

## 说明

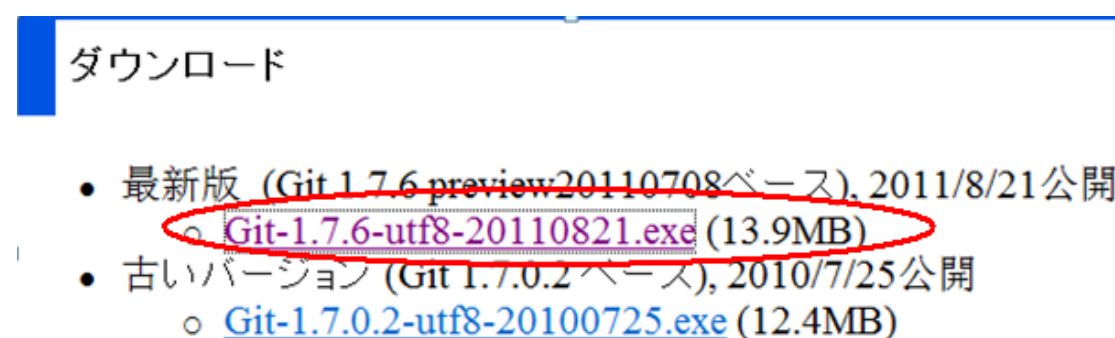
本文档只是日常工作中一些使用步骤的简单归纳，旨在帮助大家快速用上 GIT，先能够用起来，知其然后再去究其所以然，如有疏漏，请大家发邮件 [xukong@21cn.com](mailto:xukong@21cn.com)，会及时进行改进，谢谢。

## 安装步骤简介：

1. 安装支持中文文件名的 Git 命令行软件包 MsysGit。
2. 在 MsysGit 中生成用来访问 Git 版本控制库的公钥数据。
3. 安装 Git 图形界面客户端软件 TortoiseGit（此软件调用第 1 步中安装的 Git 命令行软件完成相关操作）
4. TortoiseGit 使用简介。

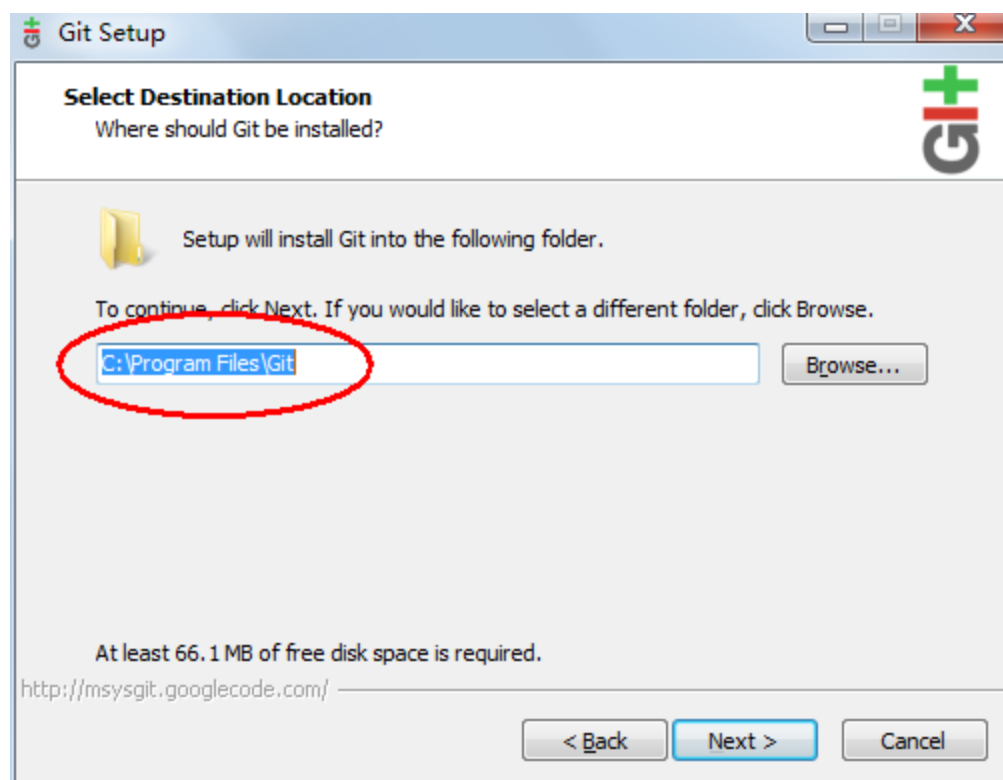
## 1 安装支持中文文件名的 GIT 软件包 MsysGit

下载支持中文文件名的 MsysGit 软件，可访问网址 <http://tmurakam.org/git/index.html>，下载其中的 <http://tmurakam.org/git/Git-1.7.6-utf8-20110821.exe>，如下图所示



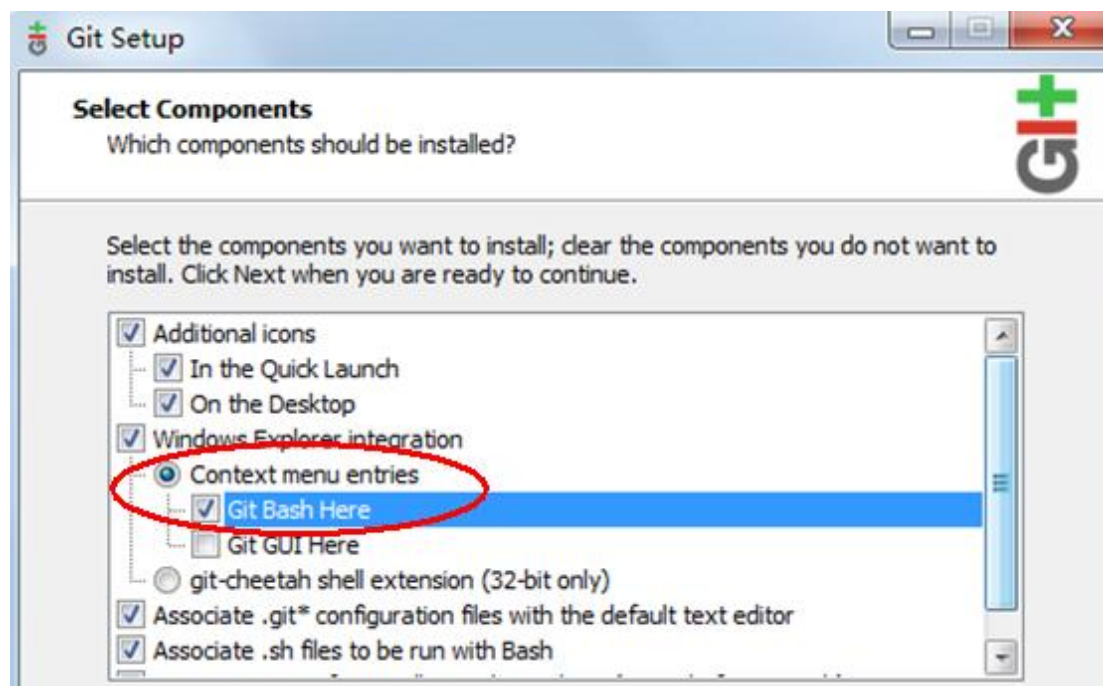
## 1 配置安装目录

使用默认的路径即可，此路径在下面配置 Git 图形界面客户端 TortoiseGit 时会用到，如下图所示：

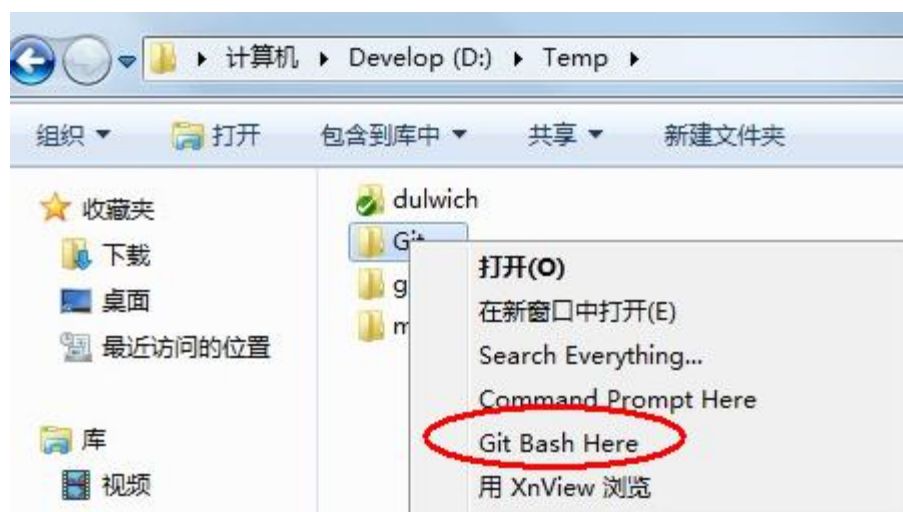


## 2 配置命令行快捷方式

在安装中会出现设置如何与 Windows 的文件浏览器结合的界面，设置成“可在目录上点右键直接进入该目录的 Git 命令行”，如下图所示：

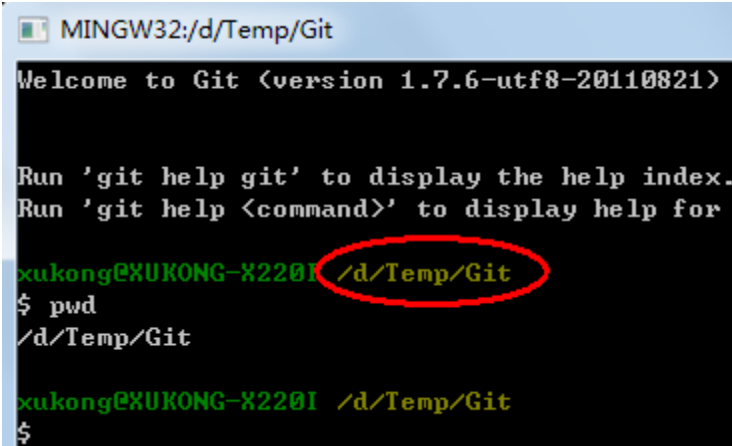


按上图的配置完成安装后，在任意目录如“D:\temp”上点右键，会出现“Git Bash Here”的菜单，如下图所示：



选择“Git Bash Here”菜单后即可进入相关目录的 Git 命令行模

式，可输入 Git 命令完成相关操作，如下图所示：



```
MINGW32:/d/Temp/Git
Welcome to Git <version 1.7.6-utf8-20110821>

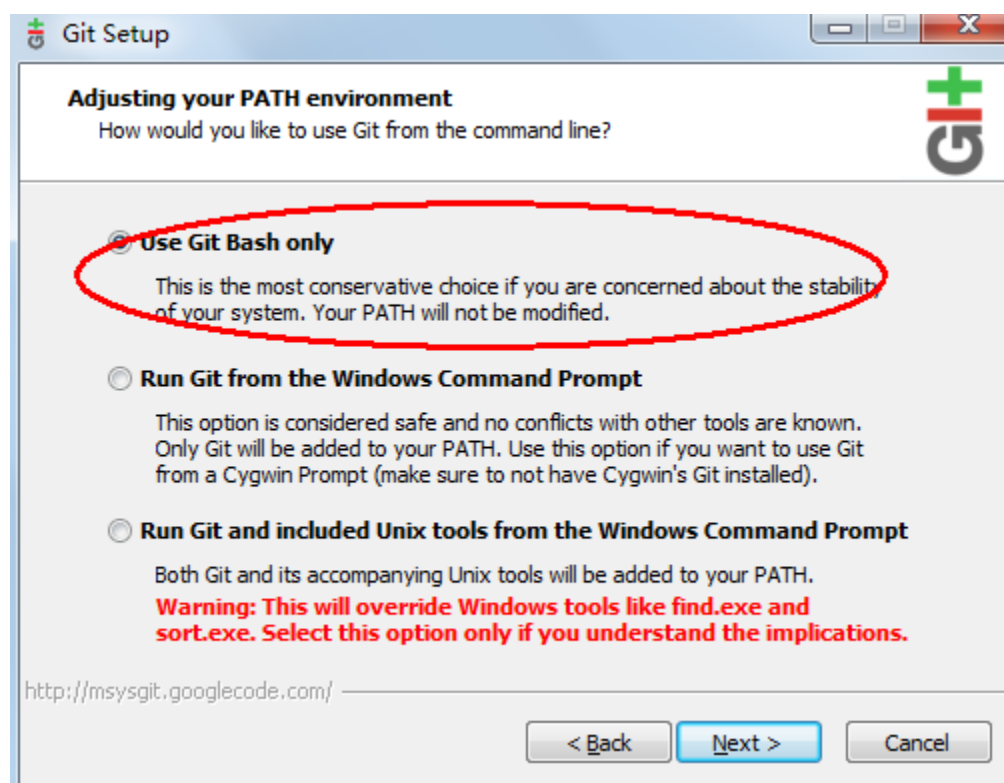
Run 'git help git' to display the help index.
Run 'git help <command>' to display help for

xukong@XUKONG-X220I /d/Temp/Git
$ pwd
/d/Temp/Git

xukong@XUKONG-X220I /d/Temp/Git
$
```

