

# TPS使用流程整理

## 概述

**TOS** 反编译适配系统（后面简称 **TPS**）是一个可以将基于源码（**Google Nexus 5**，后面简称**N5**）的 **TOS** 适配到其它机型的工具。基本思路将 **N5** 的 **TOS** 反编译代码与 **AOSP**（**GoogleAndroid** 源码）的反编译代码对比然后将改动点应用到目标机型的反编译代码上，从而完成一个反编译适配过程。

## 命令行介绍

- **make create** : 创建目标机型适配工作目录
- **make config** : 对目标机型进行公共的配置
- **make prepare** : 对目标机型进行反编译得到待处理的 smali 文件
- **make patch** : 对反编译出来的 smali 文件进行 patch（拆分 smali 文件到方法级）
- **make javapatch** : 对部分逻辑复杂难以 patch 的 smali 文件进行基于 java 源码的 patch
- **make repack** : 对拆分到方法级的 smali 文件进行合并并重新打包成 jar 包
- **make package** : 生成最终的 OTA 刷机包
- **make incpatch** : 将新版本 **TOS** patch 到已经适配好的目标机型的 **TOS** 上
- **make incpackage** : 针对新版本 **TOS** 重新将上一个 patch 好的版本重新打包
- **make clean** : 执行清理操作

## 环境要求

**Linux** : （演示环境 **ubuntu 14.04**）  
获取到最新的**TPS**工具 :  
**checkout**  
[http://tcsvn.tencent.com/mqq/mqq\\_wsrplat\\_rep/NANJI\\_ROM\\_proj/trunk/tools/TPS](http://tcsvn.tencent.com/mqq/mqq_wsrplat_rep/NANJI_ROM_proj/trunk/tools/TPS)

# 实战

- 导入环境变量

```
~/TESTROM/TPS $ source build/envsetup.sh
```

- 创建TOS机型目录

```
12:20 ~/TESTROM/TPS $ make create device=tos
device 'tos' directory tree created
now you should copy or link the ota package as 'ota.zip' to '/home/wanwang/TESTROM/TPS/devices/tos' directory
you should also link the source code directory as 'source' to the '/home/wanwang/TESTROM/TPS/devices/tos' directory
```

因为缺少ota.zip包 所以报错，解决方法是去下载个最新N5的包改名为ota.zip放在tos机型目录即可

- 配置TOS环境

```
12:21 ~/TESTROM/TPS $ make config device=tos
now you can cd to '/home/wanwang/TESTROM/TPS/devices/tos' and execute make command
```

进入到目录下，查看目录结构

```
12:22 ~/TESTROM/TPS $ cd devices/tos
12:25 ~/TESTROM/TPS/devices/tos $ ls
boot/  config/  Makefile*  ota/  ota.zip*  override/  package/  patch/  progress/  tools/
```

这个就是tos的目录，svn的tps目录里面hendy已经建好tos目录。所以我们只需要把ota.zip替换成最新N5的包就ok了

- 执行make prepare 命令

执行完毕make prepare 命令后，TOS这边的准备工作完成了

- 创建N900机型目录

在TPS根目录再次执行make create device=N900（xxx为要适配的机型名称）  
将要适配机型的N900.zip包放到device/N900目录下，重命名或链接为ota.zip，要求有system目录和boot.img文件。

注意：N900的ota包的获取，需要获取机型原底包的odex经过优化的包，而TOS在创建机型目录的时候，会在机型的根目录下生成一个N900.zip，这个是经过odex优化的。这个包里面只有一个system文件夹，这个包需要再放入一个N900原厂的boot.img文件到N900.zip包的根目录。这个就可以用来作为ota包使用。

名称	大小	压缩后大小	类型	安全
..(上层目录)				
system	2.04 GB	1.37 GB	文件夹	
boot.img	6.72 MB	6.69 MB	好压 IMG 压缩文件	

```
12:36 ~/TESTROM/TPS $ cd devices/N900/
12:46 ~/TESTROM/TPS/devices/N900 $ ls
boot/  config/  Makefile*  N900.zip*  ota/  override/  package/  patch/  progress/  tools/
```

