>

<file>AvenirNextCondensed-Bold.ttf</file>

</fileset>

</family>

步骤6：

Modify frameworks/base/data/fonts/fonts.xml

配置系统库字体，在Android 5.1中必须配置，相同字体库文件的 family name 必须和和前面system\_fonts.xml文件中的nameset一致（理论上一致，有兴趣的读者可以尝试不一致，我没有试过），textView1.setTypeface(Typeface.create("simkai",Typeface.NORMAL));

<family name="simkai">

<font weight="400" style="normal">simkai.ttf</font>

</family>

<family name="msyh">

<font weight="400" style="normal">msyh.ttf</font>

</family>

<family name="impact">

<font weight="400" style="normal">impact-Regular.ttf</font>

</family>

<family name="simfang">

<font weight="400" style="normal">simfang.ttf</font>

</family>

<family name="AvenirNextCondensed">

<font weight="400" style="normal">AvenirNextCondensed-Bold.ttf</font>

</family>

特别注意：我们修改了这几个文件还需要注意一些编译细节，比如上述几个.xml文件都会被编译到/system/etc/目录下，但是我在Android 5.1 中编译了却有一个文件没有生效，那就是fonts.xml，经过比较，我发现系统中的fonts.xml没有我添加的内容，经过在系统中查找，发现了猫腻：

vendor/qcom/proprietary/qrdplus/globalization/engine目录下的Android.mk文件中有如下语句：

include $(CLEAR\_VARS)

LOCAL\_MODULE := fonts.xml

LOCAL\_SRC\_FILES := fonts-Rev.xml

LOCAL\_MODULE\_CLASS := ETC

LOCAL\_MODULE\_TAGS := optional

LOCAL\_MODULE\_PATH := $(TARGET\_OUT)/etc

include $(BUILD\_PREBUILT)

上面这个代码的意思是把fonts-Rev.xml 编译到编译成fonts.xml，拷贝到/system/etc目录下；看到这里就知道为什么我的修改没生效了吧；我的修改先编译到/system/etc目录下，然后被这个fonts-Rev.xml编译的fonts.xml覆盖了(经过对比发现编译出来的/system/etc/fonts.xml果真和这个fonts-Rev.xml一模一样)；两种解决办法：1,将我的修改添加到vendor/qcom/proprietary/qrdplus/globalization/engine目录下的fonts-Rev.xml;2,在vendor/qcom/proprietary/qrdplus/globalization/engine的Android.mk中将上面的那几行代码注释掉；

[Android 添加字体库到系统](https://www.jianshu.com/p/296b57cf99b6)

## 系统裁剪之原生字体库精简

<https://blog.csdn.net/kanyueliang410/article/details/78975022>

<https://www.aliyun.com/jiaocheng/42326.html>

# Keyboards

# Keys defined by HID usages

key usage 0x0c006F BRIGHTNESS\_UP

key usage 0x0c0070 BRIGHTNESS\_DOWN

# Joystick and game controller axes.

# Axes that are not mapped will be assigned generic axis numbers by the input subsystem.

axis 0x00 X

axis 0x01 Y

axis 0x02 Z

axis 0x03 RX

axis 0x04 RY

axis 0x05 RZ

axis 0x06 THROTTLE

axis 0x07 RUDDER

axis 0x08 WHEEL

axis 0x09 GAS

axis 0x0a BRAKE

axis 0x10 HAT\_X

axis 0x11 HAT\_Y\

# Sounds

# videos