**软件集成方案书**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **序号** | **变更（+/-）说明** | **作者** | **版本号** | **日期** | **批准** |
| 1 | 大纲 | 梁石麟 | V1.0 | 2015-4-14 |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

[1. 引言 4](#_Toc32192)

[1.1. 编写目的 4](#_Toc21893)

[2. 打包工程管理 4](#_Toc18859)

[2.1. 打包工程结构 4](#_Toc1800)

[2.1.1. 主目录结构： 4](#_Toc7411)

[2.2. 分支版本 4](#_Toc24291)

[2.3. 打包工程的版本管理 5](#_Toc1263)

[3. 软件版本管理 5](#_Toc10852)

[3.1. Normal版与分支版的递增管理 5](#_Toc8618)

[3.1.1. 从版本号读SDK和平台型号 5](#_Toc14742)

[3.1.2. 版本管理的递增 6](#_Toc6787)

[3.2. PID & VID 6](#_Toc22490)

[4. 平台调试和升级手段 6](#_Toc12524)

[4.1. 平台调试的常用方法 6](#_Toc7713)

[4.1.1. 如何抓打印 6](#_Toc10802)

[4.1.2. 如何下uboot下烧录固件 6](#_Toc5150)

[4.1.3. 远程调试 7](#_Toc20592)

[4.1.4. 常用命令举例 7](#_Toc32748)

[4.1.5. Amlogic平台客户可定制选项说明 8](#_Toc3091)

[4.2. 升级手段 8](#_Toc32762)

[4.2.1. 存储设备界面升级 8](#_Toc25252)

[4.2.2. 存储设备recovery升级 8](#_Toc10553)

[4.2.3. 配置文件自动升级 8](#_Toc19629)

[4.2.4. OTA网路升级 8](#_Toc29669)

[5. DVB集成 9](#_Toc9203)

[5.1. DVB各模块的集成 9](#_Toc21337)

[5.2. DVB配置参数 9](#_Toc13407)

[6. CoreSS 9](#_Toc31190)

[6.1. SVN源码路径，VSS发布路径 9](#_Toc3400)

[6.1.1. 源代码的SVN路径与分支 9](#_Toc3872)

[6.1.2. 发布路径 9](#_Toc24064)

[6.2. 结构与编译 9](#_Toc6490)

[6.2.1. 结构简要说明 10](#_Toc16334)

[6.2.2. 取代码和编译 10](#_Toc4502)

[6.3. APK开发与打包 10](#_Toc8386)

[7. EMS集成 11](#_Toc10725)

[8. 工厂测试模式 11](#_Toc24059)

[9. 其他配置 11](#_Toc14575)

[9.1. Dolby/DTS 12](#_Toc20828)

[9.1.1. Dolby/DTS简介与集成 12](#_Toc8037)

[9.1.2. Dolby/DTS功能的测试 12](#_Toc30156)

[9.2. Rooted权限 12](#_Toc23998)

[9.3. CPU最小频率 12](#_Toc7784)

[9.4. SN 12](#_Toc1309)

[9.4.1. SN的存储方式 12](#_Toc24658)

[9.4.2. SN通过prop读取配置 12](#_Toc17509)

[9.5. ADB 13](#_Toc3081)

[9.5.1. 系统默认打开/关闭ADB 13](#_Toc6833)

[9.5.2. 客户端如何使用ADB调试 13](#_Toc14957)

[9.6. StreamMediaTools 13](#_Toc17104)

[10. DRM & SecureOS & SecureBoot 13](#_Toc21202)

[11. SDK更新 13](#_Toc18301)

[11.1. SDK更新有哪些工作？ 13](#_Toc23256)

[11.1.1. 公版的patch的选择 13](#_Toc28259)

[11.1.2. SDMC 产品化移植 14](#_Toc9699)

[11.2. SDK更新流程 14](#_Toc22571)

# 引言

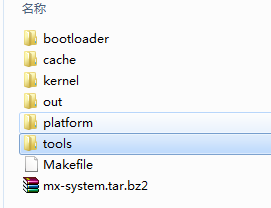
## 编写目的

* 规范版本管理和发布；
* 整理软件集成的相关要点；
* 整理平台调试升级要点；
* 用于新员工培训；
* 内部规范和总结。

# 打包工程管理

## 打包工程结构

### 主目录结构：



说明：如上图所示，是现有打包工程的目录结构。依次是 uboot.bin存放目录, cache临时目录，内核以及root,recovery存放目录，out生成目录，平台和客户私有划分目录，打包的工具目录，打包脚本，framework的压缩包。

