GIT 客户端配置和使用手册

大智慧Git版本库运行在Atlassian Bitbucket。URL：http://git.gw.com.cn:7990

Git是什么?

- 开源的代码管理软件
- 分布式版本控制系统
- 分散式协作存储工具
- Git方便地在本地进行版本管理，选择适当的本地版本推送到统一的版本管理服务器

GIT	SVN
Git 是分布式代码管理系统； 大部分的操作都在客户端本地完成，只有少数的操作需要服务器参与；	SVN 是集中式代码管理系统； 几乎所有的操作都需要服务器参与；
GIT把内容按元数据方式存储；	SVN 按文件；
Git 用指针为分支；	SVN 用目录为分支；
GIT用SHA值版本号；	SVN有全局版本号；

Git 基础

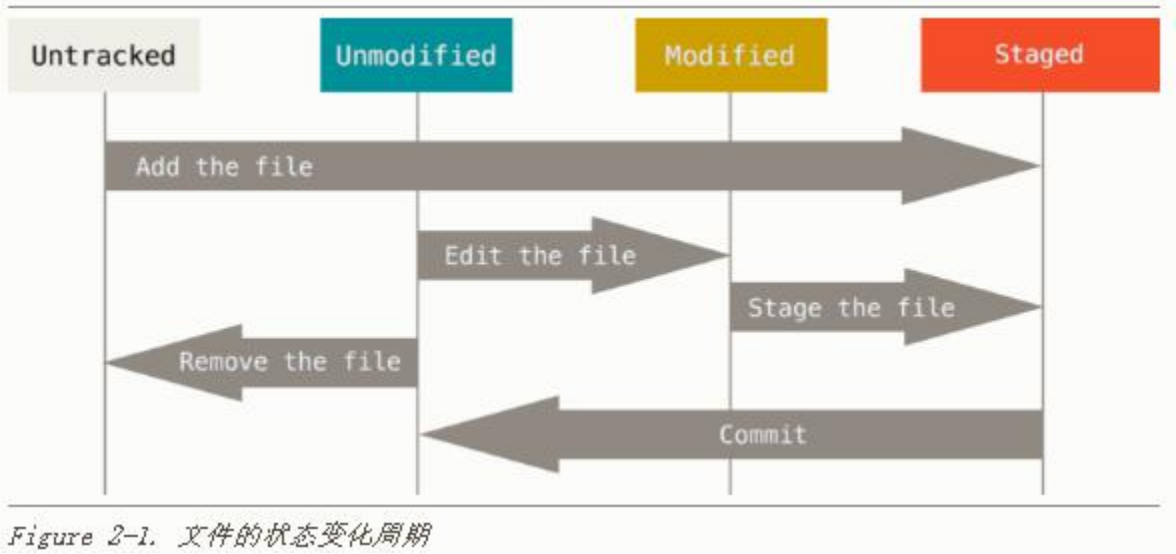


Figure 2-1. 文件的状态变化周期

对于任何一个文件，在 Git 内都只有四种状态：

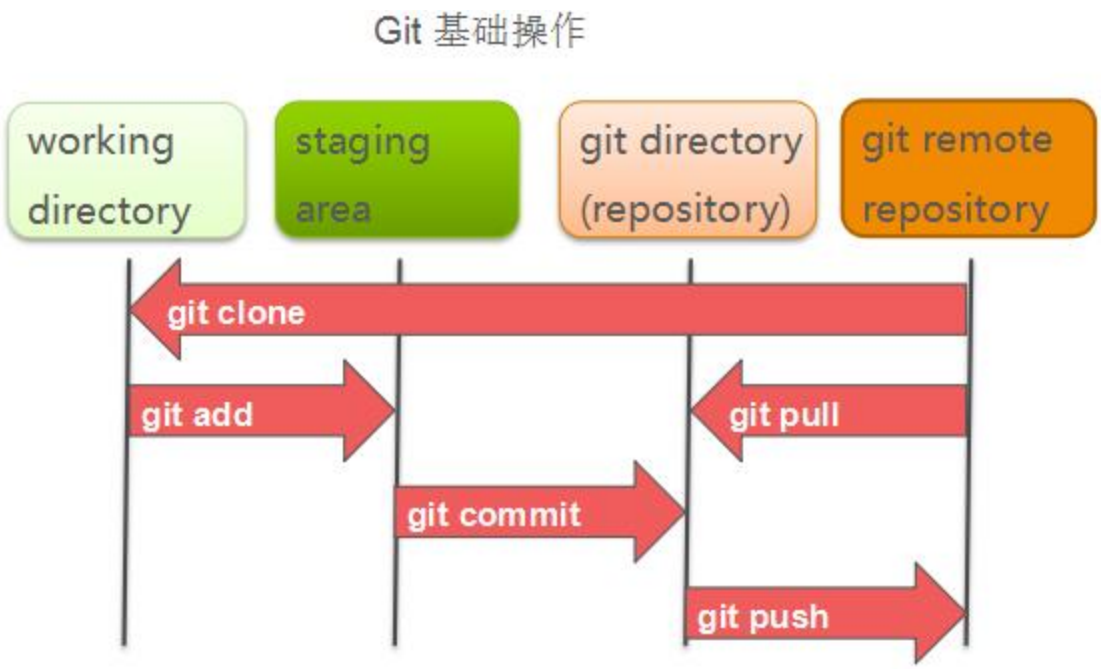
未被跟踪（untracked）表示文件未被跟踪，没有加入版本管理中；

未修改（unmodified）表示该文件已经被安全地保存在本地数据库中了；

已修改（modified）表示修改了某个文件，但还没有提交保存；

已暂存（staged）表示把已修改的文件放在下次提交时要保存的清单中。

由此我们看到 Git 管理项目时，文件流转的工作区域：Git 的工作目录，暂存区域，以及本地仓库。



每个项目都有一个 Git 目录（注：如果**git clone**出来的话，就是其中**.git**的目录。），它是 Git 用来保存元数据和对象数据库的地方。

