

Windows 系统 Git 和 TFS 的结合配置和使用简明教程

吴睿 wurui7@mail3.sysu.edu.cn

2020.09.17

目录

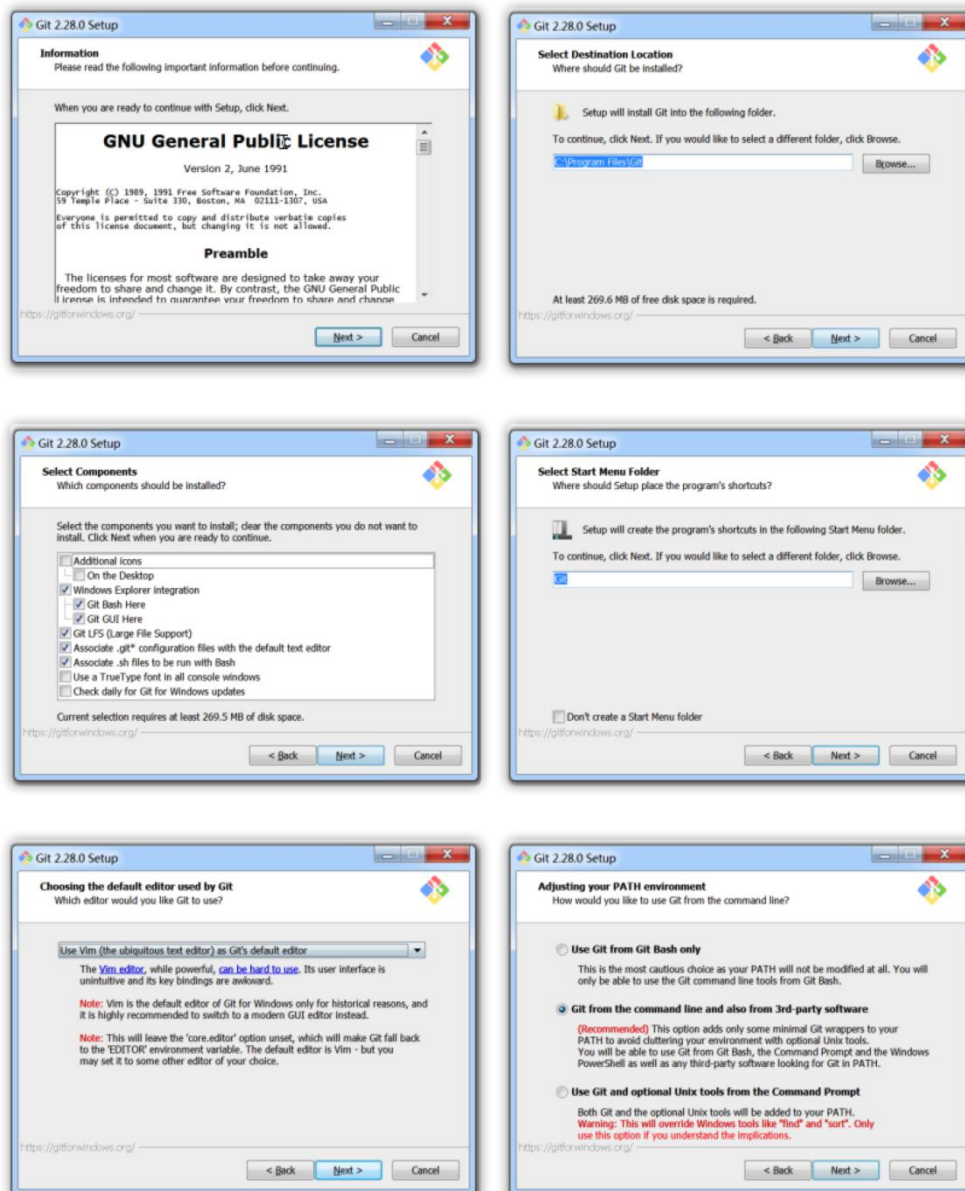
1. 基本概念.....	2
2. 安装 Git 客户端并配置 SSH 密钥.....	2
3. 克隆 Clone 项目到本地.....	6
4. Git 基本工作流程概念.....	7
5. TortoiseGit+TFS 基本操作.....	7
6. 参考资料.....	14

