<http://hi.baidu.com/tp100/item/b48debf7f6fa8909d99e7253>

## TortoiseGit + GitHub 使用指南

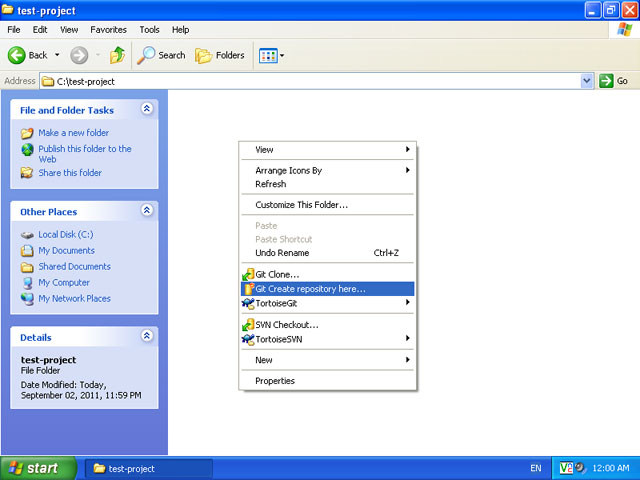
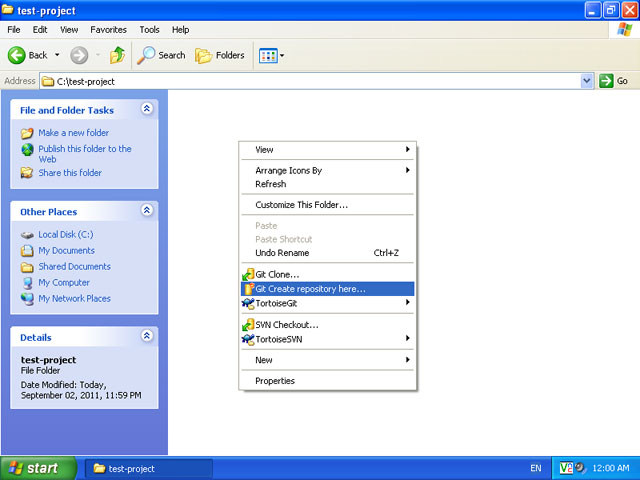
前言

对于周围很多研究程序设计的人不会使用版本控制系统(Version Control System)这点很无言，所以写这篇教学。本篇教学的主要观众是没用过版本控制系统或是没用过git的Windows使用者。如果这篇的回应良好，我下次再补上Linux上CLI界面的git指令使用入门教学。  
  
首先，为什么要用版本控制系统？版本控制系统有以下重要的功用：

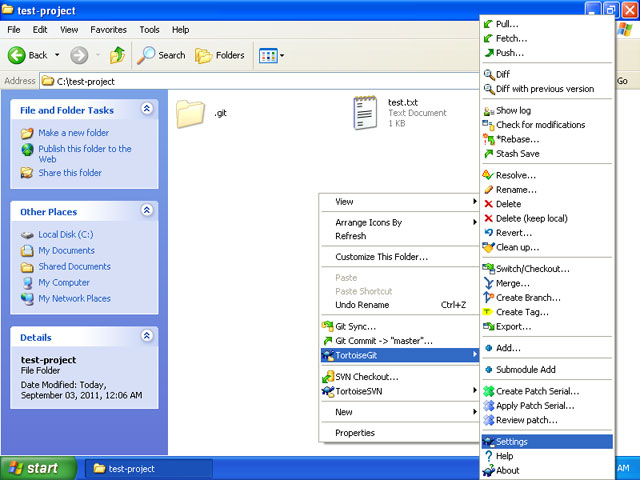
记录所有的原始码更改过程并作备份检视原始码的修改历史或复原旧的版本与人协力时可以有效率地同步修改的内容使用分支(Branch)来同时维护多个程序版本  
这些功用带来的好处非常多，其中包含：  
  
不用担心修改的东西会搞砸一个程序不需要手动备份各个版本不需要在原始码里使用注释(Comment)来保留没用到的语法在团队中与人分享原始码的时候不需要zip来zip去，然后e-mail来e-mail去 (备注1)  
因为这些优点，专业的程序开发人员没有不用版本控制系统的。所以，麻烦还不会使用版本控制系统的花点时间学习一下。  
  