该目录非常重要，每次克隆镜像仓库的时候，实际拷贝的就是这个目录里面的数据。

从项目中取出某个版本的所有文件和目录，用以开始后续工作的叫做工作目录。

这些文件实际上都是从 Git 目录中的压缩对象数据库中提取出来的，接下来就可以在工作目录中对这些文件进行编辑。

暂存区域只不过是个简单的文件，一般都放在 Git 目录中。有时候人们会把这个文件叫做索引文件，不过标准说法还是叫暂存区域。

基本的 Git 工作流程如下：

1. 在工作目录中修改某些文件。
2. 对修改后的文件进行快照，然后保存到暂存区域。
3. 提交更新，将保存在暂存区域的文件快照永久转储到 Git 目录中。

Git Windows客户端安装和配置

- MsysGit，Windows下的Git，命令行工具
    1. 下载和安装Git， 下载地址：https://www.git-scm.com/downloads
    2. 安装程序，选择默认设置

- TortoiseGit，Windows下的Git，图形界面工具
    1. 下载和安装Msysgit 下载和安装，如上
    2. TortoiseGit 下载地址：https://download.tortoisegit.org/tgit/
    3. 选择适合你系统的位数的安装包和汉化包
    4. 安装程序，选择默认设置

Git – 基础配置

1. 启动MsysGit
2. 操作命令： 图-1

```
$ git config —global user.name 'user-id'
$ git config —global user.email'xx@gw.com.cn'   # 注：请用IMS用户名为user.name并用公司邮箱
```

