Git配置过程

说明：该文档适用于在开发时配置Git和TortoiseGit，克隆仓库代码的指南文档，并未涉及代码提交，分支切换等功能。

文档修改记录

|  |  |  |  |
| --- | --- | --- | --- |
| **版本号** | **修改人** | **修改时间** | **修改内容简述** |
| V1.0 | 宋全恒 | 2018-08-16 | 新建 |
|  |  |  |  |
|  |  |  |  |

# 简介

在程序员的世界里，版本管理是无法缺少的一部分，因为在一个普通程序员一天的工作中，代码是就像建筑工人搬的砖，会非常多，积攒出来的数以千百行的代码如果没有版本管理工具，无疑是一场耗费时间的灾难。在问题出现时，有可能你的本地代码已经修改的面目全非而不得不回退，也有可能，你一次要提交的代码太多，结果你发现需要解决的冲突太多，也许，有一天你想回退到某个特定的版本重新开始开发，也许在本地开发的时候，代码越写越有问题，会想要重新从远端仓库克隆一份新的代码继续使用。或许开发了许久的代码却被自己修改的无法正常工作，这都是非常可能的情况，因为一份正常的可以运行版本的代码是非常重要的。而Git就是这样的一个程序开发的版本管理工具。Git是一个开源的分布式版本控制系统，可以有效，告诉的处理从很小到非常大的项目版本管理。Git是Linus Torvalds为了帮助管理Linux内和开发的一个开放源码的版本控制软件。一般开发人员使用的Git功能有：

1. 从服务器上克隆完整的Git仓库（包括代码和版本信息）
2. 在自己的机器上根据不同的开发目的，创建分支，修改代码
3. 生成补丁(patch)，把补丁发给主开发者。
4. 在提交时，如果包括冲突，需要先解决冲突。
5. 从Master分支拉取新的分支，多个分支并行开发，最后解决冲突合入主线。

# 材料

在使用Git管理程序代码是，需要下载的文件清单如下：

1. Git-2.18.0-64-bit
2. TortoiseGit-2.6.0.0-64bit
3. TortoiseGit-LanguagePack-2.6.0.0-64bit-zh\_CN

这些文件可以在如下的URL一键获取：

<https://download.csdn.net/download/lk142500/10566885>

注意：因为现在一般的电脑系统位数均为64位，所以仅提供了64位，并且版本需要自己下载。上述的三个文件，曾在多个电脑上部署，可以正常工作。

TortoiseGit是一个开放的Git版本控制系统源客户端，就像一个普通的文件服务器，因为具有界面并且能够以图示的方式显示文件和目录的变化，并能够提示何时改变了那些历史数据，改变人是谁，因此非常方便。

Git则是版本管理的基础功能，通过使用Git的命令行界面，几乎所有的功能均能在Bash命令行下得到实现。唯一的缺点就是不直观。

TortoiseGit其实就是对Git的一个封装起来的程序，用来简化Git的操作。而Language则是TortoiseGit的汉化包。

# 配置步骤

## 程序安装

确保程序的安装顺如为Git、TortoiseGit、TortoiseGit-LanguagePack-2.6.0.0-64bit-zh\_CN。

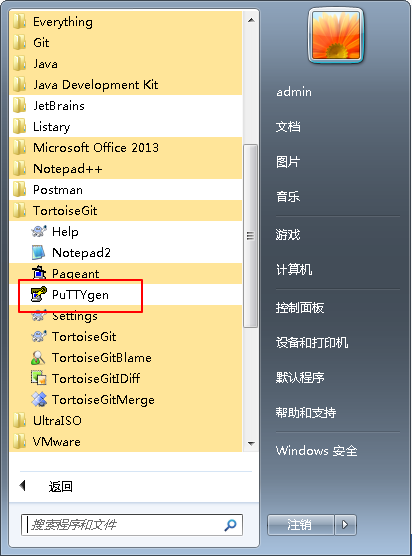
安装过程问题不大，根据提示默认安装即可。

## 生成SSH Keys

ssh keys是远程ssh连接的一种基于秘钥方式安全连接起来的秘钥文件。ssh keys是ssh中基于秘钥的安全验证，你可以通过创建公钥和私钥对的方式来完成ssh keys方式的ssh登录验证。首先需要执行的操作是必须在本地开发机上创建一对秘钥，并把公钥的内容放在需要访问的服务器上。如果你连接到ssh服务器上，客户端软件就会向服务器发出请求，请求用你的秘钥进行安全验证。

