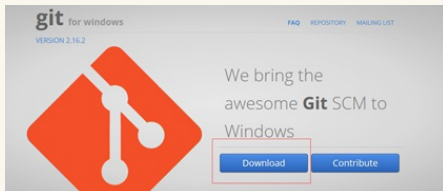


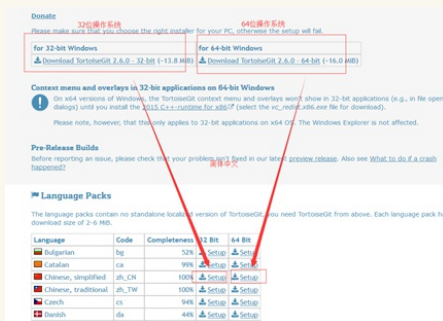
Git客户端(TortoiseGit)基本使用详解

1、环境安装

Git最新版下载地址: <https://gitforwindows.org/>



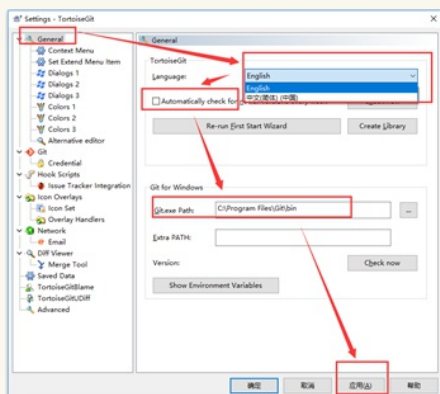
TortoiseGit, Git客户端, 32/64位最新版及对应的语言包下载地址: <https://tortoisegit.org/download/>



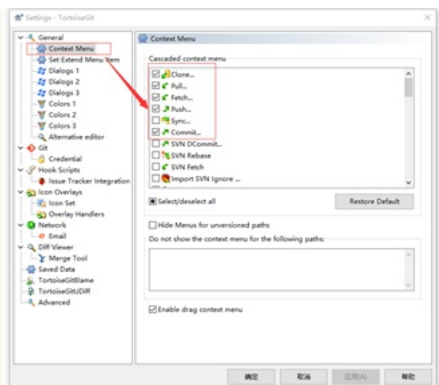
安装的方法, 一直下一步就行, 具体做法省略。

2、配置

1. 首先,请选定一个存放Git项目的目录,这样管理方便. 如: D:\test , 然后在资源管理器中打开.
2. 在空白处点击鼠标右键, 可以看到右键菜单中多了几个选项. 选择 --> TortoiseGit --> Settings, 然后就可以看到配置界面, 如下图所示:



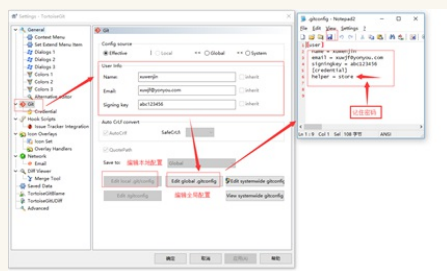
3. 选中General, Language中选择中文. 不勾选自动升级的复选框, 可能还需要指定 Git.exe 文件的路径, 如 "D:\DevlopPrograms\Git\bin". 完成后, 点击应用, 确定关闭对话框(楼主比较习惯英文).
4. 再次点击鼠标右键, 可以看到弹出菜单中已经变成中文. 原来的 Settings 变成 设置, Clone 变为 克隆.
5. 配置右键菜单. 在设置对话框中, 点选左边的"Context Menu", 设置常用的右键菜单. 楼主比较常用的是如下选项:



6. 设置记住密码

!!!!密码会明文保存在 C:\Users\Administrator\.git-credentials 这种文件中, 请小心使用.

进入设置, 点选左边的Git标签. 可以发现, 右边可以配置用户的名字与Email信息. 如下图所示:



因为当前还没有本地项目, 所以“Edit local .git/config(L)”按钮处于灰色不可用状态, 如果在某个本地Git项目下打开配置对话框, 那么这个按钮就可用, 然后就可以编辑此项目的一些属性。

点击“Edit global .git/config(O)”按钮, 会使用记事本打开全局配置文件, 在全局配置文件中, 在后面加上下面的内容:

```
[credential]
    helper = store
```

