**中国矿业大学计算机学院**

**2017 级本科生课程报告**

课程名称 源代码版本管理

报告时间 2019-12-27

学生姓名 郭子杨

学 号 08173042

专 业 计算机科学与技术

任课教师 刘迎春

目录

[1. 实验软件情况 4](#_Toc28936050)

[1.1 Visual SVN Server、Tortoise SVN 4](#_Toc28936051)

[1.2 Git、Github 4](#_Toc28936052)

[2. 利用Visual SVN server和Tortoise SVN实现源代码版本管理 4](#_Toc28936053)

[2.1 创建仓库 4](#_Toc28936054)

[2.2 创建用户 5](#_Toc28936055)

[2.3 创建群组 5](#_Toc28936056)

[2.4 用户权限分发 6](#_Toc28936057)

[2.5 本地创建工作目录Checkout 6](#_Toc28936058)

[2.5 更新本地工作目录Update 7](#_Toc28936059)

[2.6 更新服务端仓库Commit 7](#_Toc28936060)

[2.7 版本回退 10](#_Toc28936061)

[2.8 代码冲突检测 13](#_Toc28936062)

[3. 利用Git与Github实现源代码版本管理 13](#_Toc28936063)

[3.1 Git与Github常用关系图 13](#_Toc28936064)

[3.2 安装git 14](#_Toc28936065)

[3.3 配置git全局环境 14](#_Toc28936066)

[3.4 初始化git本地仓库 14](#_Toc28936067)

[3.5 提交文件 15](#_Toc28936068)

[3.6 推送到远程仓库 16](#_Toc28936069)

[4. 利用idea内置git实现源代码版本管理 18](#_Toc28936070)

[4.1 本地创建本地工作目录 18](#_Toc28936071)

[4.2 利用git-add将本地项目源码添加至处于暂存区，并没有真正进入到版本库中 19](#_Toc28936072)

[4.3 利用git-commit将本地项目源码添加至本地版本库 19](#_Toc28936073)

[4.4 登陆 20](#_Toc28936074)

[4.5 成功将源代码Push到github远程仓库中 20](#_Toc28936075)

[4.6 idea push成功反馈信息 20](#_Toc28936076)

[5. 在Visual Studio环境下对所选系统的源代码进行版本管理 21](#_Toc28936077)

[5.1 Visual Studio安装源代码管理插件 21](#_Toc28936078)

[5.2 VS中使用git 21](#_Toc28936079)

[5.3 VS使用subversion 23](#_Toc28936080)

[6. 实验感悟 26](#_Toc28936081)

# 实验软件情况

## Visual SVN Server、Tortoise SVN

**VisualSVN Server**:免费的服务器端的版本管理软件

**TortoiseSVN**:客户端的版本管理软件。

TortoiseSVN 是 Subversion 版本控制系统的一个免费开源客户端，文件保存在中央版本库，除了能记住文件和目录的每次修改以外，版本库非常像普通的文件服务器。你可以将文件恢复到过去的版本，并且可以通过检查历史知道数据做了哪些修改，谁做的修改。

## Git、Github

Git是一个开源的分布式版本控制系统，可以有效、高速地处理从很小到非常大的项目版本管理。Git 是 Linus Torvalds 为了帮助管理 Linux 内核开发而开发的一个开放源码的版本控制软件。从一般开发者的角度来看，git有以下功能：

