[FAQ13697]L 版本如何将第三方so库打包到apk

[DESCRIPTION]

1、如何判断第三方库文件是32 bit/64 bit?

2、如何将没有源码的第三方库打包到apk ？

3、对于没有root权限的user 版本，如何确保第三方so库可以打包到apk 里面, 采用adb 命令进行install ？

[SOLUTION]

1、如何判断第三方库文件是32 bit/64 bit?

使用Linux 命令： file xxx.so

可以看到拿到的xxx.so是哪种格式

2、 如何将没有源码的第三方库打包到apk ？
请把此so 放在apk 目录下，也可以存放在vendor/meidatek/libs/$project/$folder/xxx.so
采用prebuilt 的方式，在当前so 所在目录下写 Android.mk ，内容类似如下：

```
LOCAL_PATH :=$(call my-dir)
include $(CLEAR_VARS)
LOCAL_MODULE := 此so 库名(不加so后缀)
LOCAL_SRC_FILES_32 := xxx.so (表示是32 bit 的so)
LOCAL_SRC_FILES_64 := xxx.so (表示是64 bit的so)
LOCAL_MULTILIB := 32/64/BOTH (只编译32bit/64bit/both)
LOCAL_MODULE_CLASS := SHARED_LIBRARIES
LOCAL_MODULE_SUFFIX := .so
include $(BUILD_PREBUILT)
```

然后此so 就可以被直接在apk 的Android.mk 使用，例如：
在alps\packages\apps\Tester添加一个apk, Android.mk写法

```
LOCAL_PATH := $(call my-dir)
include $(CLEAR_VARS)
LOCAL_MODULE_TAGS := tests
```

```
LOCAL_SRC_FILES := $(call all-java-files-under, src)
#需要引用的静态库
LOCAL_STATIC_JAVA_LIBRARIES := \
jxl \
AChartEngine_fat
LOCAL_PACKAGE_NAME :=Tester
LOCAL_CERTIFICATE := platform
LOCAL_PROGUARD_FLAG_FILES := proguard.flags
LOCAL_DEX_PREOPT := false
LOCAL_PROGUARD_ENABLED := disabled
# 对前面prebuilt 的动态库的引用
LOCAL_JNI_SHARED_LIBRARIES +=xxx
# 此apk 限制为32 bit
LOCAL_MULTILIB := 32
include $(BUILD_PACKAGE)
##################################################
# 此apk 引用的静态库（位置：alps\packages\apps\Tester\libs\jxl.jar 与
AChartEngine_fat.jar），
对静态库采用prebuilt 的编译方式
include $(CLEAR_VARS)
LOCAL_PREBUILT_STATIC_JAVA_LIBRARIES :=jxl: libs/jxl.jar \
AChartEngine_fat: libs/AChartEngine_fat.jar\
LOCAL_MODULE_TAGS := optional
include $(BUILD_MULTI_PREBUILT)
# Use the following include to make our test apk.
include $(callall-makefiles-under,$(LOCAL_PATH))
```

3、对于没有root权限的user 版本，如何确保第三方so库可以打包到apk 里面, 采用adb 命令进行install ？

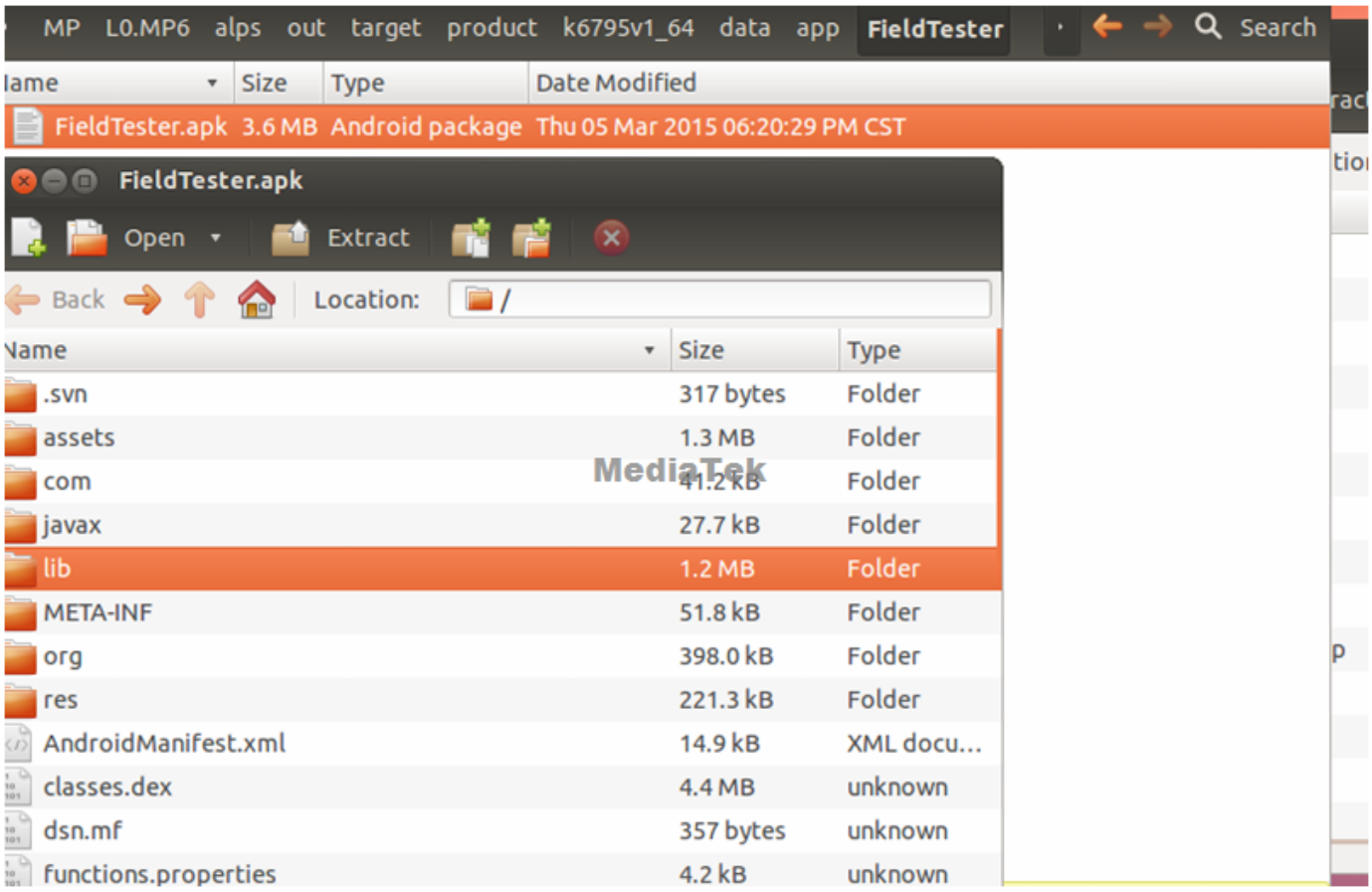
请在apk 对应的Android.mk 将

```
LOCAL_MODULE_TAGS := tests ，
```