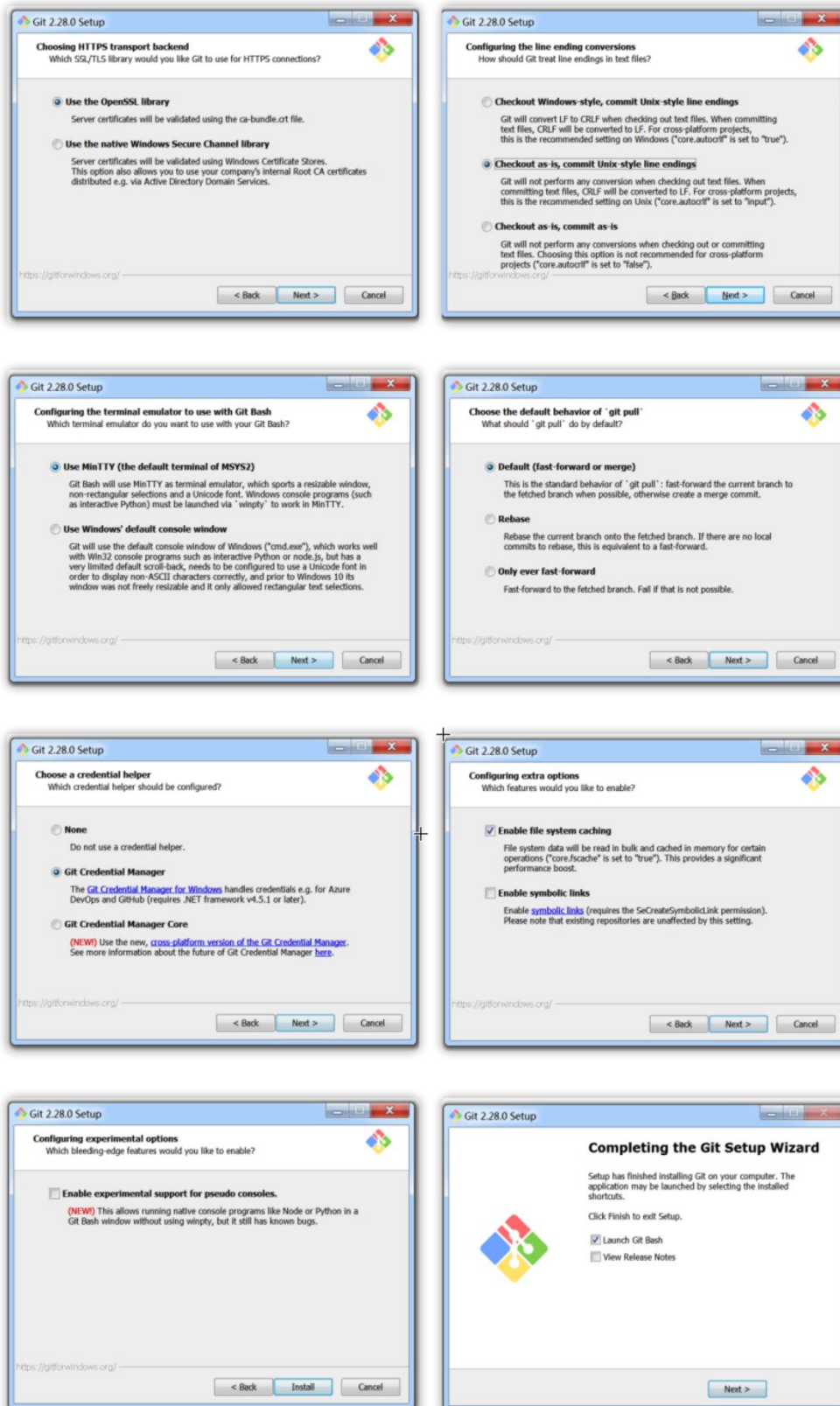
1. 基本概念

- a) 版本控制系统(Version Control System, VCS)是一种记录一个或若干文件内容变化, 以便将来查阅特定版本修订情况的系统
- b) Git 是目前最流行的版本控制系统, Github、Gitee、TFS (Team Foundation Server, 现改名为 Azure DevOps Server) 均使用了 Git
- c) TFS 是微软出品的协作式软件开发工具, 可以使跨职能团队有效处理各种规模的项目
- d) 注意: 各类目录路径尽量不要出现中文, 下面示例中, 目录路径有出现中文, 是错误示例

2. 安装 Git 客户端并配置 SSH 密钥

- a) Windows7 要先安装 [.NET Framework 4.5](#)
- b) 打开[网页](#)下载安装包, 安装勾选项请跟随下方图片, 注意勾选 OpenSSL 和 Git Credential Manager