1、从服务器上克隆完整的Git仓库（包括代码和版本信息）到单机上。

2、在自己的机器上根据不同的开发目的，创建分支，修改代码。

3、在单机上自己创建的分支上提交代码。

4、在单机上合并分支。

5、把服务器上最新版的代码fetch下来，然后跟自己的主分支合并。

6、生成补丁（patch），把补丁发送给主开发者。

# 利用Visual SVN server和Tortoise SVN实现源代码版本管理

## 创建仓库

在服务器端创建一个中央仓库，各个用户分别向中央仓库commit和update

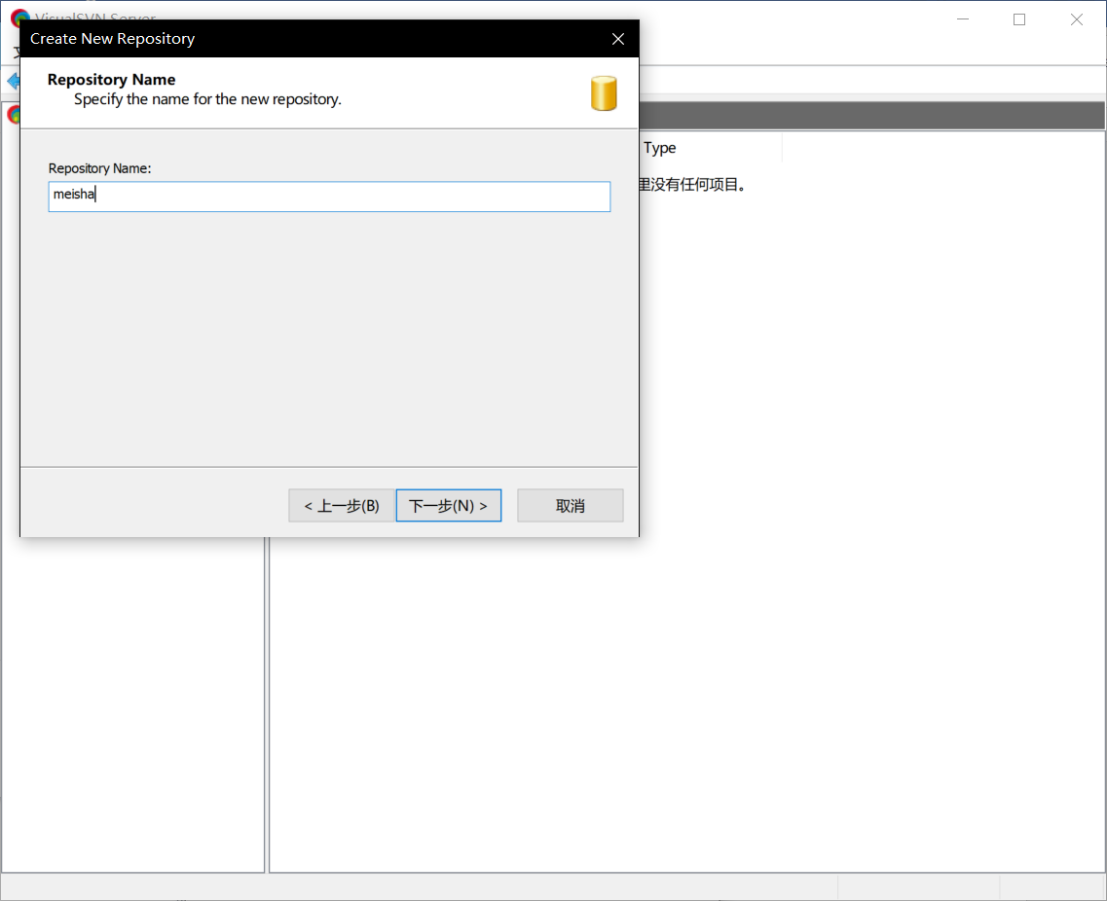


图2-1 创建仓库

## 创建用户

用户创建，创建的用户可以用于之后的群组的创建。

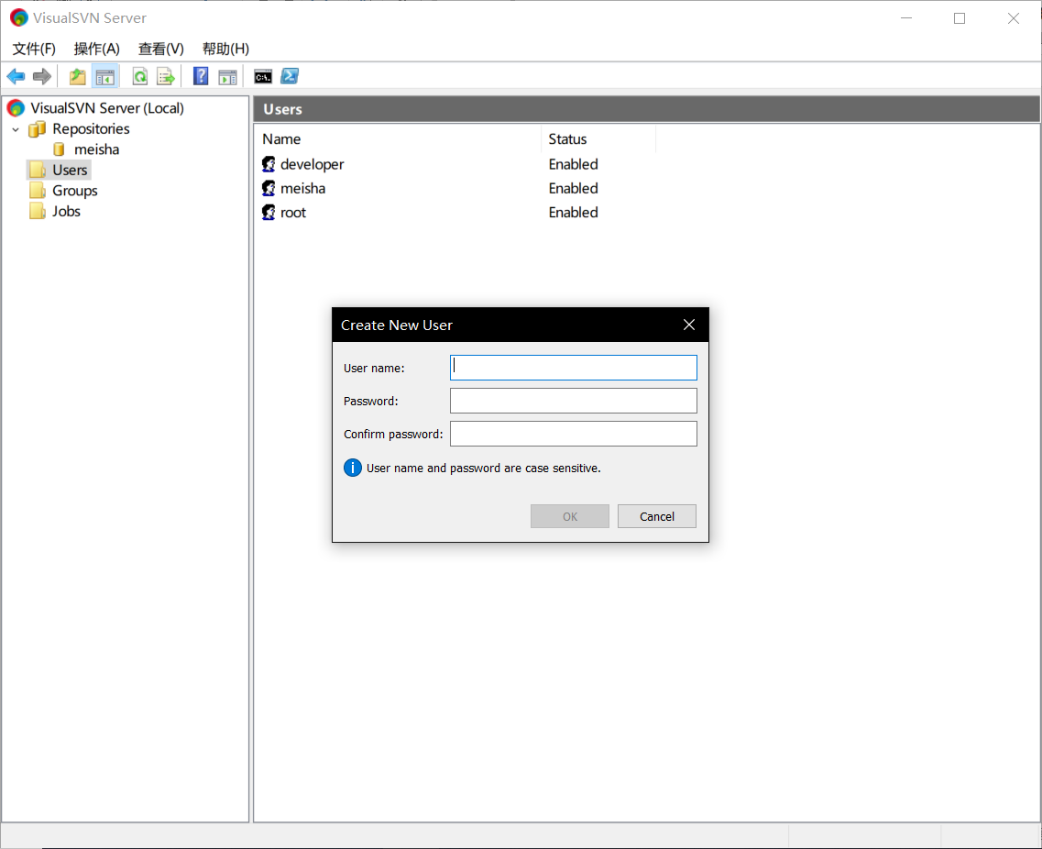


图2-2 创建用户

## 创建群组

群组的创建，创建一个群组用于管理群组中的用户。

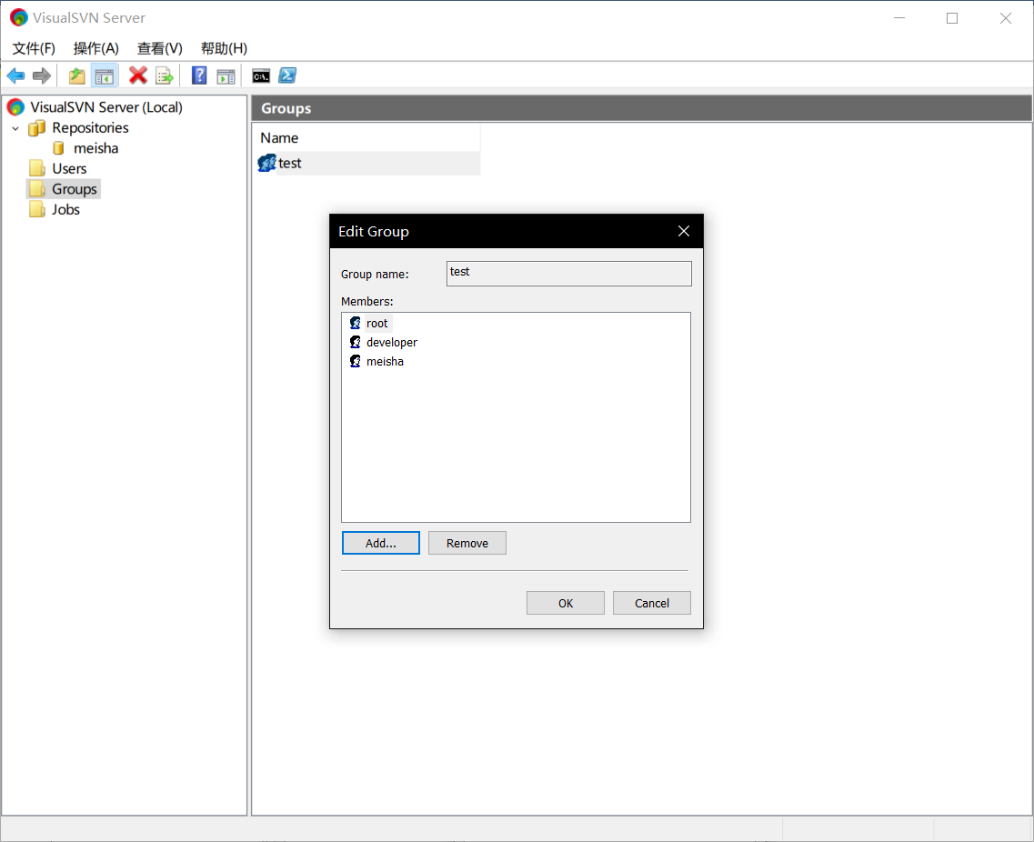


图2-3 创建群组

## 用户权限分发

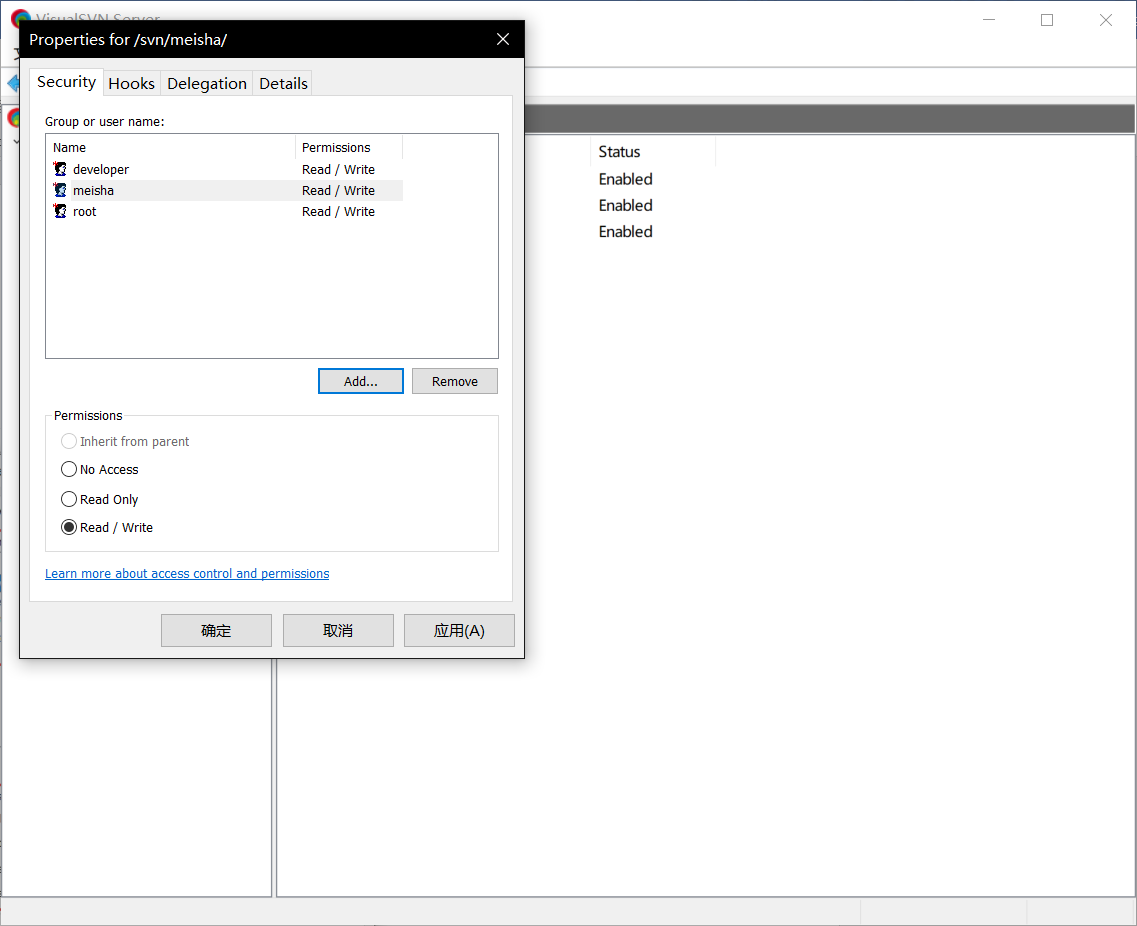


图2-4 用户权限分发

## 本地创建工作目录Checkout

本地新建一个文件夹，然后利用之间创建的用户去登入，第一次可以利用SVN Checkout操作，将远程服务器端仓库内容更新到本地的新建文件夹内。

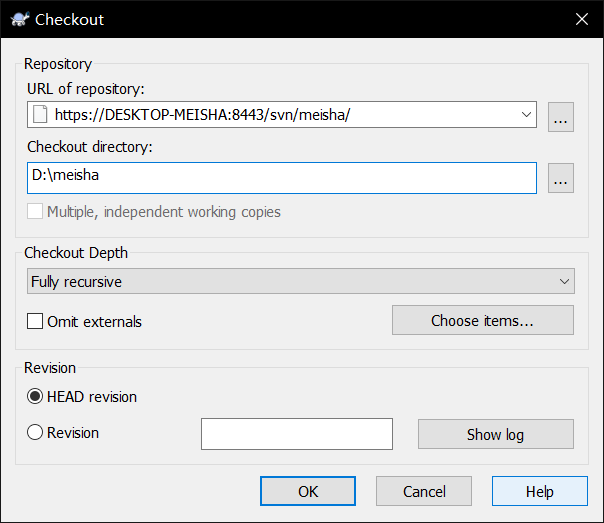


