A153 版本发布流程

[版本分类： 2](#_Toc492989516)

[准备工作： 2](#_Toc492989517)

[一．拉代码 2](#_Toc492989518)

[二．代码编译 3](#_Toc492989519)

[三．文件检查 4](#_Toc492989520)

[四．修改文件名称 5](#_Toc492989521)

[五．自检 5](#_Toc492989522)

[六．版本放入 9](#_Toc492989523)

[七．发送邮件 9](#_Toc492989524)

[八．打TAG 10](#_Toc492989525)

[九．量产版本拉分支 10](#_Toc492989526)

[十、版本归档 11](#_Toc492989527)

[十一、手动简易编译 12](#_Toc492989528)

以4049D\_MEA\_BASE\_V1.3\_20170907为例:

# 版本分类：

正式版本

临时版本

临时版本按正式版本编译

# 准备工作：

版本号：

确认版本号

提交版本号（注意提交的分支）

空间：

确认空间大于150G

确认编译时间：

和开发经理确认必须合入问题

SCM Check是否已经merge

服务器选择：

确保没有同时编译

# 一．拉代码

拉新代码，不能在之前编译过的代码上编译。

1. 本地创建文件夹（内部版本号命名）

获取最新的auto脚本

Check auto脚本内容：代码路径、项目信息、版本类型（user、eng）、GMS包路径

1. Auto脚本，选择每个定制对应的Auto脚本，如：4049D\_MEA\_BASE\_V1.3\_20170907对应A153\_DMEA定制。由于A153脚本不完善需：因A153版本Auto脚本暂不完善，因此需到目录下拷贝Auto脚本到根目录并进行修成对应的版本: 如ALWE版本
2. 利用Auto脚本拉代码会自动下拉GMS和代码

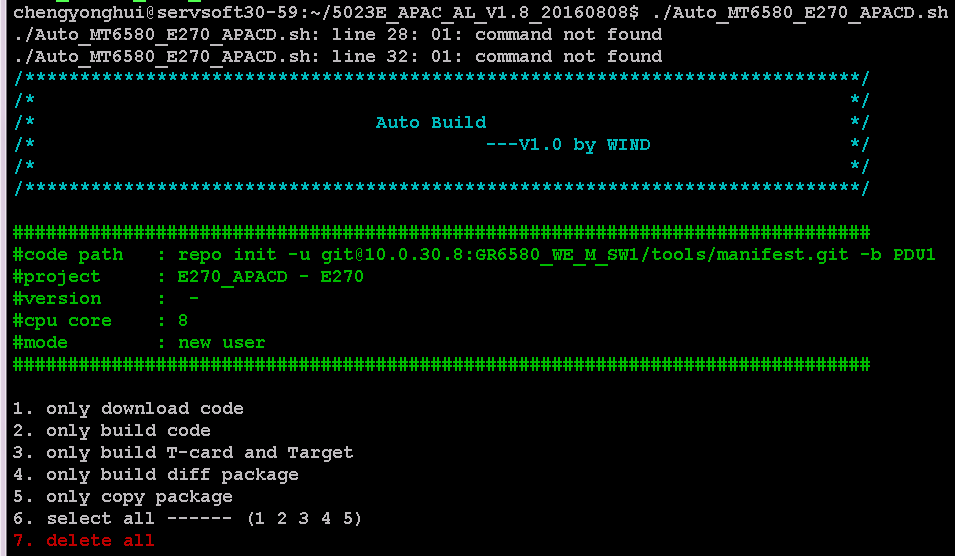


图1.1 Auto脚本拉代码

Check GMS包是否拉取成功,代码同级目录中：

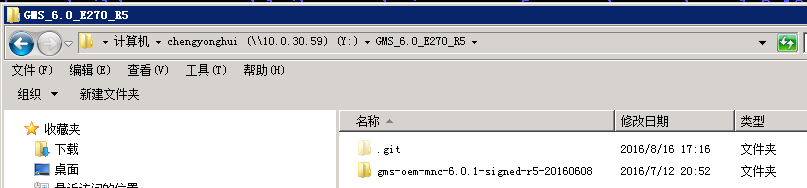


图1.2 GMS包

# 二．代码编译

1. 底包的copy。将服务器中的上个版本的target包拷贝至代码根目录，分别命名为update\_a.zip。（PS：B99 暂不需要）
2. 修改版本号。进入wind/custom\_files/device/ginreen/A153/下找到对应的版本信息文档，修改版本号、日期、TCL的版本号等。如versionDMEA

