济南大学

软件工程管理

济南大学信息学院　刘鹗　ise_liuk@ujn.edu.cn

济南大学信息科学与工程学院
School of Information Science and Engineering

初识Git

- 什么是版本管理
- 版本管理有哪些
- 为什么需要版本管理
- 为什么选择 Git

选择Git的理由

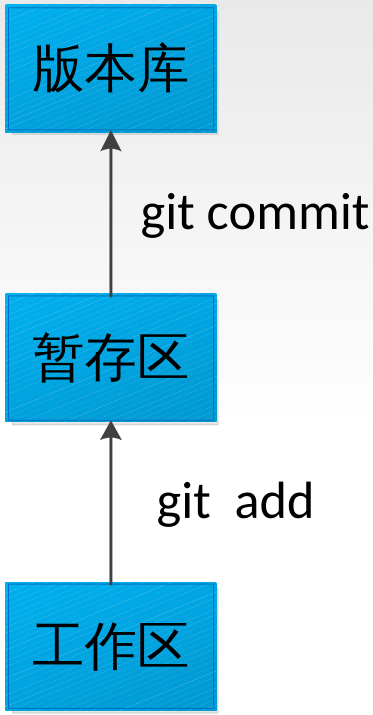
- 每日工作备份
- 异地协同工作
- 现场版本控制
  - 将工作成果创建补丁文件，保存打包走。

（1）SVN 很难针对每次提交逐一创建补丁，一般使用比较命令和最早的提交进行比较，以创建一个大补丁文件。

（2）使用 Git 来生成补丁文件打包走是很容易的事情

Git暂存区

- 暂存区的设计是 Git 设计最成功的地方之一，同时也是最难理解的
- .git/index 跟踪工作区文件的时间戳和长度信息是否改变，从而实现对工作区状态进行快速扫描的目的，这也是 Git 很高效的原因之一。
- 文件内容发生改动，不是记录在 .git/index 中，而是记录在 .git/objects 目录下的一个新对象中
- 如果执行了 git add 操作，暂存区目录树将被更新，文件内容会被记录在对象库中的一个新的对象中



Git对象

- Git 采用 40 位十六进制的 SHA1 哈希值作为提交的 commit-id

- 哈希（hash）是一种数据摘要算法（或称散列算法），是信息安全领域中重要的理论基石。比较著名的摘要算法有： MD5 和 SHA1 。

- Subversion 是集中式版本 控制系统，容易实现递增的全局唯一提交号。 Git 作为分布式版本控制系统，开发可以是非线性的，每个人都可以克隆不同版本到本地进行开发，与版本库交互（拉回很推送）而互相分发，采用唯一数字编号，容易冲突。这就不能要求是本地局部唯一，而是“全球唯一”了。所以 Git 采用哈希算法，可以实现及时输入数据量非常大，差异非常小，两者的哈希值也能显著不同。

Git的诞生

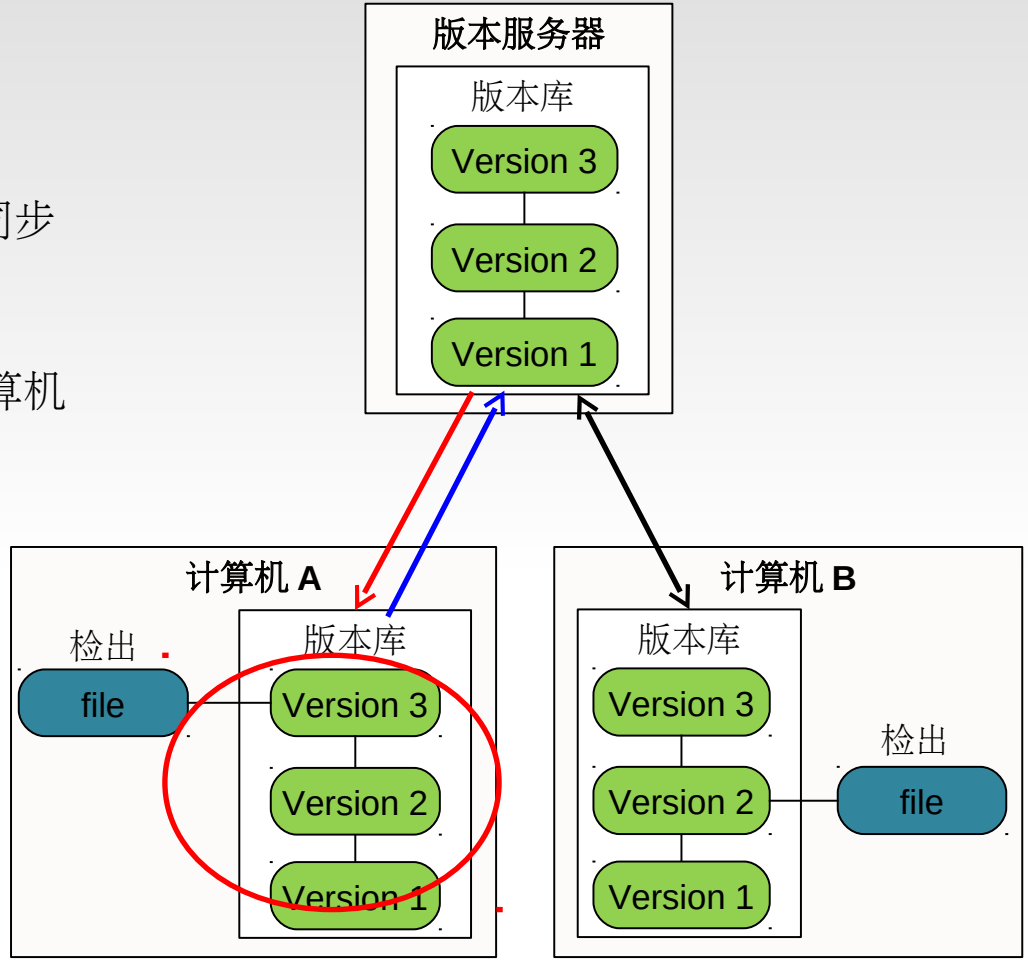
- 诞生于2005年，因“穷”而生： Bitkeeper收回了Linux社区免费使用的权利，因此Linux社区（主要是Linus）自己开发了Git；
- 特点：速度快、设计简单、分布式、有能力高效管理超复杂的项目（Linux内核）

Git的发展

- 大部份开源软件
- Github.com
- Gitcafe.com
- Code.csdn.net
- Coding.net
- 开源中国（拥抱变化，好好学习，天天向上）

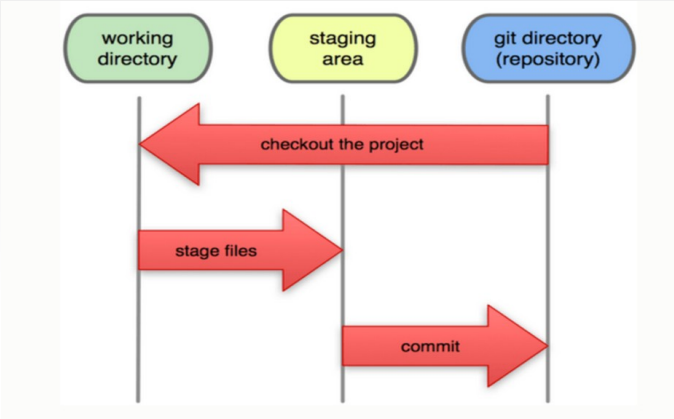
Git结构

- ◆版本库初始化
  - 个人计算机从版本服务器同步
- ◆操作
  - 90% 以上的操作在个人计算机
  - 添加文件
  - 修改文件
  - 提交变更
  - 查看版本历史等
- ◆版本库同步
  - 将本地修改传送
  - 到版本服务器