图2-5 checkout

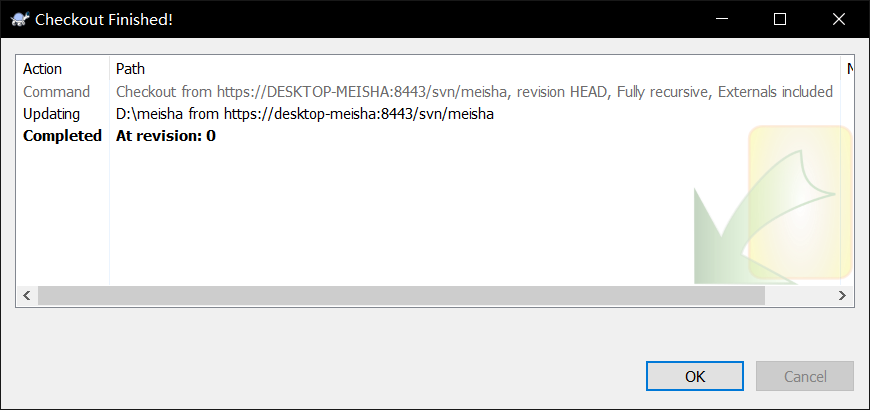


图2-6 checkout finished

## 更新本地工作目录Update

本地工作目录一旦和远程仓库连接之后，在本地就可以通过SVN update操作来获取当前远程仓库的内容了。

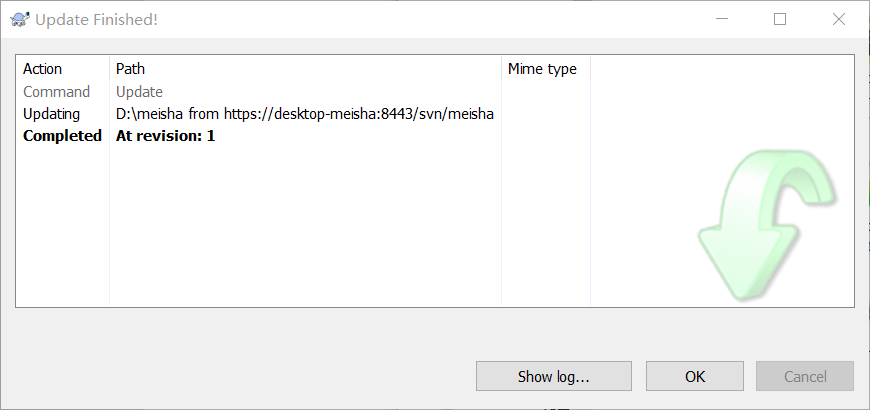


图2-7 update

## 更新服务端仓库Commit

当本地要对远程仓库进行更改的时候，就需要用到SVN Commit 操作。

1. 首先将要放入文件放到本地工作目录中，然后再利用Add命令将要加入仓库的文件进行登记。

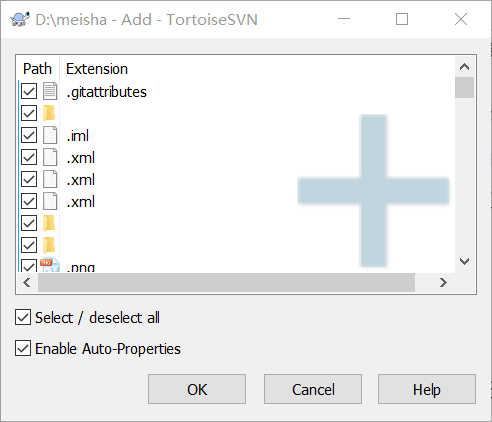


图2-8 add

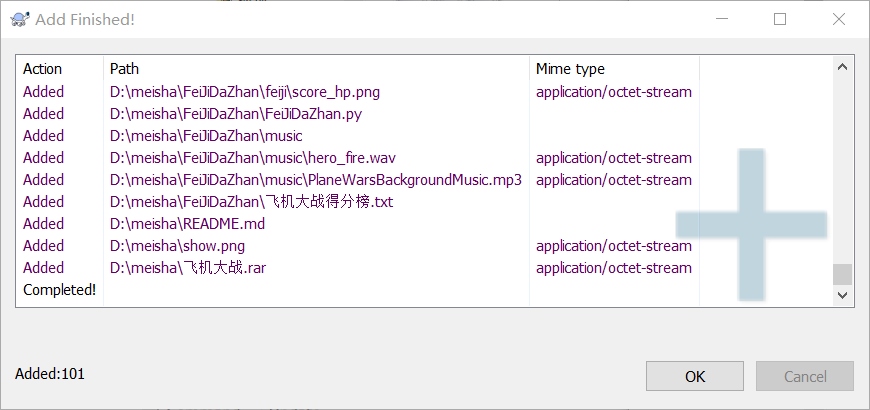


图2-9 add finished

1. 利用SVN Commit将本地工作目录更新到远程服务器仓库中

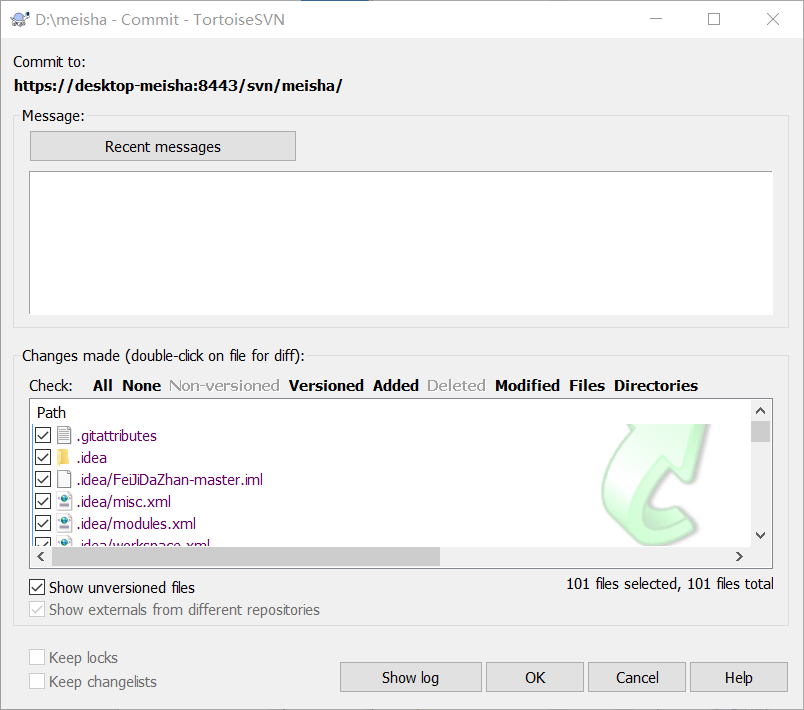


图2-10 commit

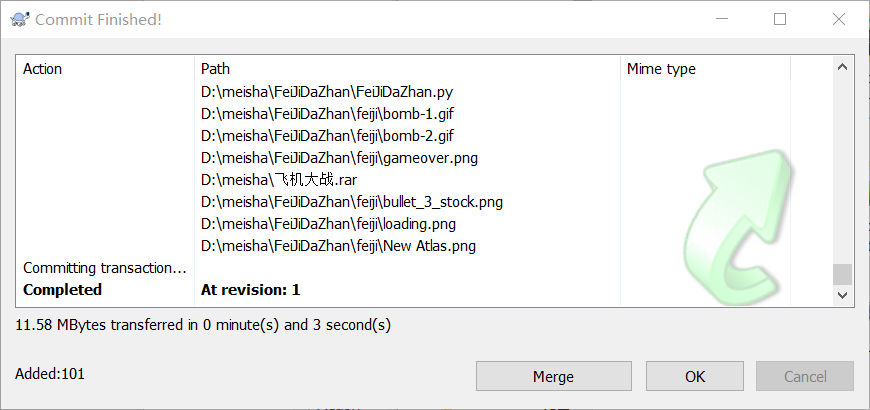


图2-11 commit finished

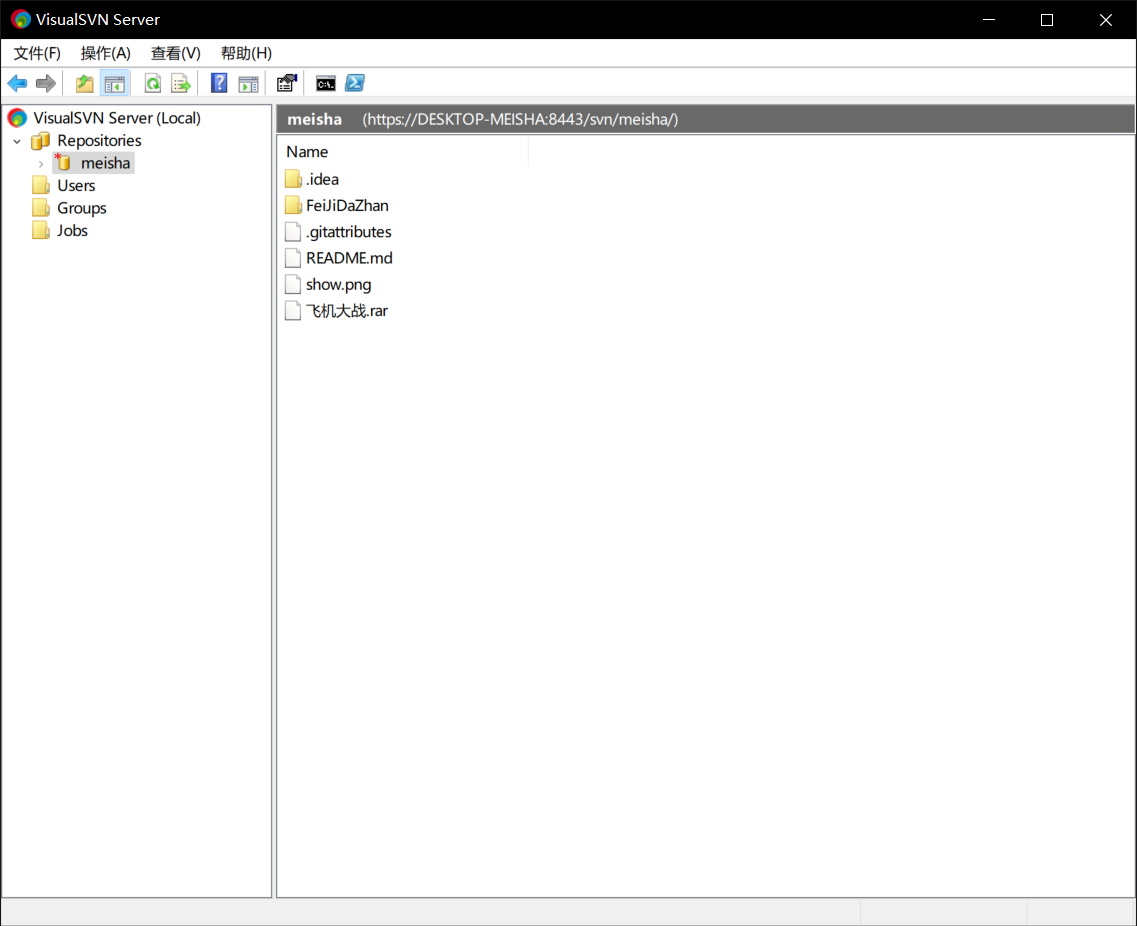


图2-12 查看仓库

## 版本回退

利用TortoiseSVN – Revert to this revision来进行版本回退，版本回退只会影响当前工作目录，不会改变服务器端仓库的内容。