c) 按照上面步骤安装 Git 之后，会打开 Git Bash，做法参考下面图片，注意把用户名和邮箱替换成自己的用户名和邮箱

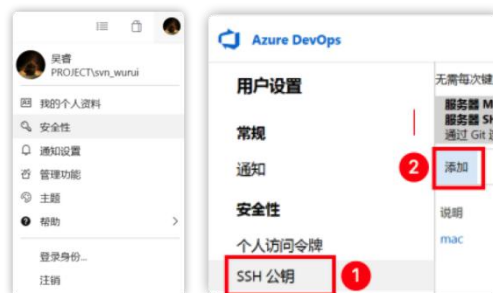
```
MINGW64~/c/Users/吴睿
吴睿@WIN-6S84G6FJ043 MINGW64 ~
$ git config --global user.name "wurui" 全局设置用户名

吴睿@WIN-6S84G6FJ043 MINGW64 ~
$ git config --global user.email "wurui7@mail3.sysu.edu.cn" 全局设置邮箱

吴睿@WIN-6S84G6FJ043 MINGW64 ~
$ ssh-keygen 输入这行命令并回车，开始生成 SSH 密钥
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/吴睿/.ssh/id_rsa): 直接回车
Enter passphrase (empty for no passphrase): 输入密码或直接回车，这里直接回车
Enter same passphrase again: 同上
Your identification has been saved in /c/Users/吴睿/.ssh/id_rsa
Your public key has been saved in /c/Users/吴睿/.ssh/id_rsa.pub
The key's fingerprint is:
SHA256:UIDKotYs7rJkv+YEPVXewV6vOIJh8o9sJK8giia/ise 吴睿@WIN-6S84G6FJ043
The key's randomart image is:
+---[RSA 3072]-----+
| ..oo. |
| . o..... |
| . . . . . |
| .+.o. . |
| .ooo+ o S . |
| .o.+ + . o |
| E+. = o . |
| X=+o = . |
| X+*+*+ |
+---[SHA256]-----+ 生成 SSH 密钥成功

吴睿@WIN-6S84G6FJ043 MINGW64 ~
$ clip < ~/.ssh/id_rsa.pub 复制 SSH 公钥
```

d) 打开 [IFS](#)，点击右上角个人头像，选择“安全性”，然后选择“SSH 公钥”，点击“添加”并添加 SSH 公钥



e) 安装 TortoiseGit

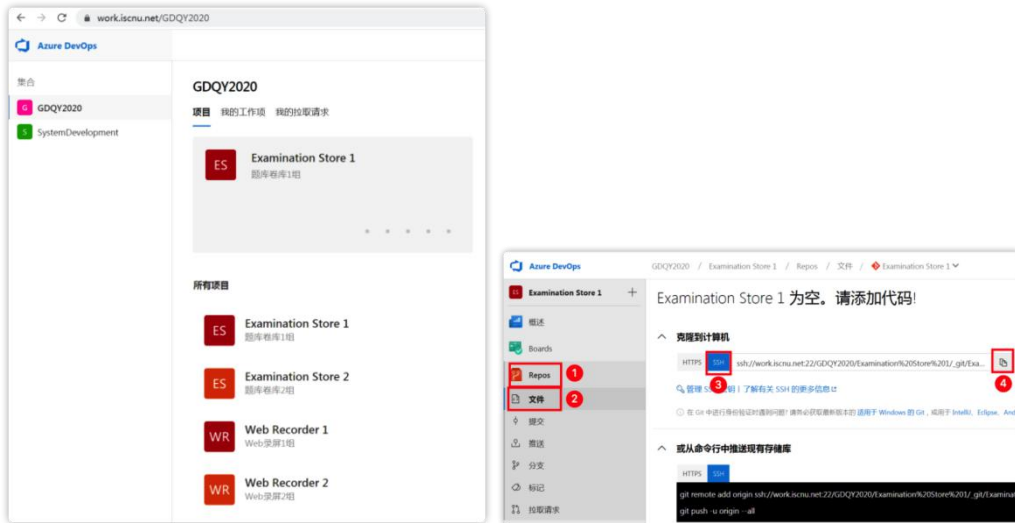
打开 [网页](#) 手动下载安装包和语言包，安装项勾选参考下方图片，注意勾选 OpenSSH