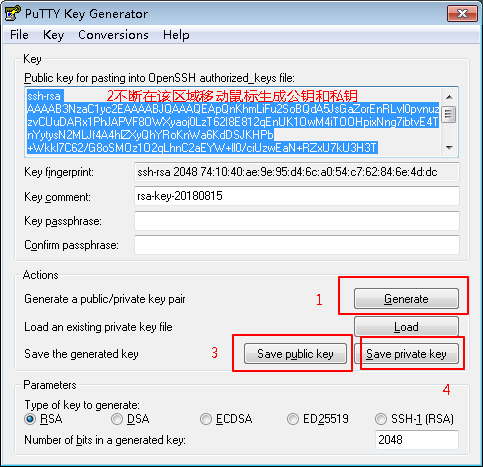
服务器收到请求后，先在该服务器上的主目录下寻找你的公用秘钥，然后把它和你发送过来的公用秘钥进行比较。如果两个秘钥一致，服务器就用公用秘钥加密“质询”并把它发送给客户端软件。客户端软件收到质询之后就可以用你的私有秘钥解密再把它发送给服务器。基于ssh keys的登录验证方式可以避免假冒服务器的问题，因为假冒问题获取不到你的秘钥。它比基于用户密码的口令方式更加安全。

### 为TortoiseGit生成SSH Keys



点击PuTTYgen即可，倘若在工作机上安装了Everything或者Listary等工具，直接搜索该文件即可。该文件的默认目录位于：

C:\Program Files\TortoiseGit\bin



在Key区域根据提示

Please Generate some randomness by moving the mouse over the blank area

快速移动鼠标产生随机数以产生Public key。生成Key之后通过点击Save public key把公钥文件保存在当前目录[TortoiseGit\bin]下，命名为id\_rsa.pub,然后点击Save private key，把私有文件保存在当前目录下，命名为id\_rsa.pri即可。

注意：私有秘钥文件是需要保存的，在使用TortoiseGit克隆代码时，需要勾选私钥文件进行加载的，不然会一直提示输入Git服务器的密码

然后需要拷贝生成的公钥文件，全选区域2中的内容即可。把该内容添加到Git 服务器网页个人设置中。参见3.3节

### 为Git生成公钥

一个重要的问题是，git生成的公钥和TortoiseGit生成的公钥是不一致的。

在3.2.1节生成的秘钥对针对TortoiseGit客户端而言的。因此，需要为Git生成公钥。

首先需要确认你自己是否已经拥有Git秘钥。默认情况下，用户的SSH秘钥保存在~/.ssh目录下，进入该目录下，并列出其中的内容，你便可以快速确认自己是否已拥有秘钥：

$ cd ~/.ssh

$ ls

Authorized\_keys2 id\_dsa known\_hosts

Config id\_rsa\_pub

需要找到一对以id\_dsa或者id\_rsa命名的文件名，其中一个带有pub扩展名，另一个则是私钥。.pub文件是公钥，另外一个则是私钥。如果找不到这样的文件，则可以进行下述操作生成秘钥对。在我的工作机上运行上述命令

admin@admin-PC MINGW64 ~/.ssh

$ cd ~/.ssh/

admin@admin-PC MINGW64 ~/.ssh

$ ls

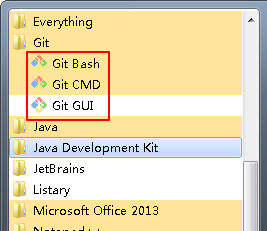
id\_rsa id\_rsa.pub known\_hosts

admin@admin-PC MINGW64 ~/.ssh

$

可以看到秘钥文件对id\_rsa和id\_rsa.pub

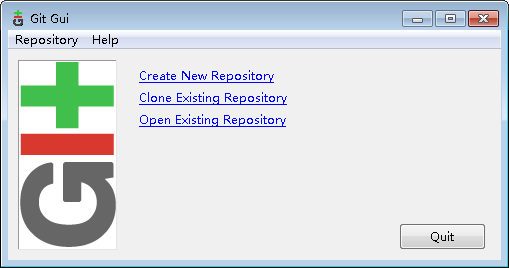
在安装Git软件之后，安装包会自动安装三个应用程序



可以通过Git Bash 或者Git GUI生成秘钥对。

#### Git UI生成SSH Key

点击Git GUI应用程序，弹出如下界面



点击Help->Show SSH Key



在点击Generate Key之后，会弹出两个对话框，直接Enter即可。全选生成的秘钥，拷贝，需要把该内容加入到Git服务器网站个人SSH Keys的设置中。可以看到通过该界面方式，程序会自动的把秘钥内容保存在目录