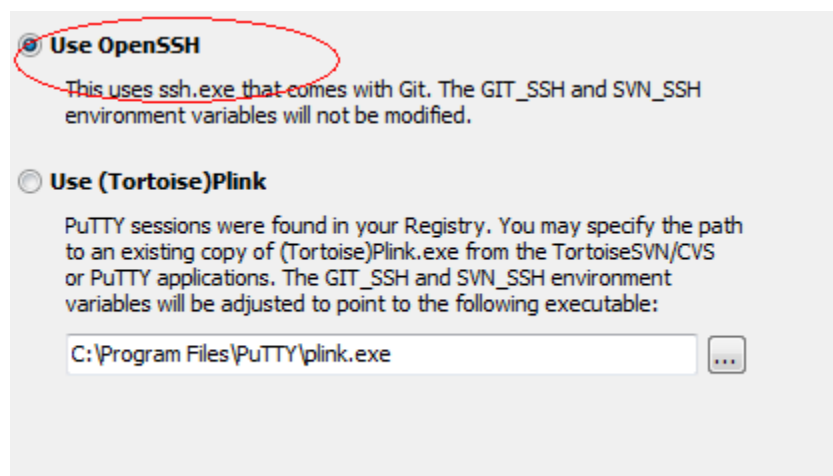
### 3 设置如何使用 GIT 命令行

按默认设置，使用软件自带的 BASH 风格的命令行即可，如下图所示：



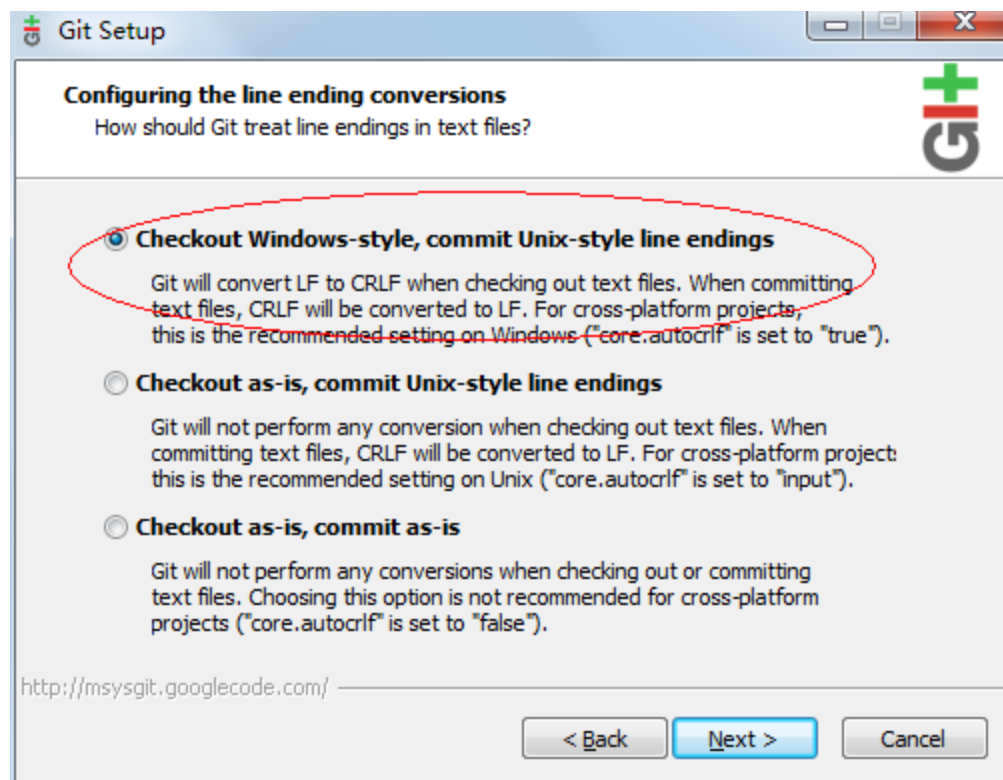
## 4 配置使用什么 SSH 客户端

配置使用软件自带的 SSH 客户端来完成与 GIT 服务器的 SSH 安全交互，确认是可信的用户才能对版本库进行操作，按系统默认设置即可，如下图所示：



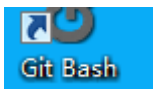
## 5 配置版本控制中文本文件的换行风格

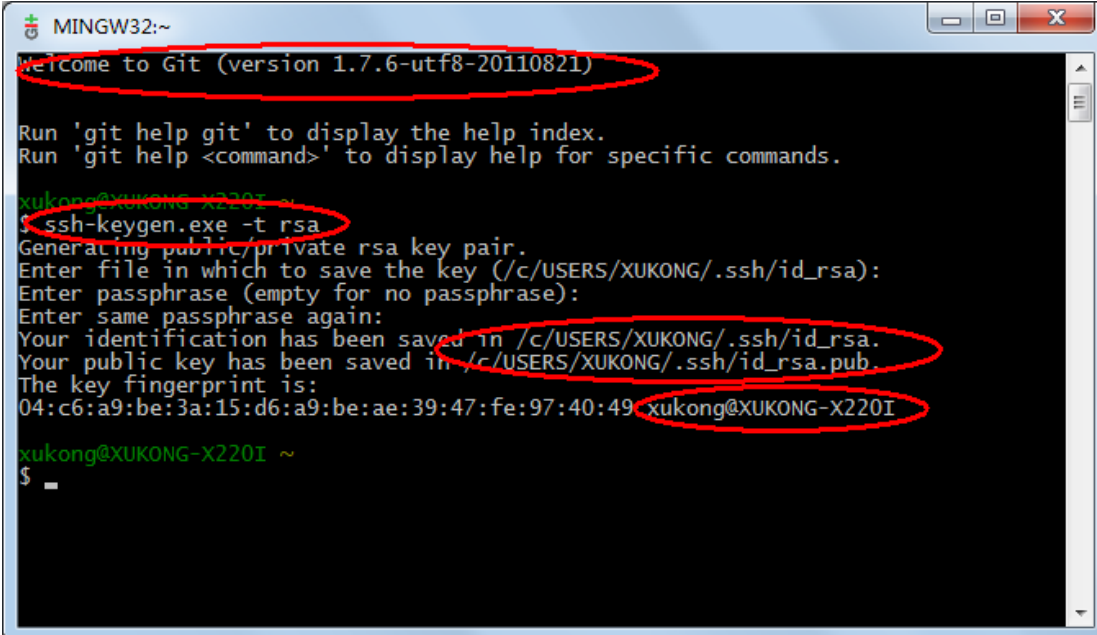
按系统默认设置即可（即从版本库中取出时使用 Windows 系统的“回车+换行”风格，存入版本库时使用 Unix 系统的“换行”风格），如下图所示：



其他的一路点击“Next”，即可完成软件安装。

## 6 生成 SSH 安全密钥，提供给 GIT 版本库管理员以访问 Git 版本库

点击桌面上生成的  图标，默认进入用户主目录，然后执行 `ssh-keygen.exe -t rsa`，一路输入回车即可完成操作。

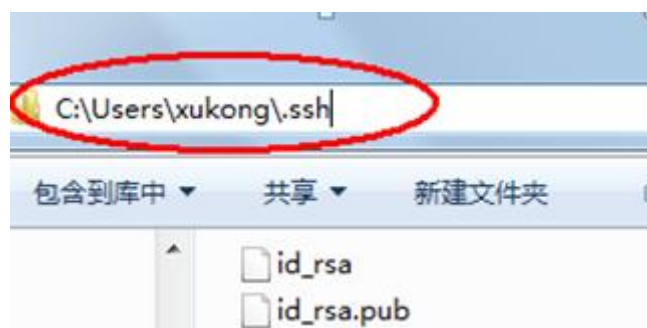


```
MINGW32:~
Welcome to Git (version 1.7.6-utf8-20110821)

Run 'git help git' to display the help index.
Run 'git help <command>' to display help for specific commands.

xukong@xukong-X220I ~
$ ssh-keygen.exe -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/c/USERS/XUKONG/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/USERS/XUKONG/.ssh/id_rsa.
Your public key has been saved in /c/USERS/XUKONG/.ssh/id_rsa.pub.
The key fingerprint is:
04:c6:a9:be:3a:15:d6:a9:be:ae:39:47:fe:97:40:49 xukong@XUKONG-X220I
xukong@XUKONG-X220I ~
$
```

在用户主目录下，例如用户名为 `xukong`，则在目录 `c:\Users\xukong` 下会出现 `.ssh` 目录，里面会生成 2 个文件：`id_rsa` 为私钥文件，`id_rsa.pub` 为公钥文件，如下图所示：



在 `id_rsa.pub` 文件内容的最后，即 2 个 `=` 号的后面，如下图所示：

`== xukong@XUKONG-X220I`，是此用户的鉴权名称（根据用户在



WINDOWS 中的用户名和主机名称的不同 , 每个 id\_rsa.pub 最后的内容也会不同 ) 将 id\_rsa.pub 复制一份 按=号后的名称重新命名 , 例如上图所示的内容 , 命名成 [xukong@XUKONG-X220I.pub](#) , 发送给 GIT 版本库管理员即可。

## 2 安装 PUTTY 软件包

为使用SSH协议与Git版本库进行交互 需安装免费软件PUTTY , 软件主页在

<http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>

下载完整的安装包

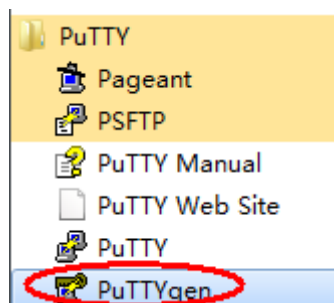
<http://the.earth.li/~sgtatham/putty/latest/x86/putty-0.61-installer.exe>

