Android Makefile 文件讲解

android编译系统的makefile文件Android.mk写法如下

1. Android.mk文件首先需要指定LOCAL\_PATH变量，用于查找源文件。由于一般情况下  
   Android.mk和需要编译的源文件在同一目录下，宏函数“my-dir”右编译系统提供的，用于返回当前路径。所以定义成如下形式：  
   LOCAL\_PATH:=$(call my-dir)  
   上面的语句的意思是将LOCAL\_PATH变量定义成本文件所在目录路径。
2. Android.mk中可以定义多个编译模块，每个编译模块都是以include $(CLEAR\_VARS)开始  
   以include $(BUILD\_XXX)结束。  
   include $(CLEAR\_VARS)  
   CLEAR\_VARS由编译系统提供，指定让GNU MAKEFILE为你清除除LOCAL\_PATH以外的所有LOCAL\_XXX变量，  
   如LOCAL\_MODULE，LOCAL\_SRC\_FILES，LOCAL\_SHARED\_LIBRARIES，LOCAL\_STATIC\_LIBRARIES等。  
   include $(BUILD\_STATIC\_LIBRARY)表示编译成静态库  
   include $(BUILD\_SHARED\_LIBRARY)表示编译成动态库。  
   include $(BUILD\_EXECUTABLE)表示编译成可执行程序
3. LOCAL\_MODULE\_TAGS := optional 表示该模块在什么版本下才编译

Eng: 只在Eng版本下才编译  
User: 只在user版本下才编译  
Tests: 只在Tests版本下才编译  
Optional: 在所有的版本下都编译

1. LOCAL\_SRC\_FILES: 编译的源代码文件列表 添加所有的java 源文件LOCAL\_SRC\_FILES := $(call all-java-files-under, src)
2. LOCAL\_SDK\_VERSION := current 该标志会使应用程序不能访问hide的API
3. LOCAL\_MODULE:= jar 包名称
4. LOCAL\_PACKAGE\_NAME := APK 包的名称
5. LOCAL\_JAVA\_LIBRARIES := platform core ext 依赖的共享jar
6. LOCAL\_STATIC\_JAVA\_LIBRARIES ：= 依赖的静态jar包 （jar 包是使用 include $(BUILD\_STATIC\_JAVA\_LIBRARY) 生成的。）
7. LOCAL\_CERTIFICATE: 签名认证   
   1.系统中所有使用android.uid.system作为共享UID的APK，都会首先在manifest节点中增加android:sharedUserId="android.uid.system"，然后在Android.mk中增加LOCAL\_CERTIFICATE := platform。   
   2. 系统中所有使用android.uid.shared作为共享UID的APK，都会在manifest节点中增加android:sharedUserId="android.uid.shared"，然后在Android.mk中增加LOCAL\_CERTIFICATE := shared。   
   3.系统中所有使用android.media作为共享UID的APK，都会在manifest节点中增加android:sharedUserId="android.media"，然后在Android.mk中增加LOCAL\_CERTIFICATE := media

Examples.

1. 编译一个应用程序(APK)  
     LOCAL\_PATH := $(call my-dir)  
     include $(CLEAR\_VARS)  
        
     # Build all java files in the java subdirectory  
     LOCAL\_SRC\_FILES := $(call all-subdir-java-files)  
        
     # Name of the APK to build  
     LOCAL\_PACKAGE\_NAME := LocalPackage  
        
     # Tell it to build an APK  
     include $(BUILD\_PACKAGE)
2. 编译静态库

#清除java环境变量

include $(CLEAR\_VARS)

LOCAL\_MODULE\_TAGS := user

#添加aidl文件

LOCAL\_SRC\_FILES += src/net/sunniwell/download/aidl/IDownload.aidl

# SDK 版本

LOCAL\_SDK\_VERSION := current

#jar包的名字

LOCAL\_MODULE:= DownLoadServerapi

#编译 DownLoadServerapi jar 包

include $(BUILD\_STATIC\_JAVA\_LIBRARY)

1. 编译一个依赖于静态Java库(static.jar)的应用程序  
     LOCAL\_PATH := $(call my-dir)  
     include $(CLEAR\_VARS)  
        
     # List of static libraries to include in the package  
     LOCAL\_STATIC\_JAVA\_LIBRARIES := DownLoadServerapi 第三方jar包   
        
     # Build all java files in the java subdirectory  
     LOCAL\_SRC\_FILES := $(call all-subdir-java-files)  
        
     # Name of the APK to build  
     LOCAL\_PACKAGE\_NAME := LocalPackage  
        
     # Tell it to build an APK  
     include $(BUILD\_PACKAGE)
2. 编译一个需要用平台的key签名的应用程序  
     LOCAL\_PATH := $(call my-dir)  
     include $(CLEAR\_VARS)  
        
     # Build all java files in the java subdirectory  
     LOCAL\_SRC\_FILES := $(call all-subdir-java-files)  
        
     # Name of the APK to build  
     LOCAL\_PACKAGE\_NAME := LocalPackage  
        
     LOCAL\_CERTIFICATE := platform  
        
     # Tell it to build an APK  
     include $(BUILD\_PACKAGE)
3. 添加一个静态JAVA库  
     LOCAL\_PATH := $(call my-dir)  
     include $(CLEAR\_VARS)  
        
     # Build all java files in the java subdirectory  
     LOCAL\_SRC\_FILES := $(call all-subdir-java-files)  
        
     # Any libraries that this library depends on  
     LOCAL\_JAVA\_LIBRARIES := android.test.runner  
        
     # The name of the jar file to create  
     LOCAL\_MODULE := sample  
        
     # Build a static jar file.  
     include $(BUILD\_STATIC\_JAVA\_LIBRARY)
4. 引用第三方JAR 包方式

#引入第三方包

include $(CLEAR\_VARS)

# 表示在当前目录下的lib/iptv.jar进行预处理到android jar包的目录中去

#LOCAL\_PREBUILT\_JAVA\_LIBRARIES := libiptv:lib/iptv.jar

LOCAL\_PREBUILT\_STATIC\_JAVA\_LIBRARIES := \

libiptv:lib/iptv.jar

include $(BUILD\_MULTI\_PREBUILT)

需要在编译apk 中进行、并添加

LOCAL\_STATIC\_JAVA\_LIBRARIES := libiptv