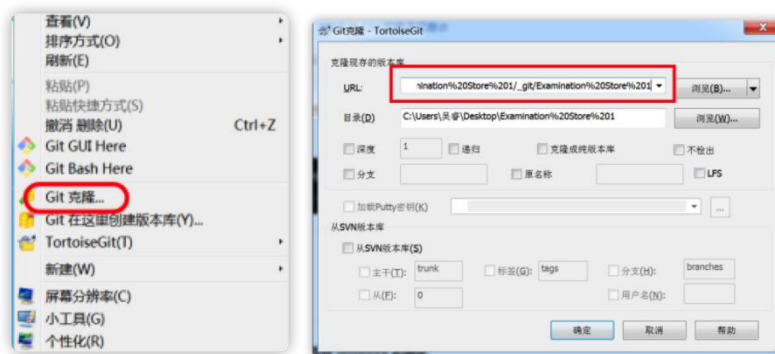


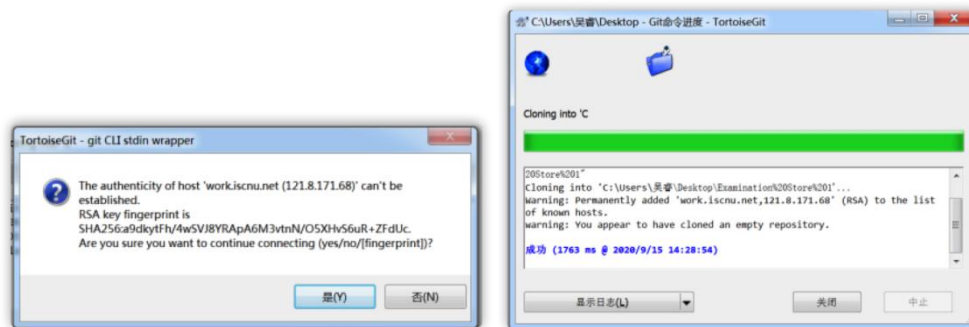
3. 克隆 Clone 项目到本地

a) 选择项目，获取项目的 Git 地址（注意是 SSH 方式的 Git 地址）



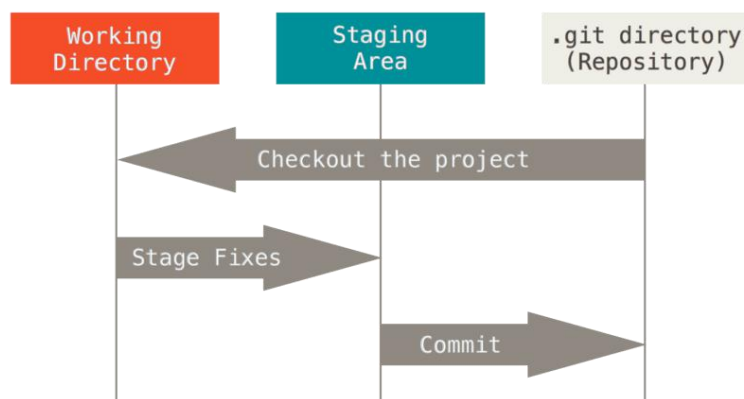
b) 在你想要放项目的地方右键选择“Git 克隆”，然后在 URL 一栏填入刚才获取的 Git 地址，点击“确定”，第一次会出现提示，选择“是”，如果一切顺利，将会显示“成功”



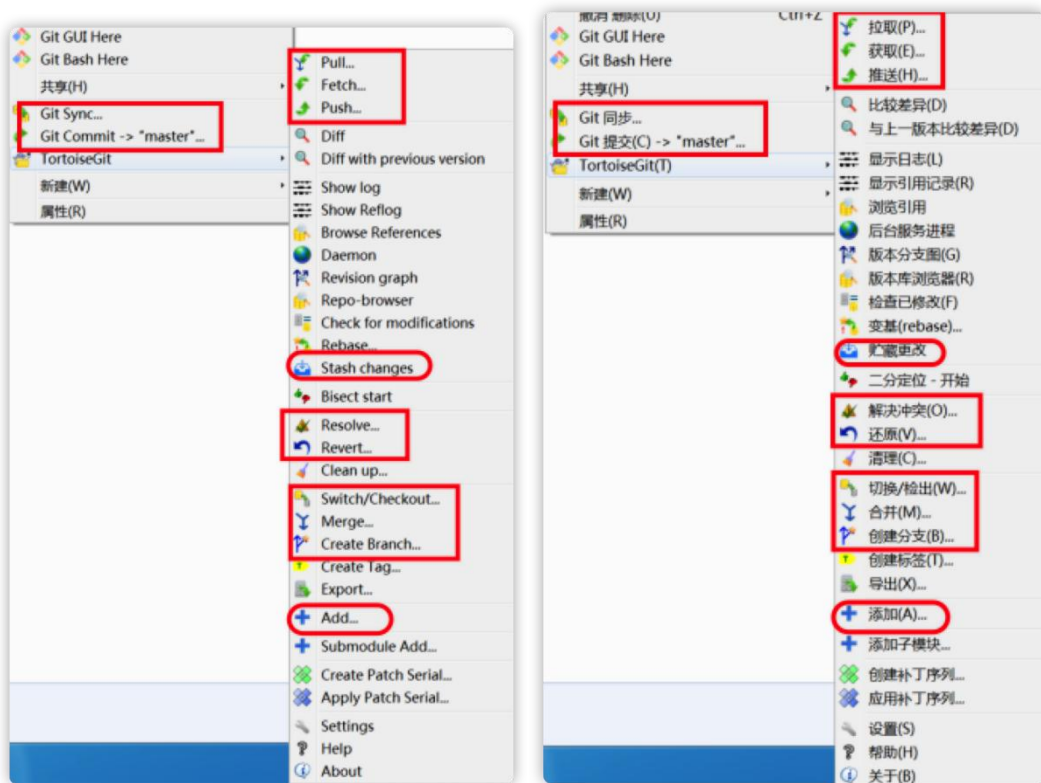


4. Git 基本工作流程概念

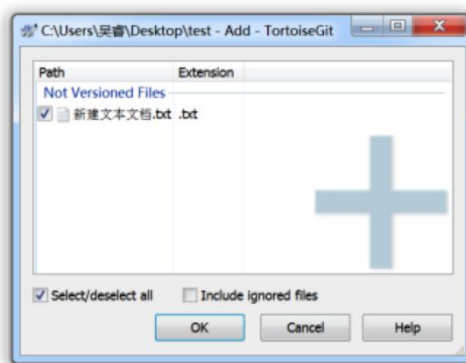
- 克隆远程 Git 仓库.git directory(Repository)到本地，得到工作区 Working directory，也就是项目目录。
- 修改项目目录内文件。
- 将想要提交的更改暂存，这样会将更改的部分添加到暂存区 Staging Area。
- 提交更改到远程 Git 仓库。
- 实际上，Git 内部要比上面描述的更复杂，上面描述的流程做了适当的简化，下面的图可以帮助理解基本工作流程