配置ssh访问

1. 启动MsysGit
2. 命令：$ ssh-keygen -t rsa

3. MsysGit 克隆版本库 将公钥（C:\Users\<用户名>\.ssh\id_rsa.pub）内容添加到Git服务器，图-2
4. 获取版本库链接，从Git服务器， 图-3

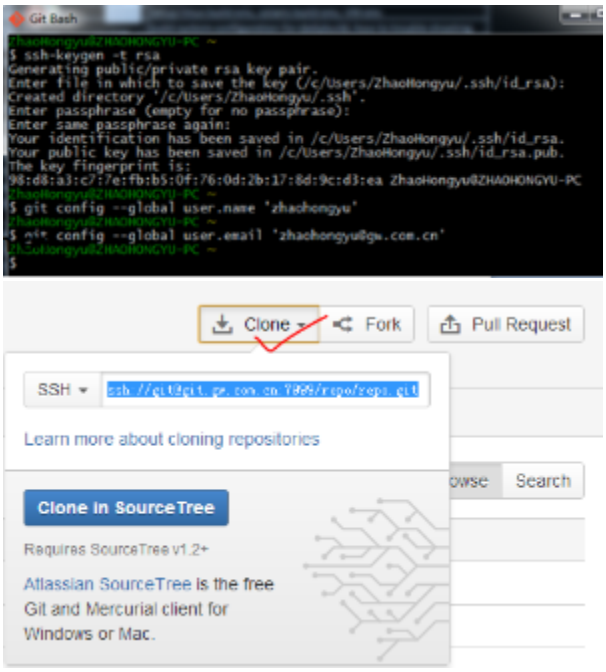
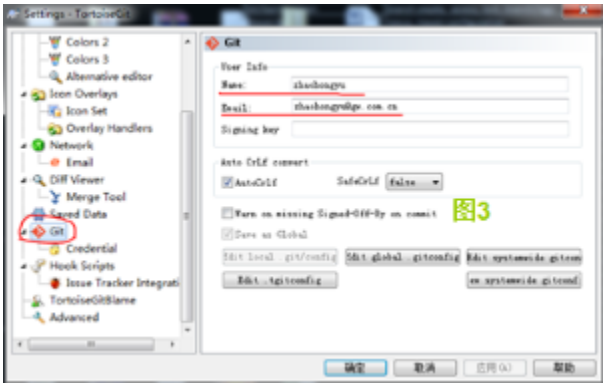
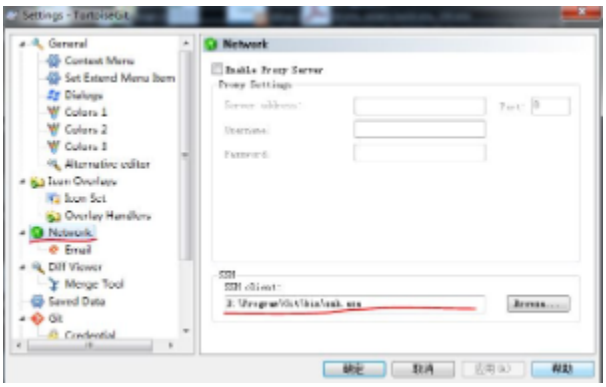


图-1 图-2 图-3

TortoiseGit – 配置用户信息和设置SSH Client

1. TortoiseGit -> Settings -> Git ， 如下图;
2. Network，输入MsysGit 的ssh.exe路径;





使用http/ssh协议克隆/推送/同步

- 登录Git服务器，页面左上角有http/ssh链接地址

Git基础练习

请参照 git-scm.com进行Git练习。

https://git-scm.com/book/zh/v2/Git-%E5%9F%BA%E7%A1%80-%E8%8E%B7%E5%8F%96-Git-%E4%BB%93%E5%BA%93