MX打包更多详细 请参考《MX打包说明》，M8上的结构和MX基本一致。下载打包工程后，有对应的.doc说明文档。

## 分支版本

打包工程现在的版本，主要有Amlogic8726-MX，Amlogic-M8两个不同的打包工具分支。在服务器的git维护路径如下：

1. Amlogic\_MX:

username@10.10.121.100:/home/svn/amlogic\_jb\_mr1\_mirror/otapackage.git

1. Amlogic\_M8:

## 打包工程的版本管理

打包工程，除打包工具之外的目录，比如platform 目录下的客户配置，uboot，内核，system.tar.bz2这些文件也是需要进行版本管理的。现在版本管理有两种途径，

一：在原有git的工程中，新建git 分支。进行git版本管理。

二：在本地安装SVN的服务端和客户端，对这些文件进行单独的版本管理。

这两种版本各自有优缺点，可以选择使用。Git的版本管理，优点是上传下载速度快。既有本地版本管理，也有远程版本管理记录。缺点是，无界面操作，需要记忆一些命令行，且与当前的打包工程混合了。SVN的优点是，界面可视化操作，独立于打包工具的管理。缺点是，上传下载更新速度慢。

# 软件版本管理

## Normal版与分支版的递增管理

### 从版本号读SDK和平台型号

在现在的打包版本管理中，MX由于平台是分开打包的。OTT和DVB的版本号各自递增维护。在M8中，所有的平台打包都集中在一起。因此对版本号进行了一些意义赋值，如下：

V4.10.1.1 代表V.android\_version + sdk\_version + platform\_model + customer\_pkg\_version

现有SDK版本对应关系如下：

|  |  |
| --- | --- |
| SDK 版本号 | 软件包版本号代号 |
| 130823 | 8 |
| 141104 | 9 |
| 150313 | 10 |
| 150414 | 11 |

现有平台对应的对应关系如下：

|  |  |
| --- | --- |
| 平台型号 | 软件包版本代号 |
| DV7905,DV6063 | 1 |
| DV7904-T2，  DV7908-TH,  DV7904-C,  DV7904-ATSC | 3 |
| DV7800,  DV7819 | 2 |
| DV7810-T2，  DV7810-TH | 5 |

### 版本管理的递增

以normal版本的迭代为基准，相同的framework，kernel的不同客户，平台。在版本号的最后一位保持一致。如果期间，由于某个客户的修改版本一直持续。其余的平台客户在下次更新时，可以跨版本号，跟上这个客户。这样可以基本了解不同客户的软件版本，基本是在软件开发的什么时期。便于版本跟踪和维护。

## PID & VID

产品的PID，VID 编号由产品部门根据一定规则制定。打包时，VID,PID的配置在客户私有目录的platform.cfg 中。例如下，是DV6800的某个客户平台型号配置。如果是normal客户，PID,VID 均为0x0000。

MODEL=dv6800

PID=0x6810

VID=0xd201

CPU=AML8726MX

DDRSIZE=4P256M

SPISIZE=4M

NANDSIZE=4096M

EMMCSIZE=NULL

CTRLFLAG=0x1

在生产前，或者新建一个明确的客户时，需要从产品部门了解到该客户的PID,VID信息。

# 平台调试和升级手段

## 平台调试的常用方法

### 如何抓打印

可以通过串口线，或者ADB查看平台的打印信息。如何使用我们的串口？

请参考文档《adb调试方法》，《如何使用串口操作文件》。

### 如何下uboot下烧录固件

MX：

#

# Ronny create it for sdmc user Jul 02, 2013

#

# use sdcard burn image in boot cmd

# first link your net line

# second config mbox ip and server ip and ensure that both can ping your computer gateway

# config server ip (your computer ip)

setenv serverip 10.10.x.x

# config mbox ip , pls ensure the ip not used

setenv ipaddr 10.10.x.x;