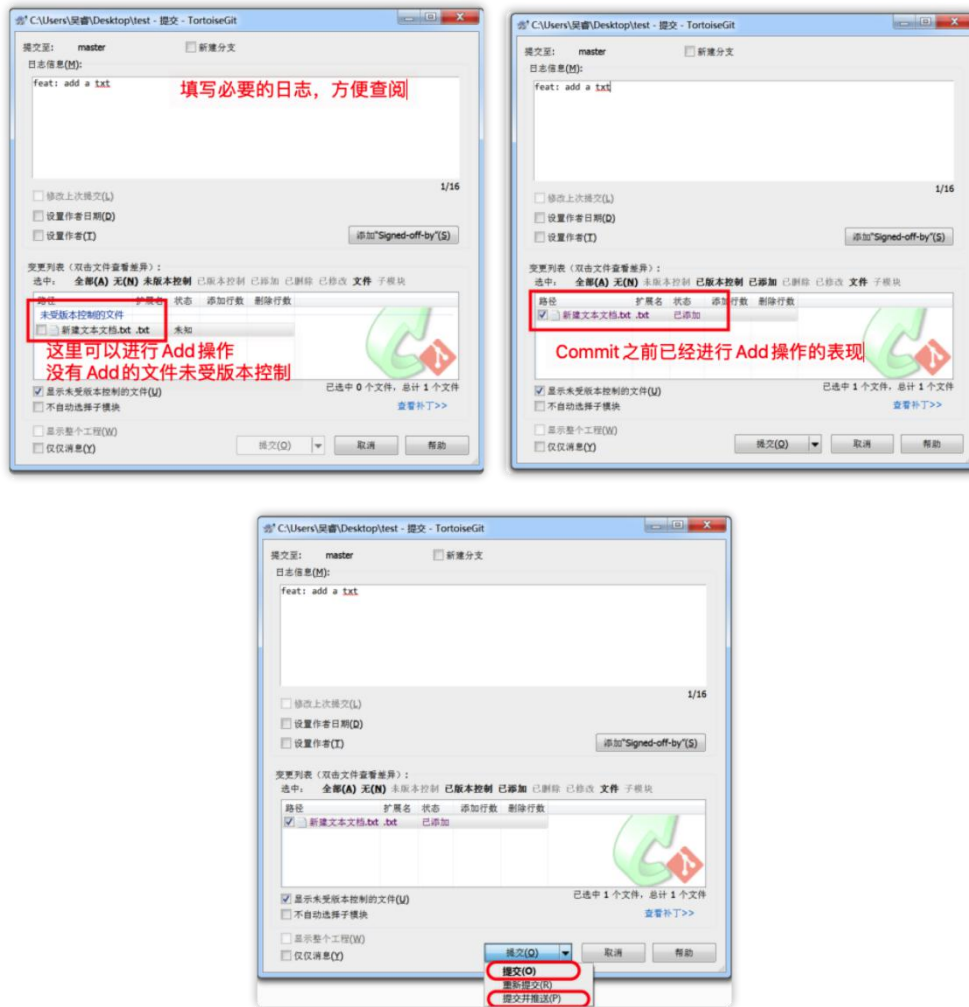
5. TortoiseGit+TFS 基本操作



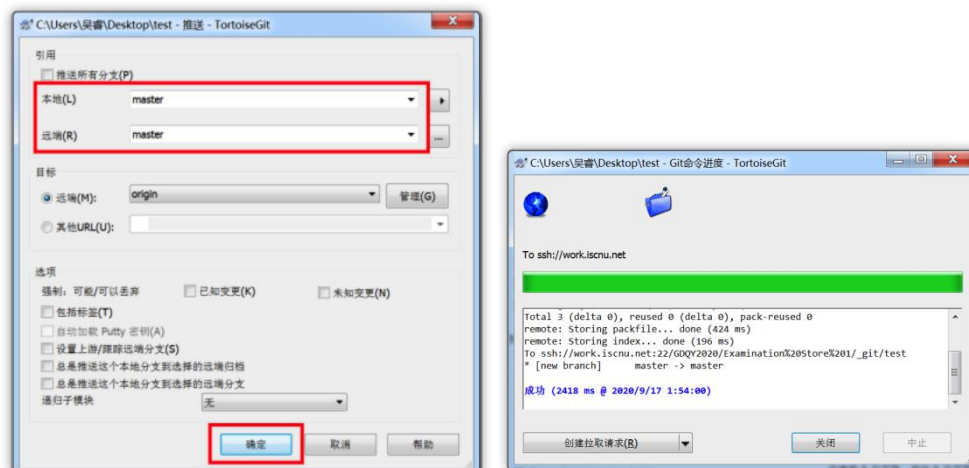
a) 添加 Add: 把更改添加到工作区，这里我新建了一个 txt 文件并写入了 123 做演示