\*备注1：没错，我对于这点非常无言。这篇教学贴出去以后，我会考虑拒收zip起来的原始码。

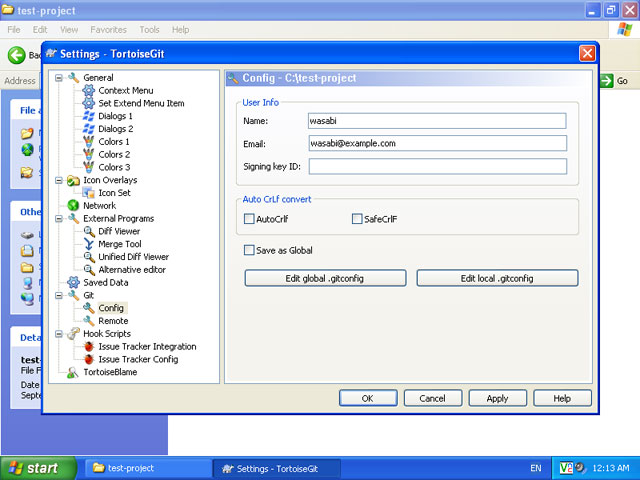
git简介  
  
git是由著名Linux内核(Kernel) 开发者Linus Torvalds为了便利维护Linux而开发的。git有以下几项特色：  
  
属于分散式版本控制系统(Distributed Version Control System)速度非常快，而且在传输档案时使用gzip来减小档案大小有非常多指令可以使用，甚至能与非git系统的版本控制系统互连  
细谈这些特色就属于中阶git教学的内容了，简介方面我先就此打住。

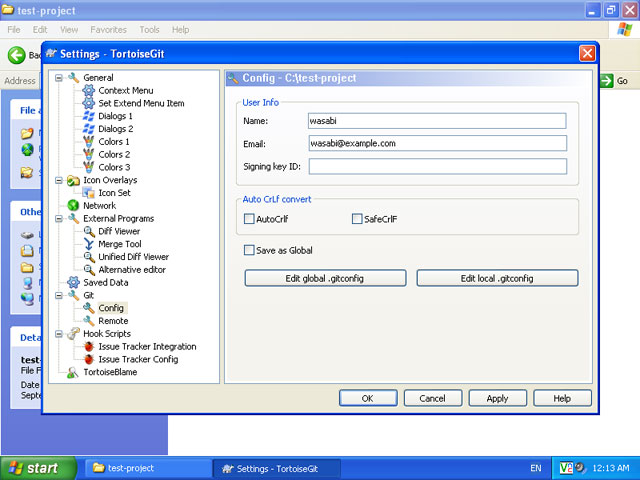
GitHub简介  
  
GitHub是个提供在线git空间的平台，能让多名开发人员很方便地进行协力软件开发。提供的方案包含免费和付费的账号，而其中的差异在于能否使用私人空间。虽然免费的账号只能使用公开的空间并且需要公布所有原始码，但对许多项目来说不是个问题。除此之外，免费的账号没有其他限制。  
  
使用GitHub的优点包含：  
  
拥有在线原始码和修改历史浏览，能让开发人员便利地分享原始码有人创立分支(Fork)时能够追踪所有的分支越来越多的知名企业在征才时会要求申请者提供GitHub账号，以用来观察在开源软件社群中的参与，还有程序写作风格和技巧等等 (备注2)  
所以，要参与开源软件开发或只是要单纯备份/集中整理自己的原始码时，GitHub是个非常好的平台。  
  
\*备注2：不过如果你的GitHub账号上的东西太空洞或潦草的话，还是不要给雇主知道，免得扣分。我也因此拒绝过一些有提供GitHub账号但是没有什么内容的求职者。  
  
TortoiseGit  
  
虽然说要用git指令才能完整地体验git的强大功能，TortoiseGit对于指令界面感到陌生的人可能会比较友善，也能够在功能和易用性上有个好的平衡。  
  
TortoiseGit可到此下载： <http://code.google.com/p/tortoisegit/downloads/list>。根据系统的类型，选择下载64bit或是32bit的版本；下载好以后安装然后重开机。  
  
重开好后，安装msysgit： <http://code.google.com/p/msysgit/downloads/list>。下载Full installer for official Git 1.7.6，或是更新的版本。装了这个以后TortoiseGit才能正常使用。  
  
建立git项目(Init)并且储存修改(Commit)