# burn bootloader , put u-boot.bin into tftp dir

mmcinfo;fatload mmc 0 $loadaddr u-boot.bin;sf probe 0;sf erase 0 60000;sf write ${loadaddr} 0 60000

# burn recovery, put recovery.img into tftp dir

mmcinfo;fatload mmc 0 $loadaddr recovery.img;nand erase 0x1800000 0x800000;nand write recovery $loadaddr 0 0x600000

# burn kernel, put boot.img into tftp dir

mmcinfo;fatload mmc 0 $loadaddr boot.img;nand erase 0x2000000 0x800000;nand write boot $loadaddr 0 0x600000

擦除 SPI

sf probe 0;sf erase 80000 8000

### 远程调试

如果盒子在客户手中，无法通过串口或者ADB调试。可以通过远程调试来分析问题。在没有特殊要求下，软件都会自带EMS管理系统客户端。通过获取客户机器的MAC地址，在PC上安装调试的客户端，就可以远程客户的机器了。具体操作以及调试方法见文档“”；

### 常用命令举例

1. 通过getprop ,setprop进行调试

在验证某些问题时，经常要查看系统属性，或者临时修改系统属性。此时，可以用get prop keyname; 的命令进行查询。 用setprop keyname keyvalue 的命令进行设置。

1. 修改系统默认重现率

系统默认重现率一般与/system/bin/set\_display\_mode.sh脚本有关。具体的平台的修改，要参见这个脚本本身。

1. ifconfig设置网络

通过busybox ifconfig eth0 xx.xx.xx.xx 来设置以太网的网络。

其余调试方法，见《MBOX现场分析指南》。

### Amlogic平台客户可定制选项说明

请参考文档《Amlogic平台软件定制》。

## 升级手段

### 存储设备界面升级

将升级包拷贝至存储设备，进入系统的设置界面，选择系统升级。会弹出升级包的列表选项。选择要升级的包，系统会重启升级。

### 存储设备recovery升级

将升级包拷贝至存储设备。机器先断电，长按住reset按键。并同时给机器上电。此时，系统会进入到recovery模式。选择从存储设备升级选项，并选择对应的升级包，即会开始升级。

### 配置文件自动升级

配置文件自动升级的方式，一般应用于空片升级。在工厂量产中，一般会包含以下文件：

1. u-boot.bin （用于烧录SPI）
2. factory\_update\_param.aml （升级引导配置文件）
3. recovery.img （升级引导recovery文件）
4. mx-mr1-g18ref-xx-xx-xx.zip （升级包）
5. factorytest-xx-xx.zip （用于工厂测试的安装包）
6. FactoryTester\_signed.apk （用于工厂测试的安装APK）

不需要进行工厂测试，和SPI烧录，只需要2,3,4文件，就可以完成自动升级。

### OTA网路升级

在客户手中的机器，也可以进行OTA网络升级。只需要得到机器的MAC，就可以在EMS系统中查找，并上传对应的升级包，就可以进行升级了。

# DVB集成

## DVB各模块的集成

## DVB配置参数

请参考文档《DTV-XML配置说明》

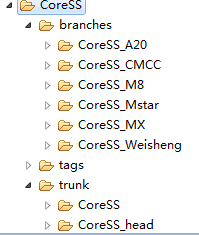
# CoreSS

## SVN源码路径，VSS发布路径

### 源代码的SVN路径与分支

svn://10.10.61.22/android/amlogic\_app/CoreSS/

分支分布如下：



### 发布路径

$Amlogic\_VSS/Test/apk/CoreSS/

## 结构与编译

### 结构简要说明

CoreSS 所有的接口类都封装在CoreSS\_head,所有的平台和分支共用相同的接口类。各个平台各自维护实现类。如上各个分支CoreSS\_A20,CoreSS\_M8等。

### 取代码和编译

#### 取代码

1.从trunk目录取CoreSS\_head作为一个普通工程；

2.从branches目录取一个分支的代码作为一个工程，假设为CoreSS\_x；

3.选中CoreSS\_x，右键菜单选择properties->java build path->source->link-source,添加CoreSS\_head的路径。