~/.ssh/id\_rsa.pub

即，通过该方式生成的秘钥对与Git Bash生成SSH Key其实是一模一样的效果。拷贝到Git服务器个人设置的过程参见3.3

#### Git Bash生成SSH Key

在确保目录

~/.ssh/id\_rsa.pub

不存在id\_rsa和id\_rsa.pub文件之后，可以通过Git bash命令行通过ssh-keygen产生秘钥对文件。

首先，在桌面或者任意文件目录下通过点击

Shift和鼠标右键

弹出如下界面：



可以看到Git GUI和Git Bash两个应用程序的入口。直接点击Git Bash即可。

admin@admin-PC MINGW64 ~/Desktop

$ cd ~/.ssh

admin@admin-PC MINGW64 ~/.ssh

$ ls

[ls命令没有列出文件，说明Git秘钥对不存在]

admin@admin-PC MINGW64 ~/.ssh

$ ssh-keygen

Generating public/private rsa key pair.

Enter file in which to save the key (/c/Users/admin/.ssh/id\_rsa):

Enter passphrase (empty for no passphrase):

Enter same passphrase again:

Your identification has been saved in /c/Users/admin/.ssh/id\_rsa.

Your public key has been saved in /c/Users/admin/.ssh/id\_rsa.pub.

The key fingerprint is:

SHA256:MfdeR1XNobRSmgAIWiVPSI4vm2lIxODMrKmatFzZzV0 admin@admin-PC

The key's randomart image is:

+---[RSA 2048]----+

|. .=+o.... o o\*|

|B =.+. . = o +|

| X . . o .+ o .|

|o.. + .. . |

|oo . S E. . .|

|+ = o o . .. . . |

|o\* o . o . . |

|=.o |

|oo |

+----[SHA256]-----+

admin@admin-PC MINGW64 ~/.ssh

$ ls

id\_rsa id\_rsa.pub

admin@admin-PC MINGW64 ~/.ssh

$ cat id\_rsa.pub

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQC1oAy1QiN9fXELTTn+ZIRk96Nz7diBrEiJeKuRtTeyUIjCzlOIxCAfP34NdzKj8FPMlHf8u0+ILNUjA26J1IJbxttef3x8IjMGNnawOmJv69SVLncfokK0QtKngzuWcnYYfffyxz4HC5D34ZrEDnicapMdIEVs4olA/O8unL+qOXrrA1gd5rG0Vc7hPHFCo/K73BJaNlxhR3PfalN7Gb0xAaAXeNw6dsHUnadSJd9nAlrn7ySh2aJF6+oGYalcAmCqkwApH/WYAX2BfjBWXg7QufwaWT5+suIdJLDuEeamjzhS0l9SAxvx9JexN0KC4MlSp4W92K8k9vONK0eCfL+h admin@admin-PC