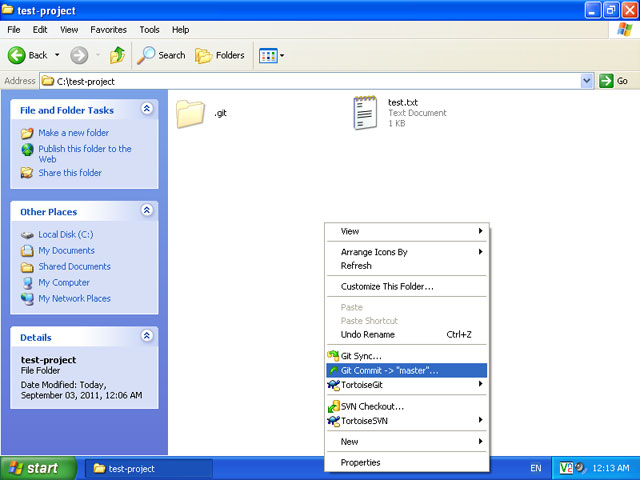
首先，在一个已存在的项目资料夹，或是一个新的资料夹，点右键并选择Git Create repository here…。这会建立一个新的git数据库，以便储存原始码资料。出现Git Init的对话框时，不要勾Make it Bare，并点OK。好了以后，如果有开啓显示隐藏档案，便可以看到一个.git的隐藏资料夹。



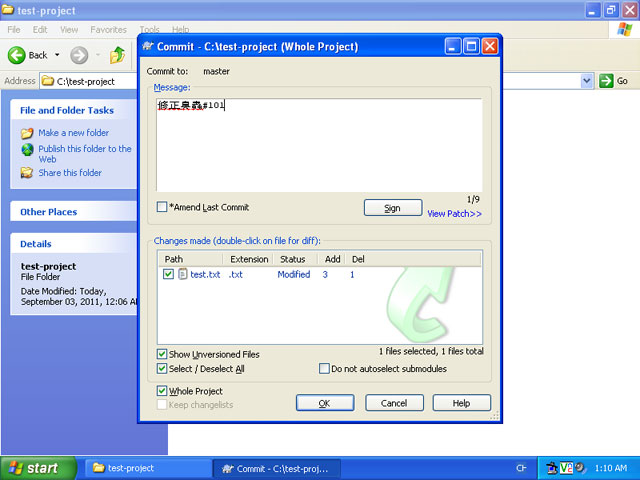




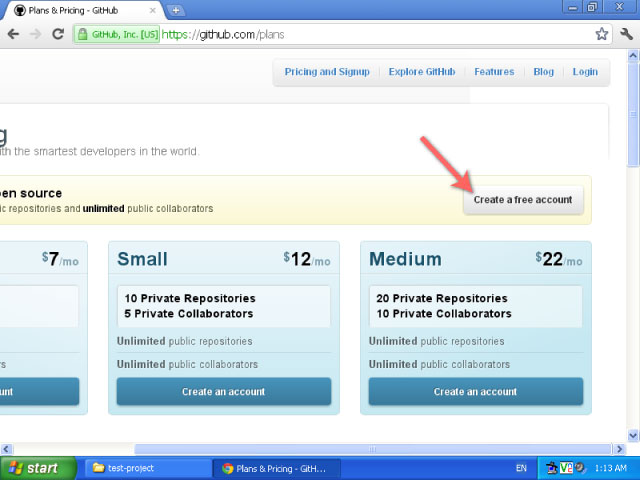
建立好git数据库以后，要设定作者信息。在同个资料夹内按右键，并选择TortoiseGit -> Settings。到Git -> Config页面，在Name输入你的名字，在Email输入你的E-mail。好了以后按OK 关闭。



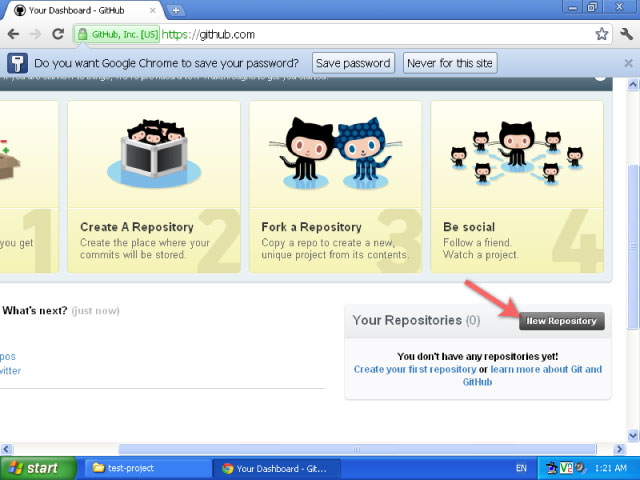
有原始码可以储存以后，在资料夹内按右键，并选择Git Commit -> "master"。master指的是当前的分支；如果本篇教学有续集的话，我会再作解释。