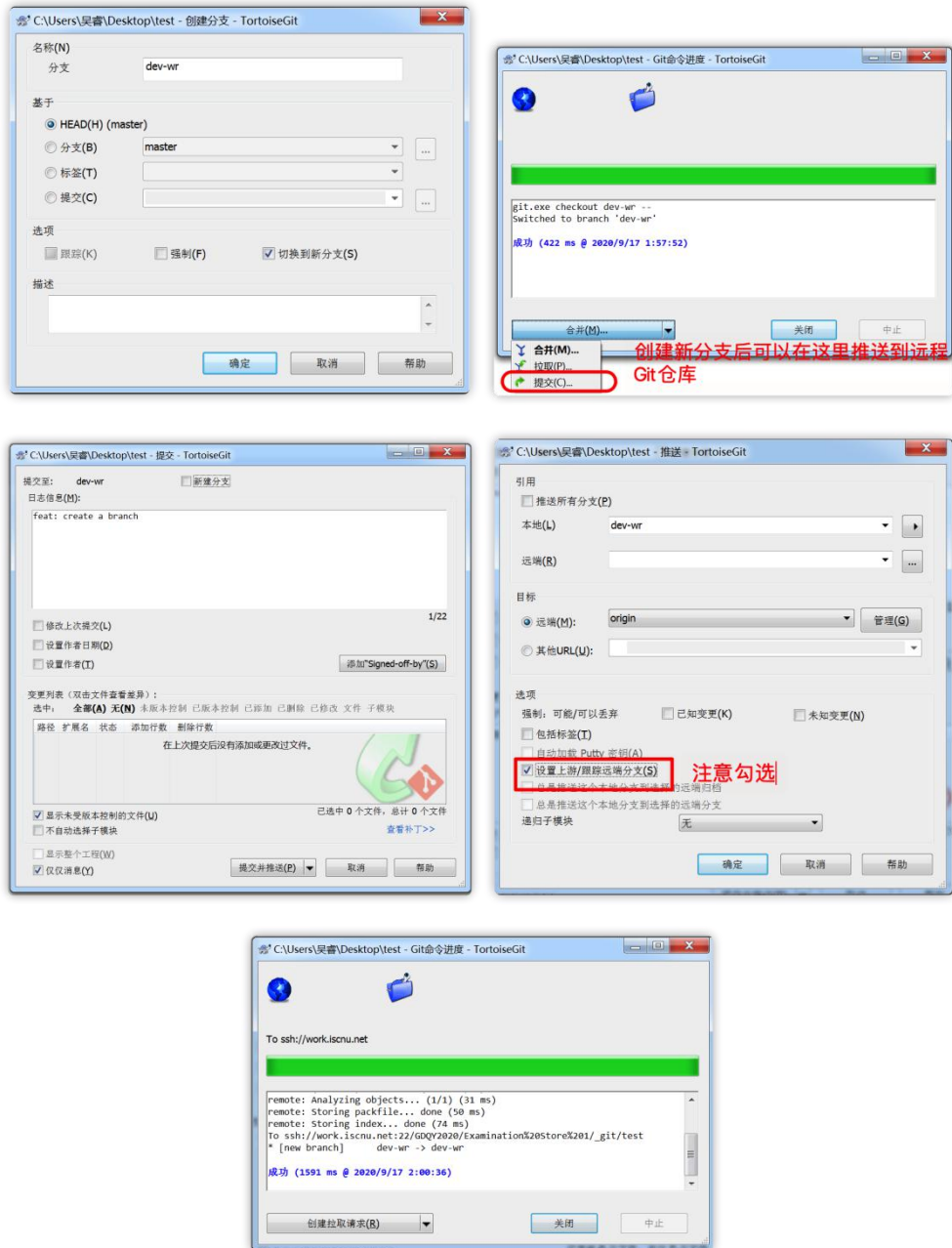
b) 提交 Commit: 把工作区更改添加到暂存区，TortoiseGit 提交 Commit 时也可以处理添加 Add 和推送 Push，注意填写便于理解的日志 Message