**安装时一路点击 “Next” 即可**

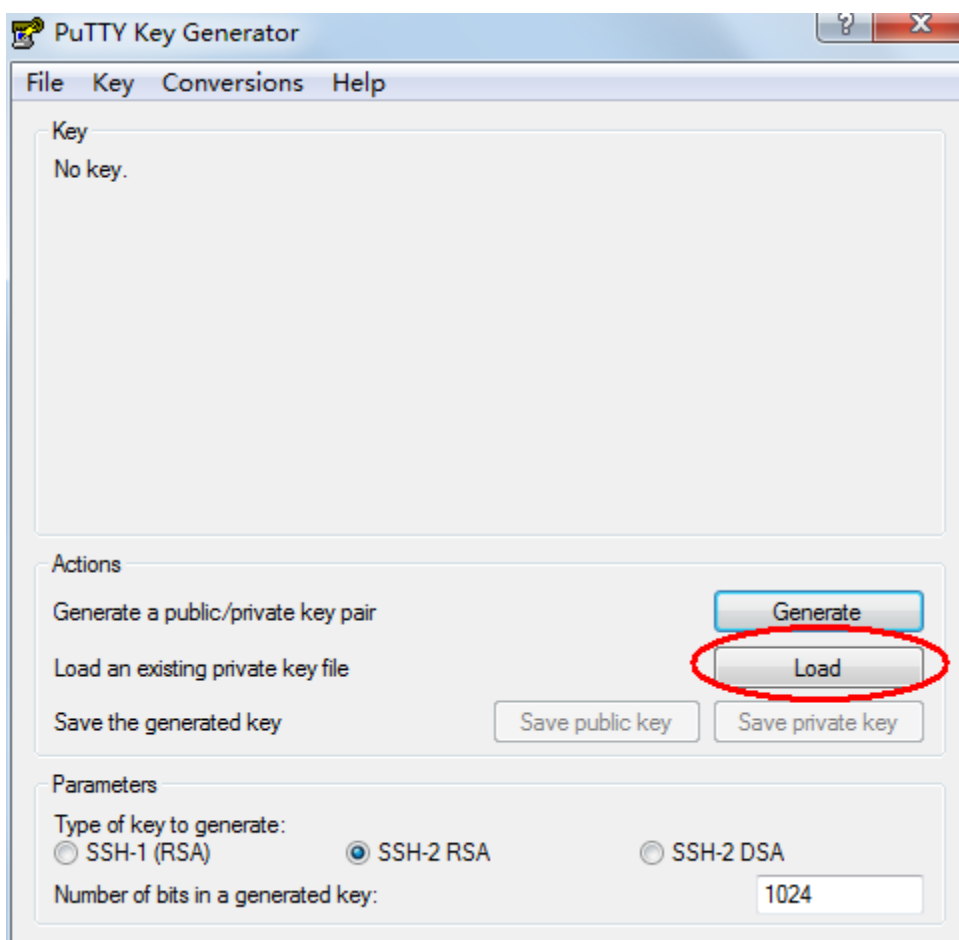
## 从 id\_rsa 私钥文件中生成.ppk 文件

因为下面安装的 Git 图形界面客户端软件 TortoiseGit 可使用 PUTTY 通过 SSH 协议与 Git 版本库进行安全验证，所以需要生成相关公钥的 ppk 配置文件。操作步骤如下：

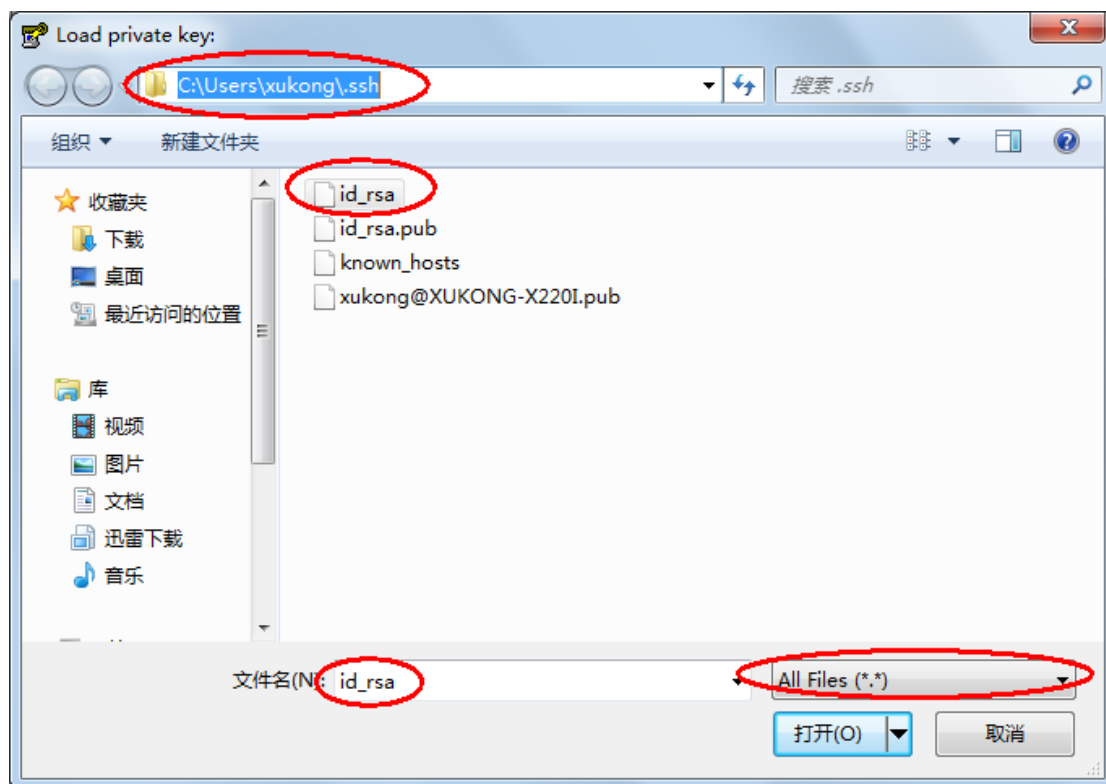
执行 puTTYgen 程序，如下图所示：



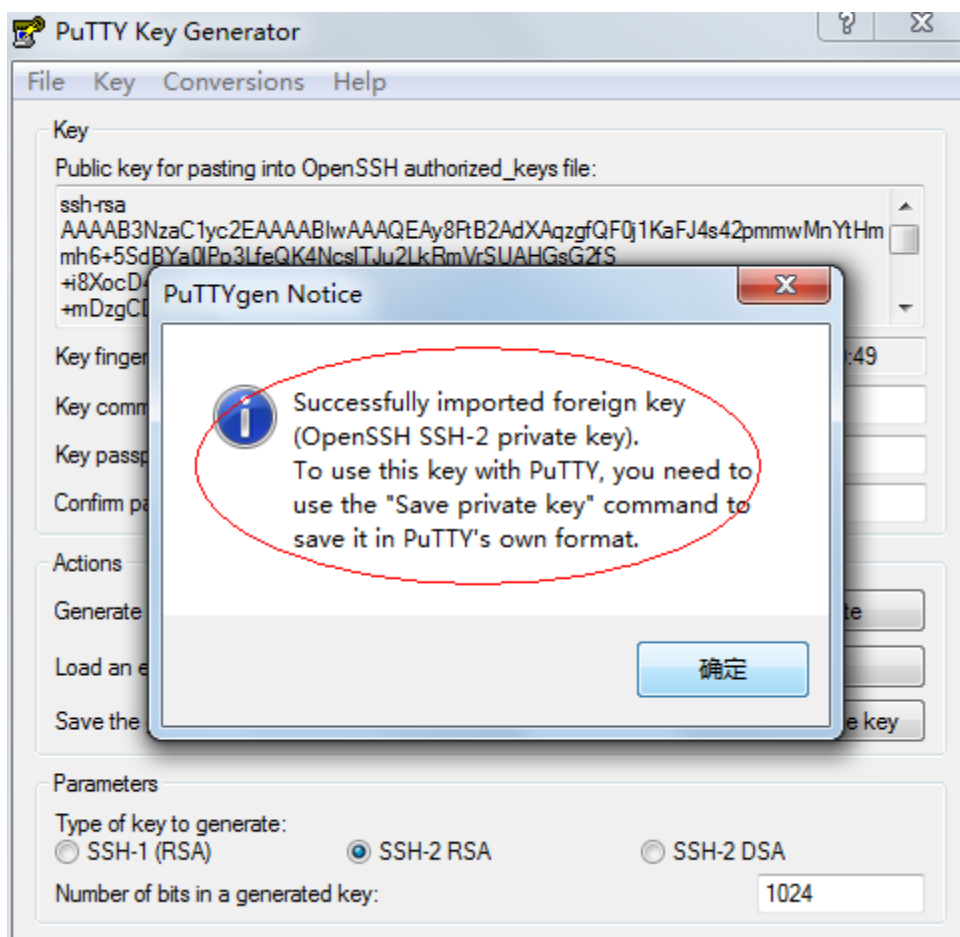
然后点击“Load”按钮，如下图所示：



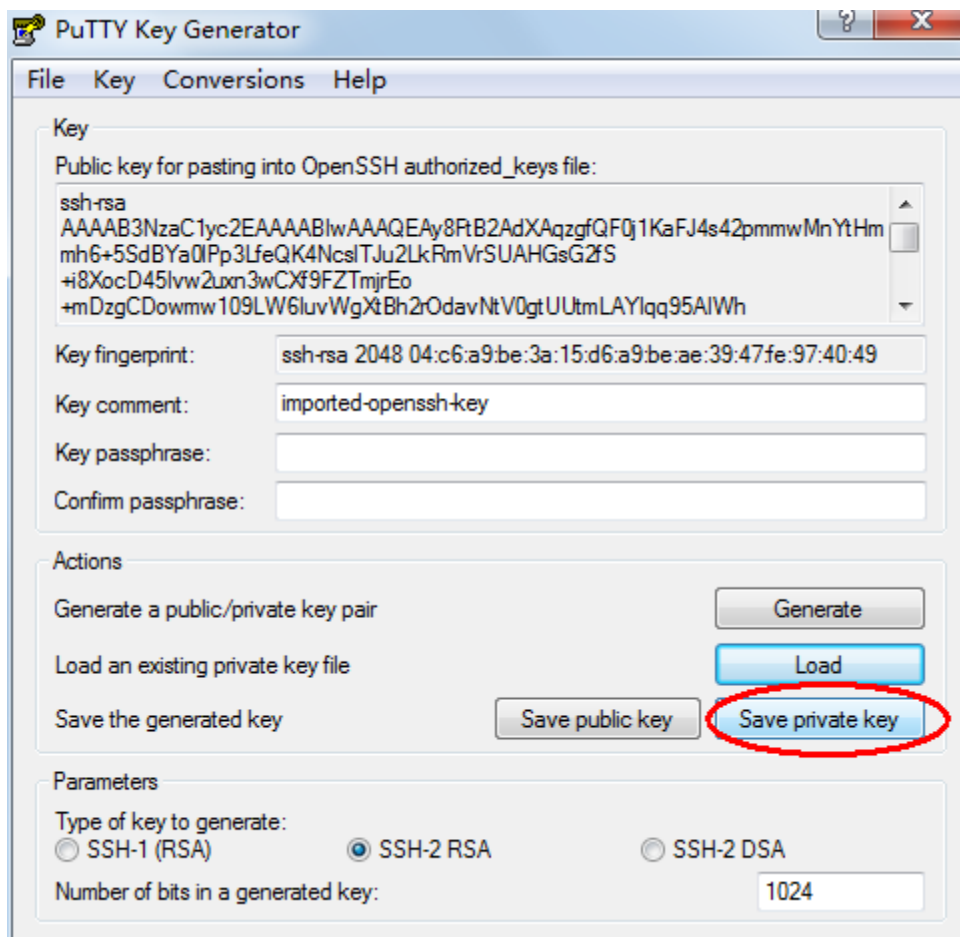
找到用户的.SSH 目录(一般是在 C 盘“用户”目录下的相关用户名目录下),载入其中“id\_rsa”文件(注意界面中默认的是只显示.ppk 文件,需要选择“显示所有文件”以看到 id\_rsa 文件),如下图所示:



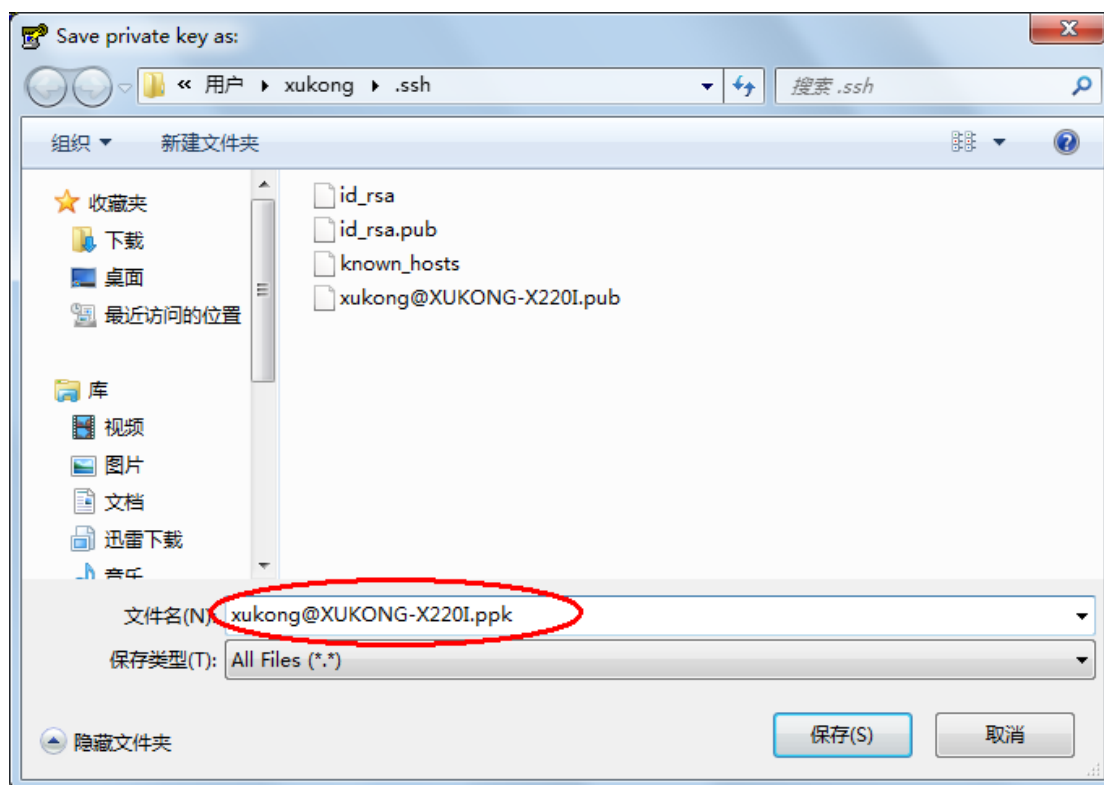
然后提示导入成功，如下图所示：



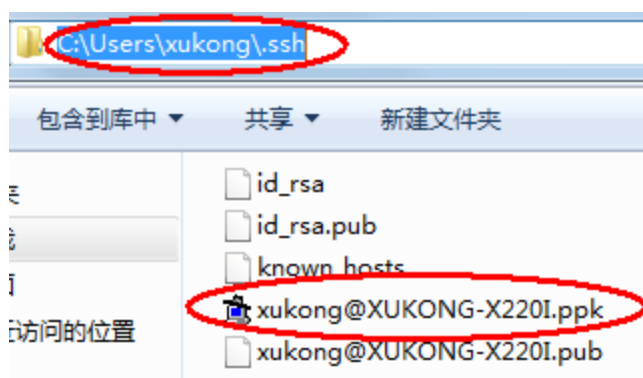
然后点击按钮 “Save private key” 另存为 ppk 文件，如下图所示：



文件名建议命名成前面提到的 id\_rsa.pub 文件中, 最后 2 个=号后出现的信息, 本例中是 xukong@XUKONG-X220I, 后缀名为 ppk 即可。



最后.SSH 目录中出现如下文件：



正确完成上述步骤, 就生成了 puTTY 软件在 SSH 验证中使用的 ppk 格式的私钥文件。

### 3 安装 Git 图形界面客户端 TortoiseGit

在 <http://code.google.com/p/tortoisegit/> 中可下载相关软件：

32 位下载地址为

<http://tortoisegit.googlecode.com/files/TortoiseGit-1.7.5.0-32bit.msi>

64 位下载地址：

<http://tortoisegit.googlecode.com/files/TortoiseGit-1.7.5.0-64bit.msi>

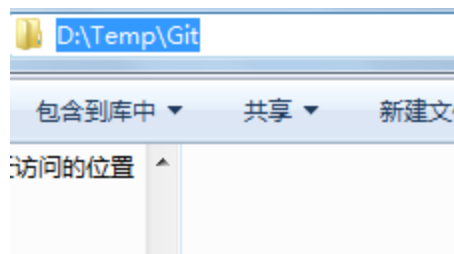
#### 1 安装注意事项

一路点击 “Next” 即可。

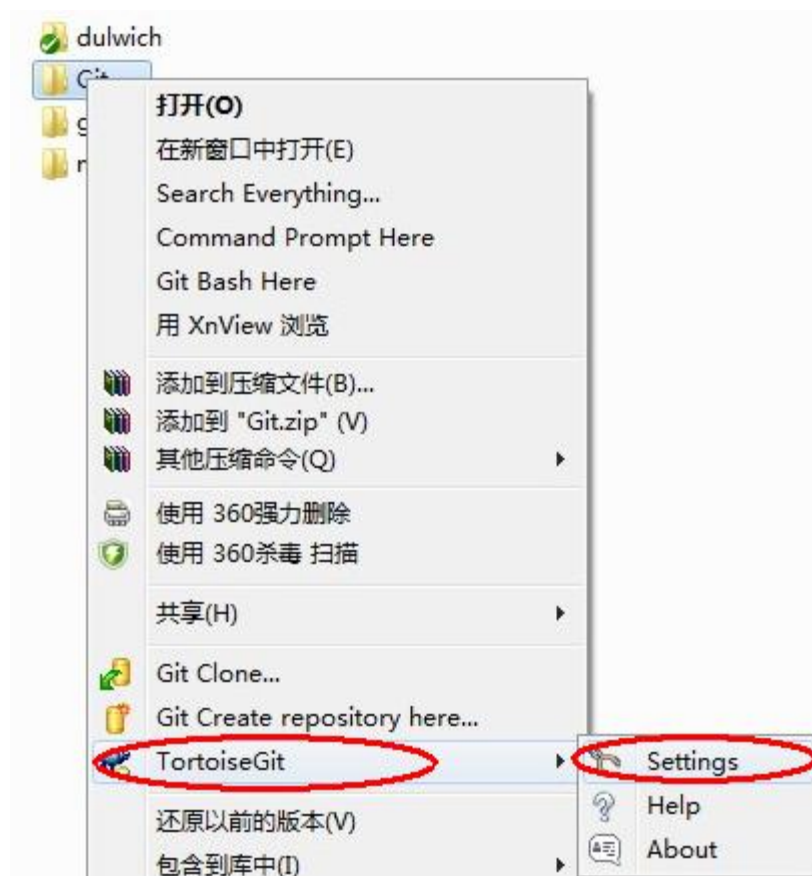


## 2 安装后的配置

随意新建一个目录，例如 d:\temp\git，如下图所示：

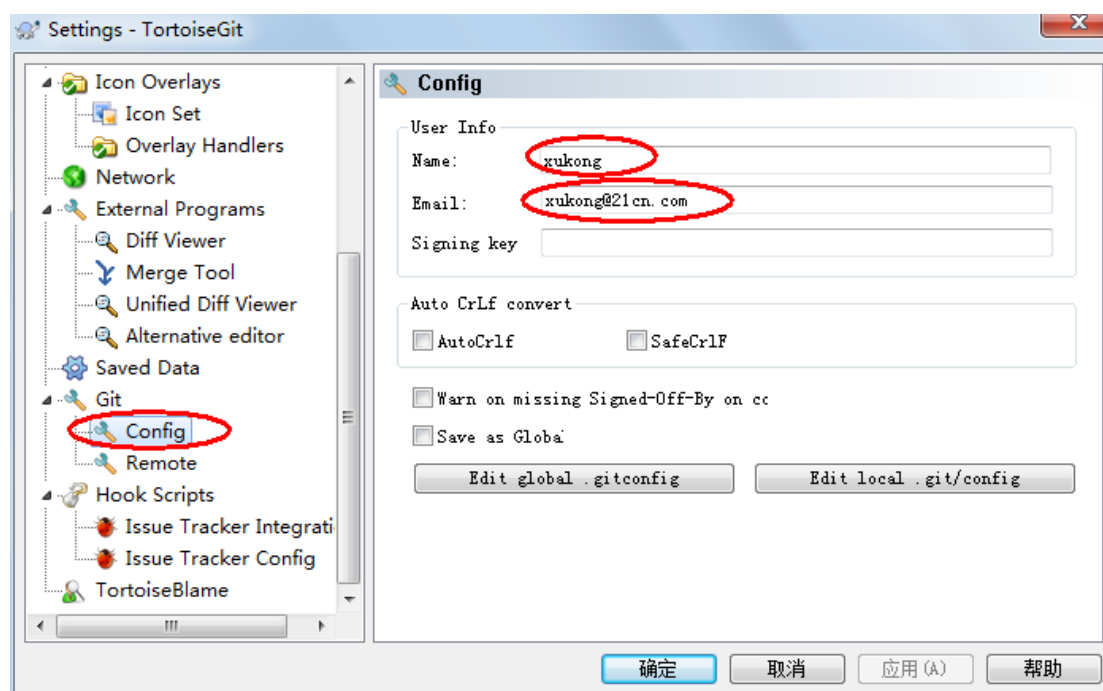


在目录“d:\temp\git”上点击鼠标右键，选择“TortoiseGit” - “Setings”进行配置，如下图所示：



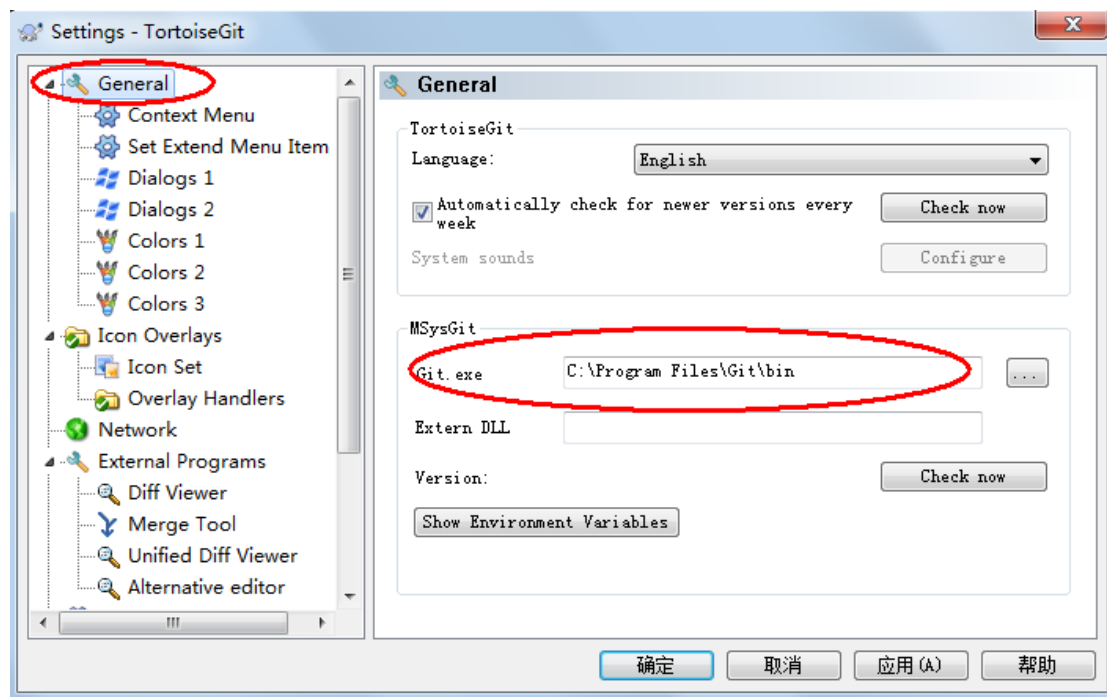
## 1 配置使用 GIT 版本库的用户信息（重要）

选择 “Git” 下的 “Config”，然后在 “Name” 和 “Email” 中填入相关用户信息即可



## 2 配置 MsysGit 软件的路径信息

### 配置 Git.exe 的路径



## 4 TortoiseGit 使用简介

### 1 克隆 ( Clone ) 签出文件

即对应其他版本控制软件的签出操作，例如 SVN 的 Checkout 操作

本例中 Git 版本库相关信息如下：

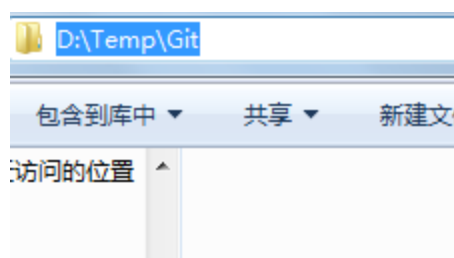
Git 版本库服务器地址：“10.196.11.242 ”