Git文件状态

- 任意一个文件，在git内只有三种状态：已提交（committed），已修改（modified）和已暂存（staged）

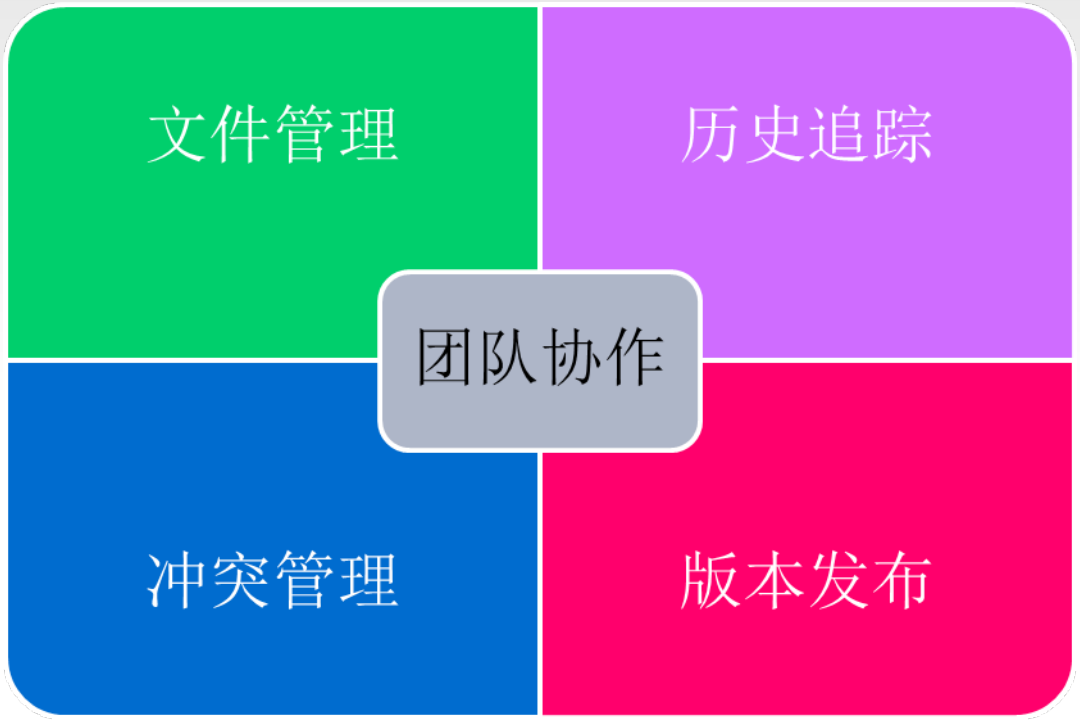


已提交：Git中有特定版本的文件

已修改：从Git中clone出来，并且做了修改

已暂存：做了修改，并放入暂存区

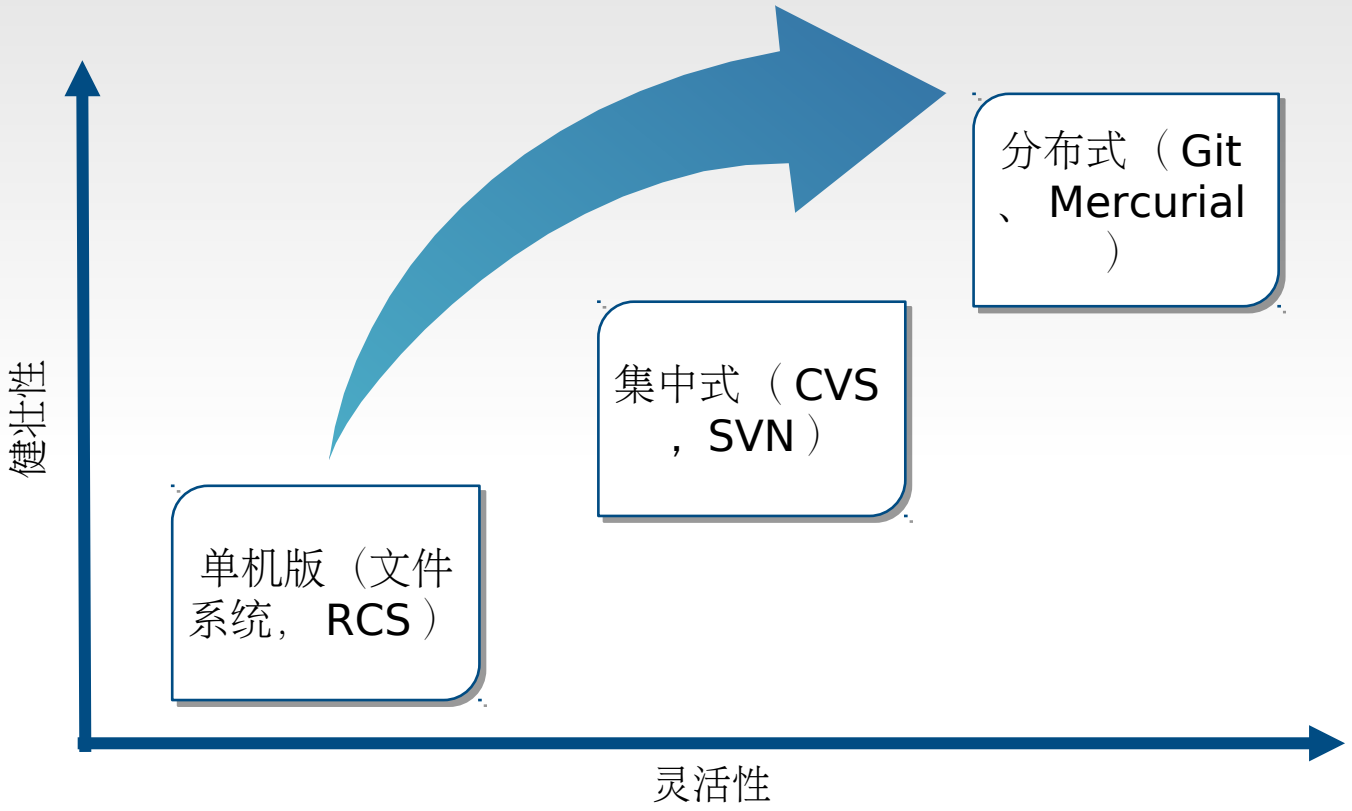
版本管理的作用



常见的版本管理软件

简　称	全　名
CVS	Concurrent Versions System
VSS	Micorosoft Visual SourceSafe
SVN	Subversion
Git	
TFS	Team Foundation Server
Mercurial	
ClearCase	IBM Rational ClearCase
Perforce	

发展过程

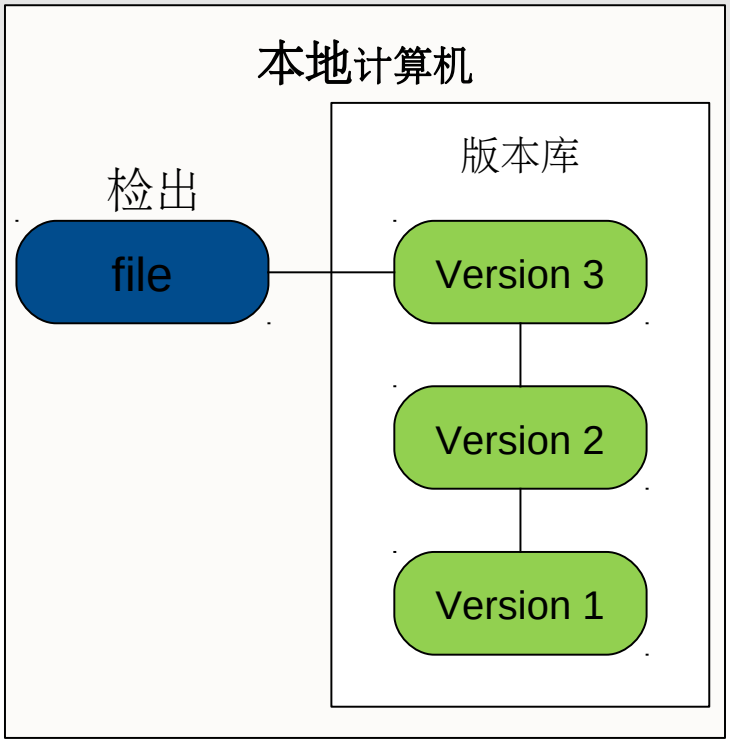


发展过程

- 版本库：个人电脑/服务器
- RCS：

  Revision Control System

  可追踪修改历史
- 问题：如何协作?