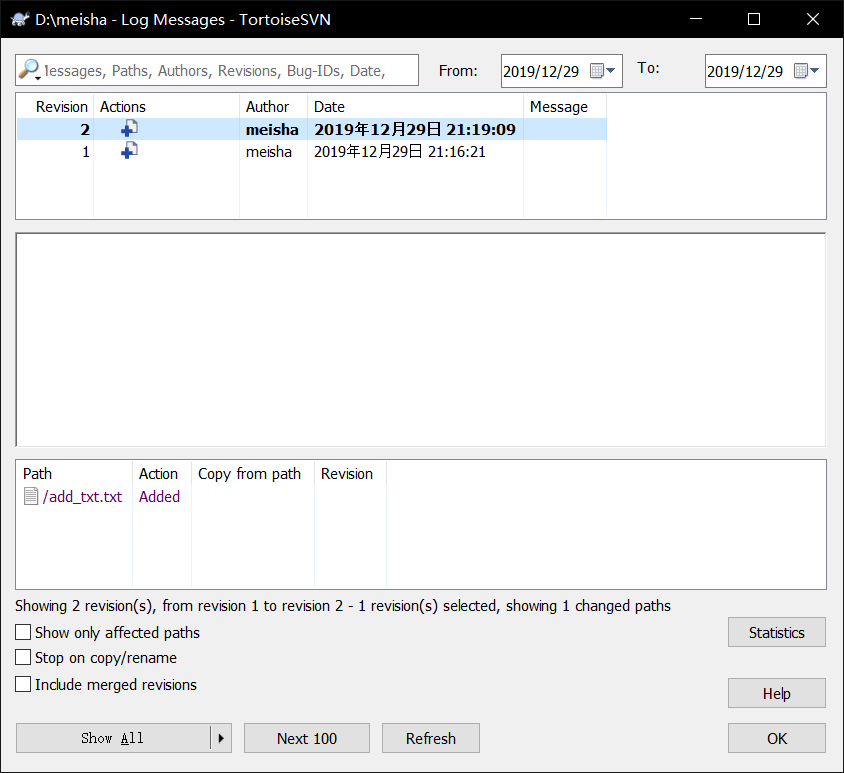


图2-13 回退

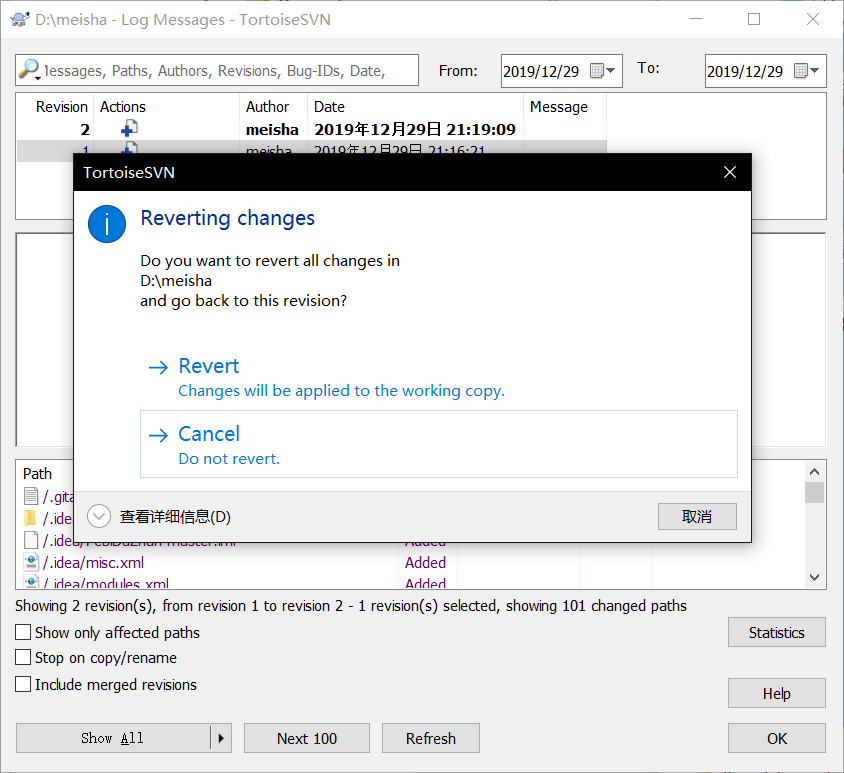


图2-14 回退确认

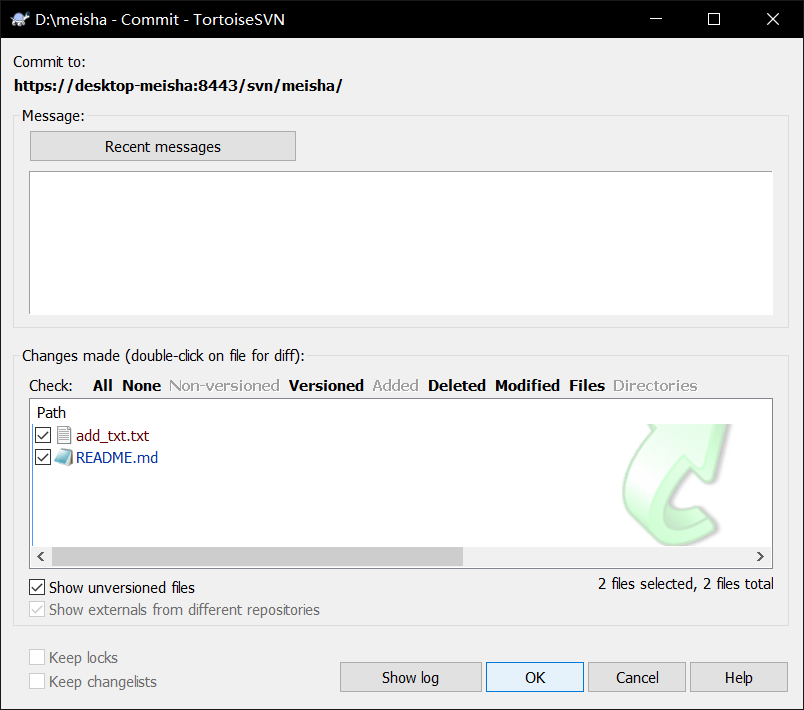


图2-15 commit

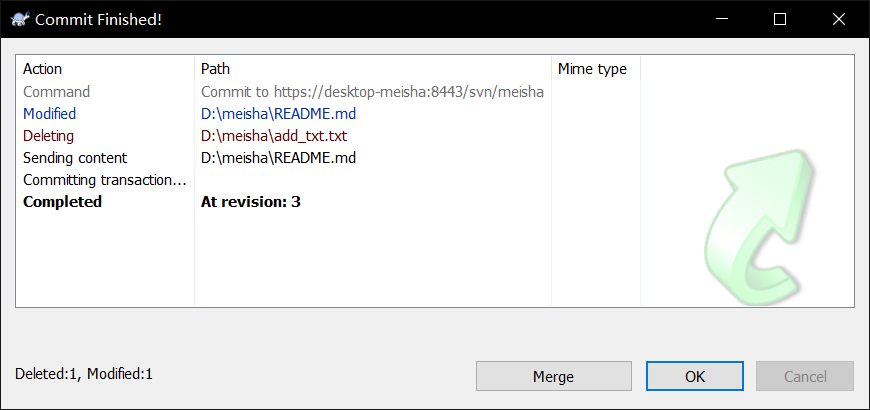


图2-16 commit finished

## 代码冲突检测

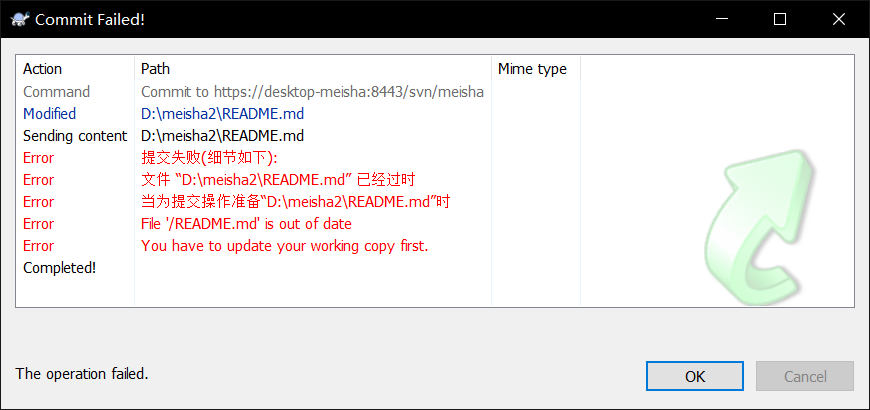


图2-17代码冲突检测

查找两段代码差异：

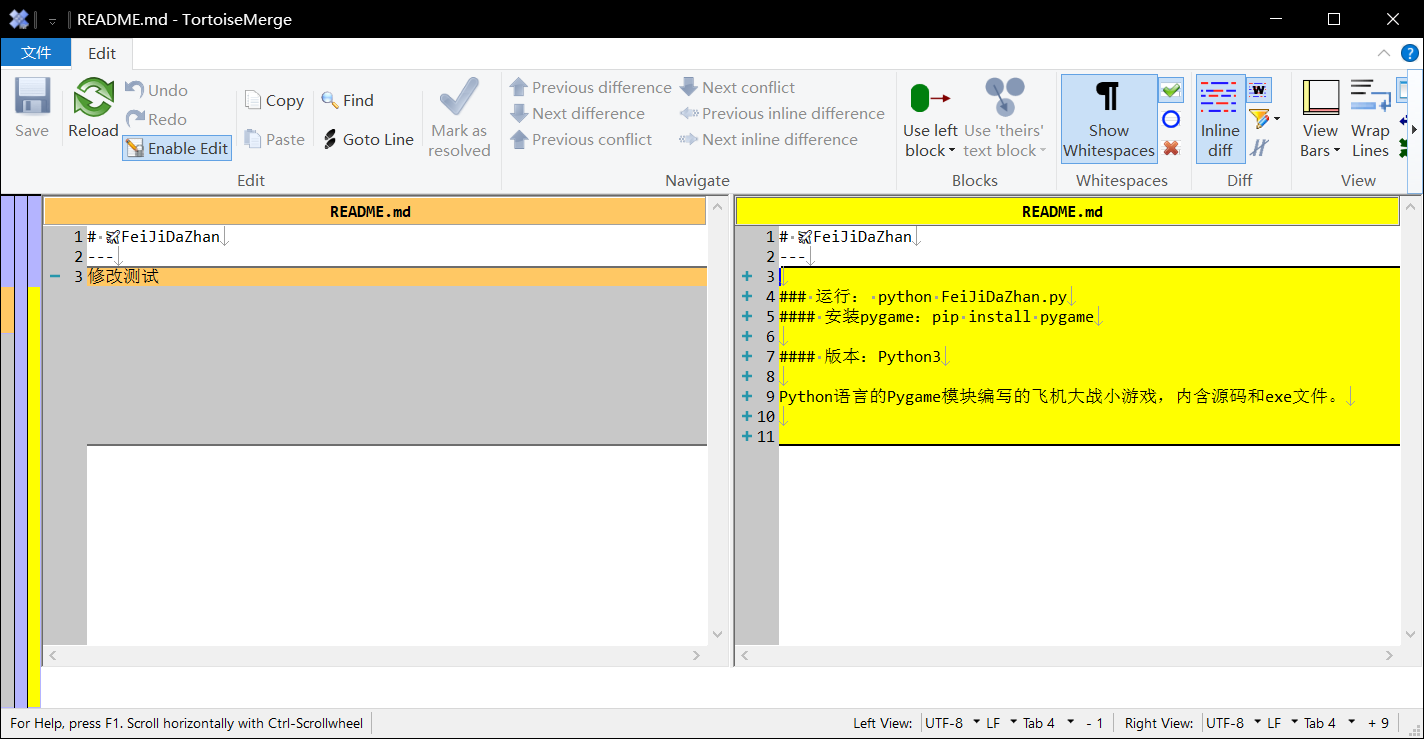


图2-18 冲突解决

修改完成后，将conflict文件标记为无冲突文件

# 利用Git与Github实现源代码版本管理

## Git与Github常用关系图

Git是客户端程序，通过Clone将Github远程仓库中的源代码克隆到本地，并将修改后源代码推送到Github远程仓库中，依此实现源代码版本管理。

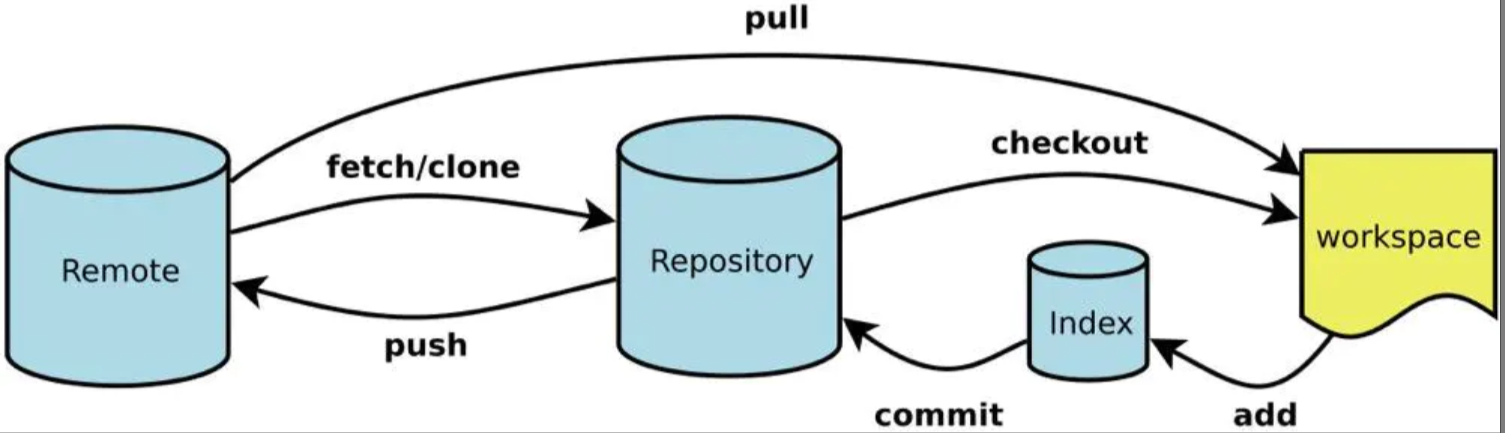


图3-1 git结构

## 安装git

Git下载链接<https://git-scm.com/downloads>

同样的打开安装包点击下一步即可。安装完成后在文件夹空白位置右键，选择Git bash here，打开命令行工具窗口。