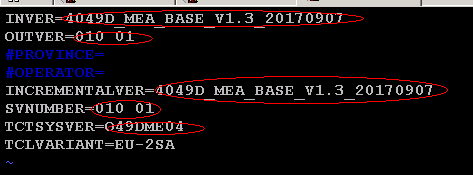


图2.1 版本信息修改

1. 提交后用Auto脚本执行2 3 4 5几步操作，先后编译DL包，T卡包和差分包，然后打包copy到版本通道。

表2.1 Auto脚本功能解释

|  |  |
| --- | --- |
| 参数 | 解释 |
| 1. only download code | 拉代码、GMS包 |
| 2. only build code | 编译DL包 |
| 3. only build T-card and Target | 编译T卡包以及Target包 |
| 4. only build diff package | 编译差分包 |
| 5. only copy package | 释放所有文件 |

# 三．文件检查

1. 版本通道文件release时间间隔不超过20分钟，超过20分钟则重新release。注意：编译完DL包时会有一次释放动作，全部编完会再次释放。两个释放只有后一次释放的才是正确的版本文件。当版本通道出错时，第二次释放没有释放完全，拷贝的文件可能是第一次的错误文件，需要比对释放时间，确认文件正确。
2. 将版本通道里的文件copy到本地，利用md5sum.bat和md5sum.exe这两个文件与编译出的checklist.md5检测文件是否齐全及文件大小是否匹配。（文件夹名为英文）
3. 文件夹准备。分别以内部版本号和内部版本号加后缀名\_SD。



图3.14049D\_MEA\_BASE\_V1.3\_20170907中文件夹

生成的DL包放入4049D\_MEA\_BASE\_V1.3\_20170907中（checklist.md5无需放入其中）。生成的T卡包改名为update.zip放入4049D\_MEA\_BASE\_V1.3\_20170907\_SD中。外加的文档以及其他生成文件放在此文件夹同级目录下即可。建议备份DL包中AP\_Database和Modem\_Database\_wg放在同级目录下，方便后面写号。

# 四．修改文件名称

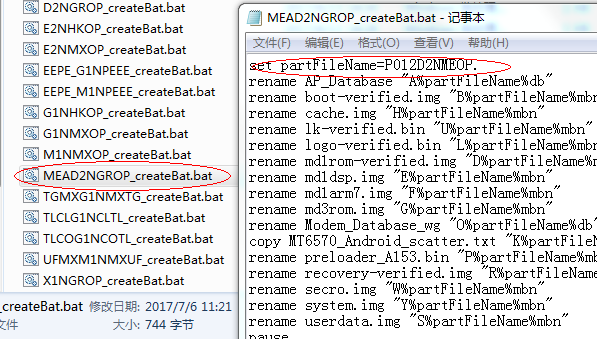
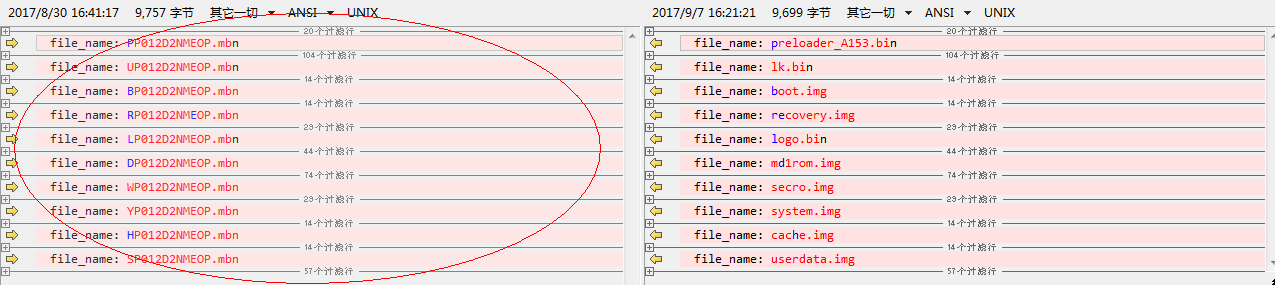
1. 进入4049D\_MEA\_BASE\_V1.3\_20170907文件夹，将对应的改名文件拷贝到此文件夹中，点击编辑此工具，保存后点击运行此文件。

图4.1 修改名称工具修改界面

如当前版本号为1.3就修改为13.

