# Linux端创建git仓库

1、切换git用户（向git仓库上传文件时需要git有写的权限，可以先创建再改权限或者用git创建git库。）

2、使用git用户创建文件夹，初始化git库

Mkdir test , git init test

或者git init –bare test

[root@localhost user1]# su git

[git@localhost user1]$ mkdir tachao

[git@localhost user1]$ ls

tachao

[git@localhost user1]$ git init tachao

Initialized empty Git repository in /home/user1/tachao/.git/

[git@localhost user1]$ ls tachao/

[git@localhost user1]$ ls -a tachao/

. .. .git

[git@localhost user1]$ ll

总用量 4

drwxrwxr-x 3 git git 4096 5月 13 04:29 tachao

# windows端创建git仓库

1. 创建库文件夹

在库文件夹右键进入git bash

或者进入git bash 通过cd 命令进入库文件

1. 初始化仓库
2. 在git bash中执行git init把这个目录变成git可以管理的仓库，这时候你当前testgit目录下会多了一个.git的目录

**图形界面:**

用repository\shared.gif,空白处点击右键出现以下菜单,创建共享仓库



在弹出的对话框勾选make it bare



第一次的话会提示是个空仓库

# 设置身份

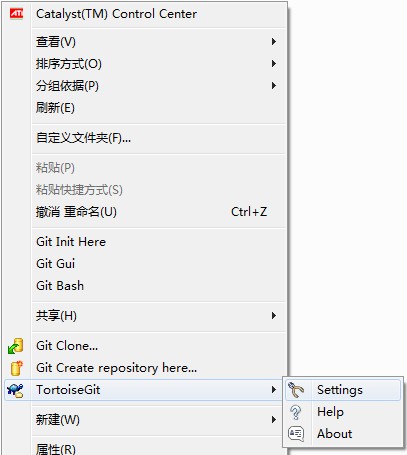
在命令行输入如下：

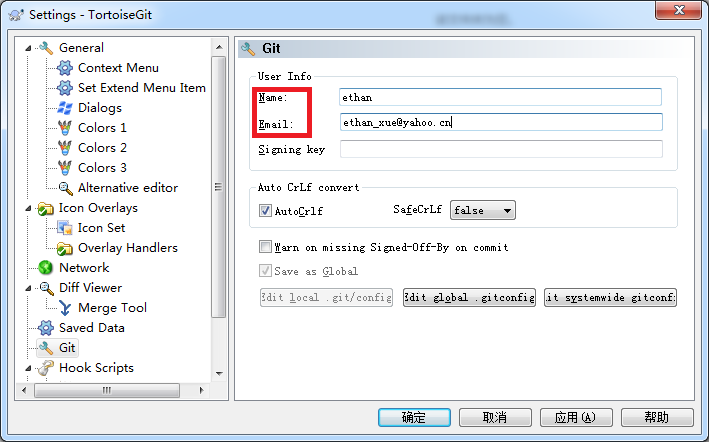
git config --global user.name "名字"

git config --global user.email "邮箱"

git config –global 参数，有了这个参数，表示你这台机器上所有的Git仓库都会使用这个配置

**图形界面:**





# 本地操作

1、Git是分布式版本控制系统，同一个Git仓库，可以分布到不同的机器上。怎么分布呢？最早，肯定只有一台机器有一个原始版本库，此后，别的机器可以“克隆”这个原始版本库，而且每台机器的版本库其实都是一样的，并没有主次之分。

分布式版本系统的最大好处之一是在本地工作完全不需要考虑远程库的存在，也就是有没有联网都可以正常工作，而SVN在没有联网的时候是拒绝干活的。

2、提交

第一步：使用命令 git add readme.txt添加到暂存区里面去；实际上就是把文件添加到暂存区（.git/stage）

第二步：用命令 git commit告诉Git，把文件提交到仓库。

提交修改和提交文件是一样的2步(第一步是git add 第二步是：git commit)。

3、查看

使用git status来查看下结果

git diff readme.txt查看变化内容

4、git checkout — file 可以丢弃工作区的修改

# 远程上传

1、创建版本库

2、在本地的仓库下运行命令：

git remote add origin git@192.168.1.17:/opt/git/test/test.git

建立连接

3、提交到本地

git add .

git commit -m "aaa"