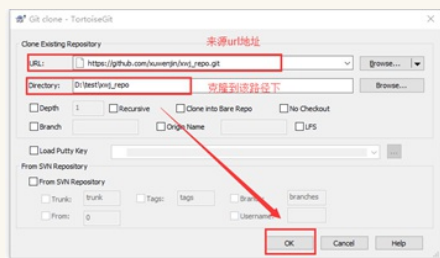
完成后保存, 关闭记事本, 确定即可。

当你推送项目到GitHub等在线仓库时, 会记住你输入的用户名和密码

3、示例

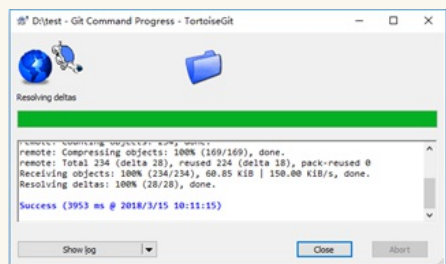
1. 克隆一个项目（楼主是从github上获取）

在工作目录下, 如D:\test, 空白处右键, 选择: Git clone, 则会弹出克隆对话框, 如下图所示:



在URL中填写项目的访问地址, 如: https://github.com/xuwenjin/xwj_repo.git

根据项目大小, 时间会不一样. 克隆完成后, 如果没有错误, 会给出提示:



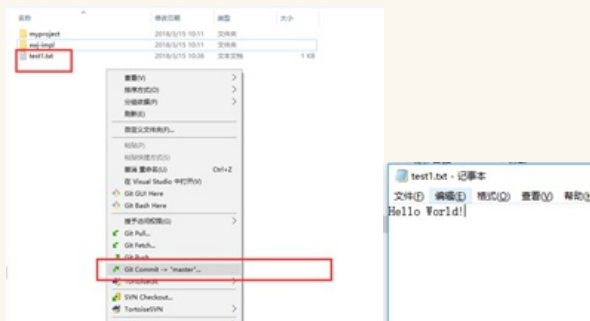
进入克隆下的文件夹中, 如D:\test\xwj_repo, 空白处右键, 弹出如下菜单:



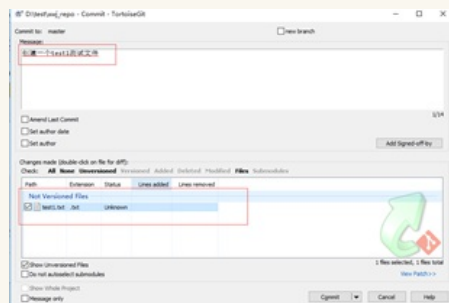
其中，Git Pull是从远端拉取最新的代码，Git Fetch是从远端拉取最新的分支，Git Push是将本地仓库的代码提交到远端
Git Commit -> "master"，将本地代码提交到本地版本库（默认的分支是master）。

2. 将本地代码提交到本地仓库

创建一个文件，如test1.txt, 然后提交(commit)到本地版本库(这个操作可以在离线状态操作)

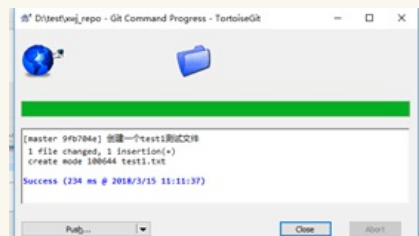


2.1 选择Git Commit -> "master"，弹出下图：



2.2 填写提交备注message（不填写不允许提交），勾选需要提交的文件，点击commit，即可将本地代码提交到本地版本库。

出现如下弹框，表示提交成功



其中，提交时，会发现上图中的Status有几种值：

Unknown: 新增的文件，也不在版本库

Added: 新增的文件，在版本库

Modified: 文件修改，在版本库

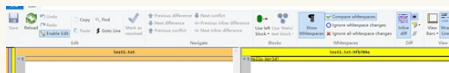
Missing: 文件被删除，在版本库

3. 查看日志

在日志中，可以通过日期、文件名、提交人等等过滤查询。



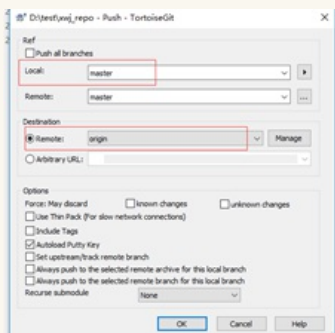
3.1 通过日志，可以很直观的看到提交相关记录。比如提交人、提交时间、提交了哪些文件等等。这些信息便于以后进行文件对比或者版本回滚(后面将会介绍)点击test1.txt，可以看到本次提交，对test1文件进行了哪些操作：