发展过程

- 版本库：版本服务器
- VCS：

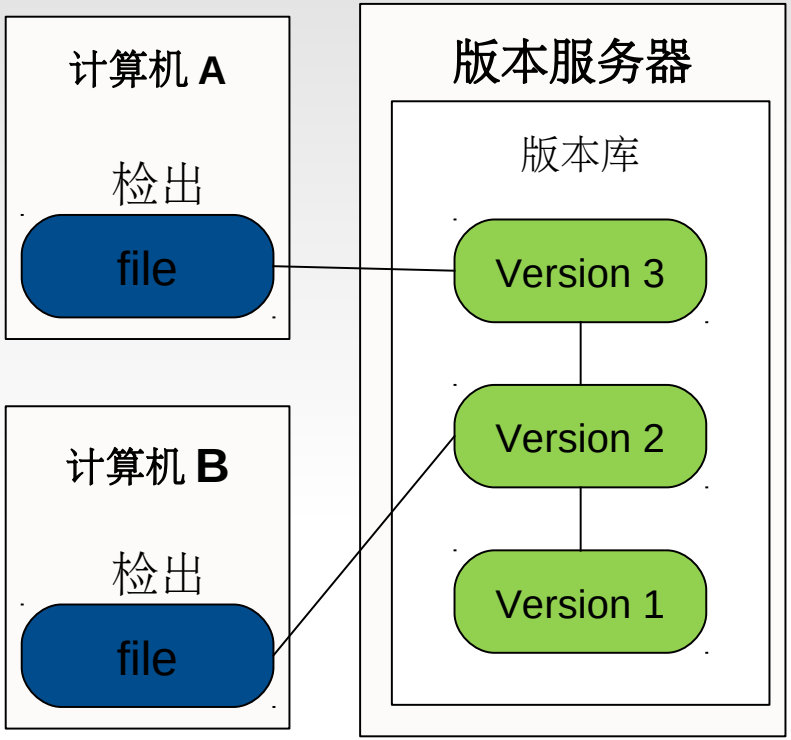
  Version Control System

  又称 CVCS （Central VCS）

  有 SVN、CVS、firefly 等产品
- 问题：

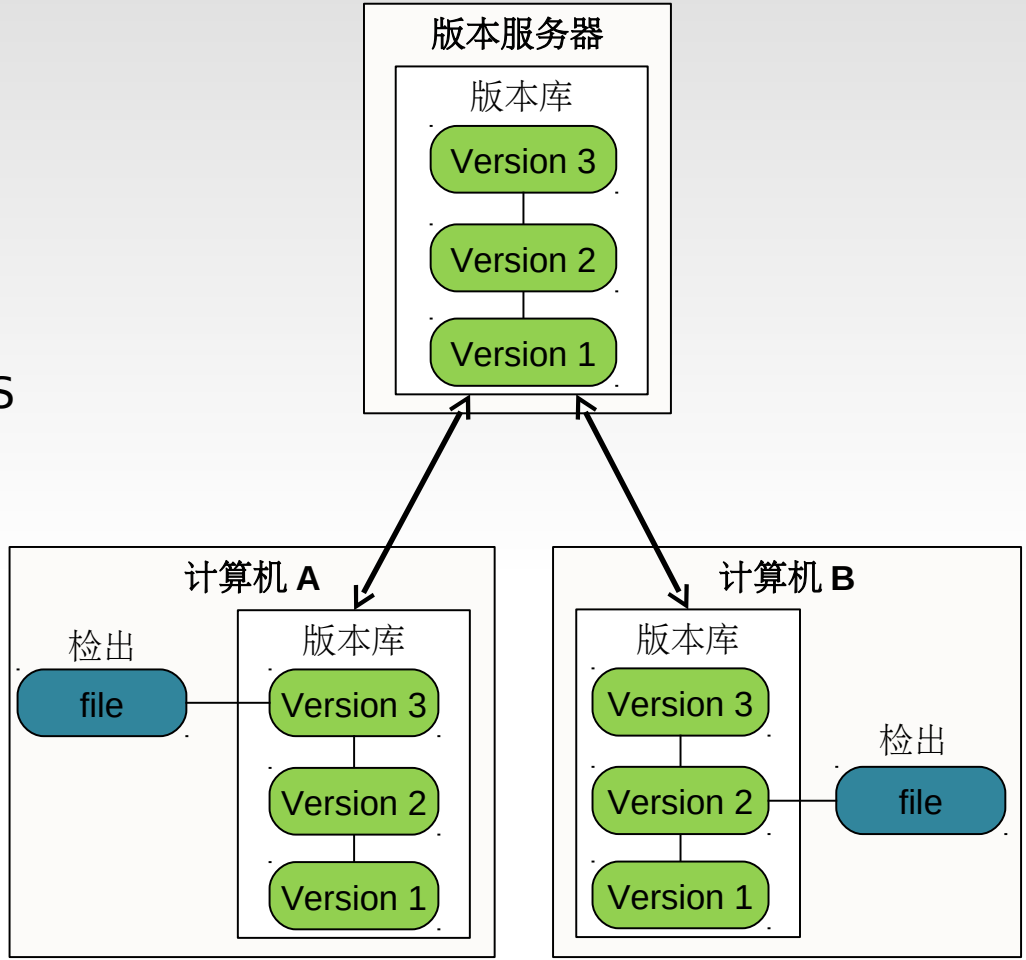
  服务器会停

  网络会不通



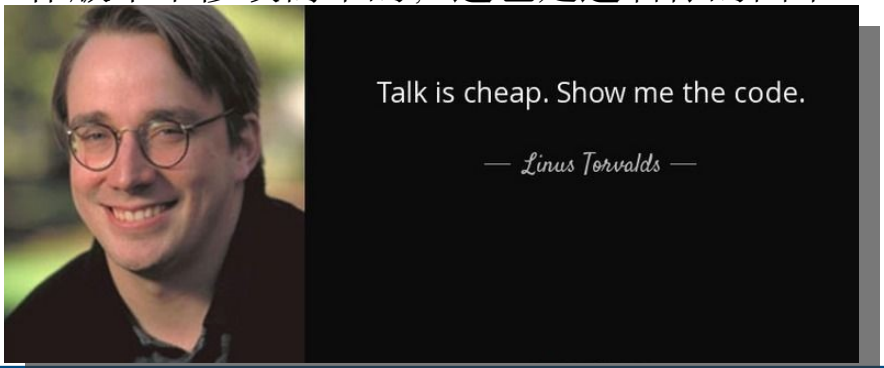
发展过程

- 版本库
  - 版本服务器
  - 个人计算机
- VCS：
  - 又称 DVCS （Distributed VCS）
  - 有 GIT、HG、bazaar 等产品
- 问题：
  - 乱成一团?
  - 分支管理机制



常用术语

- 分支 (Branch)：在一个时间点，复制一份处于版本控制之下的文件，从这之后，这两份拷贝就可以独立的互不干扰的进行各自开发。
- 取出 (Check-out)：一次“取出”，就是在本地创建一份仓库的工作拷贝。
- 提交 (Commit)：一次“提交”，将本地的修改写回到仓库或合并到仓库。
- 冲突 (Conflict)：当开发者们同时提交对同一文件的修改，而且版本系统不能把它们合并到一起，就会引起冲突，就需要人工来进行合并。
- 合并 (Merge)：合并就是把所有对文件的修改统一到文件里
- 仓库 (Repository)：仓库就是处于版本控制之下的文件所在的地方，通常在服务器端。
- 工作版本（Working copy）：从档案库中取出一个本地端的复制，所有在档案库中的档案更动，都是从一个工作版本中修改而来的，这也是这名称的由来。



常用术语

- GitHub，全球最大的代码托管网站，超过1000万人使用
- 提供了图形化的界面，可以查看代码、wiki、issue、pull request、star、fork等功能；
- 提供了社交化功能；
- 著名开源的项目都已迁移到Github上，ror，nodejs，bootstrap，jquery等；
- 一些政府也在Github上发布源代码和数据集（白宫）；
- 对普通用户免费，也提供商业服务；

GitHub

版本控制的作用

- 版本控制的作用
  - A 、版本控制
  - B 、项目的基本管理
  - C 、团队协作开发
  - D 、历史记录
  - E 、文件跟踪

两个常用的GIT服务商

- 两个常用的 GIT 服务商
  - 国外：github　https://github.com/
  - 国内：gitee　https://gitee.com/

GIT的工作原理

- 数据存储的两个位置
  - GIT 服务器，保存了一个团队成员共享的项目副本
  - 本地仓库，每个用户都会在自己的电脑上创建一个本地仓库，用户编辑的项目存储在本地仓库中，为了实现代码和文件的共享，必须及时将成果提交到 GIT 服务器
- 每个用户都有一个分支，团队成员在各自的分支上工作
  - 其中， master 分支主要用于代码整合，其他分支用于团队成员开发，因为分支相互独立，可以减少互相干扰。

GIT使用的两个方面

- GIT 使用的两个方面
  - 将项目上传到 GIT 服务器
  - 将项目从 GIT 服务器下载下来用于编辑

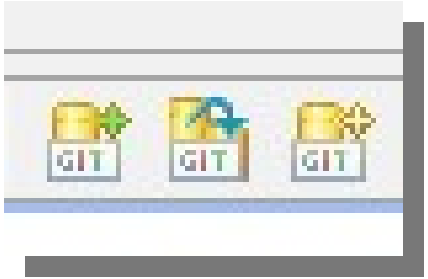
将项目上传到GIT服务器

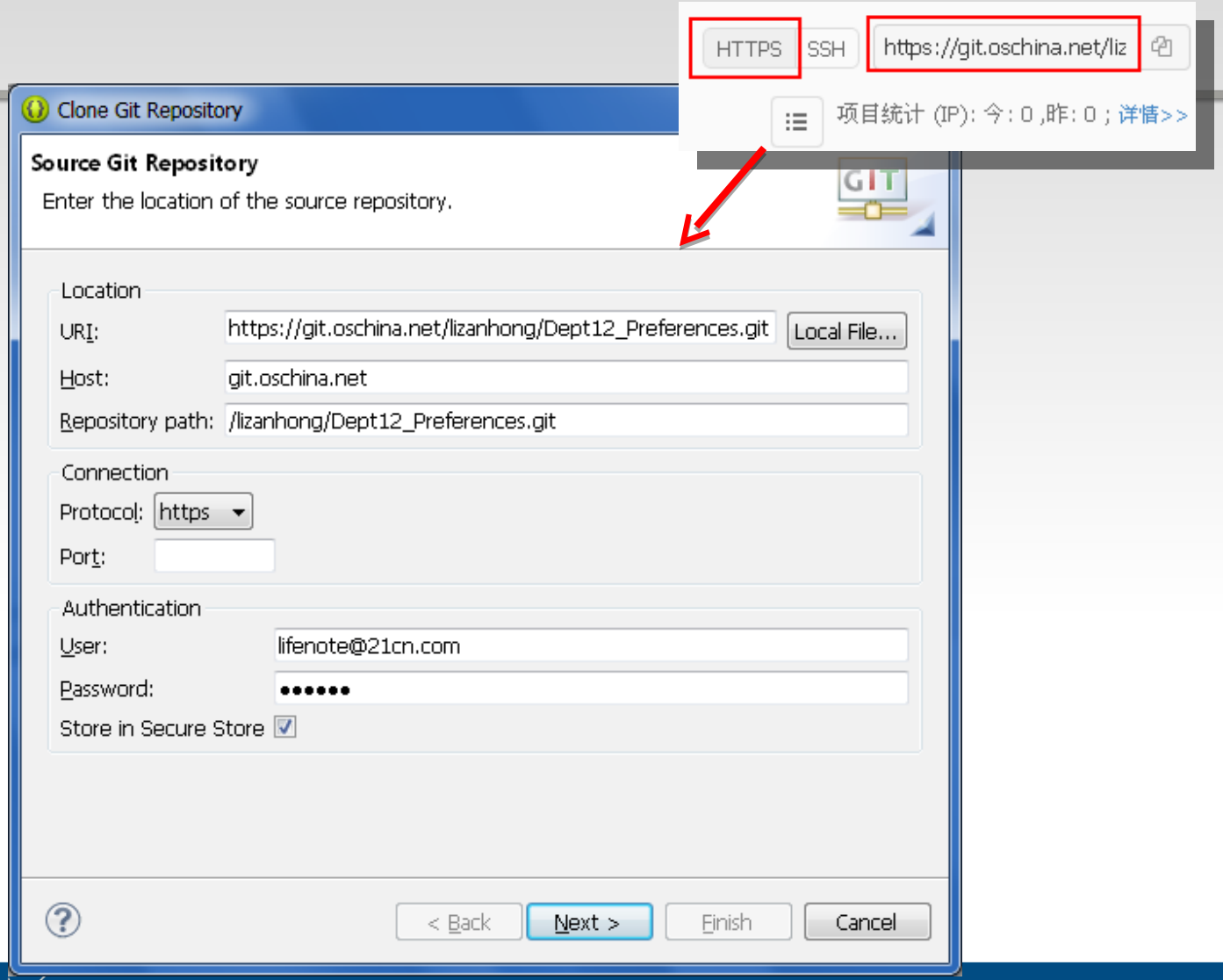
- 将项目上传到 GIT 服务器
  - 在 GIT 服务器创建项目
  - 在 Eclipse ADT 中克隆该项目
  - 将工程共享到项目中
  - 提交工程到 GIT 服务器

