repo只是google用Python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:

<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.repo init -u URL:

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***

## 3. 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。

用repo sync抓取android source code 的时候，经常会出现一些错误导致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，来自动去重复：

***repo sync***

***while [ $? -ne 0 ]***

***do***

***repo sync***

***done***

## 4. 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收到上传的代码，会转换为一个个变更，从而可以让人们来review 修改的代码。

## 5. 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。

## 6. repo forall -c command:

对所有项目执行一个command命令，这个命令相当好用。

## 7. repo start new\_branch\_name:

创建新的branch分支。 "." 代表当前工作的branch 分支。

**8. repo prune [project list]:**

删除已经merge的project 。

## 9. repo status:

显示project的状态。

## 10.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。