4、把本地库的内容推送到远程

使用 git push命令，实际上是把当前分支master推送到远程。

git push -u origin master

由于远程库是空的，我们第一次推送master分支时，加上了 –u参数，Git不但会把本地的master分支内容推送的远程新的master分支，还会把本地的master分支和远程的master分支关联起来，在以后的推送或者拉取时就可以简化命令。

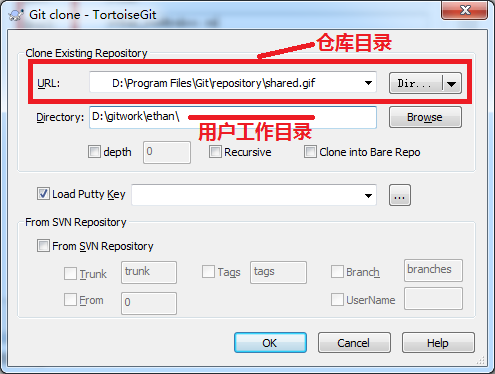
# 远程下载

1、新建库文件夹

2、在库文件夹右键进入git bash

或者进入git bash 通过cd 命令进入库文件

1. git clone [git@192.168.1.17:/opt/git/test/test.git](mailto:git@192.168.1.17:/opt/git/test/test.git)



4、更新本地文件

查看远程分支：git remote -v。

git fetch：相当于是从远程获取最新版本到本地，不会自动merge

git fetch origin master:tmp

git diff tmp

git merge tmp

首先从远程的origin的master主分支下载最新的版本到origin/master分支上，然后比较本地的master分支和rigin/master分支的差别，最后进行合并

git pull：相当于是从远程获取最新版本并merge到本地

git pull origin master相当于git fetch 和 git merge

# 分支管理

Git branch查看分支一览表

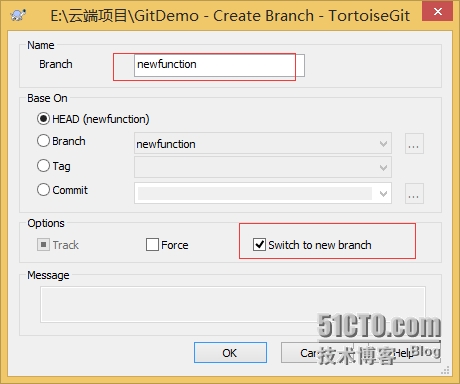
切换分支git checkout –b tmp

或Git branch tmp ; git checkout tmp

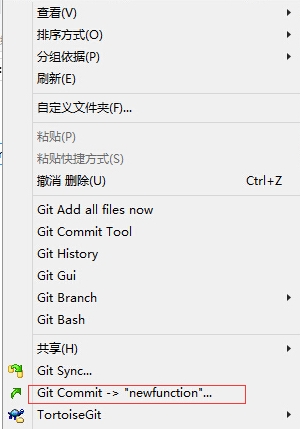
合并分支

Git checkout master ; git merge –no-ff tmp

新建一个分支，右键-->tortoiseGit-->Create Branch

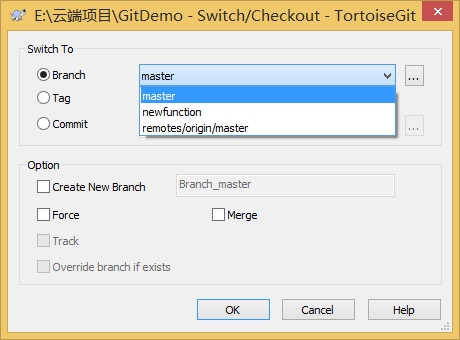
[](http://s3.51cto.com/wyfs02/M02/58/A2/wKioL1S2jeWjAmwPAAE2UQMZ6i4217.jpg)

右键，你会发现当前的分支为你新建的newfunction分支了

[](http://s3.51cto.com/wyfs02/M02/58/A6/wKiom1S2jXjCcTaPAAD1j9rKeKM309.jpg)

然后我们切换到master分支，右键-->tortoiseGit-->Switch/Checkout

选择master即可，然后确定

[](http://s3.51cto.com/wyfs02/M01/58/A6/wKiom1S2kIWR_elPAAEO8ggIGnY473.jpg)

# 获取更新到分支