改名完成后到1.2版本取ＤＬ包中取MT6570\_Android\_scatter.txt文件使用对比工具与当前1.3版本的MT6570\_Android\_scatter.txt文件对比并将1.2版本的不同之处合到1.3，

途中红圈处为1.2版本，合完后记得要将12改成13毕竟1.3版本了。最后才可以做CheckSum\_Generate\_exe。都完成后就可以刷机测试了。

图4.2 对比工具对比界面

1. 修改fota包名称。4049D\_MEA\_BASE\_V1.3\_20170907文件夹同级目录下将updateA2B.zip修改为ODM\_BUZZ6\_3G\_O49DEA03\_to\_O49DEA04\_USER\_incrementOTA.zip，每次版本的不同，注意修改标黄色部分。

# 五．自检

1. 刷机。将版本烧录到手机（如果是B01版本则全擦下载，非B01版本需先全擦上一版本写入IMEI、BT WIFI后再半擦当前版本）。
2. 写IMEI号、写地址.打开写号工具SN\_Write.此工具的详细用法可参照

\\10.0.10.2\sw01临时软件版本\A153\tools\ALCA183\_WSN\_FINAL

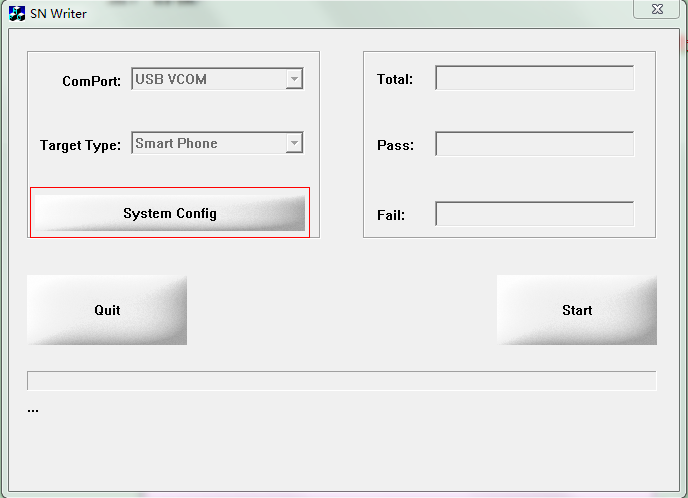


图5.1 写号工具首页

点击红色框框System Config.