操作 Git 版本库的用户名：git

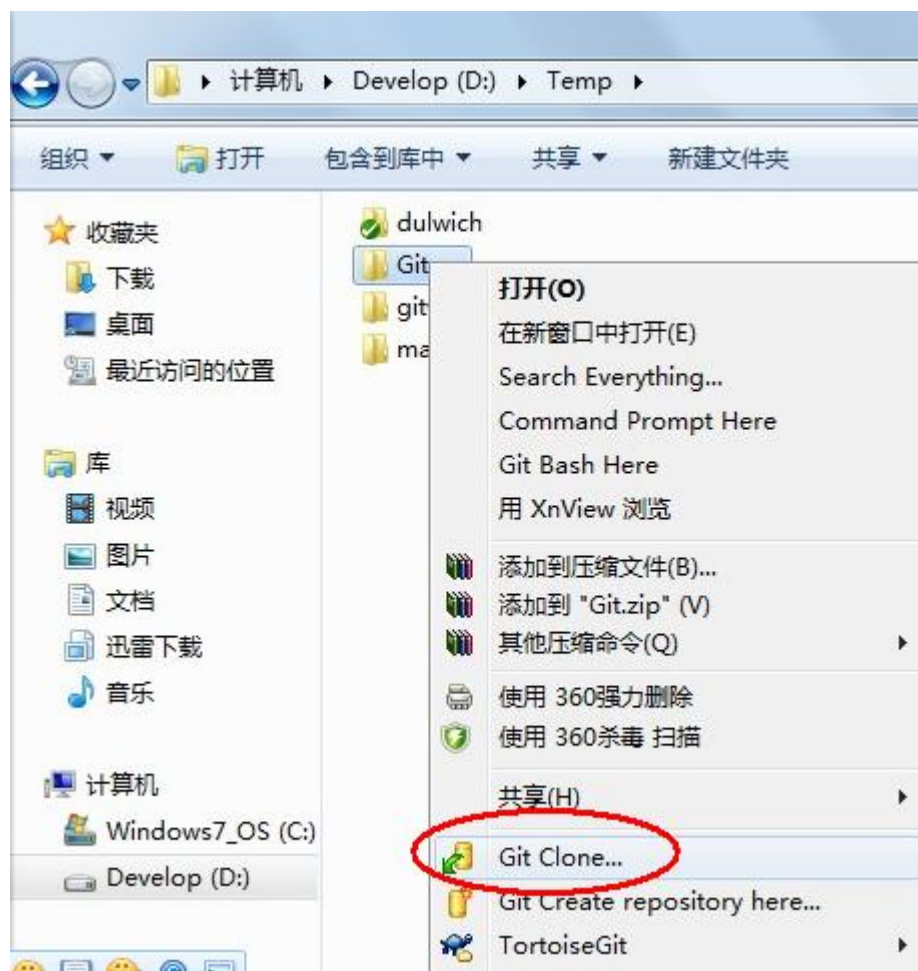
测试项目库名称：project1

本地测试目录：d:\temp\git

在前面建立的 c:\temp\Git 完成相关操作



在 c:\temp\git 目录上点击鼠标右键，选择 “Git Clone ”，



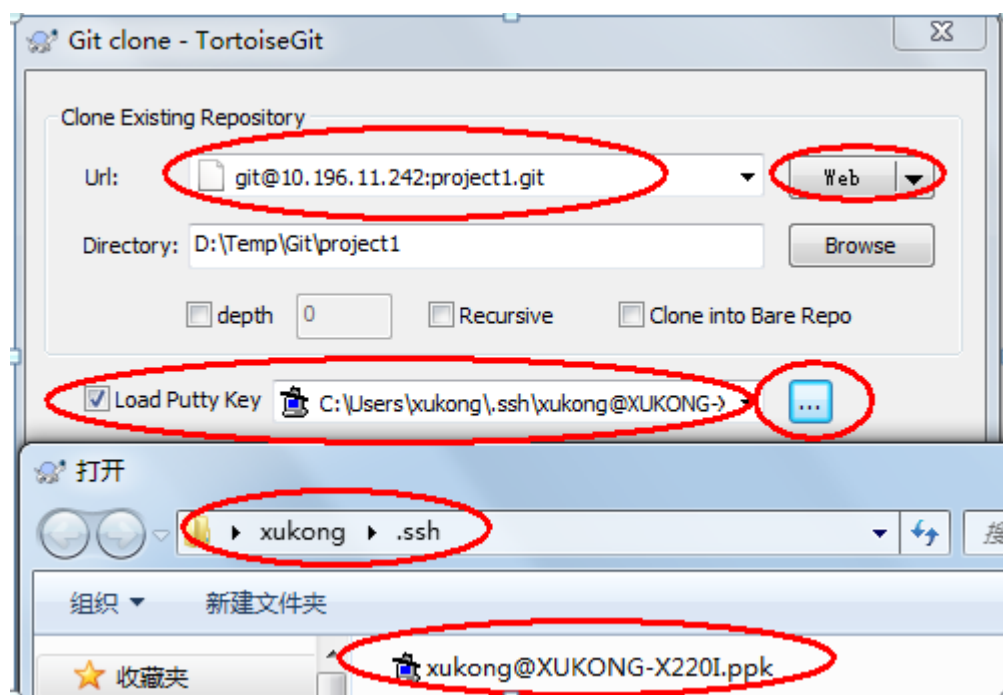
在出现的界面中按下图所示设置：

主要是

Url 输入：git@10.196.11.242:project1.git

选择 “Web”

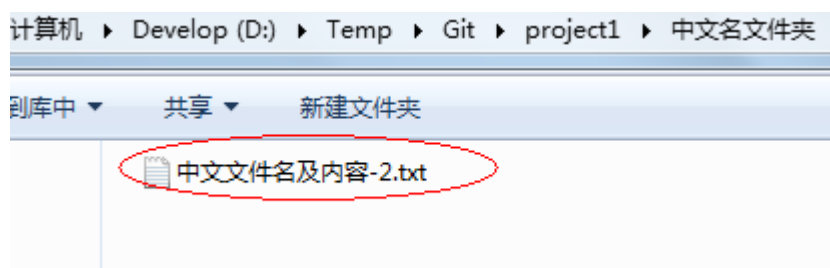
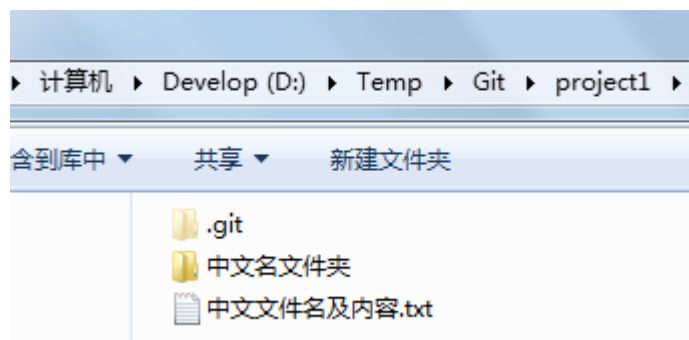
勾上 “Load Putty Key ” 并选中前面生成的 xukong@XUKONG-X220I.ppk 公钥文件。



点击按钮 “Ok ”完成签出

## 2 增加、修改、删除文件

签出后的 Project1 文件夹结构如下图所示：



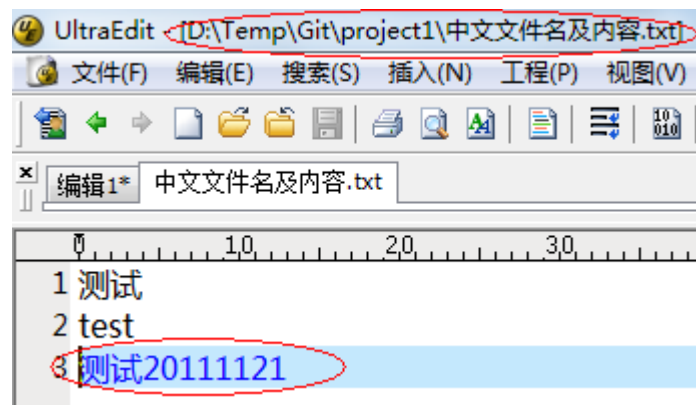
下面我们将完成如下工作：

### 1 删除“中文名文件夹”目录中的“中文文件名及内容-2.txt”文件

直接删除文件即可，在进行提交(commit)操作时 TortoiseGit 会扫描到相关变动。

## 2 修改 “project1” 下的 “中文文件名及内容.txt” 文件。

在文件最后加上一行 “测试 20111121”