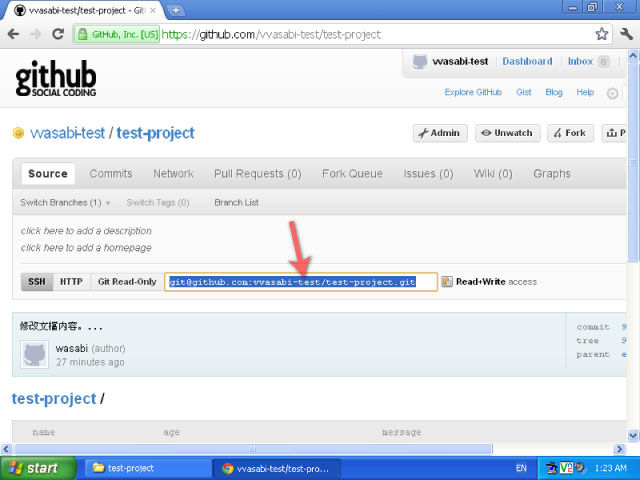
出现Commit视窗时，在Message中输入要储存变更的注释。好的开发者通常会在此大致叙述此次修改有新增或异动的内容。输入讯息后，勾取要储存的档案。要储存所有档案中的变更，勾Select / Deselect All。好了以后，点OK即可。等储存完成后，点Close来关闭视窗。  
  
建立**GitHub**账号



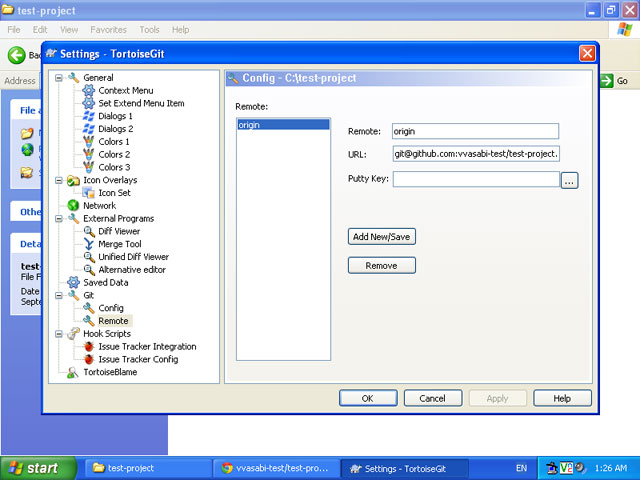
下一步骤便是建立一个GitHub账号： <https://github.com/plans>。如果要用免费方案，点Create a free account。输入使用者信息，并按Create an account。



账号建立好以后，点New Repository以建立一个源码库。输入项目名称后按Create Repository即可。

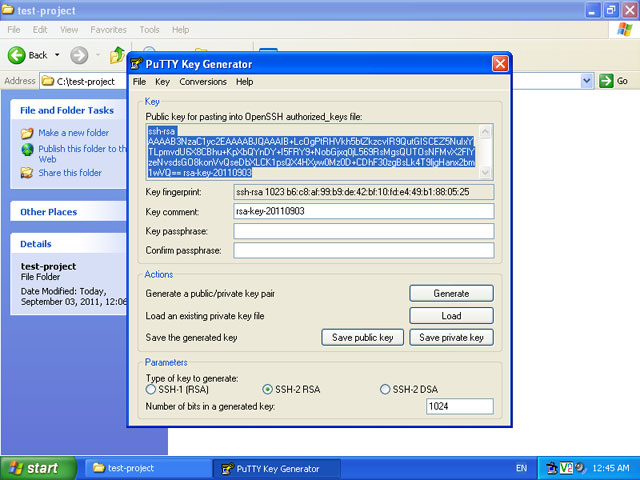


项目建立好以后，复制源码库的连结，如git@github.com:vvasabi-test/test-project.git。回到项目资料夹，点右键并选择TortoiseGit -> Settings。