## 配置git全局环境

git config --global user.name "meishax"

git config --global user.email [492895227@qq.com](mailto:492895227@qq.com)

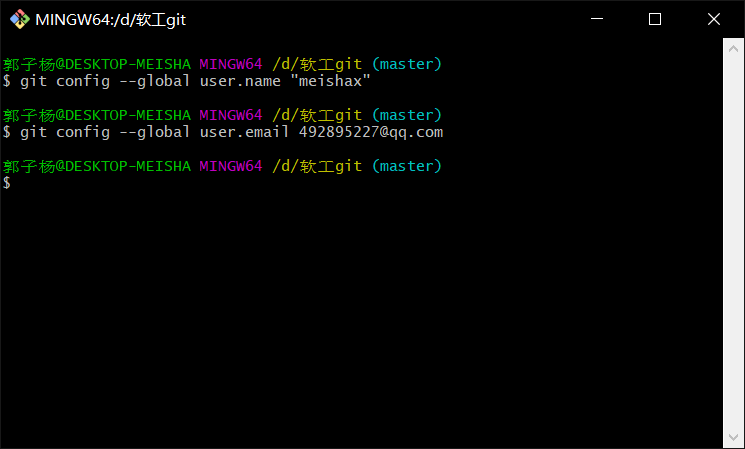


图3-2 配置全局环境

## 初始化git本地仓库

git init

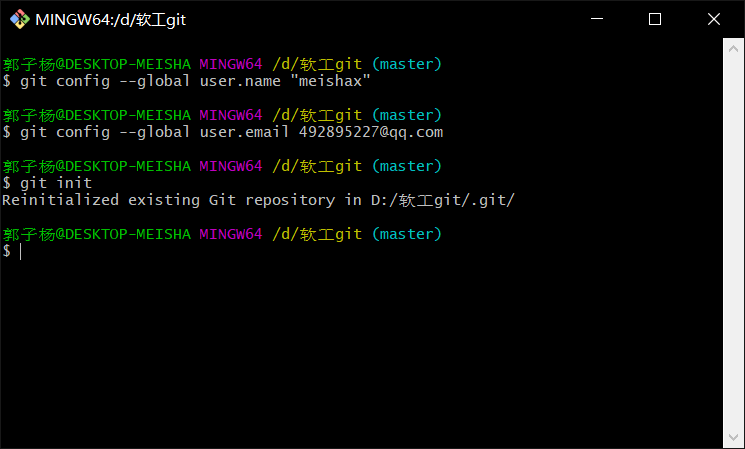


图3-3 初始化

初始化之后文件夹中多出了.git隐藏文件夹，初始化完成。新建文件readme.md。

## 提交文件

‘git add readme.md’ 添加readme.md文件。

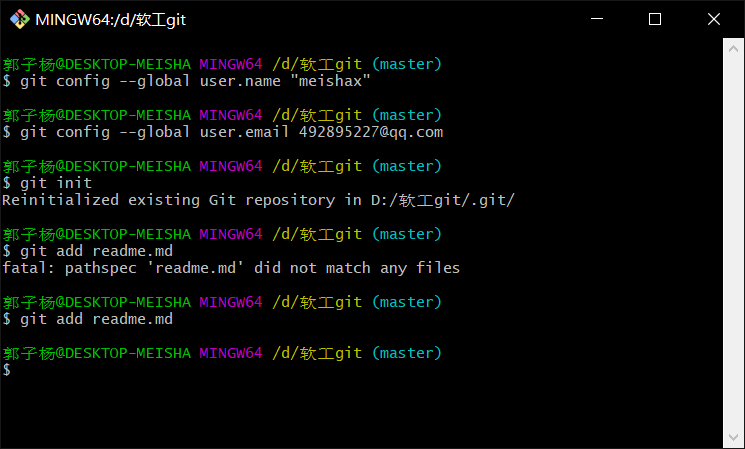


图3-4 添加文件

‘git commit -m “readme”’提交文件到版本库。

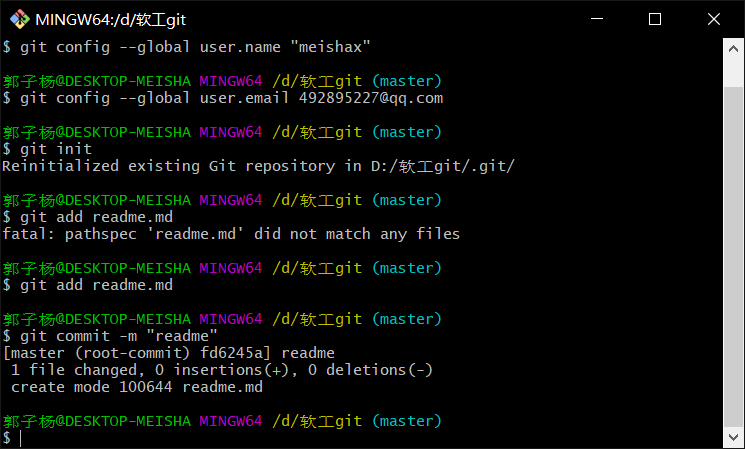


图3-5 提交文件

‘git status’查看当前状态。



图3-6 查看当前状态

## 推送到远程仓库

‘git remote add origin [git@github.com:meishax/gzy.git](mailto:git@github.com:meishax/gzy.git)’

‘git push origin master’