PS：CoreSS\_head为公用工程，修改会对所有的分支产生影响。

#### 编译给APP调用的jar包

在以上eclipse工程中，编译jar包。导出时，只选择CoreSS\_head中的接口类。生成的coress.jar即可被应用层调用。

#### 编译打包用到的com.sdmc.coress.jar包

需在android源码下编译。

1.在源码下新建CoreSS文件夹；

2.将Android.mk拷贝至该目录；

3.将CoreSS\_x工程的src文件夹拷贝至该目录；

4.新建head文件夹，并将CoreSS\_head工程的全部代码（只包含代码）拷贝至head目录；

5.如有其它依赖代码，新建ext目录并将其拷贝至ext目录；

5.编译。

## APK开发与打包

#### apk开发

classes目录下的jar包仅提供接口名，无任何实现，仅供编译时查找接口所用，切勿将其编入APK中；

使用时需在AndroidManifest.xml文件的<application>标签内添加 <uses-library android:name="com.sdmc.coress" />；

doc目录下为API标准接口说明文档；

#### 打包配置

将编译生成的framework下的com.sdmc.coress\_\*.jar 放置在打包工程的system/framework下并重命名为com.sdmc.coress.jar；

将etc/permissions下的com.sdmc.coress.xml 放置在 system/etc/permissions下；

# EMS集成

在客户没有明确提出要删除EMS的情况下，我们所有的软件都会包含这个模块。打包时，集成如下说明：在VSS下的路径是，Amlogic\_VSS/Test/apk/tools/设备管理/

**设备管理系统打包说明 V2.0:**

app目录文件放入 /system/app/下, M8平台存放在 /system/priv-app/目录下;

!!!请移除之前系统中包含的RemoteDebug应用和UpdateServer应用！！！

etc目录文件放入 /system/etc/下;

xbin目录文件放入 /system/xbin/下,请确保打包环境中无其他su;

device\_init.sh 文件中添加：

busybox chmod 4755 /system/xbin/su

busybox chmod 744 /system/etc/ems.xml

busybox chmod 644 /system/etc/ems\_private

**设备管理系统打包说明 V1.0:**

app目录文件放入 /system/app下;

lib目录文件放入 /system/lib下;

etc目录文件放入 /system/etc下;

xbin目录文件放入 /system/xbin下，请确保打包环境中无其他su;

build\_device.prop 文件中添加：

sdmc.netupdate.isSupport=1；

device\_init.sh 文件中添加：

busybox chmod 4755 /system/xbin/su

busybox chmod 744 /system/etc/UpdateServer.xml

busybox chmod 744 /system/etc/RemoteDebug.xml

# 工厂测试模式

请参考文档 《工厂测试分离版本说明v1.2》和文档《工厂生产作业指导书》

# 其他配置

## Dolby/DTS

### Dolby/DTS简介与集成

现有Amlogic 平台中，只有-B的芯片支持正版的AC3。其余芯片是不支持正版的，破解版的是不对客户开放的。除非客户提出要求，必须支持破解版的AC3和DTS。且破解版的不良后果，客户承诺承担，我们才开放AC3/DTS的支持。

### Dolby/DTS功能的测试

测试点：

1. 在PCM模式下，HDMI,SPDIF 直接输出，AC3/DTS视频文件有音频输出；

在SPDIF/HDMI PASSTHROUGH 模式下，在其他设备，如支持AC3功放设备上， SPDIF/HDMI连接，AC3/DTS视频文件有音频输出。

## Rooted权限

一些上层应用为了执行一些root权限的操作。进程需要获得root权限。在平台中需要打包预置 su的问题，以及修改相关的权限。su在源码中的路径是：

在MX 中修改这个路径下自带的su就可以使用了。具体的修改方法见下面链接

<http://blog.csdn.net/sdvch/article/details/17055167>。

在M8上，目前我们按此办法没有修改成功，直接从网上下载了一个可使用的。如果后续有开发人员在我们的源码基础上完成了破解版的su。请更新此文。

## CPU最小频率

在机器实际运行过程中，盒子的CPU会间隔使用最小频率。如果此时，正好遇到一些高清视频播放等操作，就会产生一些异常。比如视频播放卡顿。此时需要修改最小频率，在device\_init.sh的启动脚本中增加

echo 408000 > /sys/devices/system/cpu/cpu0/cpufreq/scaling\_min\_freq

这个最小频率也不能设置得过大，这样机器一直都运行在高频，可能造成死机等异常。

## SN

### SN的存储方式

SN即盒子上烧录的一串标识符。存储在加密芯片2016里面。在盒子出厂时，从规定好的条形码读入并烧录。

### SN通过prop读取配置

为了适配其他APP对于SN的读取，需要将从2016读取的接口，转为从标准的property（Build.SERIAL）来读取。要适配这个接口，系统需要修改：

1. 去掉原有的”ro.serialno”在源码/system/core/init/init.c中的初始化。因为这是个只读的property,如果已经在init里面初始化，后续将不能重新赋值。
2. 增加在系统初始化时，运行可执行程序，从2016中读取数据，再写进property中。