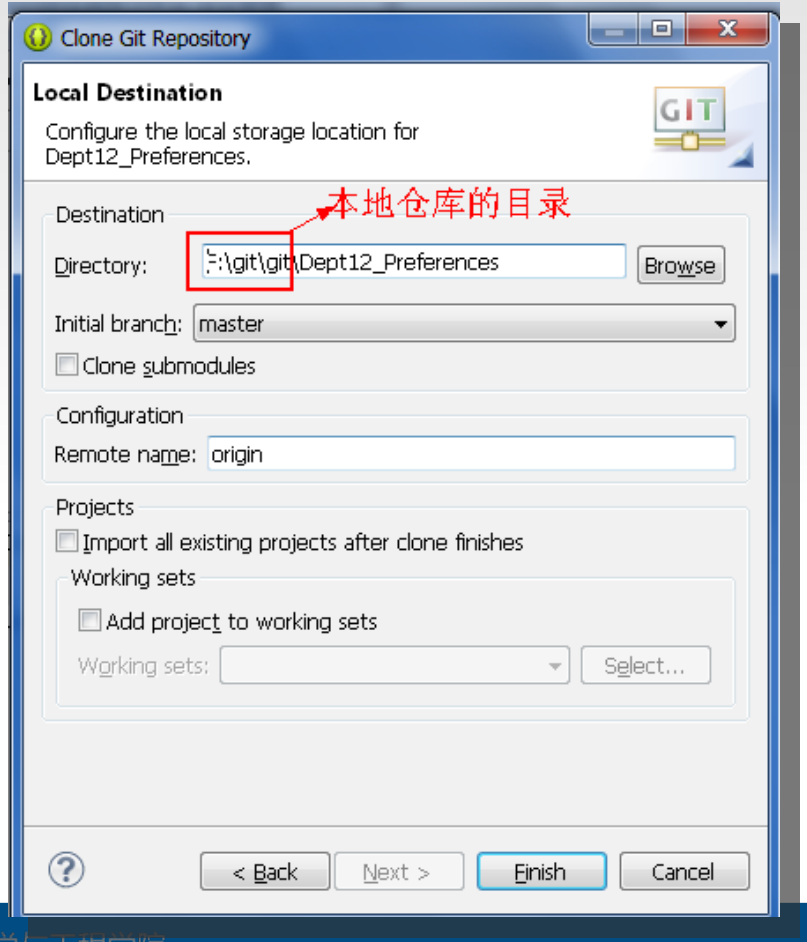
在GIT服务器创建项目

从Git服务器复制项目

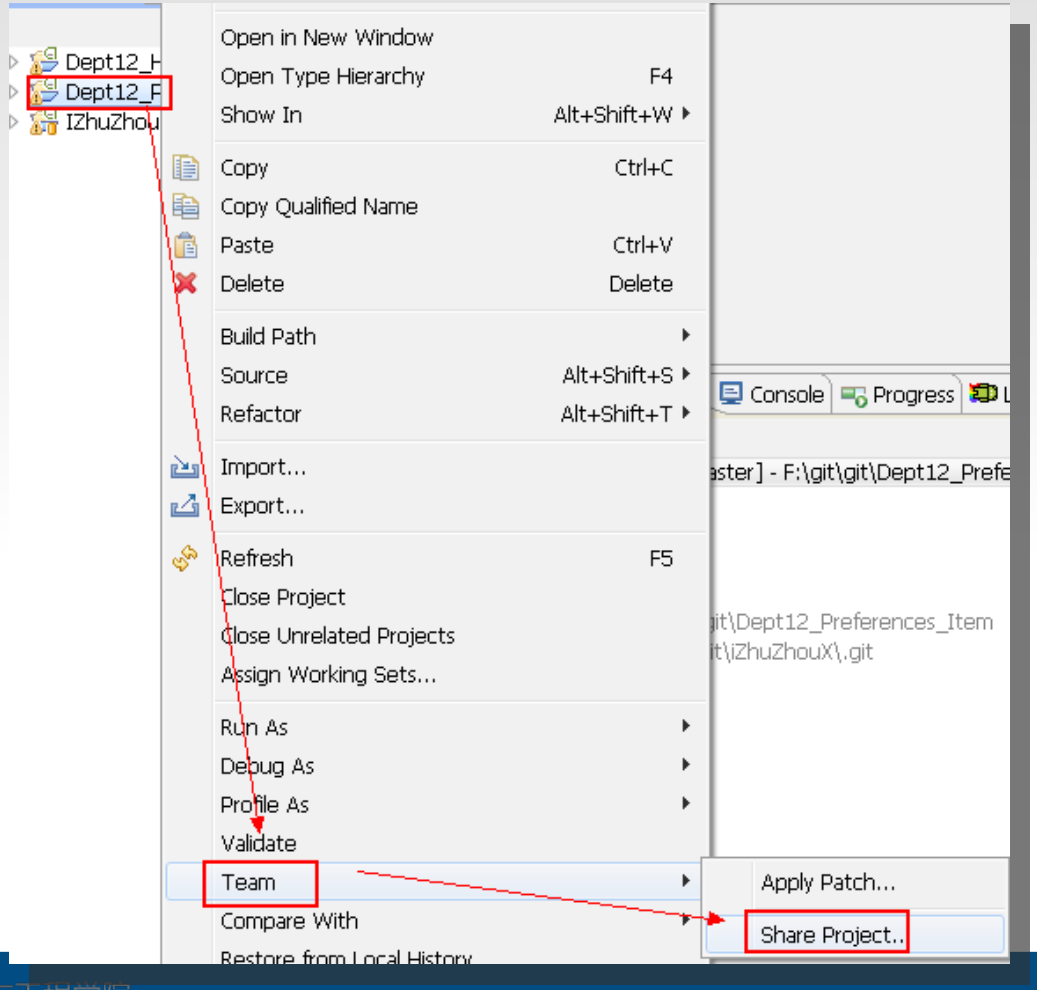
- 在 Eclipse 中打开 Git Repository 透视图
  - 第一个：将工程添加到已有的本地仓库
  - 第二个：从远程 Git 服务器复制项目到本地仓库
  - 第三个：创建本地仓库，并将工程添加到该仓库