Git日常工作流程

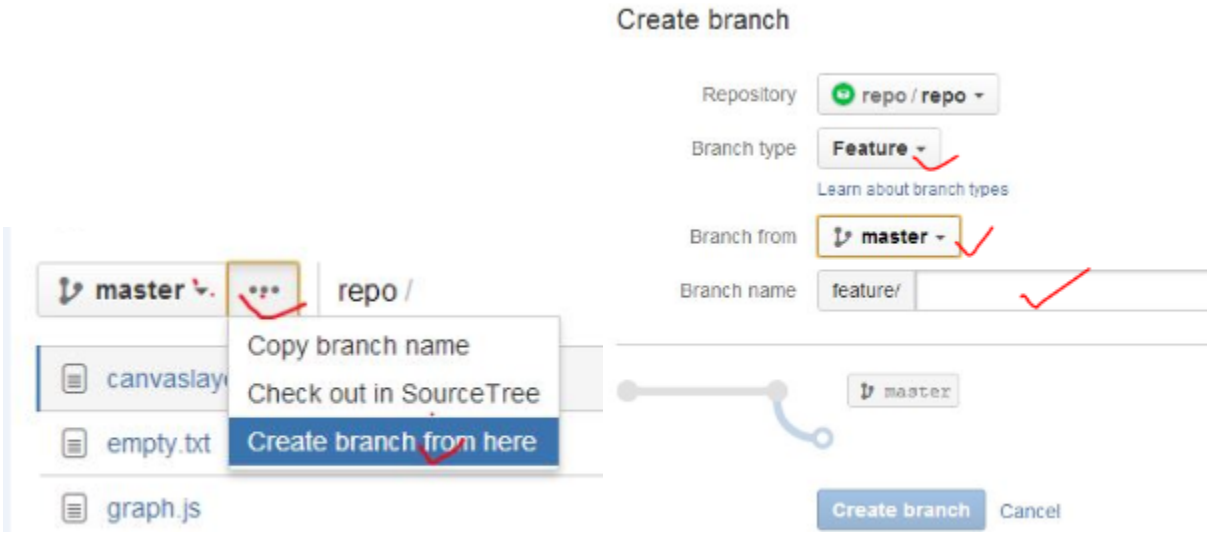
1. 登录 Git Website，依据IMS问题号从主干分支上创建特性分支；
2. 克隆/更新本地项目版本库；
3. 在本地分支上更新/修改代码，及时提交和上传到远程库；
4. 在本地完成修改后，须和主干分支负责人确认是否合并主分支及合并基点；
5. 创建 Pull Request；
6. 若特性分支基线版本低于主干分支最新版本。开发人需做git merge/ git cherry-pick，以保证特性分支基线版本与主干分支最新版本相同，使之代码合并到主干分支无冲突；
7. 主干分支负责人接到Pull Request，须审核提交代码，确认无误后，点击合并代码按钮。

分支管理

分支命名规则：

- Bugfix: Typically used for fixing bugs against a release branch
- Feature: Used for specific feature work. Typically, this branches from and merges back into the development branch
- Hotfix: Typically used to quickly fix the production branch
- Release: Used for release tasks and long-term maintenance. Typically, this branches from the development branch and changes are merged back into the development branch.
- Custom: Custom branch name.

创建远程分支，如下图：



- Bugfix： bugfix/<IMS-No.>
- Feature： feature/<IMS-No.>
- Hotfix： hotfix/<IMS-No.>
- Release：release/<IMS-No.>
- Custom：

分支合并

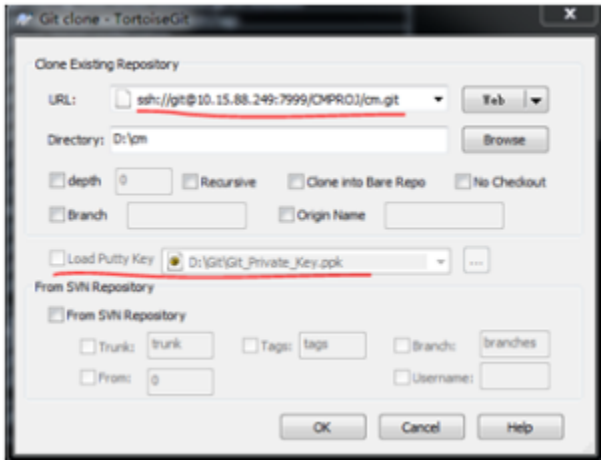
使用Pull
resquest快速和简单的方法来评估更改在开发分支上，讨论这些变化,并进行进一步的修改合并分支之前回到发布分支或主开发分支。

详细内容请见Git 开发流程 – Pull Resquest in Stash

Git常用命令

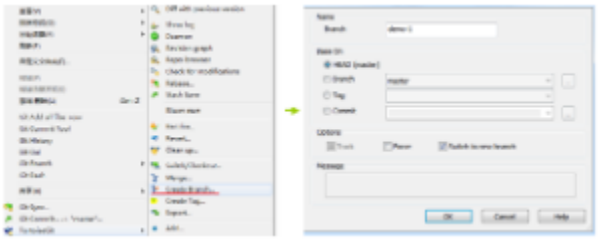
- git clone

克隆远程版本库;



