# 第三方应用集成Android.mk编写说明

一、Android.mk示例

LOCAL\_PATH := $(my-dir)

include $(CLEAR\_VARS)

LOCAL\_MODULE := ChangBa

LOCAL\_MODULE\_TAGS := optional

LOCAL\_MODULE\_CLASS := APPS

LOCAL\_MODULE\_SUFFIX :=$(COMMON\_ANDROID\_PACKAGE\_SUFFIX)

LOCAL\_SRC\_FILES := ChangBa.apk

LOCAL\_BUILT\_MODULE\_STEM := package.apk

LOCAL\_CERTIFICATE := PRESIGNED

LOCAL\_MULTILIB := 32

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

include $(TOPDIR)packages/apps/ThirdParty/EXTRACT\_SO.mk

include $(BUILD\_PREBUILT)

其中LOCAL\_MODULE、LOCAL\_SRC\_FILES、LOCAL\_MULTILIB必须存在，因为EXTRACT\_SO.mk脚本中使用到了这几个变量。

include $(TOPDIR)packages/apps/ThirdParty/EXTRACT\_SO.mk

这是提取so的脚本，写在include $(BUILD\_PREBUILT)之前。

二、基本原理

以imx8qxp的androidp平台为例，

通过查询ro.product.cpu.abilist64和ro.product.cpu.abilist32这两个属性：

[ro.product.cpu.abilist32]: [armeabi-v7a,armeabi]

[ro.product.cpu.abilist64]: [arm64-v8a]

可知，平台支持的cpu api有arm64-v8a, armeabi-v7a, armeabi。

脚本的逻辑如下：

1. 清除LOCAL\_MODULE定义的文件夹
2. 将LOCAL\_SRC\_FILES定义的apk中的lib目录解压缩到LOCAL\_MODULE目录
3. 如果LOCAL\_MULTILIB:=64，则使用lib/arm64-v8a/文件夹中的so；如果LOCAL\_MULTILIB:=32,则使用apk中lib/ armeabi-v7a/文件夹中的so,如果armeabi-v7a文件夹不存在，则使用lib/ armeabi/文件夹。