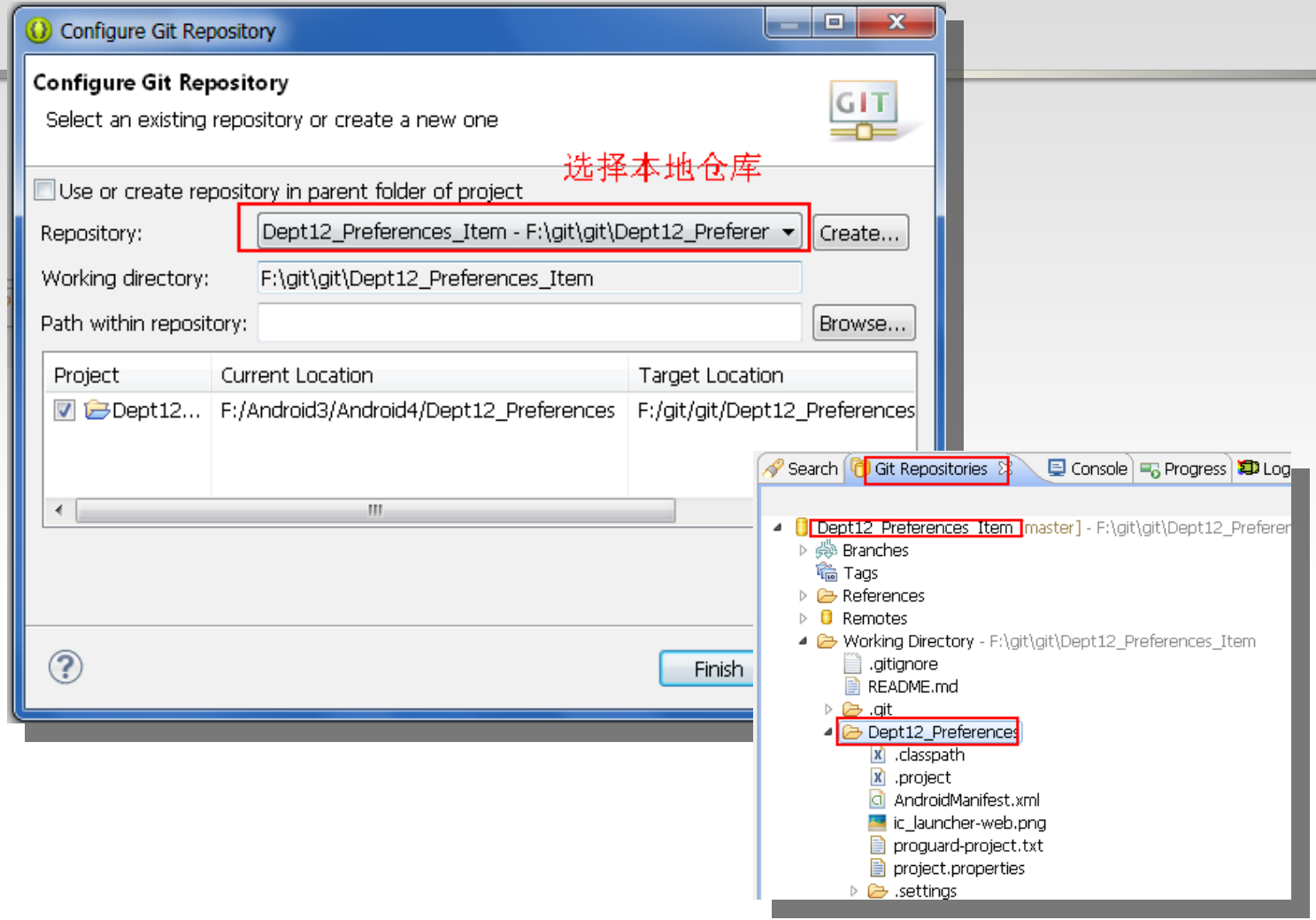



HTTPS
SSH
https://git.oschina.net/liz
项目统计 (IP): 今: 0 ,昨: 0 ; 详情>>
Clone Git Repository
Source Git Repository
Enter the location of the source repository.
GIT
Location
URI:
https://git.oschina.net/lizanhong/Dept12_Preferences.git
Local File...
Host:
git.oschina.net
Repository path:
/lizanhong/Dept12_Preferences.git
Connection
Protocol:
https
Port:
Authentication
User:
lifenote@21cn.com
Password:
Store in Secure Store
< Back
Next >
Finish
Cancel

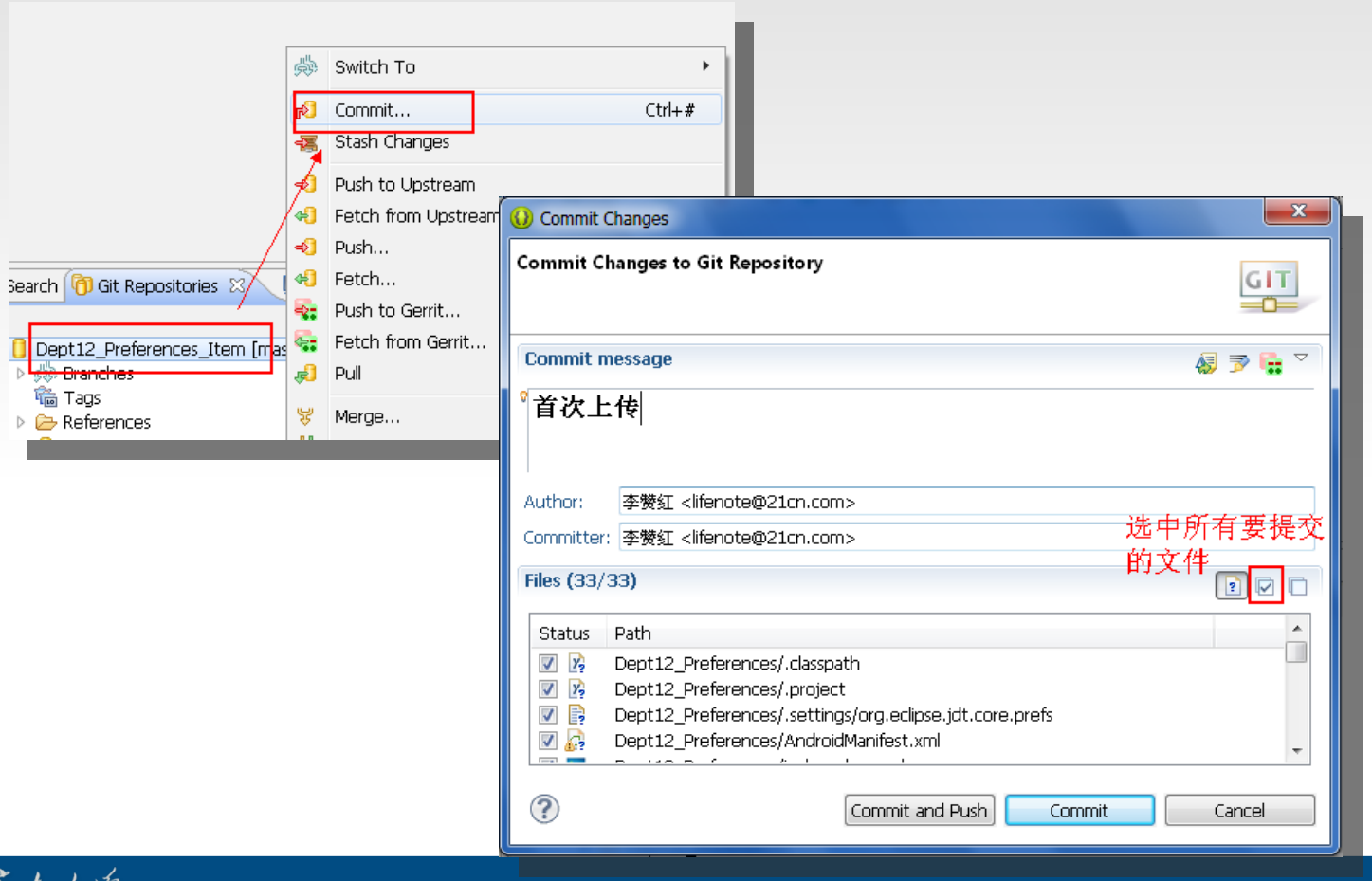

Clone Git Repository
Local Destination
Configure the local storage location for Dept12_Preferences.
GIT
本地仓库的目录
Destination
Directory:
F:\git\git\Dept12_Preferences
Browse
Initial branch:
master
Clone submodules
Configuration
Remote name:
origin
Projects
Import all existing projects after clone finishes
Working sets
Add project to working sets
Working sets:
Select...
< Back
Next >
Finish
Cancel

将工程共享到本地仓库

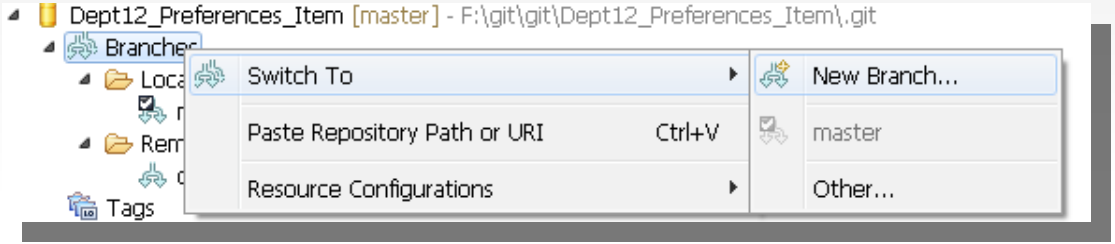

Dept12_H
Dept12_F
IZhuZhou
Open in New Window
Open Type Hierarchy F4
Show In Alt+Shift+W
Copy Ctrl+C
Copy Qualified Name
Paste Ctrl+V
Delete Delete
Build Path
Source Alt+Shift+S
Refactor Alt+Shift+T
Import...
Export...
Refresh F5
Close Project
Close Unrelated Projects
Assign Working Sets...
Run As
Debug As
Profile As
Validate
Team
Compare With
Restore from Local History
Console
Progress
aster] - F:\git\git\Dept12_Prefe
it\Dept12_Preferences_Item
t\iZhuZhouX\.git
Apply Patch...
Share Project...


Configure Git Repository
Configure Git Repository
Select an existing repository or create a new one
GIT
选择本地仓库
Use or create repository in parent folder of project
Repository:
Dept12_Preferences_Item - F:\git\git\Dept12_Preferer
Create...
Working directory:
F:\git\git\Dept12_Preferences_Item
Path within repository:
Browse...
Project
Current Location
Target Location
Dept12...
F:/Android3/Android4/Dept12_Preferences
F:/git/git/Dept12_Preferences
Finish
Search
Git Repositories
Console
Progress
Log
Dept12_Preferences_Item [master] - F:\git\git\Dept12_Preferer
Branches
Tags
References
Remotes
Working Directory - F:\git\git\Dept12_Preferences_Item
.gitignore
README.md
.git
Dept12_Preferences
.classpath
.project
AndroidManifest.xml
ic_launcher-web.png
proguard-project.txt
project.properties
.settings