## ADB

### 系统默认打开/关闭ADB

ADB的打开，关闭与功能实现相关的可执行程序是/system/bin/adbd。在MX上可以通过命令打开关闭adb.在amlogic四核上，原生不支持此操作。需要修改源码/system/core/adbd。

如果默认开启ADB，则需要在系统初始化的脚本device\_init.sh中添加”stop adbd ;start adbd”。默认关闭ADB，不用做任何修改。

### 客户端如何使用ADB调试

在电脑上安装android 的SDK，其中包含ADB的执行程序。至于ADB的一些操作方法和命令，可以参考以下链接：<http://jingyan.baidu.com/article/60ccbceb634e2364cab197c6.html>

## StreamMediaTools

请参考文档《StreamingMediaPlayer使用说明书》

# DRM & SecureOS & SecureBoot

请参考文档《DRM集成与测试大纲》

# SDK更新

## SDK更新有哪些工作？

### 公版的patch的选择

由于现在SDK的更新，从四核开始维护。之前MX的版本，Amlogic已经停止SDK的发布了。在四核上，amlogic的sdk包含3部分：

1. **基本的通用SDK（适合amlogic所有的客户）；**
2. **强制需要更新的patch（amlogic对一些问题的修改，需要客户强制更新）**
3. **建议更新的patch (针对个别平台，客户提出的问题的修改，没有经过大量测试的patch).**

以上公版的SDK包含的内容，我们必须下载更新的是1,2。第3部分的patch需要工程师和项目经理初步评估。以及在后续客户维护的过程中，关注我们的下游客户是否也遇到了同样的问题，届时可以选择更新。

### SDMC 产品化移植

我们的软件在公版上进行了一些私有化封装和产品化的工作。大概包含以下几个模块：

1. uboot ,内核，recovery，sdmc-libs的移植

包括2016的检测，升级接口的修改，WIFI模组的适配，工厂烧录库的移植，SDMC脚本（build\_device.prop, device\_init.sh）的添加，ETH MAC地址的初始化，DVB相关库的移植。

1. Framework层的移植

升级接口的移植，遥控器按键的添加。

1. DRM的移植

Playready,vmx的移植。

1. 上层APP的适配

CoreSS(主要移植对象，移植OK后，其余APP理论上是兼容的，如果不兼容需要增加新的接口适配), MetroUI, ems,hometv,filebrowser,factorytest （进行打包验证）。

1. 上个SDK版本重要解决重要Bug的patch

那些在上个版本中存在且解决的问题，在新的SDK中仍然存在。

1. 打包配置

一般新移植SDK，会重新建立一个打包工程。最初以一个平台的normal版本为基准。在稳定后，移植之前SDK版本还在持续维护的客户。新的SDK打包所需要的uboot，recovery,kernel ,system.tar.bz2等和之前的全不兼容。且随着新版本SDK的稳定，旧的版本会在迭代过程中被舍弃。在维护的过程中，最多只有2个版本的SDK的软件在同时维护，且在软件移植周期结束，最好所有的客户全部移植到最新版本的SDK。除非有特殊情况（如生产紧急，客户旧版本更改太多，移植多来成本泰国，且旧版本稳定）。

## SDK更新流程

基本流程：

1. 从Amlogic开放的代码路径下，下载Amlogic对应的SDK的代码。
2. 工程师从本地服务器更新代码。项目内所有工程师
3. 合并公版的patch,提交公版测试，如果有太多致命问题，将不会采用此版本的SDK；
4. 合并SDMC的patch,提交normal版本的测试。根据测试情况，进行版本迭代至稳定版本。
5. 根据稳定版本，移植对应的正在维护的客户型号并测试。发布对应的软件。