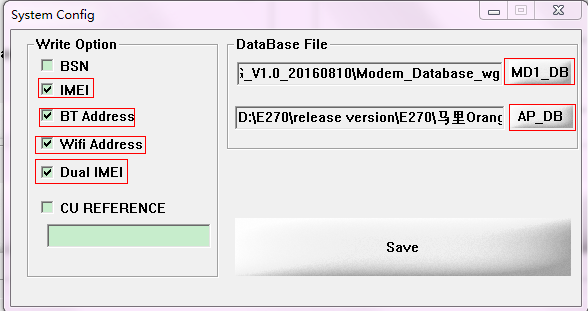


图5.2 写号配置

其中的MD1\_DB和AP\_DB分别选中之前备份的Modem\_Database\_wg和AP\_Database.点击Save。然后点击Start。

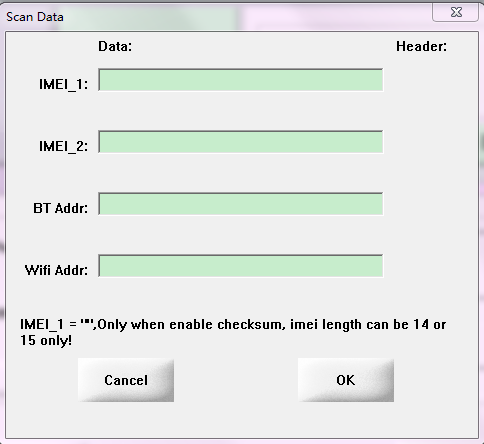


图5.3 填写imei、addr

注意：IMEI\_1和IMEI\_2可以相同。

点击OK按钮之后，插入手机。（手机为关机状态）出现绿色pass即写号成功。

1. 解锁码的生成（若版本无锁网需求，无需看此步）。打开工具SML\_Gen V1.0.

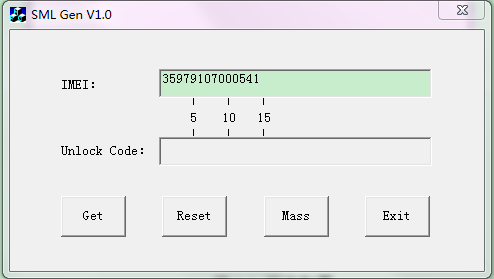


图5.4 生成解锁码

开机之后点击Dismiss，查看手机中的IMEI号（\*#06#）填入上面IMEI栏，点击Get即可得到Unlock Code，手机中输入解锁码即可正常使用。

1. 验证版本信息是否正确，运行工程模式以及其他一些基本验证。

开机之后需要验证版本号是否和需求一致，IME地址是否写入，蓝牙、wifi地址是否正确入。

查看版本号指令为\*#458#

查看IMEI号指令为\*#06#

查看蓝牙和WIFI可在设置中查看。

工程指令验证：输入\*#2886#进入工程模式。工程模式Apk为TCL自带的APK。进入工程模式之后，点击Auto，每检测一项点击Pass, 直至多有项目检测完成。

以上验证的详细步骤请参照文档：

1. 验证各个语言下的SAR值是否显示正常。各种语言下，进入手机中Settings->Regulatory&safety->SAR图标，查看对应的SAR值是否正常显示。
2. 验证T卡升级。在手机中刷上一个版本，如：4049D\_MEA\_BASE\_V1.3\_20170907，将4049D\_MEA\_BASE\_V1.3\_20170907\_SD中update.zip拷贝至手机SD卡的根目录。依次点击设置->About Phone->systemUPdate->System updates->System updates->confirm。升级完成之后，手机会自动重启，检查一下版本号为当前版本号5023E\_APAC\_AL\_V1.8\_20160808即为正确。否则T卡升级失败。
3. 验证Fota升级。

TCL的FOTA升级分本地验证和发邮件让客户在服务器上配置

a.本地验证FOTA

上个版本到当前版本的验证。在手机中刷上一个版本，如：4049D\_MEA\_BASE\_V1.3\_20170907，将ODM\_BUZZ6\_3G\_O49DEA03\_to\_O49DEA04\_USER\_incrementOTA.zip拷贝到手机SD卡根目录，并将此改名为update.zip验证方法同T卡升级。

注意：每次升级完之后要验证版本号是否正确。

b. 发邮件给客户配置服务器FOTA（只需要配置上个版本到当前版本）

本地验证没有问题之后，发送邮件给客户，内容如下：

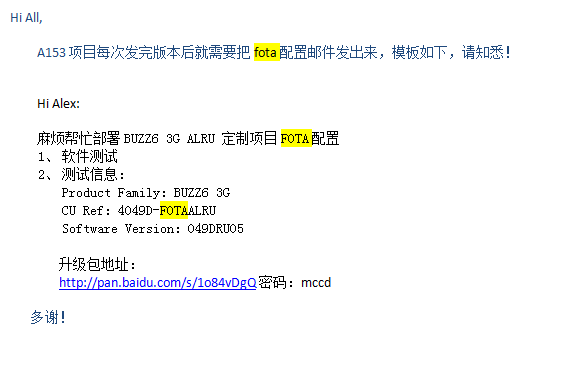


图5.5 邮件内容

将4049D\_MEA\_BASE\_V1.3\_20170907上传到百度网盘上进行分享，创建公开链接。

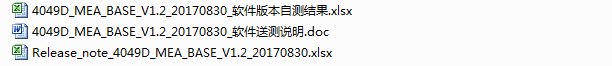
第二条CU Reference值后面两位值为对应的定制的缩写，需要和开发经理确认清楚。