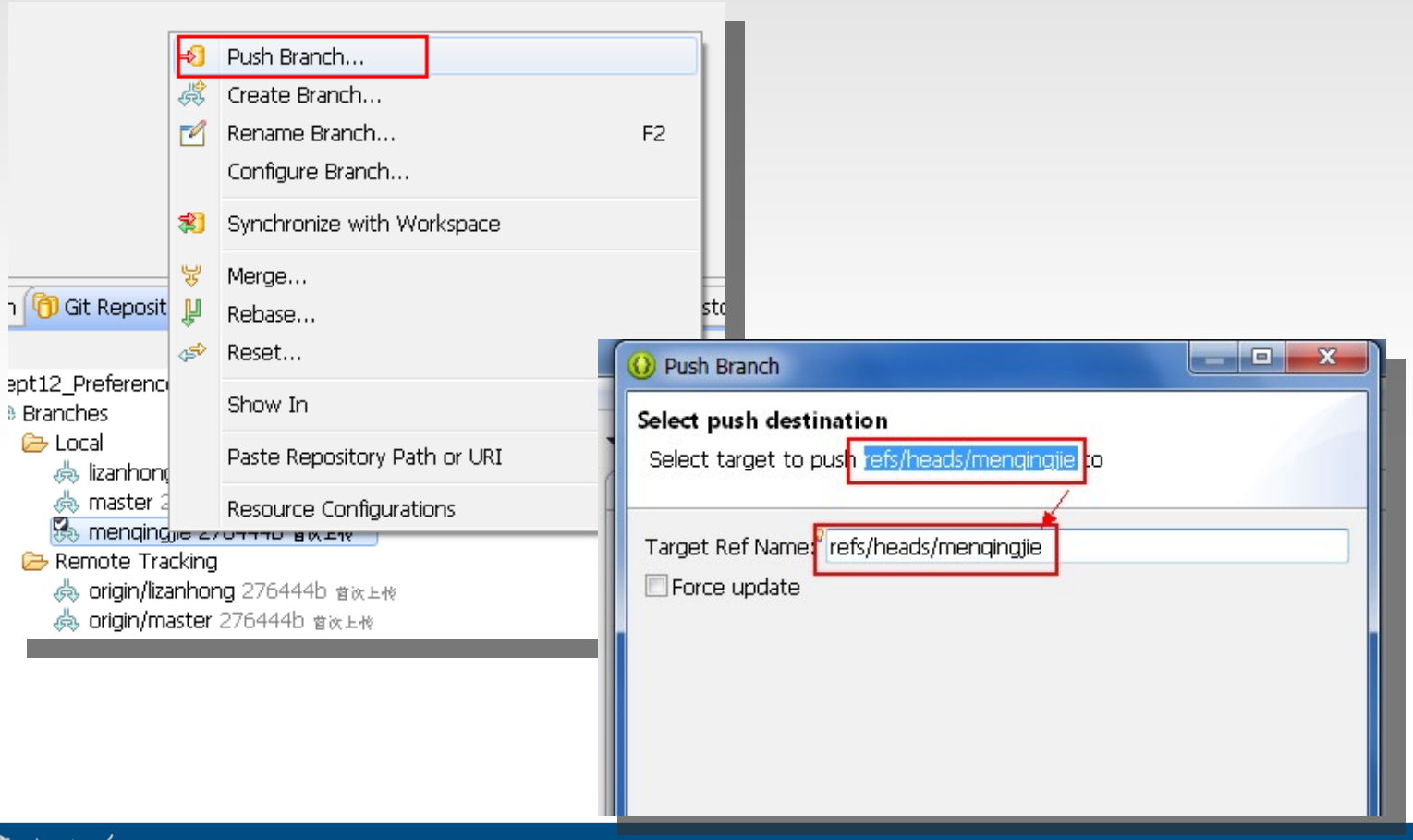
本地仓库同步到Git服务器



创建分支


Dept12_Preferences_Item [master] - F:\git\git\Dept12_Preferences_Item\.git
Branches
Loca
Rem
Tags
Switch To
Paste Repository Path or URI Ctrl+V
Resource Configurations
New Branch...
master
Other...