添加远程仓库地址

推送版本库到远程仓库

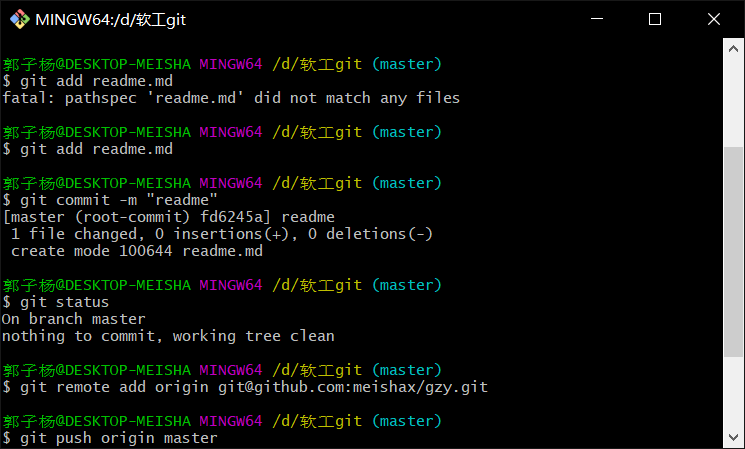


图3-7 push

远程仓库显示文件。

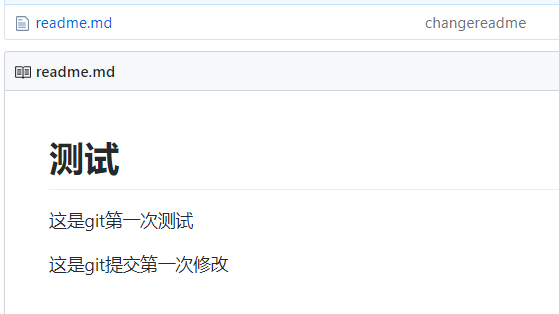


图3-8 github

# 利用idea内置git实现源代码版本管理

## 本地创建本地工作目录

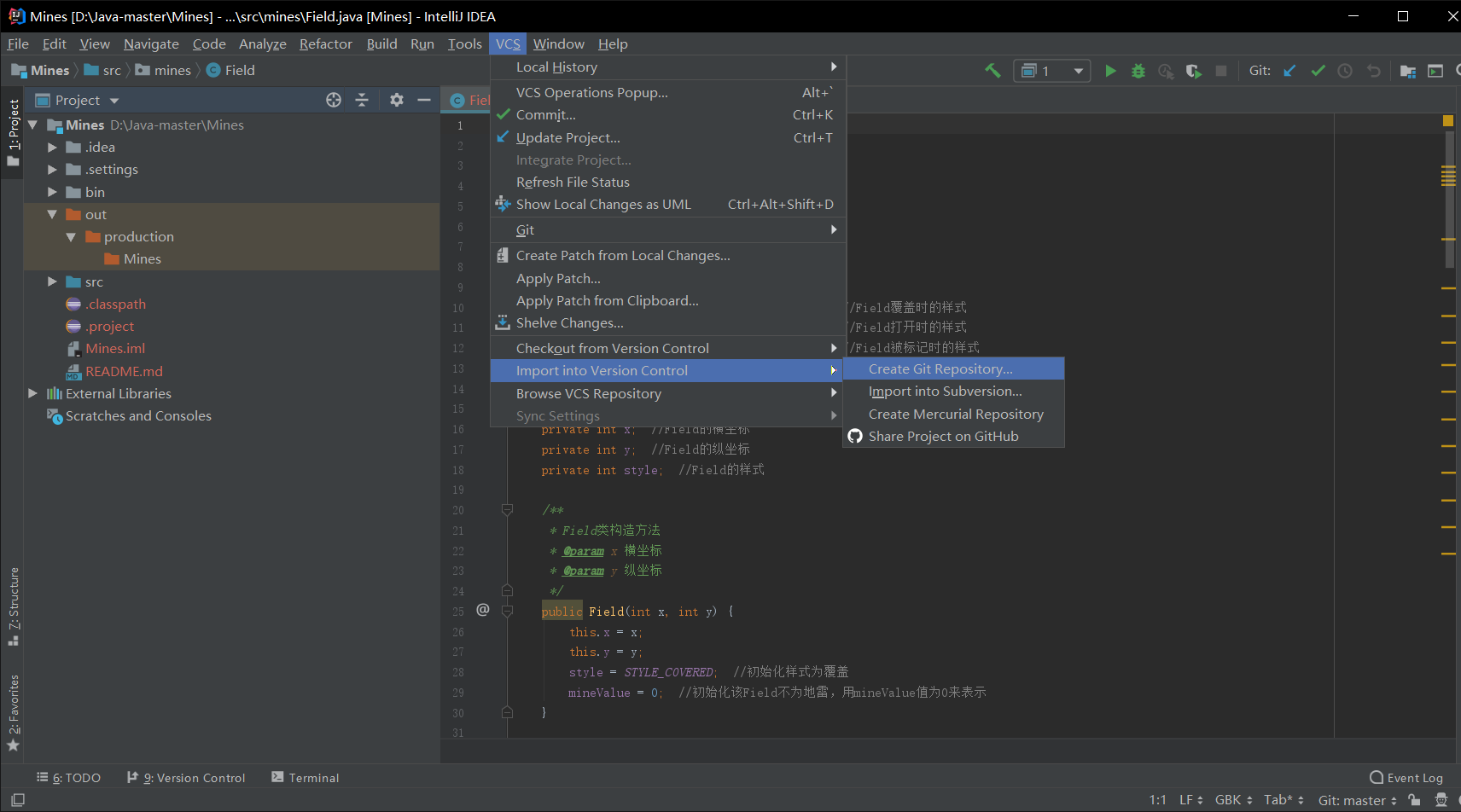


图4-1本地创建工作目录

## 利用git-add将本地项目源码添加至处于暂存区，并没有真正进入到版本库中

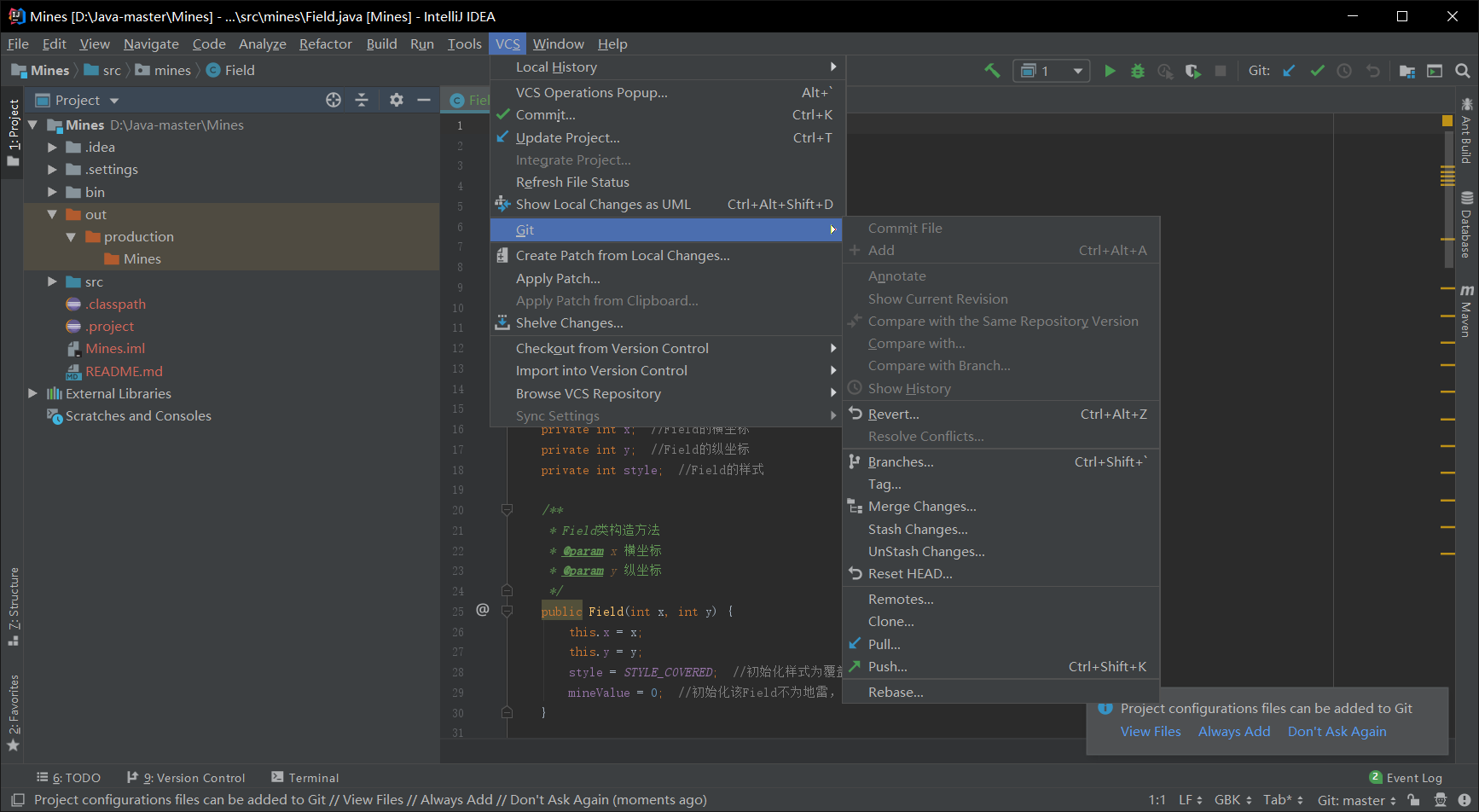


图4-2 git-add

## 利用git-commit将本地项目源码添加至本地版本库

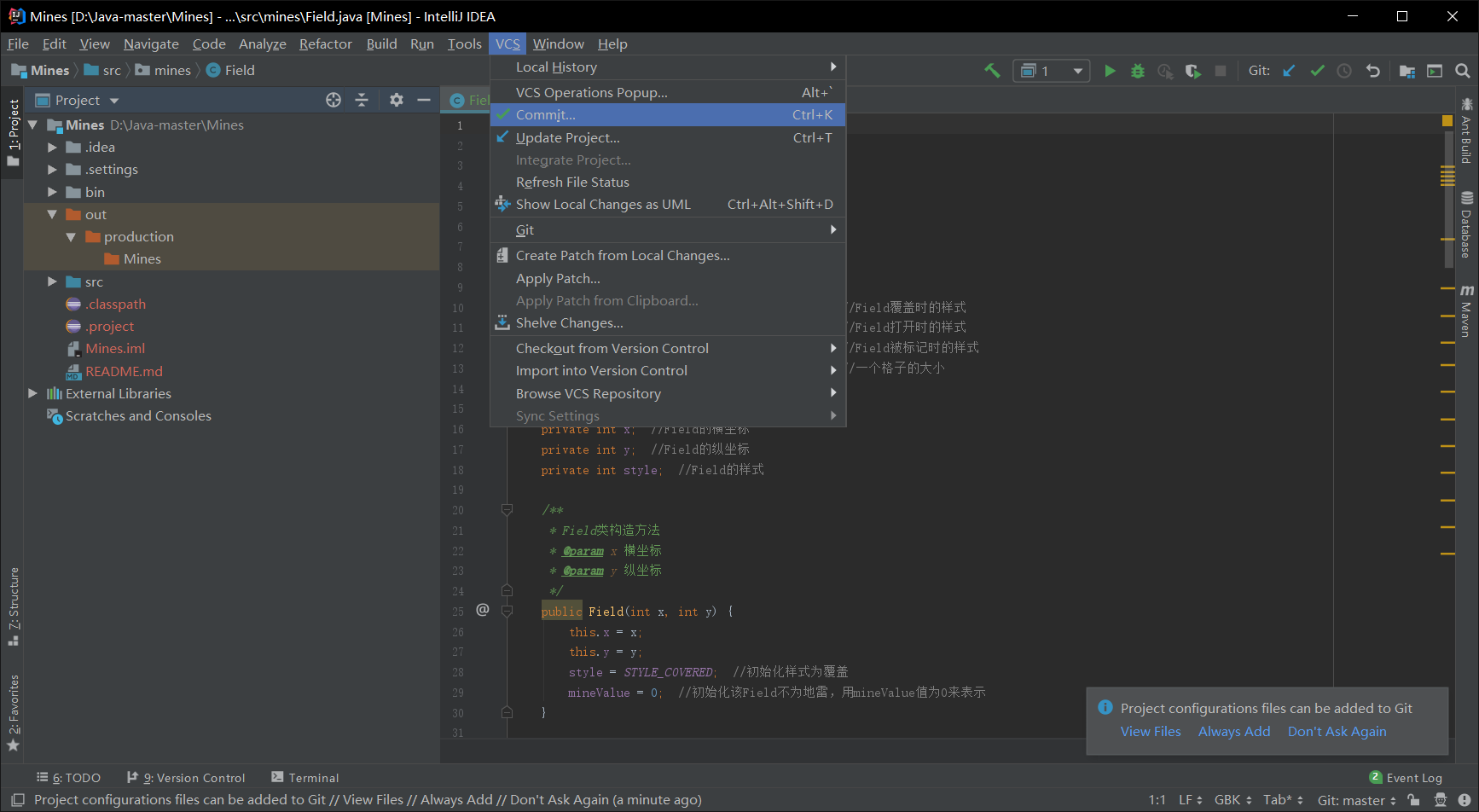