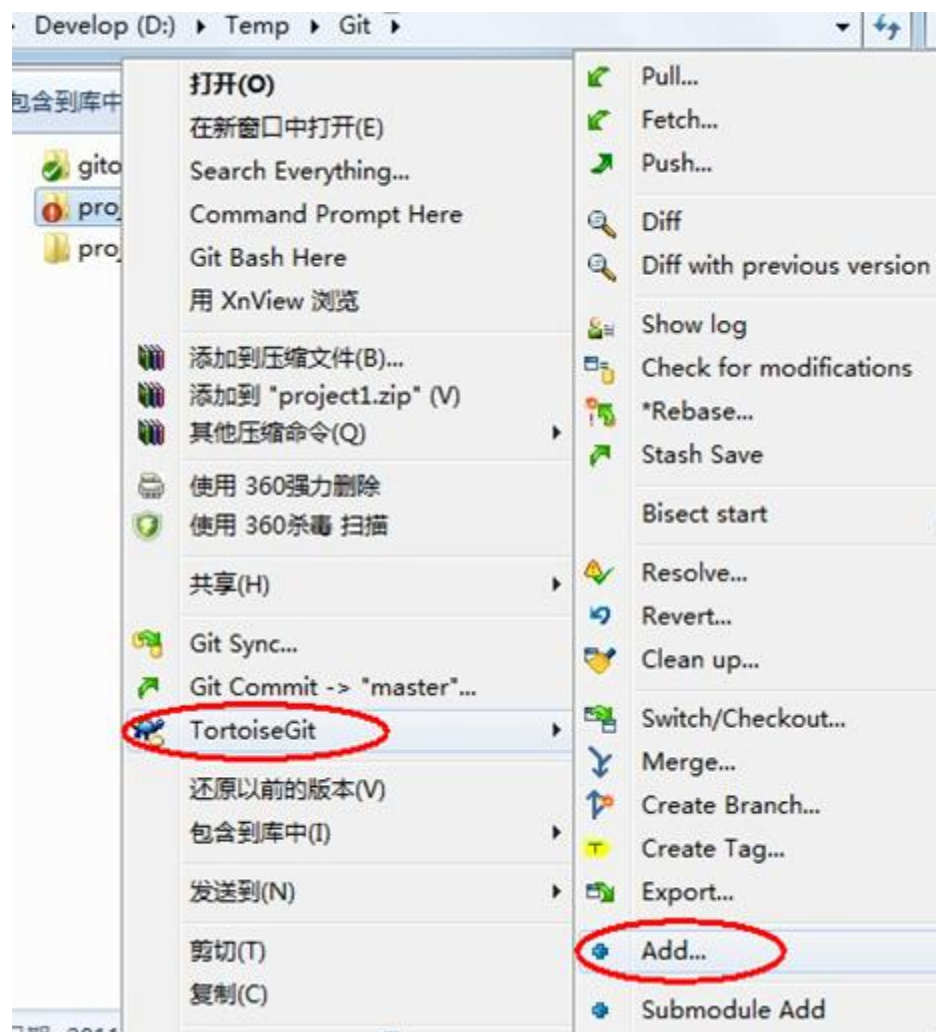


## 3 在 “中文名文件夹” 中新建一个文件 “中文文件名及内容-3.txt”。

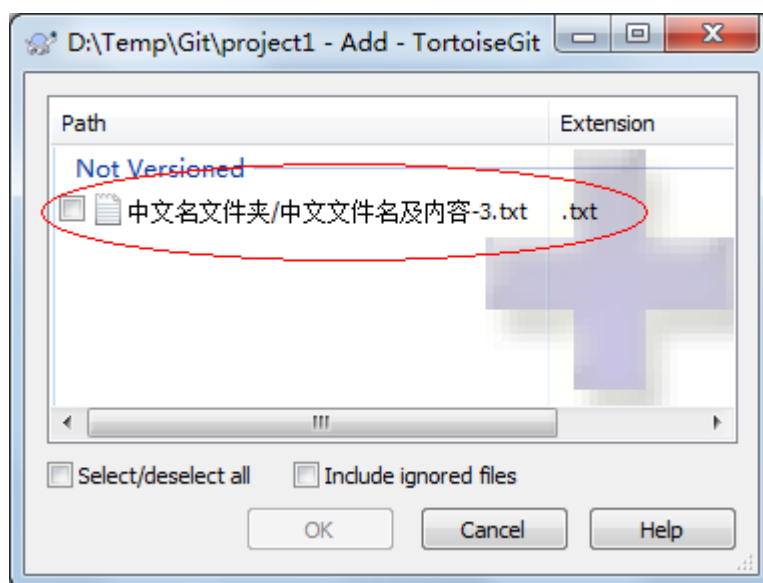
新建一个文件 “中文文件名及内容-3.txt”，在其中输入内容 “测试 3”。

因为新增了文件，需要告诉 GIT 我们已经新增了文件，在目录 “d:\temp\git\project1” 上点击鼠标右键，在弹出的菜单中选择执行 “ Add... ” 命令，如下图所示：

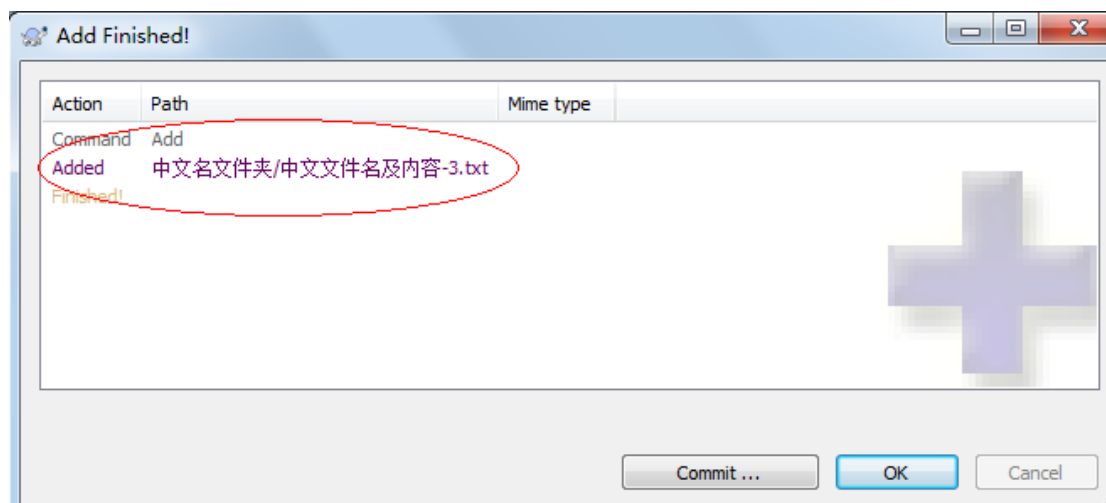




在弹出的窗口中可以看到已经列出了新增加的文件，如下图所示：



但是这时文件默认是没有选中，“OK”按钮不可用，因为 Git 不能确定新增的文件是否是我们需要的，这时我们就打勾选上并点击“OK”，确认是我们需要新增的文件，结果如下图所示：



虽然这里已经出现“COMMIT”按钮，但是一般我们新增文件后都要进行输入，修改等操作，这里就不执行 COMMIT 操作了。

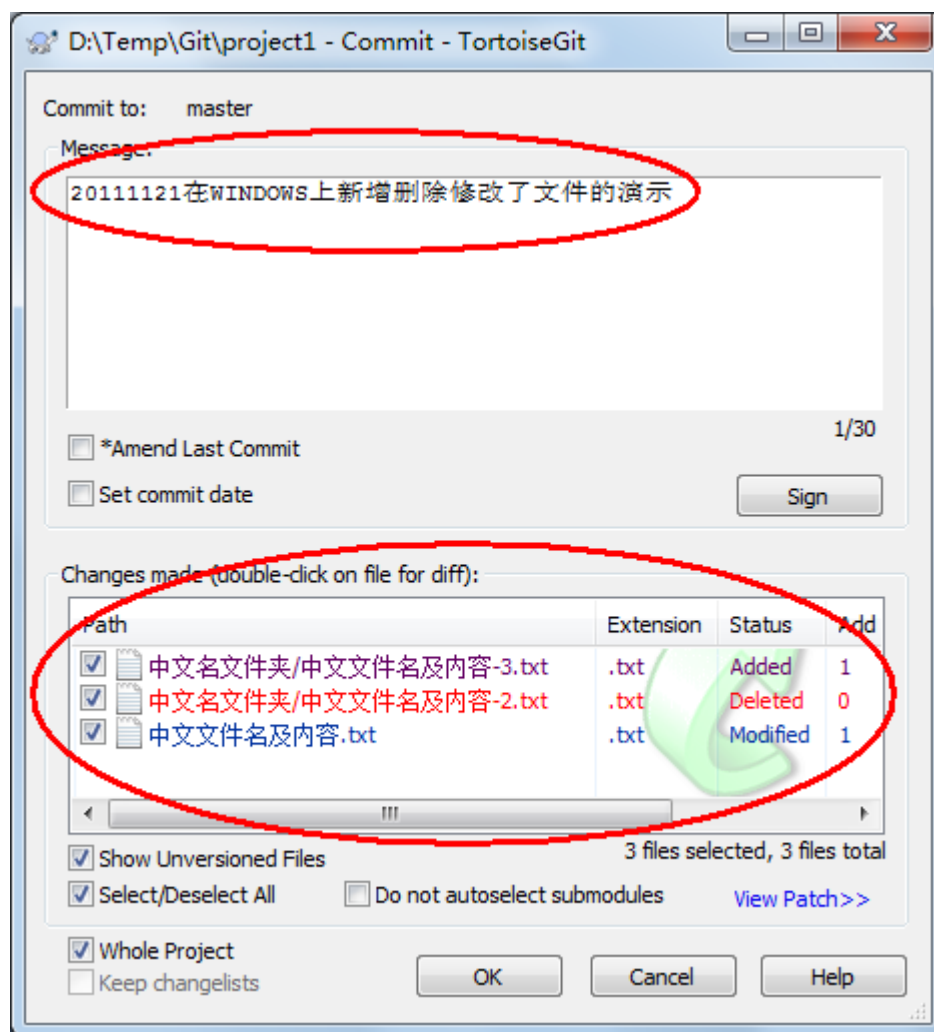
### 3 将更新提交 ( commit ) 到版本库

注意：由于 Git 的设计特性，Commit 只是在本地版本库中记录了此次提交操作，此时远程版本库，（即前面签出时使用的地址 `git@10.196.11.242:project1.git`）也就是 10.196.11.242 上的 `project1.git` 版本库后是没有任何数据保存进去的，需要执行后面的“将更新推送（Push）到版本库”才能将本地修改保存到远程版本库。

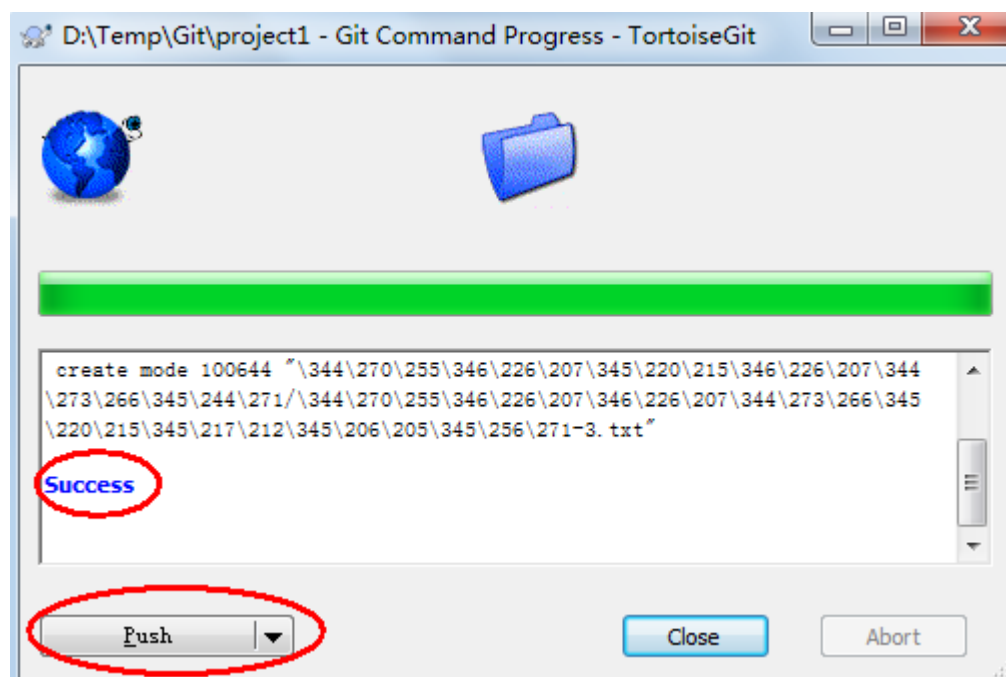
在确认已经完成某个阶段性的工作后，就可以将本地的更新同步到版本库中去了，在目录“`d:\temp\git\project1`”上点击鼠标右键，执行“Git Commit-> “master””，如下图所示：



可以看到 TortoiseGit 已经扫描到我们前面进行的新增删除修改操作了，在“Message”中输入我们这次提交的说明，注意，一定要养成输入提交说明的习惯，最好加上日期，提交人等信息。



