|  |
| --- |
| 上海与德通讯技术有限公司 |
| 新建repo仓库流程 |
| 新建代码仓库标准流程 |

|  |
| --- |
|  |

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Issue** | **Description** | **Author** |
| 2016/01/14 | 1.0 | 初始版本 | 张贺廷 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

[1. Abstract 4](#_Toc440547815)

[2. 准备工作 4](#_Toc440547816)

[2.1 取代码到本地 4](#_Toc440547817)

[2.2 必要的脚本和工具 5](#_Toc440547818)

[3. 建立repo仓库具体步骤 5](#_Toc440547819)

[3.1 新代码命名 5](#_Toc440547820)

[3.1.1名称需包含代码平台信息 5](#_Toc440547821)

[3.1.2加上部门名称后缀（如SW1） 6](#_Toc440547822)

[3.1.3 避免名称重复 6](#_Toc440547823)

[3.2删除代码中的.gitignore文件 6](#_Toc440547824)

[3.3生成代码路径 6](#_Toc440547825)

[3.3.1 修改default.xml 6](#_Toc440547826)

[3.3.2修改initRepos\_git30.8.py的路径 7](#_Toc440547827)

[3.3.3执行脚本initRepos\_git30.8.py生成路径 7](#_Toc440547828)

[3.4修改提交gitolite-admin里的gitolite.conf 7](#_Toc440547829)

[3.5执行脚本initRepos\_git30.8.py 上传代码 9](#_Toc440547830)

[3.6上传default.xml文件 9](#_Toc440547831)

[3.7 检查上传代码完整性 9](#_Toc440547832)

[3.8 上传modem端代码 9](#_Toc440547833)

[4 总结 9](#_Toc440547834)