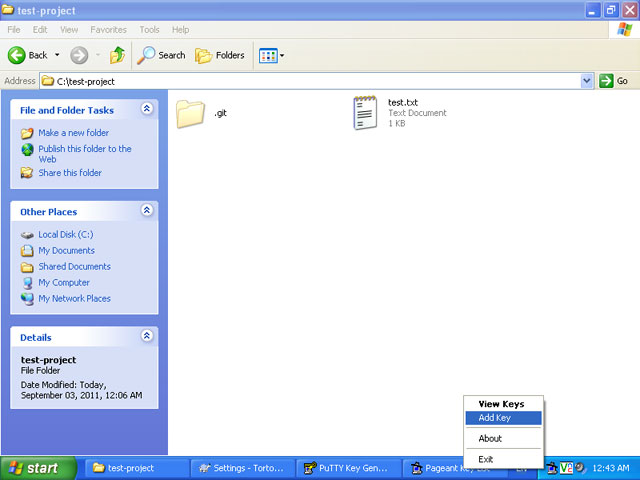


在这个视窗中，选择Git -> Remote，在Remote中输入origin，Url中贴上刚刚复制的连结然后点Add New/Save。好了以后按OK。  
这些步骤每个项目只需要做一次。

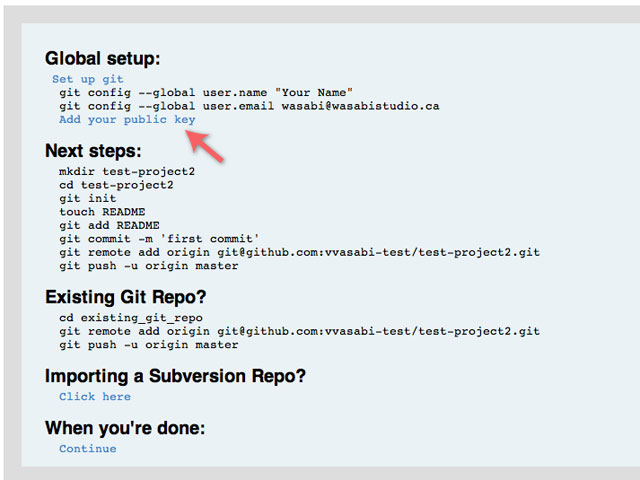
上传修改内容(Push)

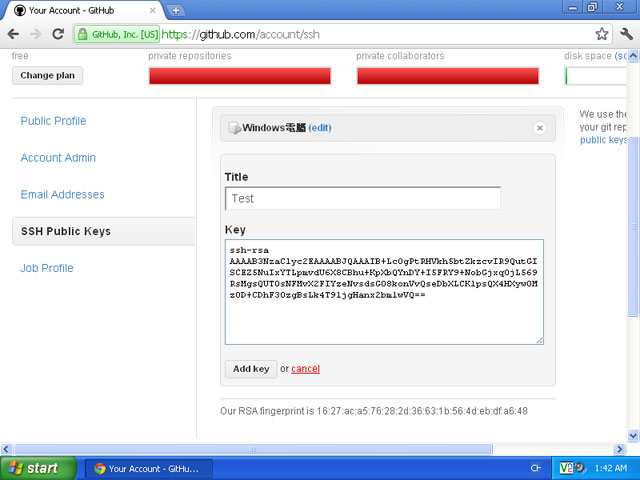


现在要把修改的内容上传到GitHub。首先要建立认证密钥，开启开始选单 -> 所有程序 -> TortoiseGit -> Puttygen。点Generate，等待程序跑完，然后点Save private key，将此密钥存到任意处。如果需要保障密钥不会被盗用，可在储存前输入Key passphrase来用密码保护密钥(同样的密码需在Confirm passphrase内输入一次)。如果没有输入密码，出现PuTTYgen Warning时点Yes即可。



