[FAQ07293]如何用jar包代替java代码编译

[DESCRIPTION]

有时不希望某个模块的java代码被人看到，希望是以jar包release，可以按以下方式修改

注意：最终是编译成jar包的模块才能这样做

[KEYWORD]

jar

java

编译

[SOLUTION]

1. 先完整编译一遍工程（任意eng/user），然后保存要替换java代码的jar包，在alps/out/target/common/obj/JAVA_LIBRARIES/xxx__intermediates目录下的
classes.jar
javalib.jar

2. 请将以下文件保存成java_library_prebuilt.mk，并放在alps/build/core/目录下面：

注意：有11处需要tab键开头（make语法对命令需要tab键开头，但网站不支持tab键），请保存成文件后打开编辑，将11处(以下有标志)重新用tab键处理一遍。

```
##Writen by guangye.yang@mediatek.com
##
LOCAL_SRC_FILES :=
ifdef LOCAL_IS_HOST_MODULE
$(error $(LOCAL_PATH): Just for target java library)
endif

LOCAL_MODULE_SUFFIX := $(COMMON_JAVA_PACKAGE_SUFFIX)
LOCAL_MODULE_CLASS := JAVA_LIBRARIES
LOCAL_BUILT_MODULE_STEM := javalib.jar

intermediates.COMMON := $(call local-intermediates-dir,COMMON)
full_classes_jar := $(intermediates.COMMON)/classes.jar
common_javalib.jar := $(intermediates.COMMON)/$(LOCAL_BUILT_MODULE_STEM)

LOCAL_INTERMEDIATE_TARGETS += $(full_classes_jar) $(common_javalib.jar)
```

```
ifneq (true,$(WITH_DEXPREOPT))
LOCAL_DEX_PREOPT :=
else
ifeq (,$(TARGET_BUILD_APPS))
ifndef LOCAL_DEX_PREOPT
LOCAL_DEX_PREOPT := true
endif
endif
endif
ifeq (false,$(LOCAL_DEX_PREOPT))
LOCAL_DEX_PREOPT :=
endif

include $(BUILD_SYSTEM)/base_rules.mk

$(cleantarget): PRIVATE_CLEAN_FILES += $(intermediates.COMMON)

$(full_classes_jar) : $(LOCAL_PATH)/classes.jar | $(ACP)
$(call copy-file-to-target) #注意：这里必须以tab键开头

$(common_javalib.jar) : $(LOCAL_PATH)/$(LOCAL_BUILT_MODULE_STEM) | $(full_classes_jar)
$(ACP)
$(call copy-file-to-target) #注意：这里必须以tab键开头

ifdef LOCAL_DEX_PREOPT
dexpreopt_boot_jar_module := $(filter $(LOCAL_MODULE),$(DEXPREOPT_BOOT_JARS_MODULES))
ifneq ($(dexpreopt_boot_jar_module),)
dexpreopted_boot_jar :=
$(DEXPREOPT_BOOT_JAR_DIR_FULL_PATH)/$(dexpreopt_boot_jar_module)_nodex.jar
$(LOCAL_BUILT_MODULE) : $(dexpreopted_boot_jar) | $(ACP)
$(call copy-file-to-target) #注意：这里必须以tab键开头

dexpreopted_boot_odex :=
$(DEXPREOPT_BOOT_JAR_DIR_FULL_PATH)/$(dexpreopt_boot_jar_module).odex
built_odex := $(basename $(LOCAL_BUILT_MODULE)).odex
$(built_odex) : $(dexpreopted_boot_odex) | $(ACP)
$(call copy-file-to-target) #注意：这里必须以tab键开头

else
built_odex := $(basename $(LOCAL_BUILT_MODULE)).odex
$(built_odex): PRIVATE_MODULE := $(LOCAL_MODULE)
$(built_odex) : $(DEXPREOPT_BOOT_ODEXS)
$(built_odex) : $(common_javalib.jar) | $(DEXPREOPT) $(DEXOPT)
@echo "Dexpreopt Jar: $(PRIVATE_MODULE) ($@)" #注意：这里必须以tab键开头
$(hide) rm -f $@ #注意：这里必须以tab键开头
@mkdir -p $(dir $@) #注意：这里必须以tab键开头
$(call dexpreopt-one-file,$<,$@) #注意：这里必须以tab键开头
```

```
$(LOCAL_BUILT_MODULE) : $(common_javalib.jar) | $(ACP) $(AAPT)
$(call copy-file-to-target) #注意：这里必须以tab键开头
ifneq (nostripping,$(LOCAL_DEX_PREOPT))
$(call dexpreopt-remove-classes.dex,$@) #注意：这里必须以tab键开头
endif
endif
else
$(LOCAL_BUILT_MODULE) : $(common_javalib.jar) | $(ACP)
$(call copy-file-to-target) #注意：这里必须以tab键开头
endif
```

3. 在alps/build/core/config.mk里添加：

```
BUILD_JAVA_LIBRARY_PREBUILT:= $(BUILD_SYSTEM)/java_library_prebuilt.mk
```

4. 删除对应的java代码及对应的Android.mk，然后在当前目录创建新的Android.mk，并复制第1步备份的2只文件到Android.mk旁边(注意，文件名不能修改，否则无法识别!!!)，新的Android.mk编写如下：

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

LOCAL_MODULE := services
LOCAL_MODULE_PATH := $(TARGET_OUT_JAVA_LIBRARIES)

include $(BUILD_JAVA_LIBRARY_PREBUILT)
```

5. 重新完整的编译工程(任意eng/user)，下载查看效果

6. 重要提示，由于第1步保存的classes.jar是没有经过proguard，流出去很容易被反编译，请自行做proguard，但是请不要把接口proguard，否则编译失败！