至于怎么去看这些颜色表示什么意思，可以去百度一下，这里不做介绍

4. 将代码推送(push)到远端。这里是github

右键空白处，选择Git Push，出现如下弹框：



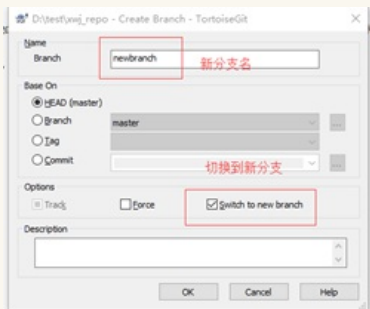
这里可以看到是本地哪个版本库提交到远端。至此，文件的整个提交过程就完成了

5. 分支

发现问题：你代码写了很多，运行OK；但是突然想加个新功能进去，这个功能你也不知道能否正常运行，而且修改过程中，除了新加代码和文件进去，还会修改以前的代码。要是万一失败，修改回来也是一种很麻烦的事情。这种时候很多人就用备份方式。来看看git是怎么优雅的处理这个问题的。

git的处理方式：当你想加一个新功能进去的时候，你可以新建一个分支，例如名字叫newbranch，然后在分支中把新功能加上去，如果OK，将代码合并到master分支上，如果新功能失败，切换回master分支上来，在newbranch写的代码，又全看不到了。

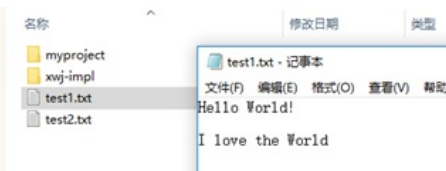
5.1 新建一个分支，右键-->TortoiseGit-->Create Branch



右键，你会发现当前的分支为你新建的newbranch分支了



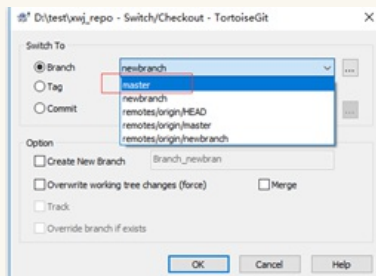
5.2 接下来，就开心新增你的功能，比如我们在test1.txt文件中新增一行文字，同时新增一个文件test2.txt



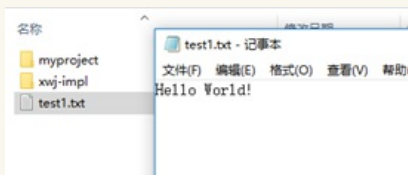
5.3 提交我们的代码到newbranch分支的本地仓库，参考之前的2.2

5.4 突然发现该功能有漏洞，想回到之前的master分支，怎么办呢？

右键-->TortoiseGit-->Switch/Checkout，选择master即可。如下图：



5.5 空白处右键，会发现已经切换到master分支了。此时发现test2.txt并没有带过来，再来看看test1.txt文件是否回到了从前呢？



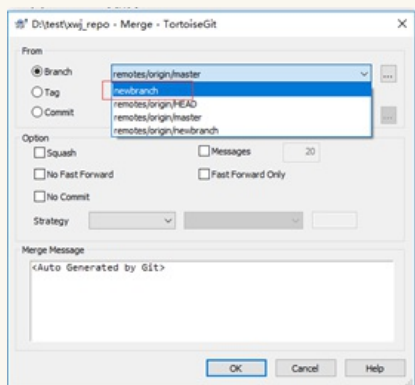
结果显示，无论是修改还是添加，都没有影响master分支上的文件，这就是分支的作用。

6. 分支合并

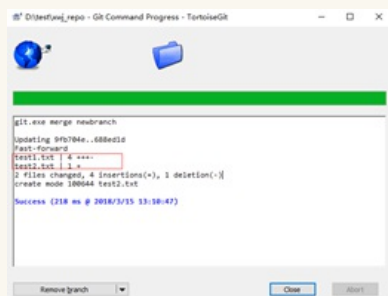
如果此时发现newbranch分支上的功能是有用的，希望能合并到master，又该怎么操作呢？