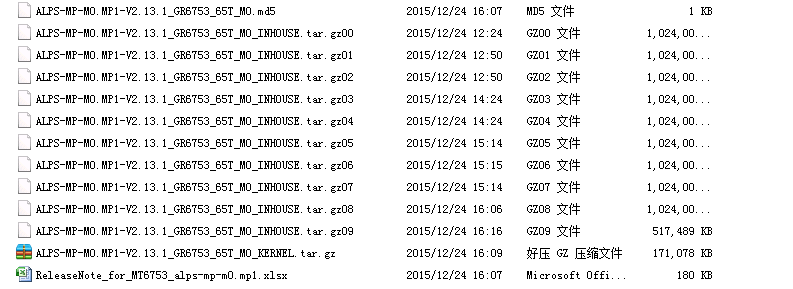
# Abstract

此文档介绍在10.0.30.8服务器上新建代码repo仓库的具体步骤以及注意事项。

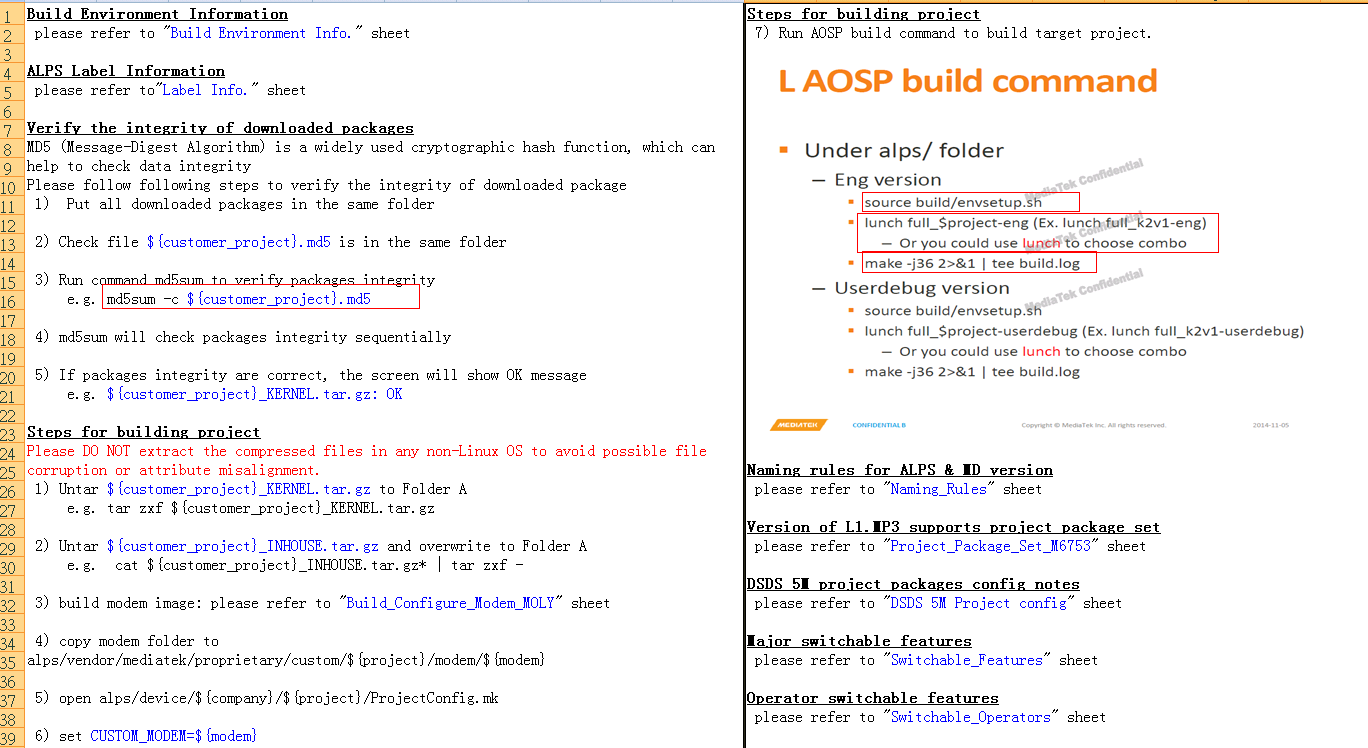
# 准备工作

# 2.1 取代码到本地

拿到MTK代码后，一般的代码文件如下图所示（以MT6753 M0平台为例）：



拷贝到本地后，先阅读ReleaseNotes，这个文档对拿到代码之后做的每一步都介绍得很清楚。根据文档，校验拿到代码的完整性。并解压代码。



解压完AP端代码为一个alps的文件夹，拷贝一份出来，编译一下，确保能通过后，就可以对源代码进行上库操作了。

# 2.2 必要的脚本和工具

本地安装了python

脚本：initRepos（可以从/Exchange/public/zhanghetin拷贝过来）

gitolite-admin : (需要开通gitolite-admin的读写权限，根据服务器git clone 一个)

git clone git@10.0.30.8:gitolite-admin.git

# 建立repo仓库具体步骤

## 3.1 新代码命名

### 3.1.1名称需包含代码平台信息

一般根据MTK代码名称，如MT6753 M0版本的代码为GR6753\_65T\_M0,这个名字包含了平台信息：6753平台, M0代表Android 6.0版本。所以新代码的命名需包含 6753 L1 这些关键信息。

### 3.1.2加上部门名称后缀（如SW1）

由于公司不断壮大，同平台同版本的代码会存在多个部门共用，为方便区分，需要加上自己部门缩写的后缀。

综合以上两点，软件一部MT6753 M0这套代码最初命名可定为：GR6753\_65T\_M0\_SW1。

### 3.1.3 避免名称重复

名字大略定好后，可以git pull 一下gitolite-admin，去gitolite.conf搜一下，是否这个名称已存在了，如果存在，就得另外加关键字去区分。这点要特别注意，否则上库时新的代码会覆盖同名的代码，后果严重!!!

# 3.2删除代码中的.gitignore文件

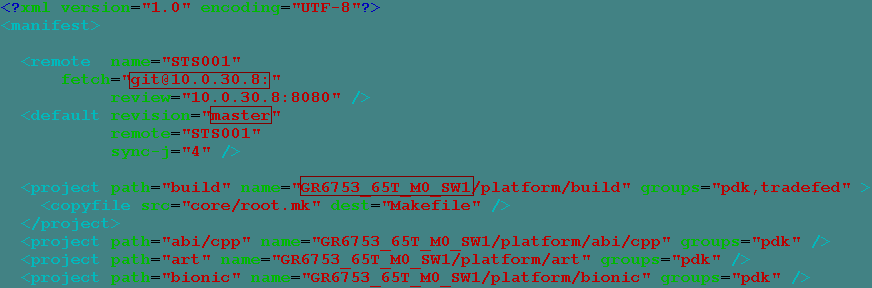
MTK给过来的代码中包含了很多.gitignore文件，如若上库前不删除，会导致很多代码被忽略，无法成功上传，因此代码上库之前要将所有的.gitignore删掉。