admin@admin-PC MINGW64 ~/.ssh

$

首先ssh-keygen会确认秘钥的存储位置(默认是./ssh/id\_rsa)，然后它会要求你输入两次秘钥口令。如果你不想在使用密钥时输入口令，将其留空即可。

然后拷贝id\_rsa.pub文件的所有内容到Git服务器的个人设置即可，参见3.3

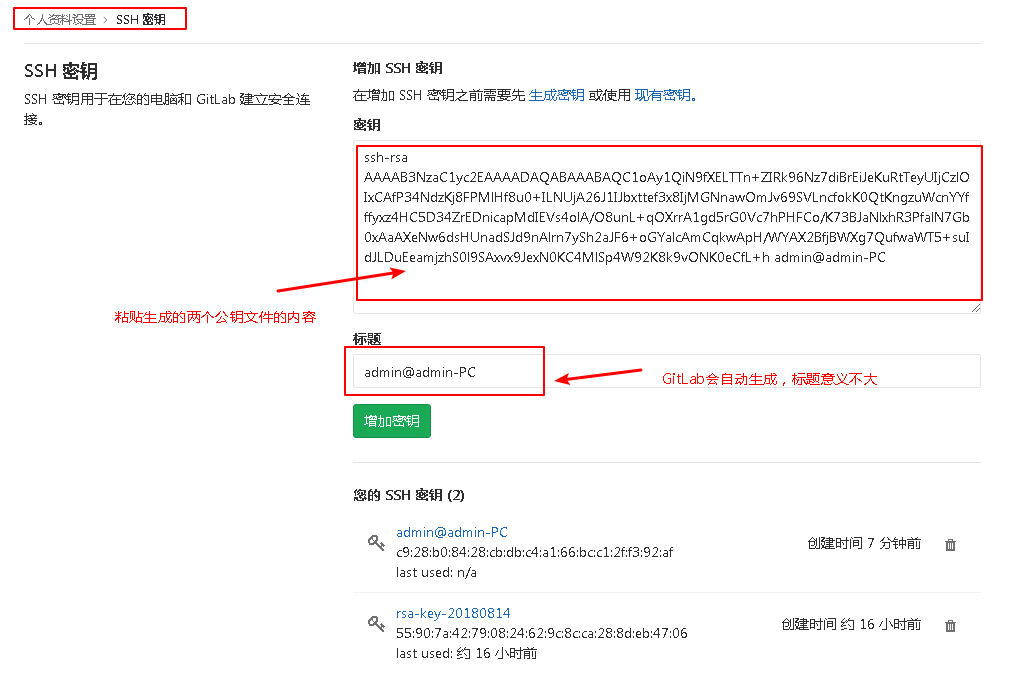
## Git网站添加SSH Keys

### 登录Git

使用个人的Git账户登录Git服务器网址，点击头像下方的设置，然后选择出现页面左侧的SSH密钥。

### 添加SSH Key

拷贝TortoiseGit和Git生成的两个公钥文件的内容拷贝。



保存两个公钥文件的内容到自己的个人设置即可。

## 结果验证

### 拷贝Git url

登录Git服务器，然后拷贝SSH的项目URL。



### 使用Git Bash克隆项目

进入你想要拷贝项目的文件目录，例如

D:\Git

然后点击右击空白处，选择Git Bash。键入git clone 命令如下：

admin@admin-PC MINGW64 ~/.ssh

$ git clone git@192.168.0.223:fiso/qw.git

Cloning into 'qw'...

The authenticity of host '192.168.0.223 (192.168.0.223)' can't be established.

ECDSA key fingerprint is SHA256:T/xFtMg6XVeFxyuWXohc7V6UD56eHaoTdVCkW/UsPTQ.

Are you sure you want to continue connecting (yes/no)? yes

Warning: Permanently added '192.168.0.223' (ECDSA) to the list of known hosts.

remote: Counting objects: 11618, done.

remote: Compressing objects: 100% (359/359), done.

remote: Total 11618 (delta 256), reused 501 (delta 192)

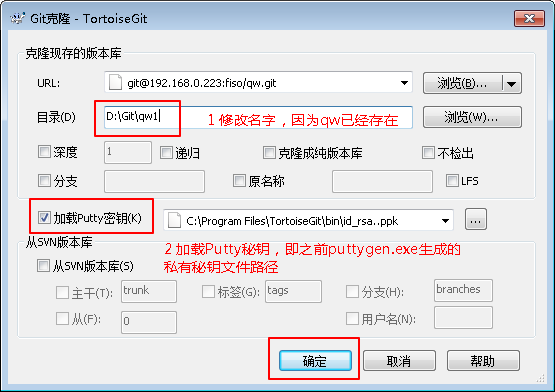
Receiving objects: 100% (11618/11618), 745.83 MiB | 26.62 MiB/s, done.

Resolving deltas: 100% (2642/2642), done.

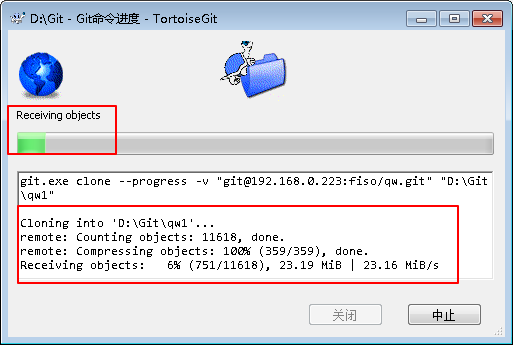
Checking out files: 100% (18760/18760), done.

### 使用Tortoise Git克隆项目

在当前目录下，右击空白处，选择Git克隆[使用Tortoise Git]，运行界面如下：



点击确定之后：运行界面如下：



注意：确保勾选加载Putty秘钥

# 结论

在使用Git和TortoiseGit应用程序进行项目代码的管理时，正确的配置Git和TortoiseGit是工作的第一步，因此，确保能Git和TortoiseGit能正常工作，是非常重要的一项入门能力。在使用过程中会遇到各种各样的问题，记录问题，并且记录问题的解决方式，总结，反复的锤炼积攒，慢慢就会对Git与TortoiseGit的使用悉熟于心。提高自己的开发效率，使得自己的主要精力放在代码编写和Bug调试上。