将创建的分支同步到GIT服务器



拉项目到本地

- 拉项目到本地
  - 将用户设置为项目的开发人员
  - 从 GIT 服务器拉取工程
  - 提交代码
  - 同步 Git 服务器

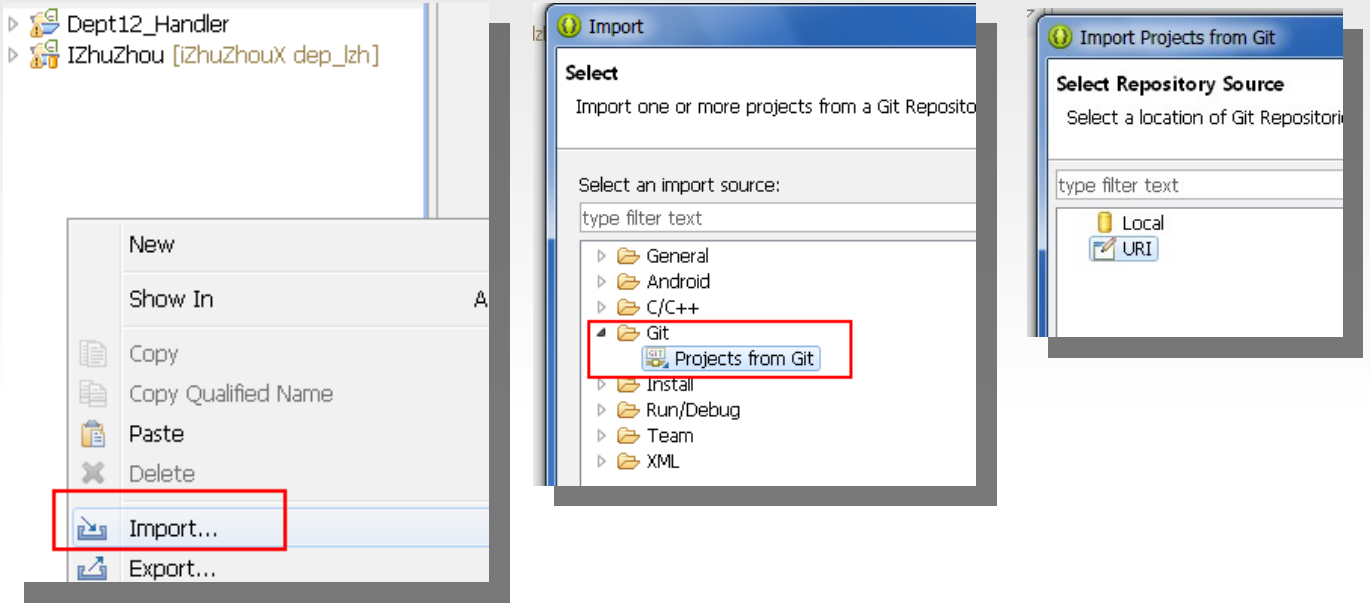
设置开发人员

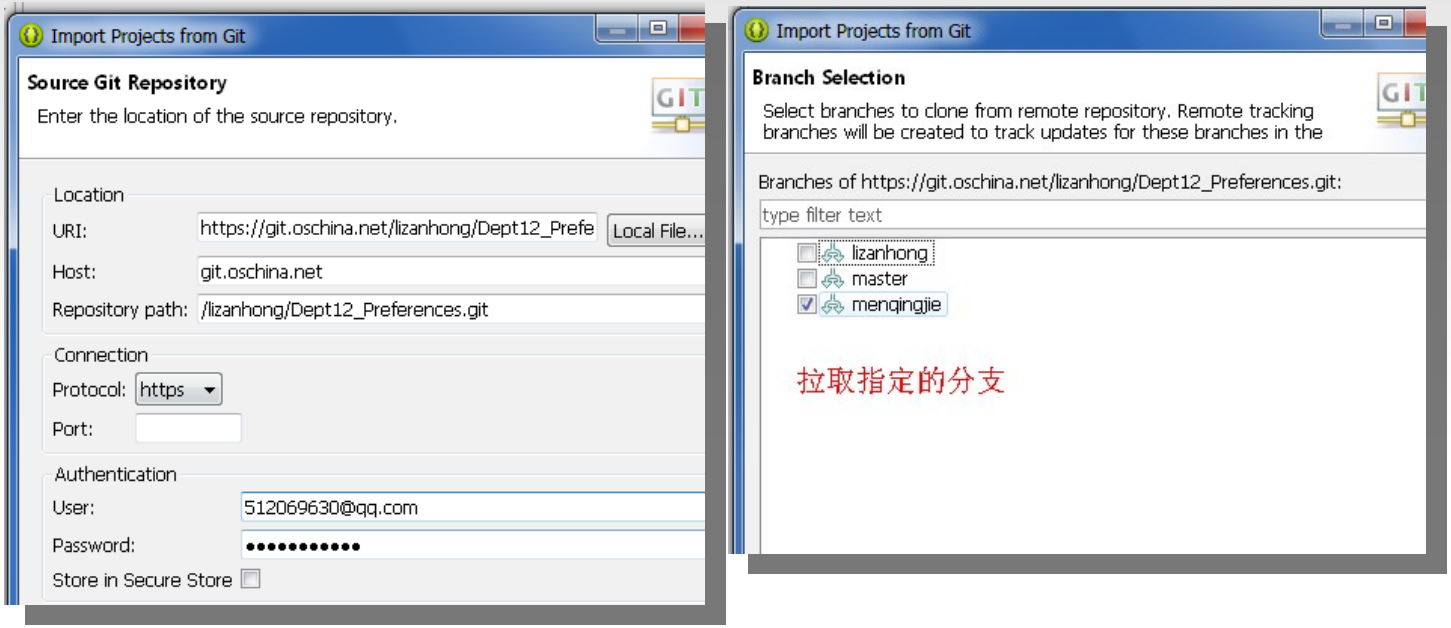

代码演示
管理
SH
https://git.oschina.net/liz

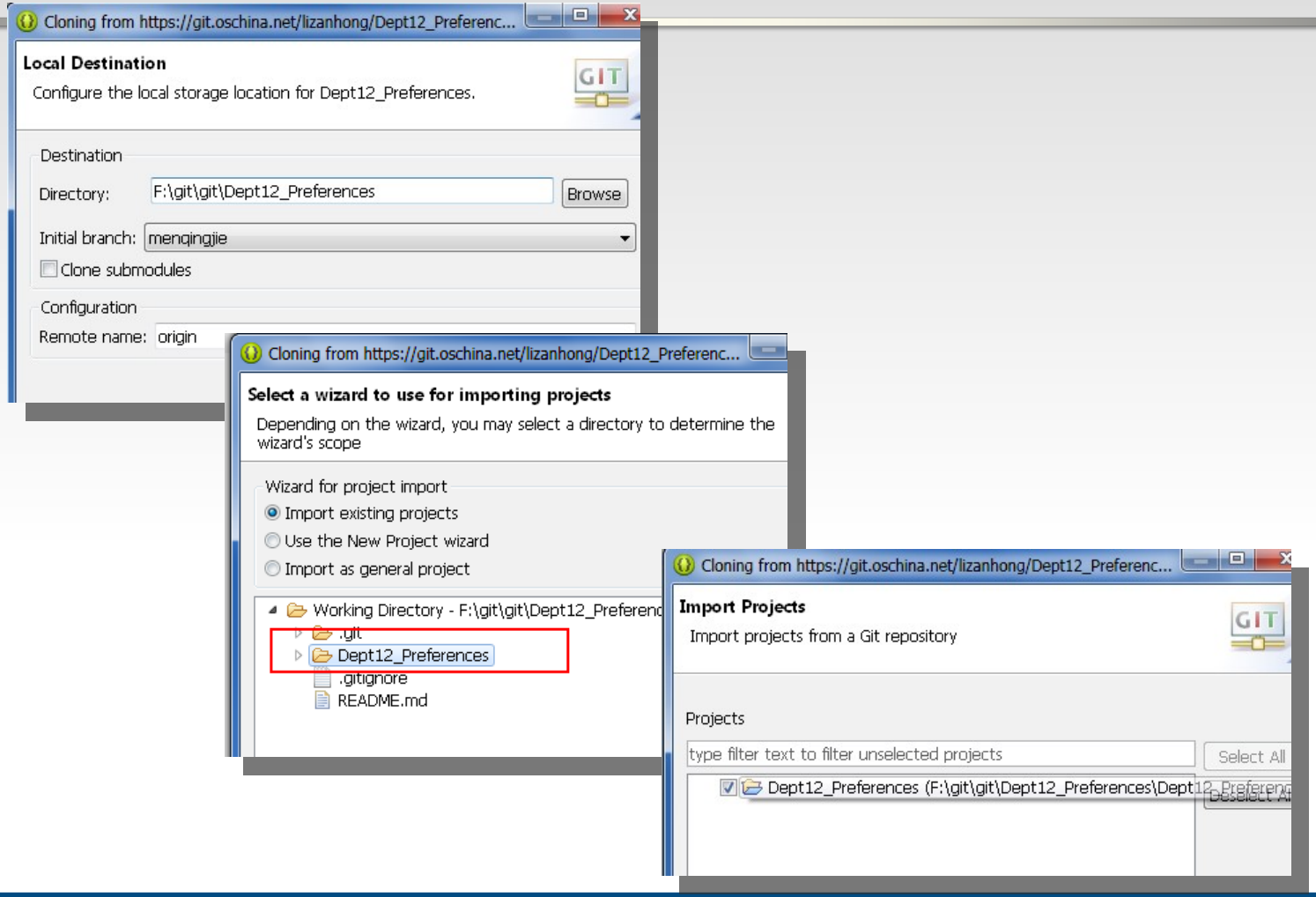

项目成员管理
所有 1
管理员 1
开发者 0
观察者 0
报告者 0


添加项目成员
选择需要添加的用户
512069630@qq.com
设置用户权限
开发者
添加用户

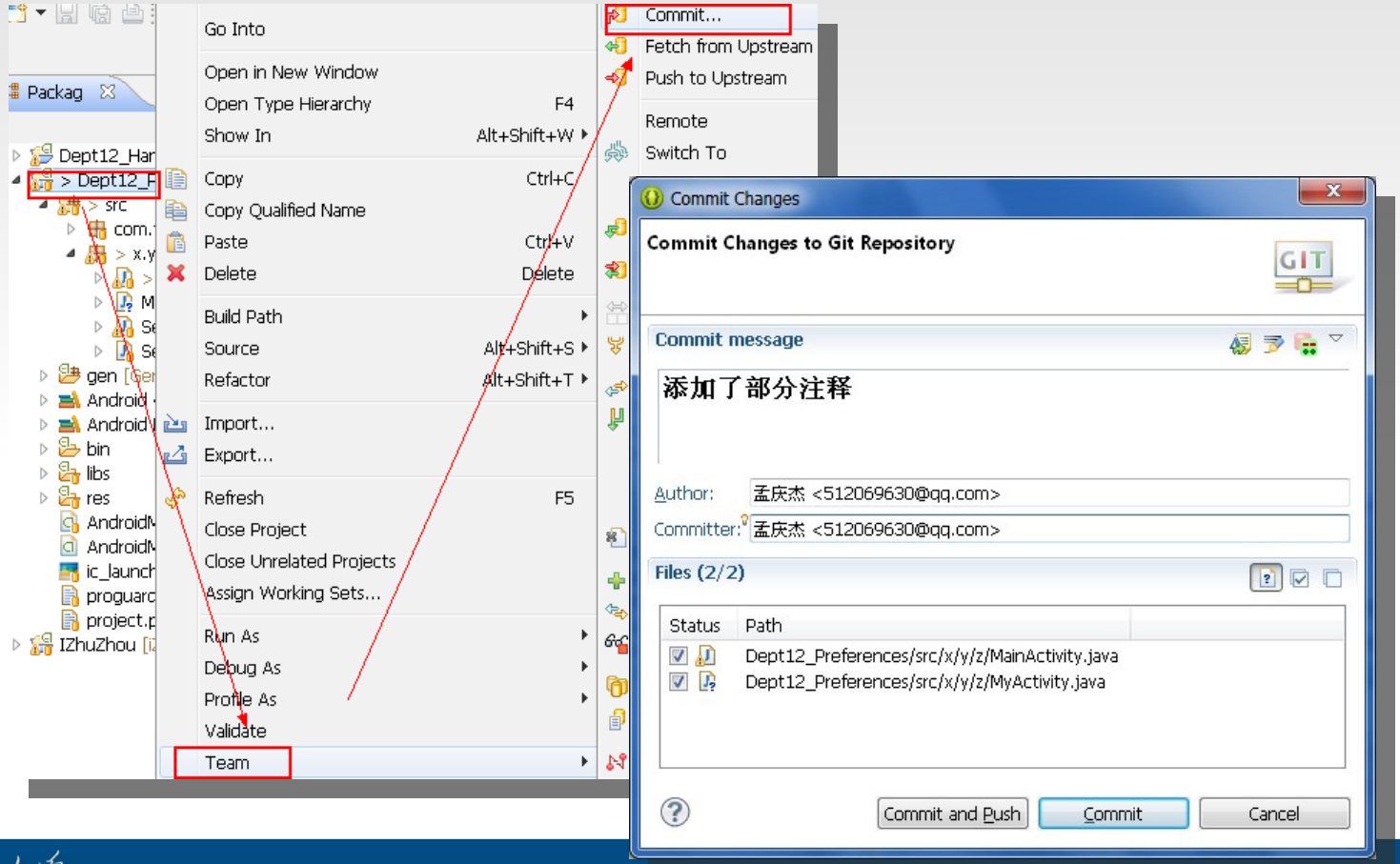
从GIT服务器拉取工程



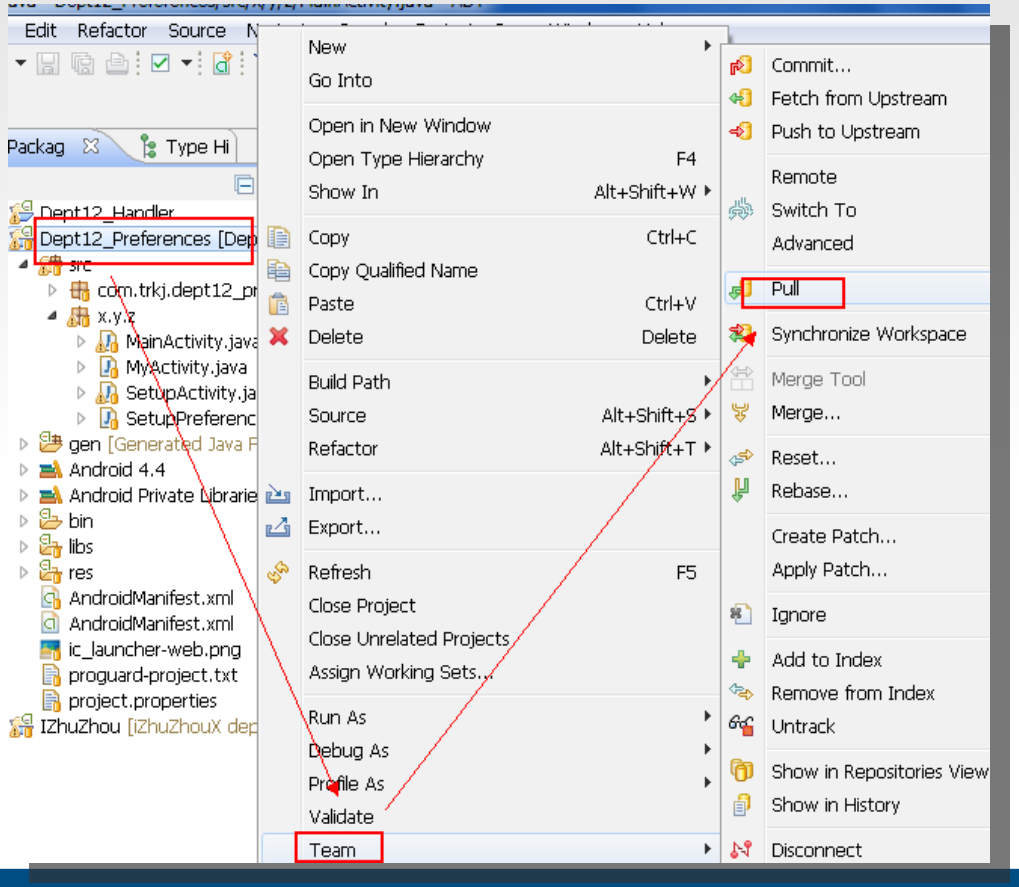

Import Projects from Git
Source Git Repository
Enter the location of the source repository.
Location
URI:
https://git.oschina.net/lizanhong/Dept12_Prefe
Local File...
Host:
git.oschina.net
Repository path:
/lizanhong/Dept12_Preferences.git
Connection
Protocol:
https
Port:
Authentication
User:
512069630@qq.com
Password:
Store in Secure Store
Import Projects from Git
Branch Selection
Select branches to clone from remote repository. Remote tracking branches will be created to track updates for these branches in the
Branches of https://git.oschina.net/lizanhong/Dept12_Preferences.git:
type filter text
lizanhong
master
menqingjie
拉取指定的分支