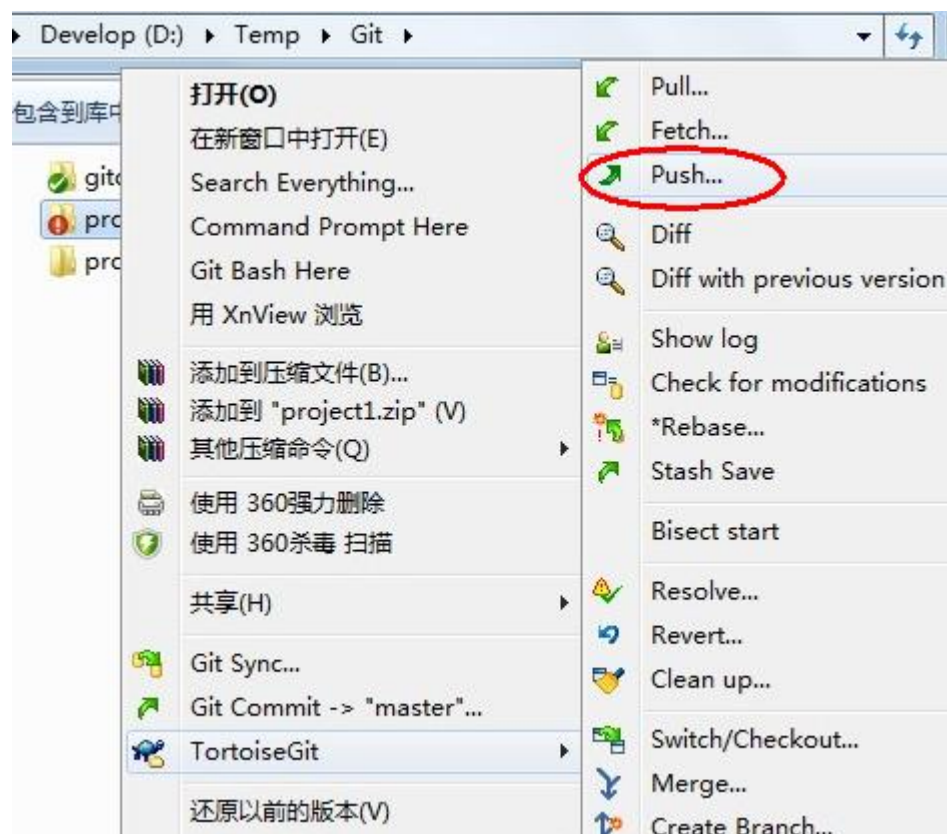
点 击 “ OK ” 按 钮 完 成 提 交 操 作 ，



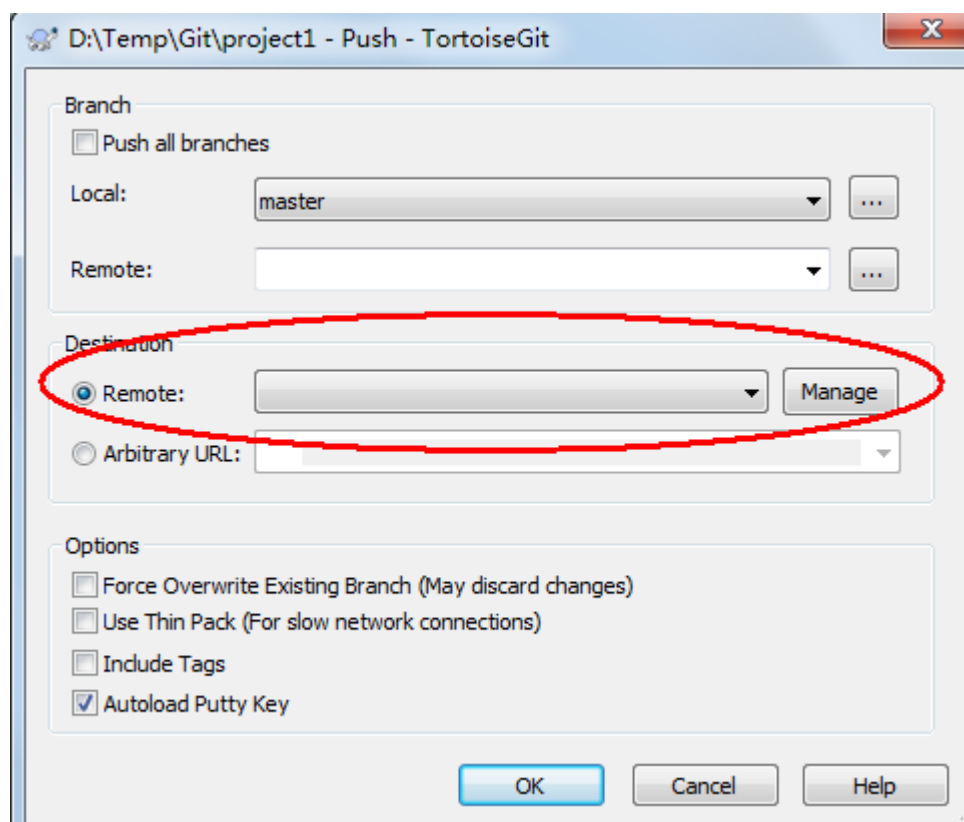
注：如果确认此次 COMMIT 的内容已经需要推送到远程版本库了，也可在此处点击“Push”按钮完成推送操作。

## 4 将更新推送 ( push ) 到版本库

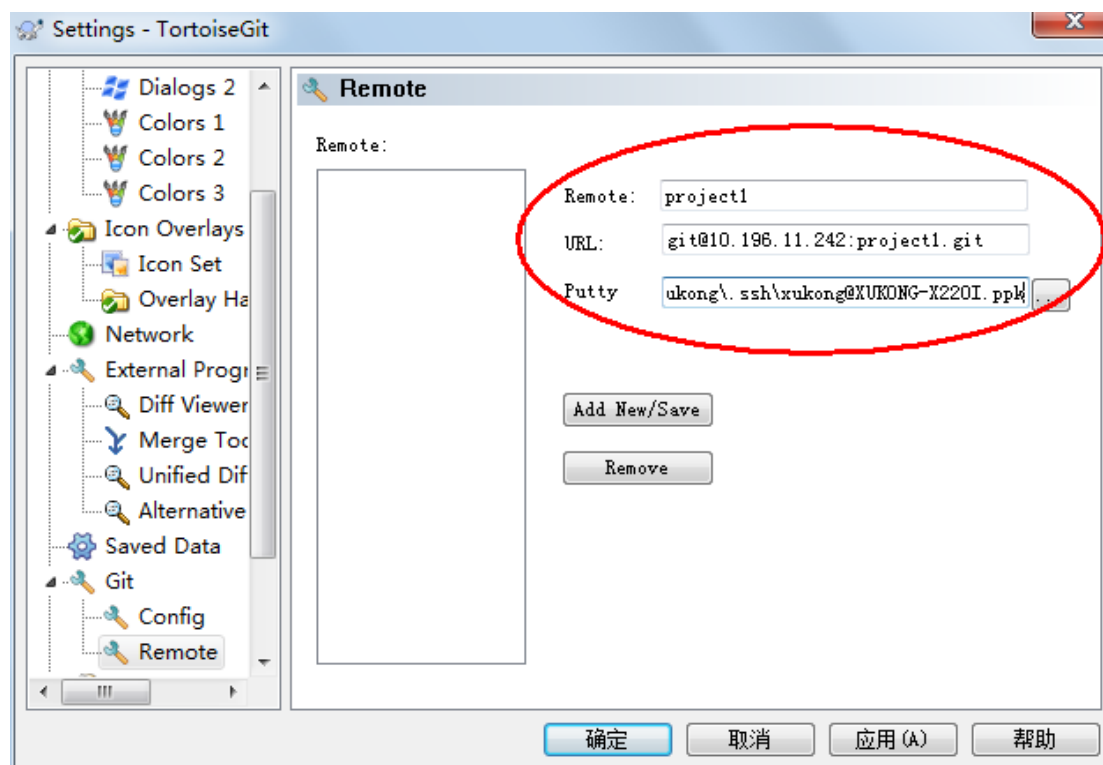
确认完成所有相关工作后，则可以将修改后的文件保存到远程版本库中了，这个操作称为“Push”（即推送）。在 project1 目录上点击鼠标右键，然后执行“Push”菜单，如下图所示：



执行“Push”后出现的界面如下图所示：



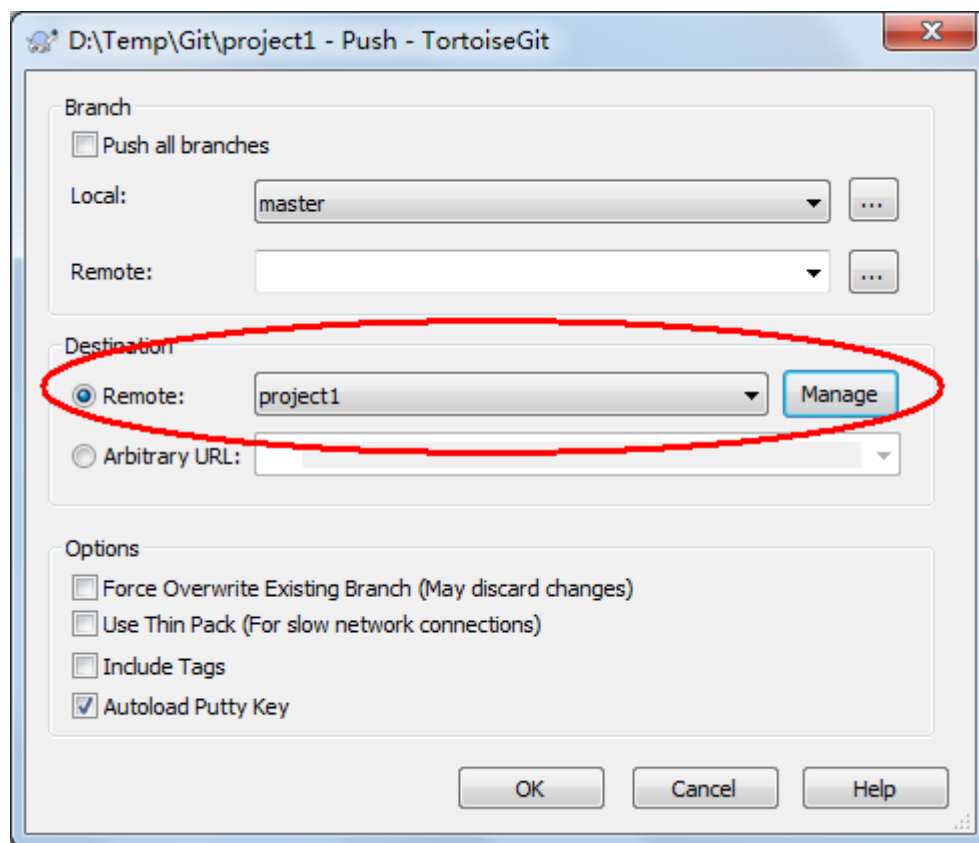
此时可能还没有远程版本库的相关信息，可自行添加，点击  
“Manage” 按钮，在出现的界面中添加 project1 的相关信息即可，  
如下图所示：