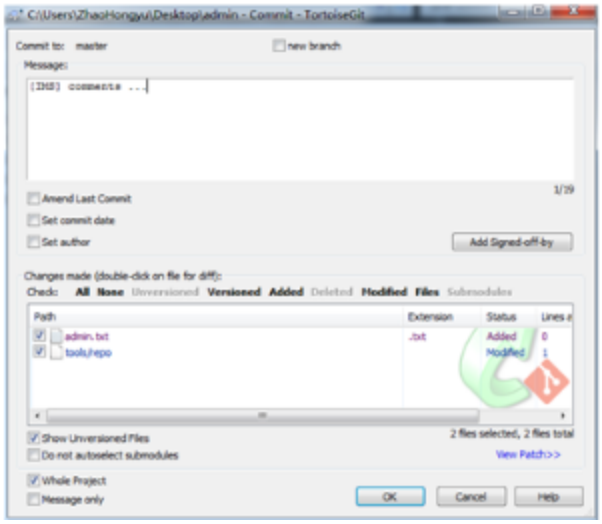
- git checkout -b <branch-name>

创建本地私有分支;



git commit

代码提交。这个时候的代码还没有提交到服务器上，只是提交到了本地;



- git merge 代码合并

1. 在主分支上，拉取主分支最新代码;
2. 选择要合并的本地分支;
3. 使用squash参数，将你本地多次提交，合并为一次修改;
4. 解决冲突，代码提交（commit），代码推送（push;

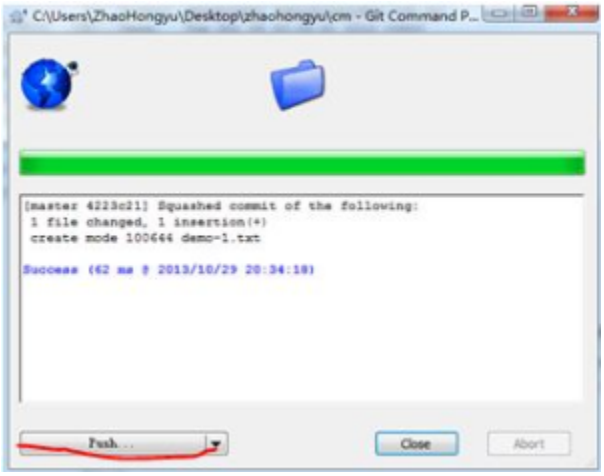
TortoiseGit -> Merge -> 选择 “特性分支”

- git cherry-pick 代码合并之挑拣

1. 在主分支上，拉取主分支最新代码；
2. show log -> choose dev branch -> select commits -> right click > cherry-pick -> commit ；
3. 使用squash参数，将你本地多次提交，合并为一次修改；

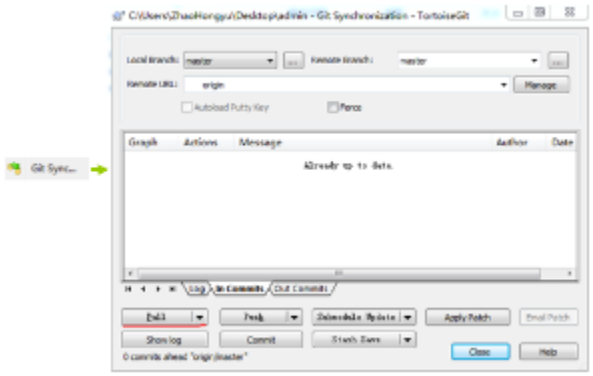
- git push

  修改提交到远程服务器



- git pull

同步远程服务器代码



- git reset – 重定位
- git revert –还原本地修改
- git delete 从版本库中删除
- git log 查看历史记录
- git status
- git stash

标签管理

发布版本须打标签，样式如下：

- v7.6-sp2-rel
- v7.50-rel

Git参考资料

1. Git 权威指南
2. Pro-Git中文版https://git-scm.com/book/zh/v1