# 3.3生成代码路径

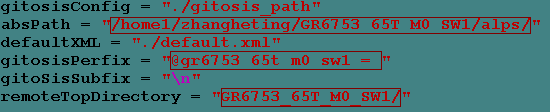
### 3.3.1 修改default.xml

找一个已有代码（最好代码路径相似）default.xml到initRepos文件夹。。打开default.xml更改路径：将name名后面的代码名称全部替换为新代码的名称。仔细核对default.xml所列路径，确保跟所传代码一一对应，不要有多加和遗漏。如MT6753 M0代码替换完如下图(30.8服务器，master分支，代码库名称GR6753\_65T\_M0\_SW1)：



### 3.3.2修改initRepos\_git30.8.py的路径

修改absPath（本地代码路径）、gitosisPerfix（远程代码库git引用参数）和remoteTopDirectory（远程代码库名称），如MT6753 M0代码修改之后如下图：



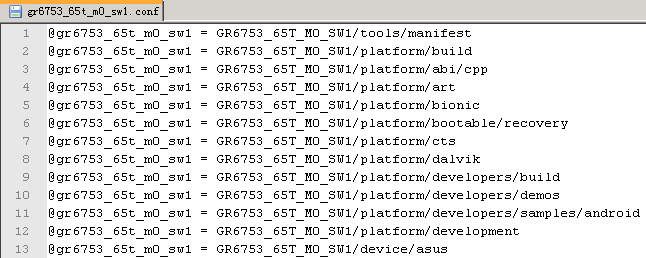
### 3.3.3执行脚本initRepos\_git30.8.py生成路径

python initRepos\_git30.8.py

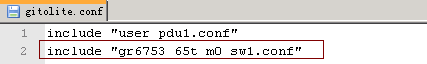
执行完成后会在gitosis\_path文件中生成新代码的路径。

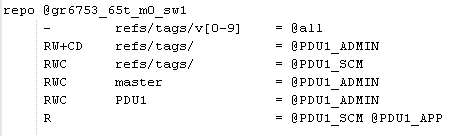
# 3.4修改提交gitolite-admin里的gitolite.conf

在gitolite-admin/conf中新建\*.conf文件，如MT6753 M0新建gr6753\_65t\_m0\_sw1.conf，将上一步生成的路径复制到conf文件夹中的gr6753\_65t\_m0\_sw1.conf里，注意不要有多余的空格,如MT6753 M0部分代码截图如下：



在gitolite-admin/conf中的gitolite.conf调用gr6753\_65t\_m0\_sw1.conf并配置相应权限，截图如下：





注：PDU1\_ADMIN具有最高权限，可以push代码，repo upload代码，打TAG,删除TAG

PDU1\_SCM 只可以repo upload代码，打TAG

PDU1\_SCM 只可以repo upload代码

修改好以后提交修改：

git add .

git commit –m ‘注释’

git push

# 3.5执行脚本initRepos\_git30.8.py 上传代码

进入intRepos目录，执行如下指令，上传代码：

python initRepos\_git30.8.py 1

等待执行完成，代码就已经上传至服务器

# 3.6上传default.xml文件

上面的步骤之后，代码已经上传到服务器，但是代码库中还缺少对应的default.xml。

1. 创建一个文件夹：mkdiradddefault
2. cd adddefault
3. gitinit
4. 拷贝default.xml 到adddefault文件夹
5. git add .
6. git ci -m 'add default'
7. git remote add -f default git@10.0.30.8:GR6753\_65T\_M0\_SW1/tools/manifest （gitolite上项目目录）
8. git push default master:master

# 3.7 检查上传代码完整性

至此，AP端代码已上传完成。为检查代码是否上传完整，一定要重新拉一套代码，并与原始代码做比对，找出漏传的文件，补传文件。存在漏传的目录和文件很正常，因为不同版本的代码，路径结构也不尽相同。

# 3.8 上传modem端代码

上传完AP端代码后，要上传对应的Modem代码，步骤与上述一致，只是路径稍有区别，在此就不再一一赘述。

# 4 总结

上面详细的介绍了新来代码新建repo仓库的具体流程和注意事项。总之，在代码上库过程中一定要仔细做好每一步，确保路径正确，代码完整。