图4-3 git-commit

## 登陆

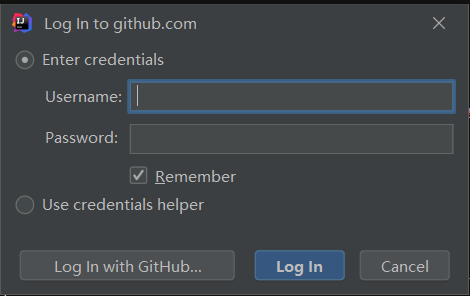


图4-4 登录

## 成功将源代码Push到github远程仓库中

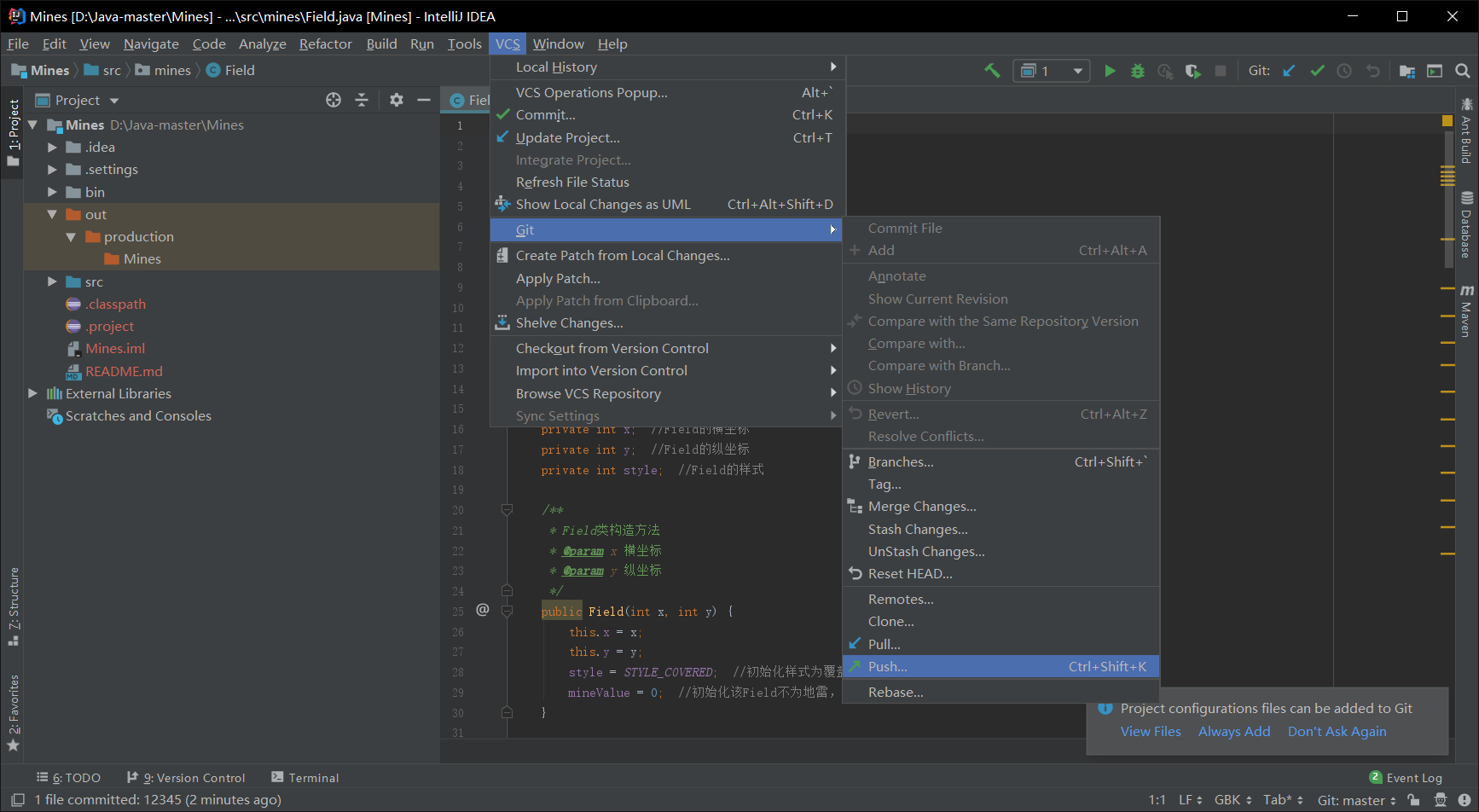


图4-5 push

## idea push成功反馈信息

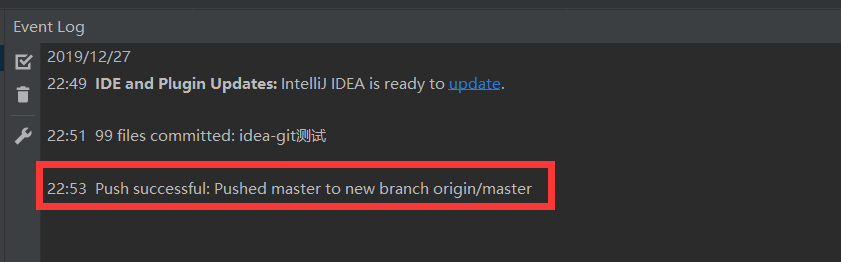


图4-6 成功信息

# 在Visual Studio环境下对所选系统的源代码进行版本管理

## Visual Studio安装源代码管理插件

直接在vs的扩展和更新中搜索SVN，可以安装ankhsvn。这里我安装的是微软开发的VisualSVN for Visual Studio 2017。下载之后重启vs即可。