PS：在合并分支前，一定要确认newbranch分支上的代码全部提交到本地版本库了

右键-->TortoiseGit-->Merge，选择被合并的分支，即newbranch。如下图



6.1 点击OK按钮，会出现下图弹框。弹框中会列出被合并的文件。如下图所示：



6.2 我们发现test2.txt文件被合并过来了，打开test1文件，发现内容正好是在newbranch分支上修改的内容。

至此，分支合并完成

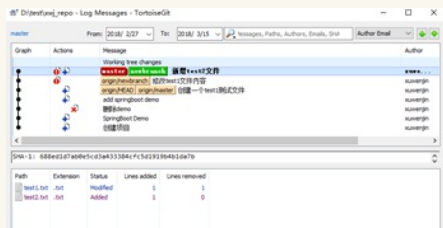
7. 拉取（pull）远端最新的修改内容

此时，我们团队还有另一个成员B，他需要获取我最新修改的内容，该怎么操作呢？

首先B切换到跟我同一分支，然后右键-->Git Pull，点击确认即可，就可将我修改的内容拉取到他的本地版本库

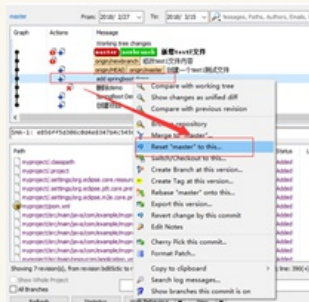
8. 版本回滚

8.1 先查看日志，确定想回滚到哪个版本

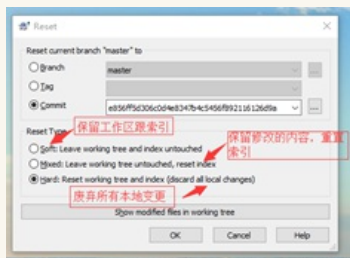


8.2 比如想回滚到创建test1.txt文件前，则点击该步操作所对应的message之前的那一个日志（add spring demo）

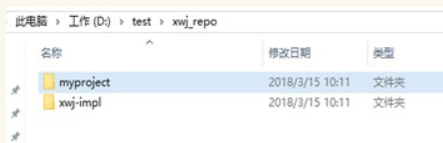
然后右键-->Reset "master" to this..., 表示将当前master分支上的文件回滚到这个版本，如下图



8.3 选择之后，弹出如下弹框，在Reset Type下选择Hard: Reset working....., 点击Ok即可



会发现，此时test1.txt跟test2.txt两个文件都消失了，如下图：



9. 解决冲突（略）

知识改变世界

分类: [Git](#)



仅此而已-远方
关注 - 0
粉丝 - 7
[+加关注](#)

5 推荐
0 反对

« 上一篇: [Git文件常见下标符号说明](#)
» 下一篇: [MySQL 临时表](#)

发表评论


#1楼 2018-08-21 13:23 | 碌云 

66

写得非常棒，不过，什么时候把解决冲突补充下。。。。。

支持(2) 反对(0)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

 注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。

【推荐】超50万VC++源码: 大型组态工控、电力仿真CAD与GIS源码库！

【活动】申请成为华为云云享专家 尊享9大权益

【腾讯云】拼团福利，AMD云服务器8元/月



腾讯云AMD云服务器

节省IT成本30%
1核1G AMD机型**0.57元/天**起

[立即抢购](#)

相关博文：

- [git for windows+TortoiseGit客户端的使用](#)
- [TortoiseGit安装与配置](#)
- [git客户端基本操作](#)
- [转。git 乌龟的使用安装](#)
- [git客户端msysGit和TortoiseGit使用](#)



最新新闻：

- [万豪面临客户集体诉讼 索赔金额最高达125亿美元](#)
 - [押金难退、专利纠纷，途歌公司再遇风波](#)
 - [微软重振雄风留给投资者的三大教训](#)
 - [SpaceX将再次发射“三手”猎鹰9号 这次是一箭64星](#)
 - [既不是业界首个也非框架，阿里妈妈的XDL到底应该怎么理解？](#)
- » [更多新闻...](#)

昵称：仅此而已-远方

园龄：4年5个月

粉丝：7

关注：0

[+加关注](#)

2018年12月						
日	一	二	三	四	五	六
25	26	27	28	29	30	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

搜索

找找看

谷歌搜索

[常用链接](#)

我的随笔
我的评论
我的参与
最新评论
我的标签
更多链接

随笔分类

ActiveMq(5)
docker(1)
elasticsearch(6)
freemarker(1)
Git(2)
java基础(13)
Java内存(3)
JPA(3)
mybatis(2)
mysql(2)
RabbitMq(2)
redis(3)
spring(4)
Spring Security(7)
springboot(5)
springcloud(3)

随笔档案

2018年11月 (1)
2018年10月 (4)
2018年9月 (7)
2018年7月 (5)
2018年5月 (12)
2018年4月 (30)
2018年3月 (5)
2018年2月 (1)

最新评论

1. Re:com.mysql.jdbc.MySQLDataTruncation: Data truncation: Truncated incorrect DOUBLE value:...
set 字符串要放在语句最后，算是个bug

--truestar

2. Re:Git客户端(TortoiseGit)基本使用详解
写得非常棒，不过，什么时候把解决冲突补充下。。。。。

--碌云

阅读排行榜

1. Git客户端(TortoiseGit)基本使用详解(29832)
2. ObjectMapper使用(10883)
3. elasticsearch-head的使用(5568)
4. springboot之异步调用@Async(3311)
5. JPA之@Entity、@Table、@Column、@Id(3101)

评论排行榜

1. Git客户端(TortoiseGit)基本使用详解(1)
2. com.mysql.jdbc.MySQLDataTruncation: Data truncation: Truncated incorrect DOUBLE value:...(1)

推荐排行榜

1. Git客户端(TortoiseGit)基本使用详解(5)
2. JPA之@Entity、@Table、@Column、@Id(1)
3. Spring JdbcTemplate详解(1)
4. ObjectMapper使用(1)