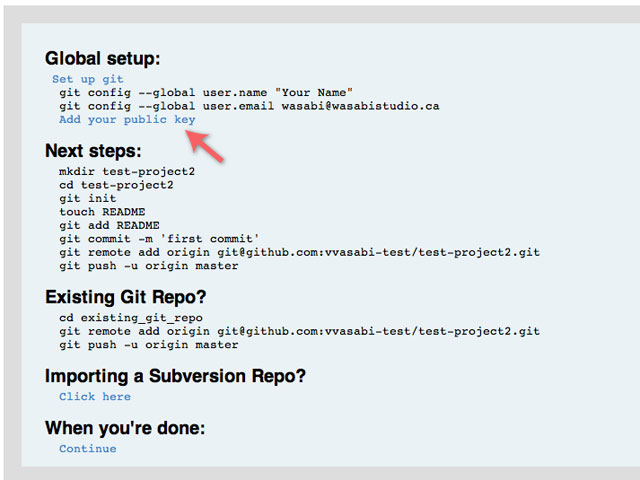
密钥储存完以后，回到开始选单，开启所有程序 -> TortoiseGit -> Pageant。这时状态列会跑出一个新图示，在这图示上点右键并选择Add Key，然后选择刚刚储存的密钥。

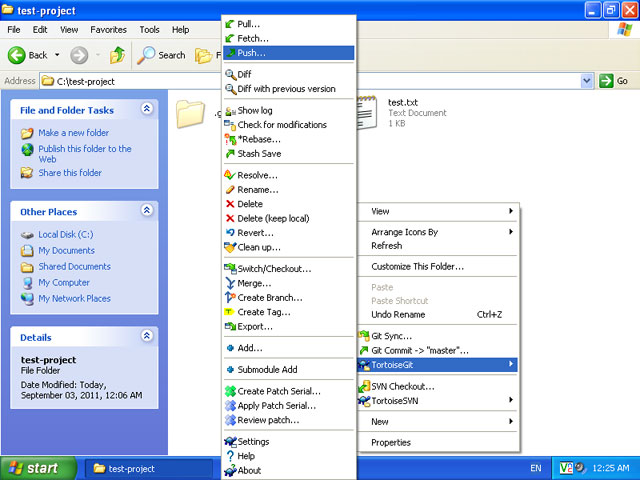




回到PuTTY Key Generator的视窗，并且选取和复制Public Key。开启GitHub网页，并点取Add your public key。点Add another public key，在Title栏输入随意名称，在Key栏贴上密钥，好了以后点Add key。

设定密钥的步骤虽然很麻烦，好消息是一台电脑只要设定一次。





回到项目资料夹，并按一右键，选择TortoiseGit -> Push…。勾Push all branches，然后点OK，出现PuTTY Security Alert时点Yes(这只会出现一次)。如果没有出现错误警告，这就代表项目已成功地上传到GitHub了。

平日工作流程  
  
GitHub一开始的设定很麻烦，懒得设定这些步骤的人也可以只使用git而不使用GitHub。不过，这些设定大多只要操作一次，所以最好可以有耐心地完成这些步骤。  
平常修改程序码时，在修改完以后需要储存变更(Commit)，在项目资料夹内点右键并点Git Commit -> "master"…。输入修改的讯息，选择要储存的档案并按OK。  
好了以后，点TortoiseGit -> Push…来将修改上传到GitHub。勾Push all branches，然后点OK。如果出现错误，请确定Pageant有开。没有的话，请参照上面的步骤来开启Pageant并加入先前储存的密钥。

结语  
虽然使用版本控制系统看起来很麻烦，但是带来的好处绝对值得花时间去学习。这篇教学由于是入门教学，先在此打住。如果本篇的回应良好，我会再找时间写续集。

/\* \*

\* Source：<http://forums.wasabistudio.ca/viewtopic.php?f=69&p=477010>  ([[教學]初心者git (TortoiseGit) + GitHub使用教學](http://forums.wasabistudio.ca/viewtopic.php?f=69&p=477010#p476727))

\* read more：

\*    <http://intl-bcds.iteye.com/blog/652594> （[git的基本用法](http://intl-bcds.iteye.com/blog/652594)）

\*    <http://www.cnblogs.com/xiaoxuanzi/archive/2011/07/30/SCM.html> （[TortoiseGit 使用快速指南: 利用GitHub做Git Server](http://www.cnblogs.com/xiaoxuanzi/archive/2011/07/30/SCM.html)）

\*   <http://hi.baidu.com/eehuang/blog/item/37af8d54242d6351564e00b5.html> （Git入门教程）

\*  <http://help.github.com/win-set-up-git/>  （github：help）

\*  <https://github.com/multitheftauto/multitheftauto/wiki/how-to-use-tortoisegit> （How to use TortoiseGit）

\* <http://hi.baidu.com/einyboy/blog/item/614d547bb7b228e62e73b30d.html>  (Cygwin使用方法)

 \*/