邮件中的收件人抄送人不知道的可以查看自己邮箱以往类似邮件。

# 六．版本放入



版本放入需要先准备以下三个文件



找到版本通道对应的项目目录，以内部版本号新建一个文件夹4049D\_MEA\_BASE\_V1.3\_20170907，将上面准备的三个文件放入

将4049D\_MEA\_BASE\_V1.3\_20170907和4049D\_MEA\_BASE\_V1.3\_20170907\_SD文件夹压缩成4049D\_MEA\_BASE\_V1.3\_20170907.zip和4049D\_MEA\_BASE\_V1.3\_20170907.zip.

打包完之后和生成的其他文件放到\\10.0.10.2\sw01临时软件版本\A153\系统版本\4049D\_EU\_双卡\4049D\_MEA\_双卡\4049D\_MEA\_BASE\_V1.3\_20170907文件夹中。

打包完后，全部文件的列表如下：



图6.1 软件版本所有文件列表

# 七．发送邮件

正式版本的发布都是需要发送邮件的，邮件的主要内容即软件送测说明文档。 ( 邮件收件人中需要添加主测，开发经理，项目组邮件群，[抄送sw1@wind-mobi.com,st@wind-mobi.com](mailto:抄送sw1@wind-mobi.com,st@wind-mobi.com))

