

# Git-SVN说明

---

## 前言

为了更好的管理项目，由于服务器是svn服务器，所以这里推荐使用git-svn协同工作，在原来的项目工程基础上可以使用git，git更适合Android这种类型的项目，接下来介绍相关命令。

## git-svn常见命令

### 1. 克隆svn项目

```
git svn clone [svn url]
```

例如：MT8163

```
git svn clone svn://192.168.12.7/MT8163_0/MT8163_01_MP5_V1
```

### 2. svn提交版本过多，采用下面的命令进行克隆，r1250表示从该版本开始克隆

```
git svn clone -r1250:HEAD --prefix=svn/ svn://192.168.12.7/MT8163_0/MT8163_01_MP5_V1
```

### 3. 查看当前版本库情况

```
git svn info
```

### 4. 取回远程仓库所有分支的变化

```
git svn fetch
```

### 5. 取回远程仓库当前分支的变化，并与本地分支变基合并

```
git svn rebase
```

### 6. 上传当前分支的本地仓库到远程仓库

```
git svn dcommit
```

### 7. 清空当前项目的所有修改包括新增文件

```
git add --all && git reset --hard
```

## 8. 同步 svn最新节点

```
git svn fetch  
git svn rebase
```

## 8. 克隆失败处理方式（跟运行环境有关）

提示 error: git-svn died of signal 11

处理方法:

```
rm -rf .git  
git svn init svn://192.168.12.7/MT8163_0/MT8163_01_MP5_V1  
git svn fetch
```

## 9. 提交修改到svn

```
git add --all #提交所有修改，包括新增到本地暂存区  
git commit -am "20210709 1.测试提交" #提交本地仓库  
git svn dcommit #提交到svn服务器
```

## 10. 常用git命令

```
git status #查看当前项目所有修改记录  
git status <dir> #查看指定目录的修改情况  
git diff <file path> #查看当前文件修改的内容  
git symbolic-ref --short -q HEAD 获取当前分支名
```

## 11.同步master代码到分支

```
git fetch  
git pull origin master  
git checkout V300  
git merge master -m "同步master修改"
```

## 12.git所有分支同步到最新

```
git fetch --all  
git pull --all
```

## 13.git stash

```
git stash save -u "描述"
    会把没有记录到的文件也保存下来
git stash list
    查看stash了哪些存储
git stash show :
    显示做了哪些改动, 默认show第一个存储, 如果要显示其他存储, 后面加stash@{ $num }, 比如第二个 git
    stash show stash@{1}
git stash show -p :
    显示第一个存储的改动, 如果想显示其他存储, 命令: git stash show stash@{ $num } -p , 比如第
    二个: git stash show stash@{1} -p
git stash apply :
    应用某个存储, 但不会把存储从存储列表中删除, 默认使用第一个存储, 即stash@{0}, 如果要使用其他
    个, git stash apply stash@{ $num } , 比如第二个: git stash apply stash@{1}
git stash pop :
    命令恢复之前缓存的工作目录, 将缓存堆栈中的对应stash删除, 并将对应修改应用到当前的工作目录下,
    默认为第一个stash, 即stash@{0}, 如果要应用并删除其他stash, 命令: git stash pop stash@{ $num } ,
    比如应用并删除第二个: git stash pop stash@{1}
git stash drop stash@{ $num } :
    丢弃stash@{ $num }存储, 从列表中删除这个存储
git stash clear :
    删除所有缓存的stash
```

## 14.svn提交步骤

### 1.添加文件

```
svn add + 文件名或文件目录
```

提示:

```
Could not add all targets because some targets are already versioned
```

使用 `svn add + 文件名或文件目录 --force`

### 2.提交文件到服务器

```
svn ci -m "提交修改的备注信息"
```

## 15.svn删除文件

```
svn delete + 文件名或文件目录
```

```
svn ci -m "提交删除的备注信息"
```

## 15.svn提交so文件

```
svn add --no-ignore + 文件名或文件目录
```

```
svn ci -m "提交删除的备注信息"
```

## 16.git 忽略提交文件，但是文件已经在服务器上了，如何解决？

```
git rm --cache -r filename
```