

repo 只是 google 用 **Python** 脚本写的调用 **Git** 的一个脚本，主要是用来下载、管理 **Android** 项目的软件仓库。

1. 下载 repo 的地址: <http://android.git.kernel.org/repo> , 可以用 wget

<http://android.git.kernel.org/repo> 或者 `curl http://android.git.kernel.org/repo >~/bin/repo` 来下载 repo , `chmod a+x ~/bin/repo`

2. 用 repo sync 抓取 android source code 的时候，会经常出现一些错误导致 repo sync 中断，每次都要手动开始。

可以在一个脚本里写如下的命令，来自动取重复：

```
repo sync
while [ $? -ne 0 ]
do
    repo sync
done
```

repo sync [project] 下载最新本地工作文件，同步 repository 中的代码。可以指定需要更新的 project，如果不指定任何参数，会同步整个所有的项目

如果是第一次运行 **repo sync**，则这个命令相当于 **git clone**，会把 repository 中的所有内容都拷贝到本地。

如果不是第一次运行 **repo sync**，则相当于 更新 和 自动合并 动作；

repo sync 会更新 **.repo** 下面的文件。如果在 merge 的过程中出现冲突，这需要手动运行 **git rebase --continue**

3. **repo help [command]**，显示 command 的详细的帮助信息内容

如：可以用命令: **repo help init** 来获取 repo init 的其他用法

4. **repo init -u URL**，在当前目录安装 repository，会在当前目录创建一个目录 **".repo"**，**-u** 参数指定一个 URL，从这个 URL 中取得 repository 的 manifest 文件。

repo init -u git://android.git.kernel.org/platform/manifest.git，可以用 **-m** 参数来选择 repository 中的某一个特定的 manifest 文件，如果不具体指定，那么表示为默认的 manifest 文件 (default.xml)

repo init -u git://android.git.kernel.org/platform/manifest.git -m dalvik-plus.xml，可以用 **-b** 参数来指定某个 manifest 分支。

repo init -u git://android.git.kernel.org/platform/manifest.git -b release-1.0

5. **repo upload[project-list]** 上传修改的代码，如果你本地的代码有所修改，那么在运行 **repo sync** 的时候，会提示你上传修改的代码，所有修改的代码分支会上传到 Gerrit (基于 web 的代码 review 系统)，Gerrit 受到上传的代码，会转换为一个个变更，从而可以让人们来 review 修改的代码。

6. **repo diff [project-list]** 显示提交的代码和当前工作目录代码之间的差异。

7. **repo forall -c command**，对所有项目执行一个 command 命令，这个命令 相当 好用；

8. **repo start newbranchname** 创建新的 branch 分支。"." 代表当前工作的 branch 分支。

9. repo prune [project list] 删除已经 merge 的 project

10. repo status 显示 project 的状态

repo 遇到的问题：

1.repo status 遇到 ascii 和 uninode 冲突，在相应代码中添加

```
import sys
reload(sys)
sys.setdefaultencoding('utf8')
```