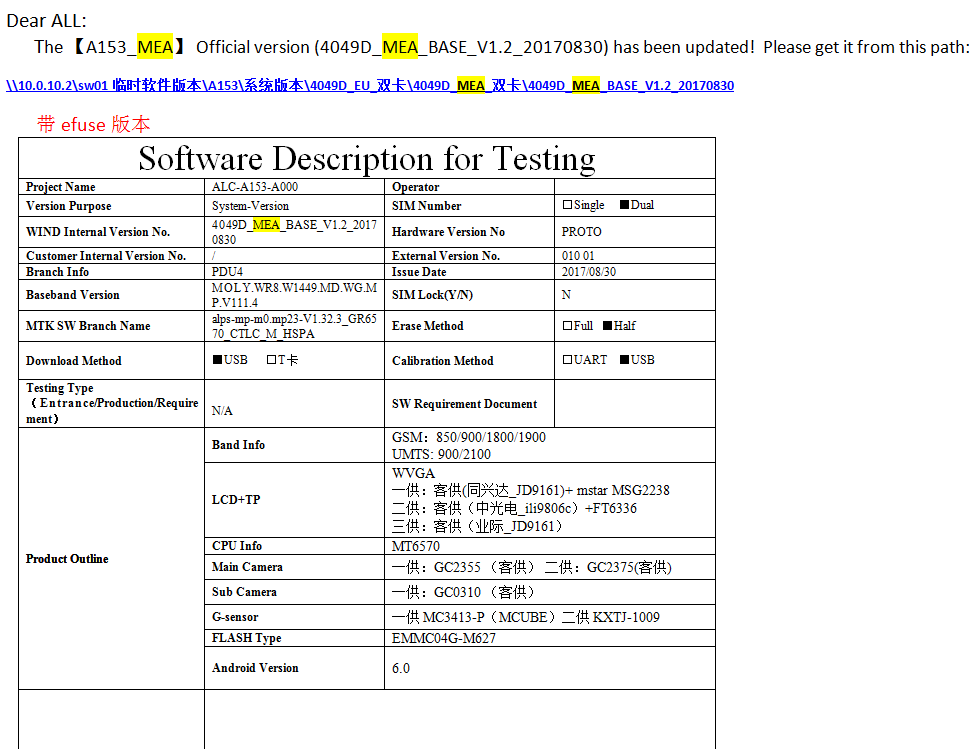


图7.1 邮件发送的主要内容

Redmine 上TASK状态改为已解决，附上软件版本自检表

填写10.2服务器上的软件版本编译记录表.xlsx（\\10.0.10.2\sw01临时软件版本\软件版本编译记录表.xlsx）。

# 八．打TAG

编完正式版本，当测试发回入口测试邮件后需要及时打TAG。打TAG的步骤如下：

1. repo forall -c "git tag -a 4049D\_MEA\_BASE\_V1.3\_20170907 -m '2016-08-08'"
2. repo forall -c "git push STS001 --tags"
3. 验证TAG是否打成功（重新拉一套代码或者别的代码去repo sync去检测）

repo forall -c "/EXCHANGE/public/hebin/scripts/repo\_check\_tag.sh 4049D\_MEA\_BASE\_V1.3\_20170907"

# 九．量产版本拉分支

当一个项目的版本已经量产之后，为了避免其他项目修改引起不必要的问题，通常需要SCM重新建立一个稳定分支。具体步骤如下：

1. 将分支切换到量产版本的tag节点上s

repo forall -c "git co –b PDU1\_Stable\_Mass\_DIS\_RUS\_P635N38V1\_BRH DIS\_RUS\_P635N38V1.0.0B10\_20160818"

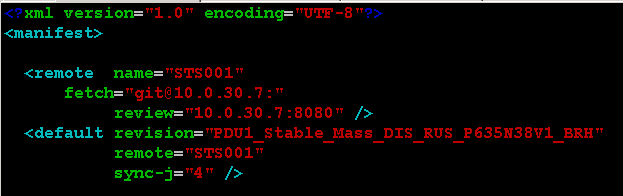
1. Push到服务器上

repoforall -c "git push STS001 PDU1\_Stable\_Mass\_ DIS\_RUS\_P635N38V1\_BRH: PDU1\_Stable\_Mass\_ DIS\_RUS\_P635N38V1\_BRH"

1. 修改default.xml文件并上传

在代码根目录下：cd .repo/manifests🡪 vim default.xml

将<default revision="PDU1"后面的值改为需要新建的分支名，如



1. 修改保存好之后，git add🡪git commit🡪git push origin default:PDU1\_Stable\_Mass\_ DIS\_RUS\_P635N38V1\_BRH
2. 检查新分支代码是否有遗漏

保留本地的代码不动，重新拉一套新建分支代码，用Beyond Compare工具跟本地代码二进制比较，检查是否有遗漏。

新拖代码路径就是将源代码-b后面的分支名换成新建分支名：

repoinit -u git@10.0.30.8:GR6753\_65T\_M0\_SW1/tools/manifest.git -b PDU1\_Stable\_Mass\_ DIS\_RUS\_P635N38V1\_BRH

# 十、版本归档

版本归档需要走OA，在流程🡪新建流程🡪文件归档申请中。



需要填写的内容为：项目名称、文件名称、文件类型、归档说明以及路径

项目名称可以在redmine上看到

文件名称为路径下全部文件

文件类型为软件版本

归档说明写出要归档到的路径以及用途

路径为正式版本现在存放的目录



确认填写无误之后，即可进行提交。

# 十一、手动简易编译

编译正式版本时需要用到的指令

1. 编译DL包和T卡包

./quick\_build.sh A153\_DALWE new user efuse && ./quick\_build.sh A153\_DALWE ota user effuse

*正式编译时替换成当前需要编译的客制化名称，可以在quick\_build脚本中找到*

1. 编译FOTA包（A153暂时不需要B99包）

./build/tools/releasetools/ota\_from\_target\_files -s ./device/mediatek/build/releasetools/mt\_ota\_from\_target\_files.py -n --block -i B04.zip B05.zip update.zip

*B04.zip为上一版本的底包，B05.zip为当前版本的底包，都需要copy到代码根目录。Update.zip则是生成的fota包，最终的fota包release出来的时候还需要重命名，命名规则参考上一版本（跟A155类似）。*

1. 释放文件

./release\_version\_efuse.sh A153

*执行这个脚本的时候只释放DL包的文件，共17个，其他的SD卡包等文件在上一步会自动释放出来。*