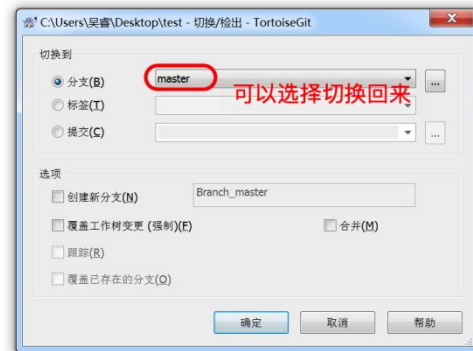
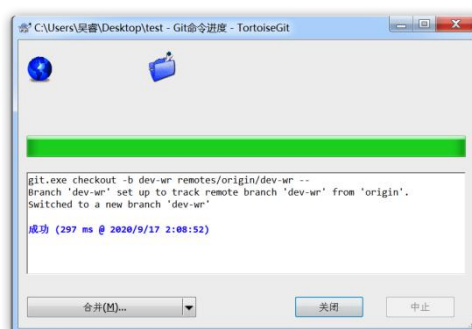
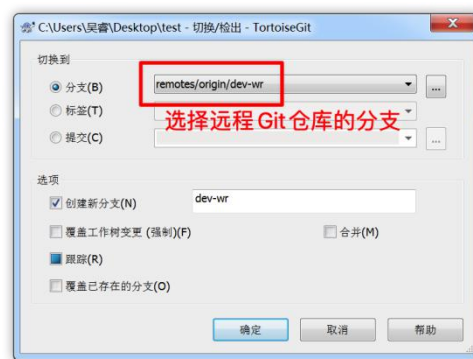
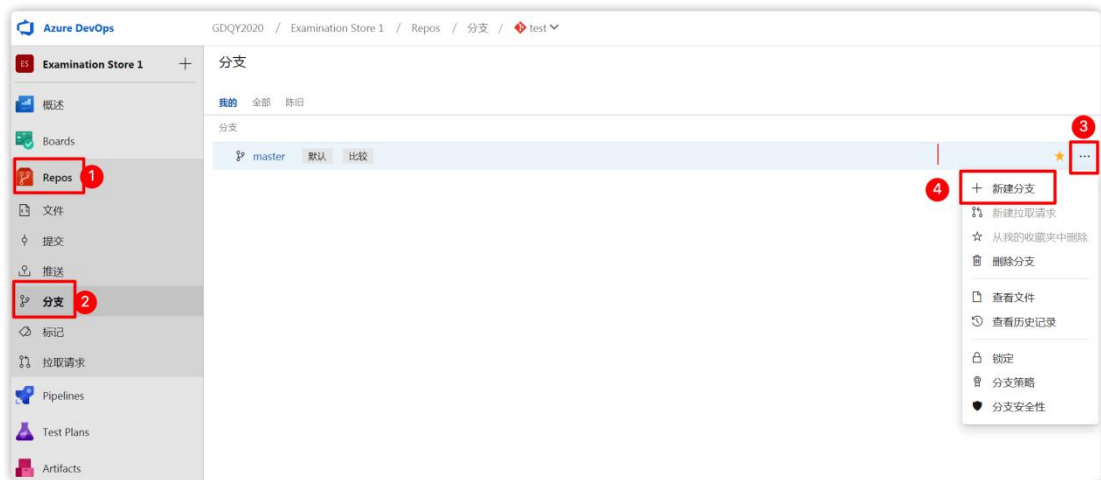
- c) 推送 Push: 把暂存区更改添加到远程 Git 仓库，一般保持本地分支和远端分支一致，推送成功一般显示“成功”



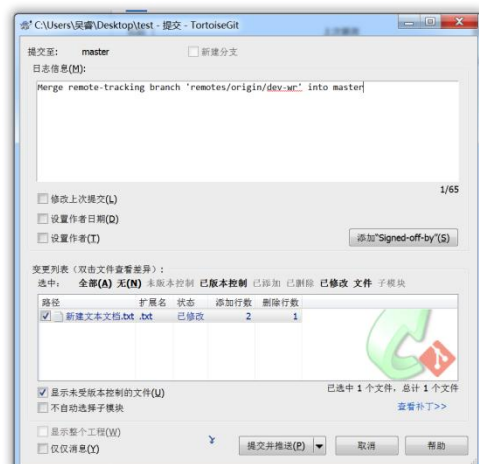
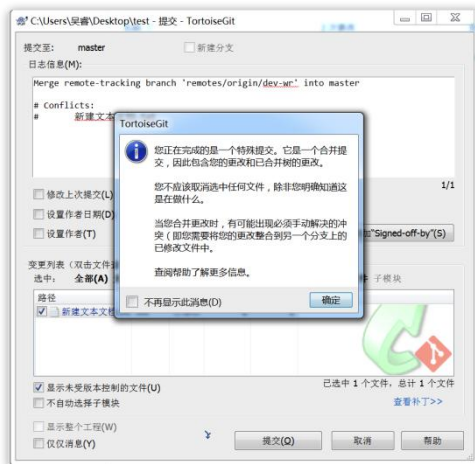
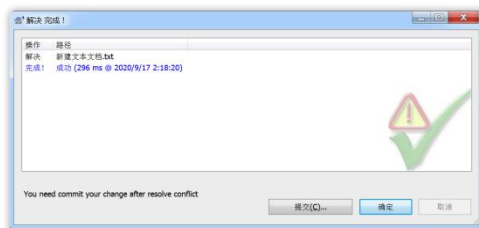
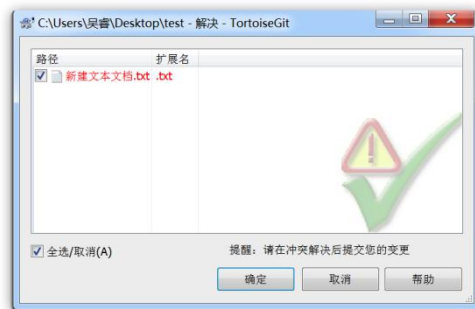
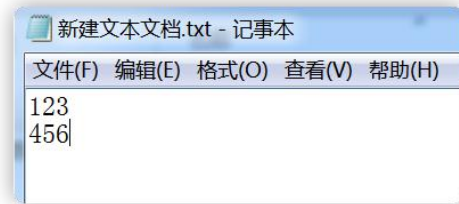
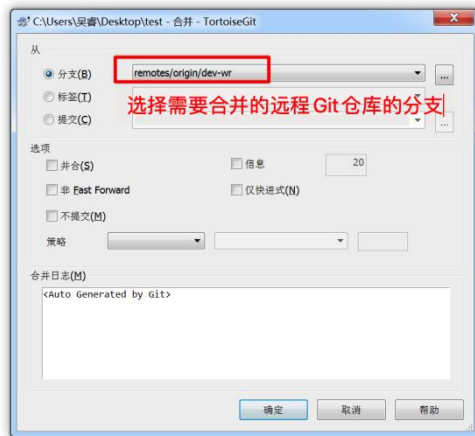
- d) 创建分支 Create Branch: 创建一个分支，便于独立工作，互不影响，可以使用 dev-xxx 命名分支，如果在本地创建分支，需要手动推送到远程 Git 仓库，最终会显示“成功”