需要注意的是： 因为各个厂商的`boot.img`差异导致打包和解包会出错 所以需要手动配置下解包和打包工具的路径（最好配置绝对路径，防止在使用的过程中出现找不到工具或者命令的情况）。 需要对`Makefile`文件进行修改。

```
# packages need to be decompiled
DECOMPILE_PACKAGES := framework.jar services.jar android.policy.jar framework2.jar telephony-common.jar ext.jar

# custom resource package
CUSTOM_RESOURCE_PACKAGE := none

# extra packages need to be decompiled
#EXTRA_DECOMPILE_PACKAGES := TOS:qgallery.apk TARGET:Camera2.apk

# unpack boot.img tool
# support relative path & variables
# e.g. $(PORT_TOOLS)/unpackbootimg.sh
UNPACK_BOOTIMG_TOOL := default

# pack boot.img too
# support relative path & variables
# e.g. $(PORT_TOOLS)/mkbootimg.sh
PACK_BOOTIMG_TOOL := default

# device list that is suitable for. these devices will be added to the ota updater-script assertion statement
SUITABLE_DEVICES :=

# apktool/smali/baksmali selection
# support relative path & variables
# e.g. $(PORT_TOOLS)/apktool/apktool.jar
APKTOOL_JAR := default
SMALI_JAR := default
BAKSMALI_JAR := default

# aapt tool
# support relative path & variables
# e.g. $(PORT_TOOLS)/android/aapt
AAPT_PATH := default

include $(PORT_BUILD)/main.mk
```

- 配置N900的环境

对N900机型目录机型配置，执行`make config`

- 执行`make prepare` 命令

进入`devices/N900`目录，执行`make prepare`  
至此,TOS跟N900两个机型目录的准备工作已经完毕

- 执行`make patch` 命令

这个是命令是整个TPS适配的核心，整个插桩的工作都在这个过程中完成

- 执行`make repack` 命令

这个步骤是将合并好的`method`重新合并成一个完整的`smali`文件，并打包

注意：`make repack` 执行完之后 就需要进入`N900`目录下的`config`文件夹为打包`package`做准备

```
13:36 ~/TESTROM/TPS/devices/N900/config $ ls
build.prop*  file_record/  misc_info.txt*  remove_files.list*
file_contexts*  keep_files.list*  recovery.fstab*  tos_system_files.ignore*
```

进入`devices/N900/config`目录，生成配置文件`misc_info.txt`，`recovery.fstab`，配置`build.prop`、`remove_files.list`

- `misc_info.txt` 的模板如下：

```
recovery_api_version=3
fstab_version=2
use_set_metadata=1
multistage_support=1
update_rename_support=1
fs_type=ext4
blocksize=4096
boot_size=11534336
recovery_size=11534336
system_size=2254438400
userdata_size=12738101248
cache_size=608174080
selinux_fc=META/file_contexts
```

最主要是配置几个分区的大小

- `recovery.fstab` 这个可以从一个可以用的目标机型的`recovery`中提取出来，路径是`/etc/recovery.fstab`
- `build.prop` 将模板中的空白字段填上，比如：

```
ro.qrom.beaconkey=0M000V5PH01B6QQD
ro.qrom.product.device=ja3g
ro.qrom.product.device.brand=samsung
ro.qrom.build.brand=tos
ro.qrom.build.os=android4.4.2
ro.qrom.otapath=/data/media/0

qrom_disposeIcon_enable=1
qrom_permission_enable=1
```

- `remove_files.list` 这个是精简列表，里面的列表均为相对路径，比如

```
###qrom.mk###needtobekeptorremovedaccordingtoqrom.mk
app/PackageInstaller.apk
priv-app/SecContacts_0Sup.apk
priv-app/SecNoteMyFiles.apk
priv-app/SecGallery2013.apk
priv-app/SecLauncher3.apk
priv-app/Keyguard.apk
priv-app/SecMms.apk
priv-app/SecPhone.apk
priv-app/SecSettings.apk
priv-app/SystemUI.apk
app/BrowserProviderProxy.apk
app/EdmSimPinService.apk
app/EdmSysScopeService.apk
app/JobManager.apk
app/MediaUploader.apk
app/PartnerBookmarksProvider.apk
app/RootPA.apk
app/SBrowser.apk
app/SBrowserTrv.apk
```

- `tos_system_files.ignore` 这个是忽略要拷贝的tos文件，目前来说，由于media注入有问题，需要配置以下两项：

```
bin/mediaserver_injector
lib/libmedia_jni.so
```

- 执行make package 命令

配置完成之后 进入N900 机型目录执行make package  
执行make package 命令生成OTA包  
在devices/N900/package/

```
14:01 ~/TESTROM/TPS/devices/N900/package $ ls
N900.zip  smali/  target_files/  target_files.zip
```