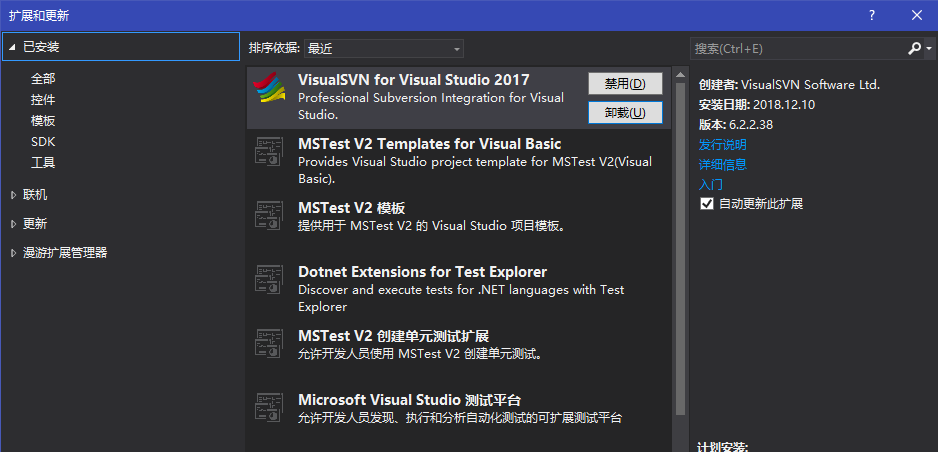


图5-1安装vs插件

新建项目之后在右下角点击按钮添加到源代码管理，先选择git。

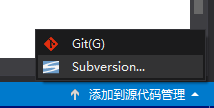


图5-2选择源代码管理工具

## VS中使用git

在github网页中将仓库地址复制粘贴过来，和远程仓库建立连接。



图5-3发布Git存储库

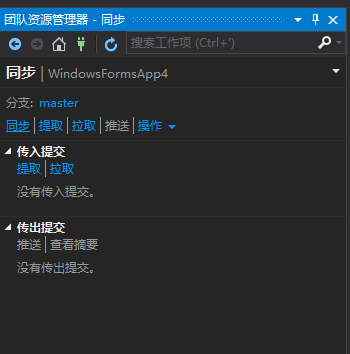


图5-4当前状态

提交当前更改信息，步骤同之前的git add和git commit。提交更改信息。

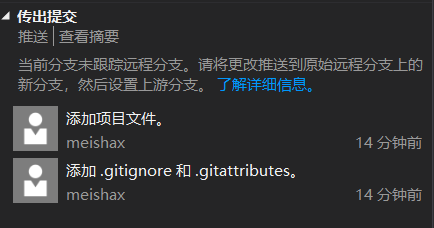


图5-5提交更改信息

提交更改信息之后就可以推送到远程仓库。

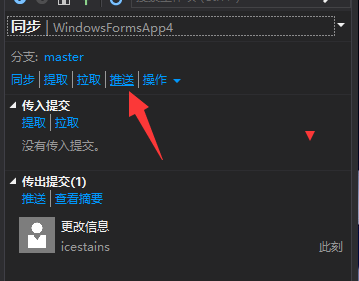


图5-6推送到远程仓库

推送成功远程仓库文件。



图表 5-7远程仓库结果

VS 2017项目同样的可以使用命令行工具进行同步操作

## VS使用subversion

选择当前文件夹：

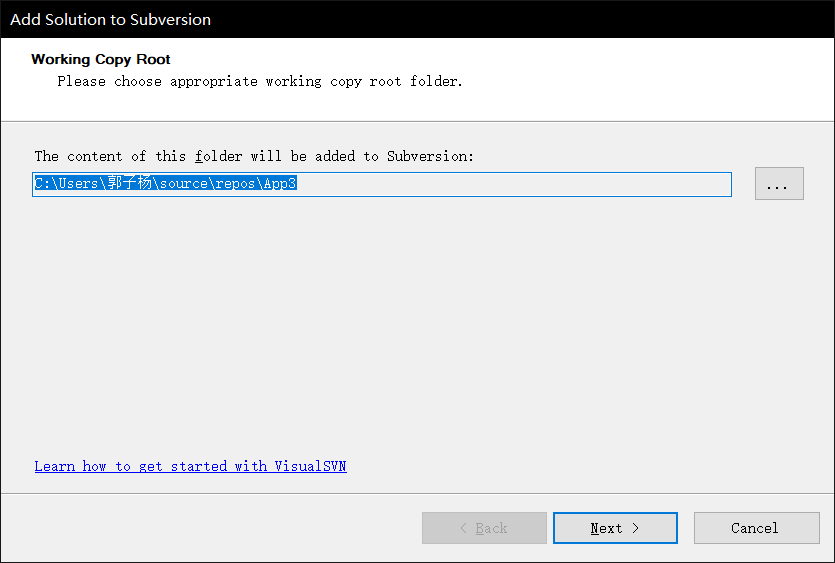


图5-8选择当前目录作为工作空间

创建新的仓库：

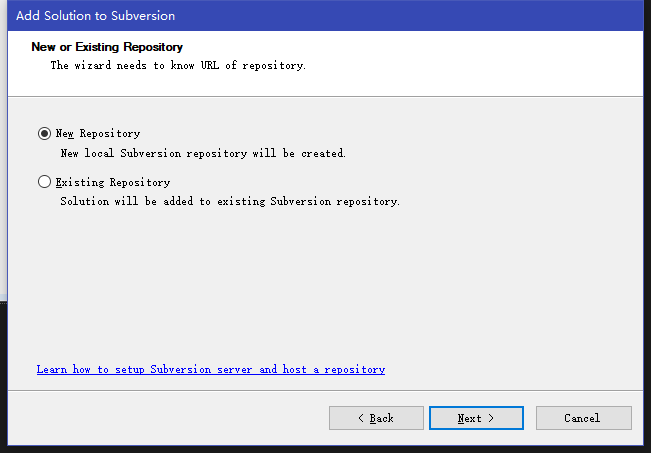


图5-9创建新的版本库

新版本库位置

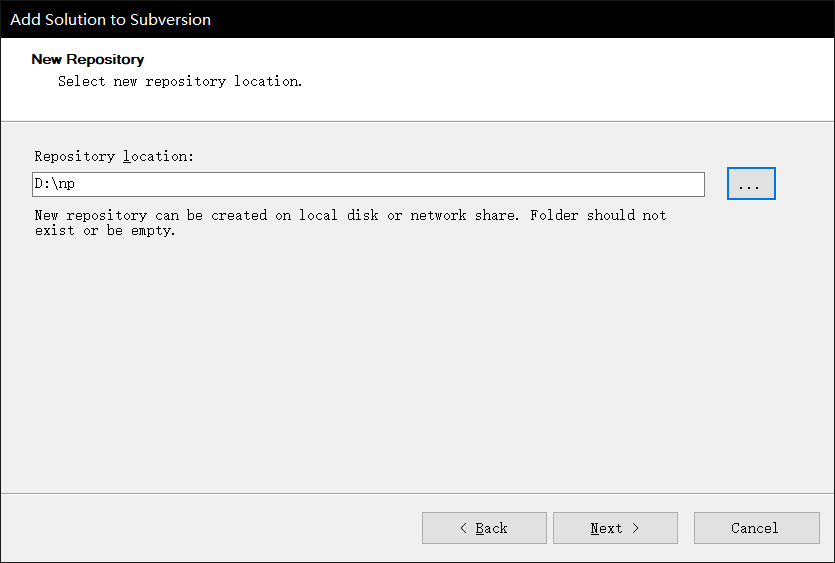


图5-10新的版本库路径

确认信息

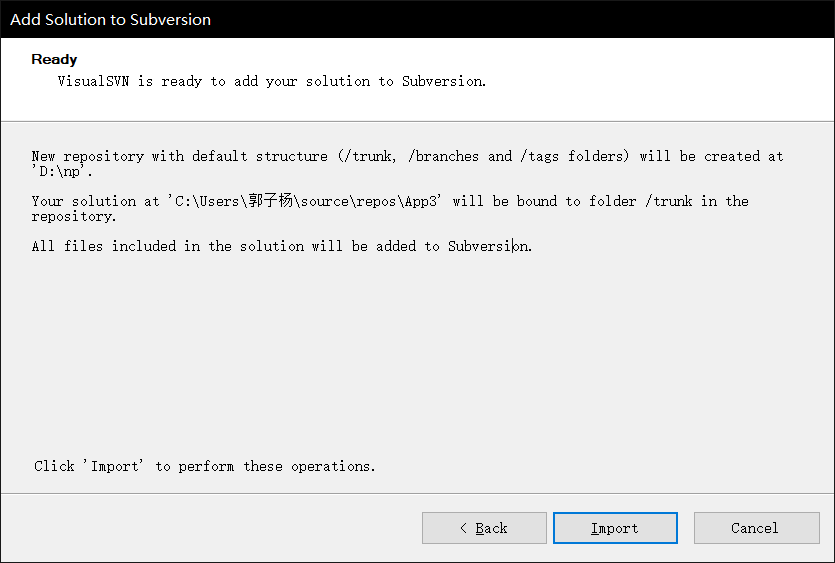


图5-11确认信息

成功

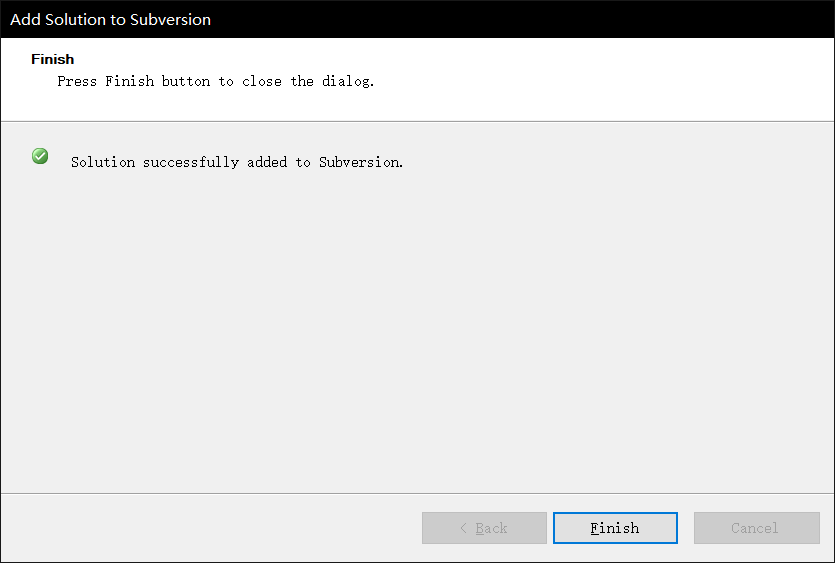


图5-12成功

# 实验感悟

通过本次实验，学习了如何使用SVN，Git等工具来实现源代码管理及版本控制。团队开发中，版本控制和分布式开发的高效率功能让一个团队的工作在效率上有极大地提升。各个编辑器通过集成或者安装插件的方式，在开发过程中给开发人员提供了极大地便利。