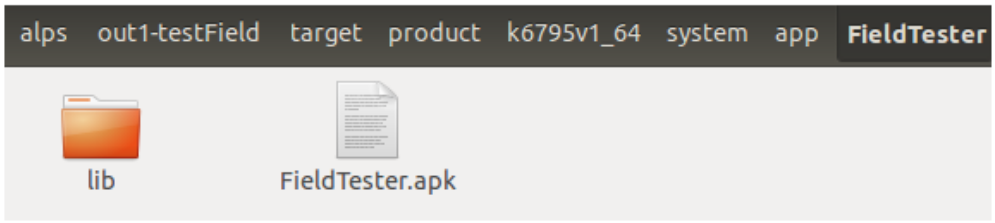
此apk 默认会被编译到out/target/product/$project/data/app/tester/tester.apk 但不会被安装到system.img，

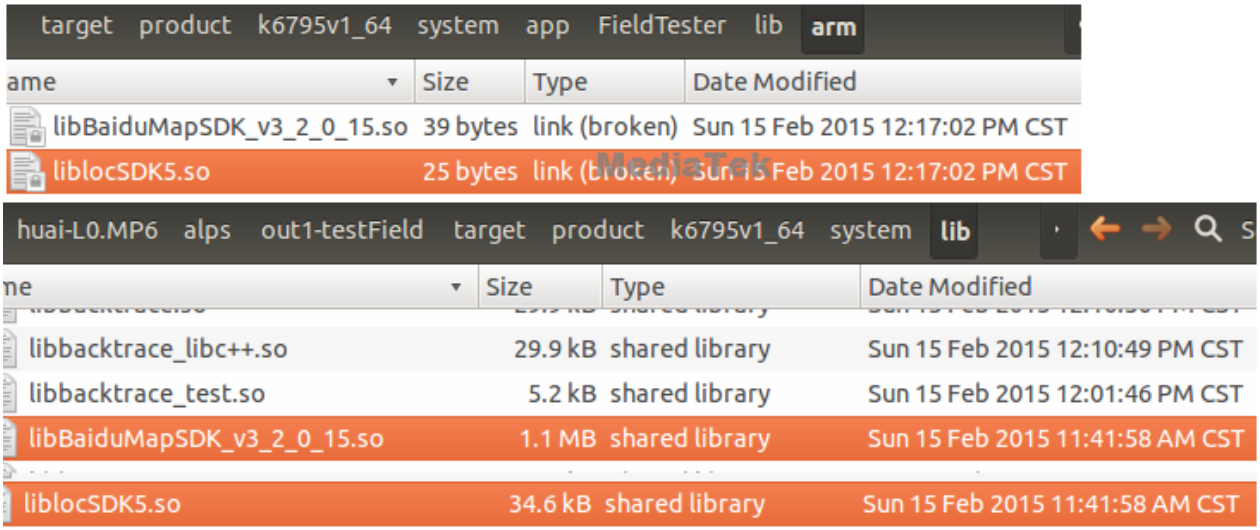
这种适应于采用adb 命令install此apk的方式，会将需要的库整体打包到apk, 类似如下apk结构：

引用的三方so 库打包在apk 的 lib 中



LOCAL_MODULE_TAGS := optional /eng 此apk可以被打包到systemimage,
而引用的三方so库，实际存放在system/lib 或lib64中，
apk结构类似如下：lib中存放三方so 库的链接，实际的so 存储在system/lib 下





可以采用flashtool download，此apk 可以正常执行
对eng 版本和有root 权限的user 版本，install apk 后，需要把so 库也push 进去，否则apk 无法运行！！！