点击 “ Add New/Save ” 保存即可。

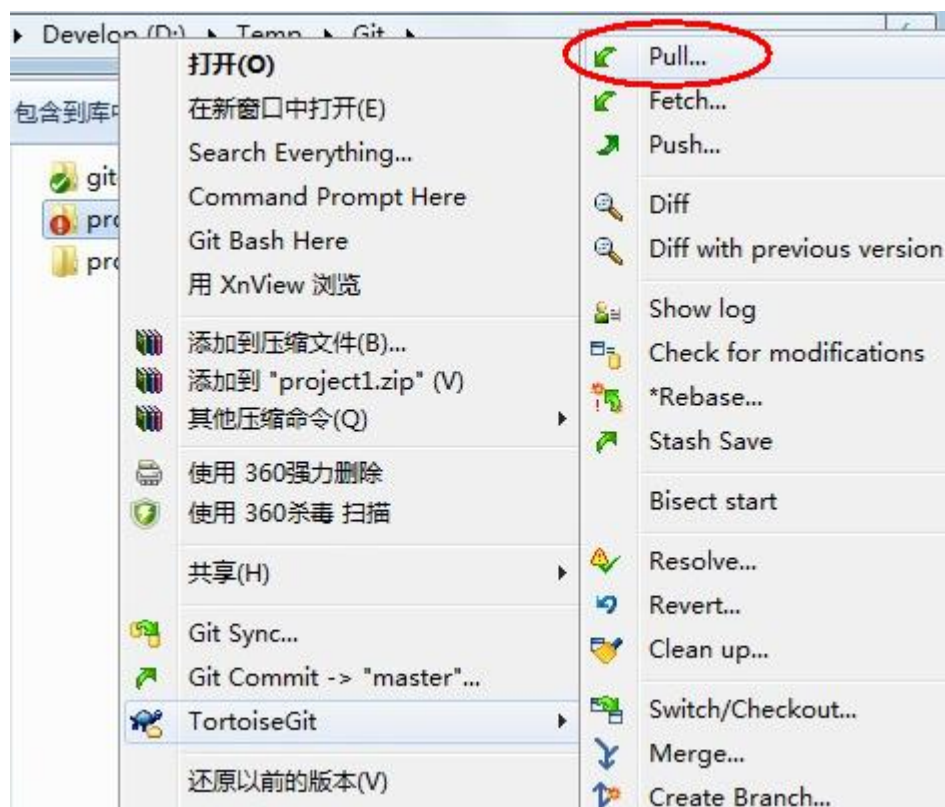
然后返回 Push 界面即可看到刚才添加的 project1 信息，选择后点击 “ok” 即可。



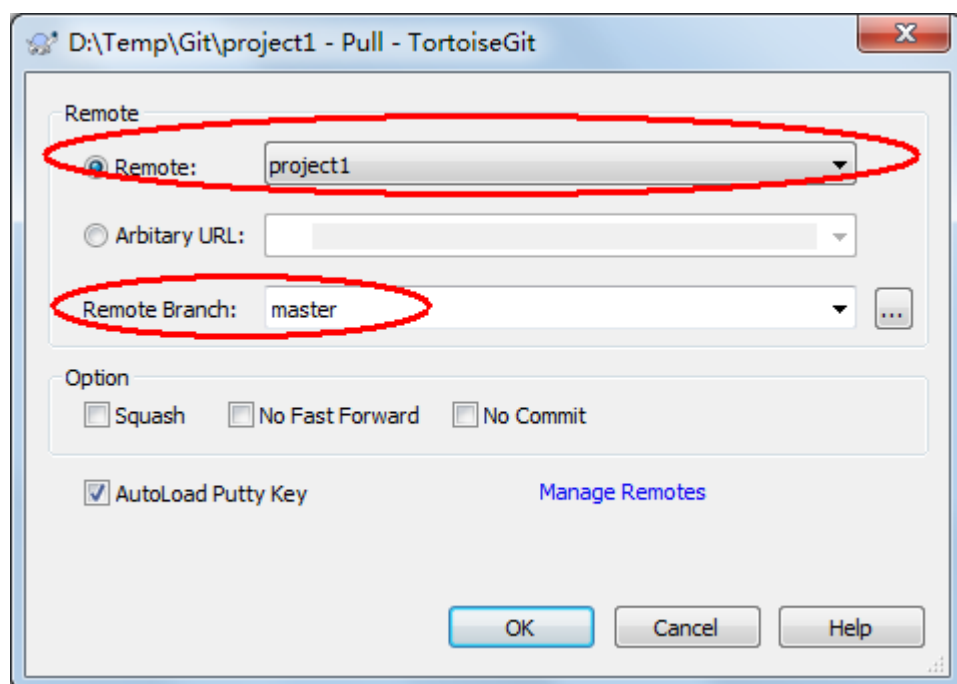


## 5 与远程版本库保持同步 ( Pull )

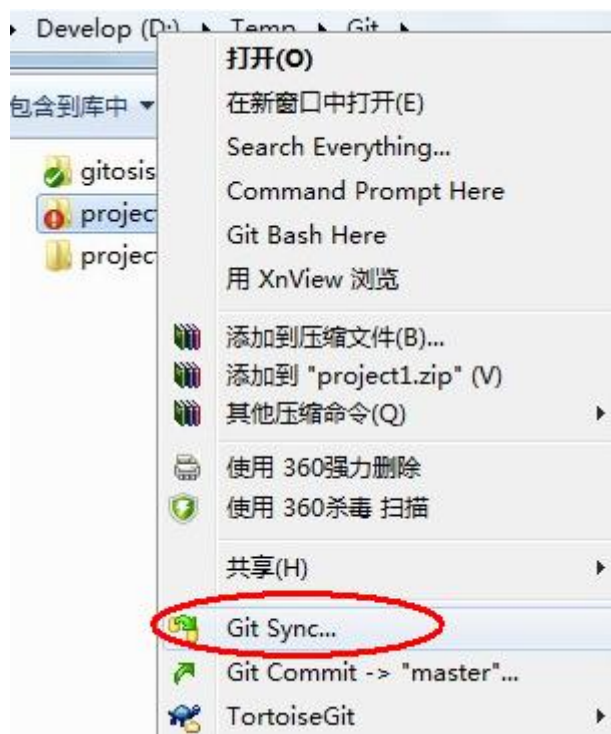
如果其他同事对版本库进行了更新，需要将他们的修改下载到本地版本库，可执行“pull”进行同步，如下图所示：



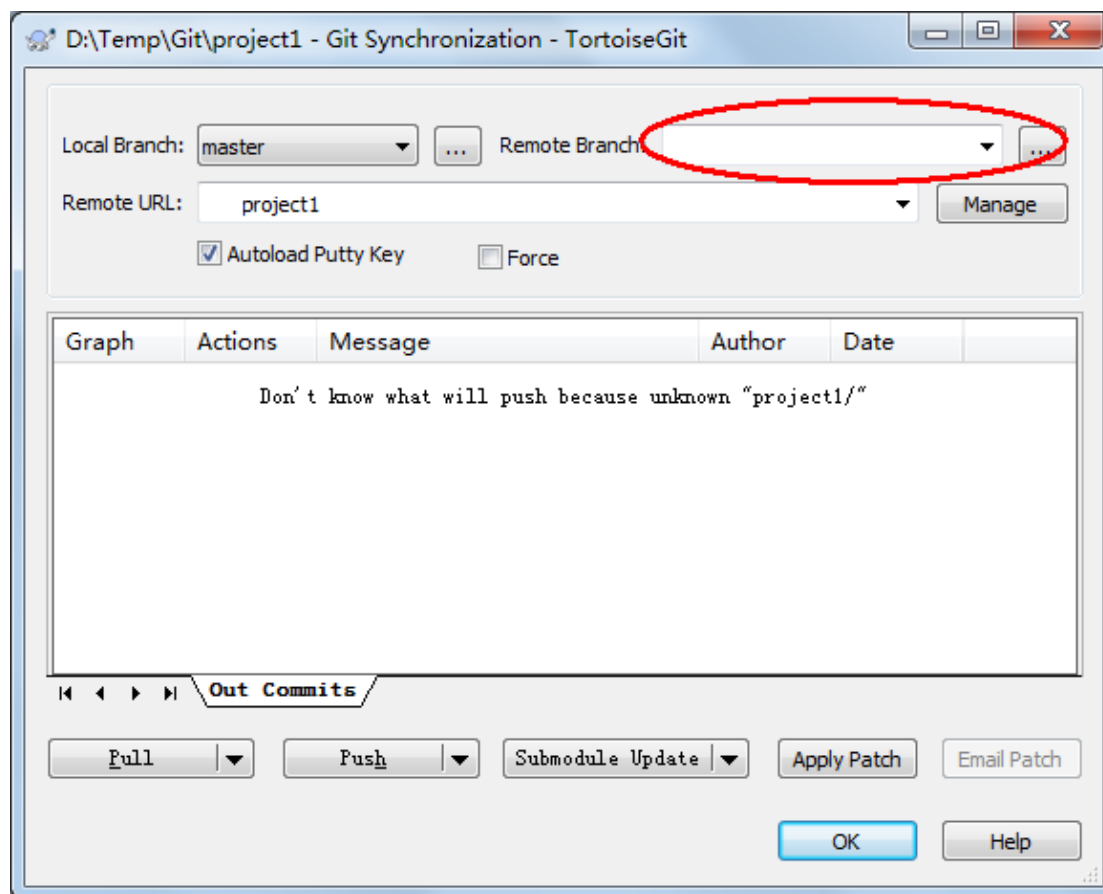
然后选择前面建立的远程版本库 “project1”，分支一般选择 “master”，点击 “OK” 即可。



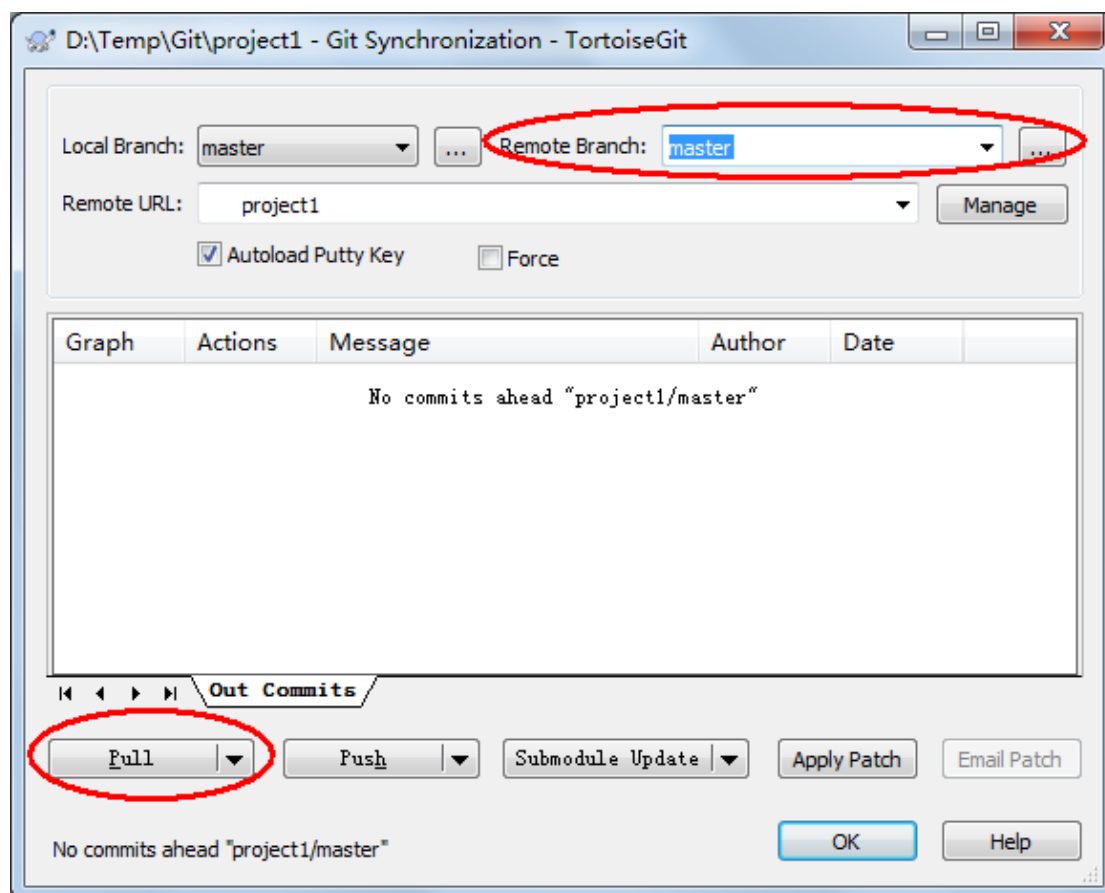
也可执行 “Git Sync...”



由于“Remote Branch”(远地分支)默认为空,一般选择为“master”



选中 “master” ( 主分支 ) 后 , 再点击 “pull” 按钮 , 即可完成同步。



## 6 查看版本库日志信息