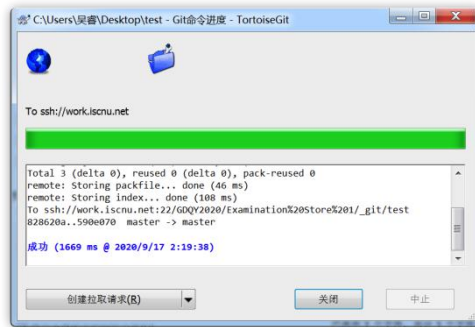


- e) 切换 Switch/检出 Checkout: 用于切换或者获取分支的内容, TFS 上也可以创建分支, 创建后需要使用切换手动获取该远程分支, 一切顺利的话会显示“成功”

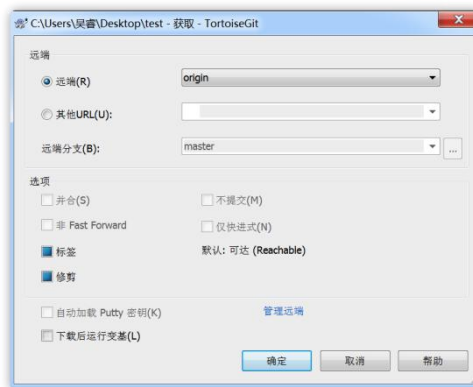


- f) 合并 Merge 和解决冲突 Resolve: 可以把其他分支的改动合并到当前分支, 假设我现在在 dev-wr 分支上修改了 txt 文件的内容为 456, 再切换到 master 分支, 这时候需要把 dev-wr 分支上的内容合并到 master 分支上, 因为 txt 的内容不一样, 所以产生了冲突, 这时候需要解决冲突 Resolve, 可以借助开发工具来进行这一步, 或者手动修改, 最后提交即可





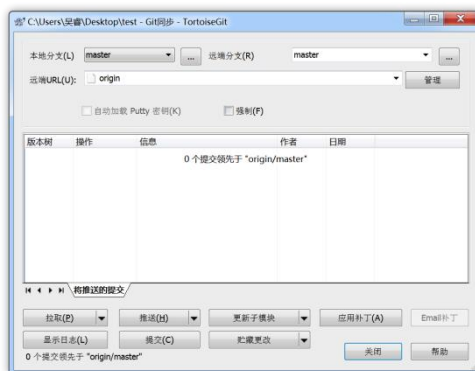
- g) 获取 Fetch：读取远程 Git 仓库的更新，可以指定分支，但不会获取到本地，如果有更新，可以手动拉取 Pull



- h) 拉取 Pull：等同于 Fetch+Merge



- i) 同步 Sync：等同于 Pull+Push，不是标准的 Git 操作，是 TortoiseGit 为了方便提供的操作



6. 参考资料

- a) [Git 协同与提交规范](#)
- b) [Pro Git](#)
- c) [廖雪峰 Git 教程](#)
- d) [易百教程 Git 教程](#)
- e) [一张脑图带你掌握 Git 命令](#)

每个人都可能犯错，如果你发现这份教程内有任何错误，请联系我修改，谢谢。