Cloning from https://git.oschina.net/lizanhong/Dept12_Preferenc...
Local Destination
Configure the local storage location for Dept12_Preferences.
GIT
Destination
Directory:
F:\git\git\Dept12_Preferences
Browse
Initial branch:
menqingjie
Clone submodules
Configuration
Remote name:
origin
Cloning from https://git.oschina.net/lizanhong/Dept12_Preferenc...
Select a wizard to use for importing projects
Depending on the wizard, you may select a directory to determine the wizard's scope
Wizard for project import
Import existing projects
Use the New Project wizard
Import as general project
Working Directory - F:\git\git\Dept12_Preferenc
.git
Dept12_Preferences
.gitignore
README.md
Cloning from https://git.oschina.net/lizanhong/Dept12_Preferenc...
Import Projects
Import projects from a Git repository
GIT
Projects
type filter text to filter unselected projects
Select All
Dept12_Preferences (F:\git\git\Dept12_Preferences\Dept12_Preferenc

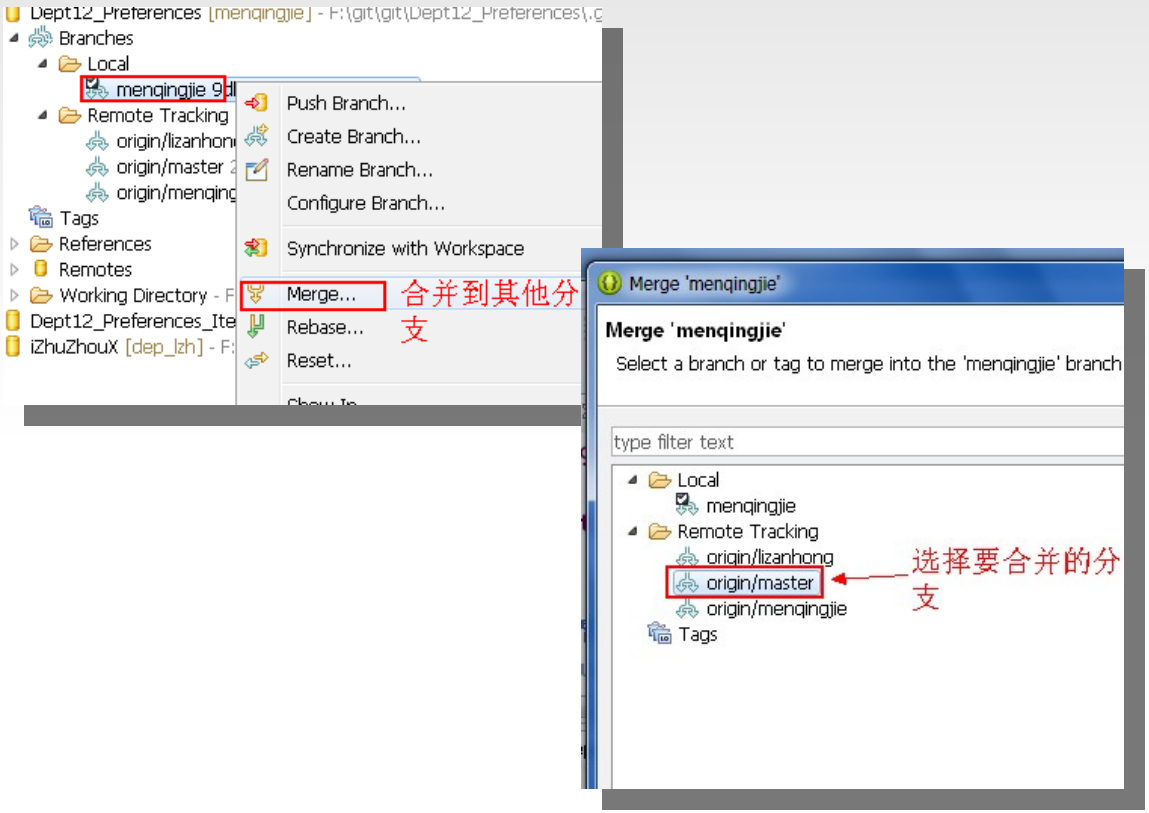
提交代码


Go Into
Open in New Window
Open Type Hierarchy F4
Show In Alt+Shift+W
Copy Ctrl+C
Copy Qualified Name
Paste Ctrl+V
Delete Delete
Build Path
Source Alt+Shift+S
Refactor Alt+Shift+T
Import...
Export...
Refresh F5
Close Project
Close Unrelated Projects
Assign Working Sets...
Run As
Debug As
Profile As
Validate
Team
Commit...
Fetch from Upstream
Push to Upstream
Remote
Switch To
Commit Changes
Commit Changes to Git Repository
GIT
Commit message
添加了部分注释
Author: 孟庆杰 <512069630@qq.com>
Committer: 孟庆杰 <512069630@qq.com>
Files (2/2)
Status
Path
Dept12_Preferences/src/x/y/z/MainActivity.java
Dept12_Preferences/src/x/y/z/MyActivity.java
Commit and Push
Commit
Cancel

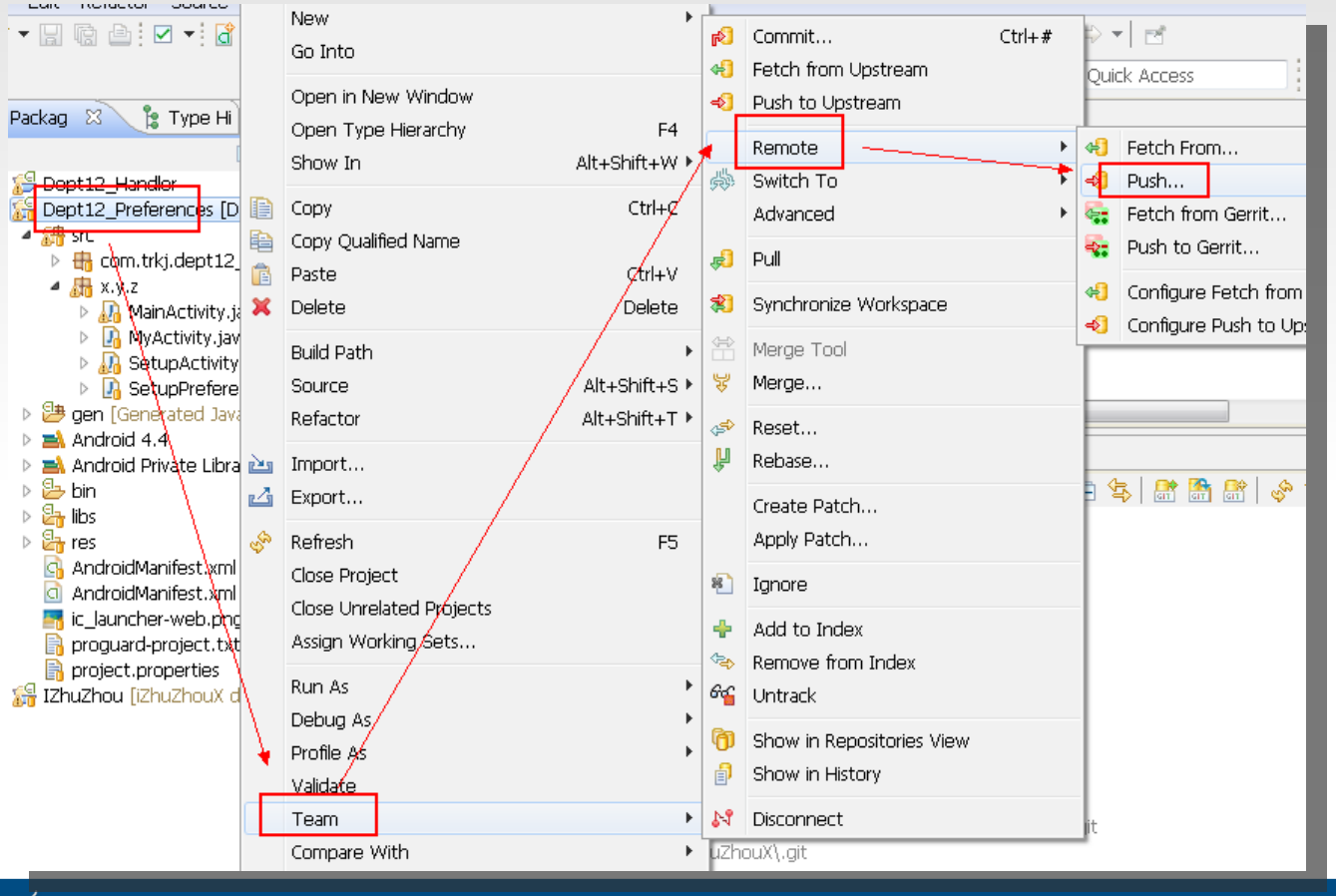
同步Git服务器

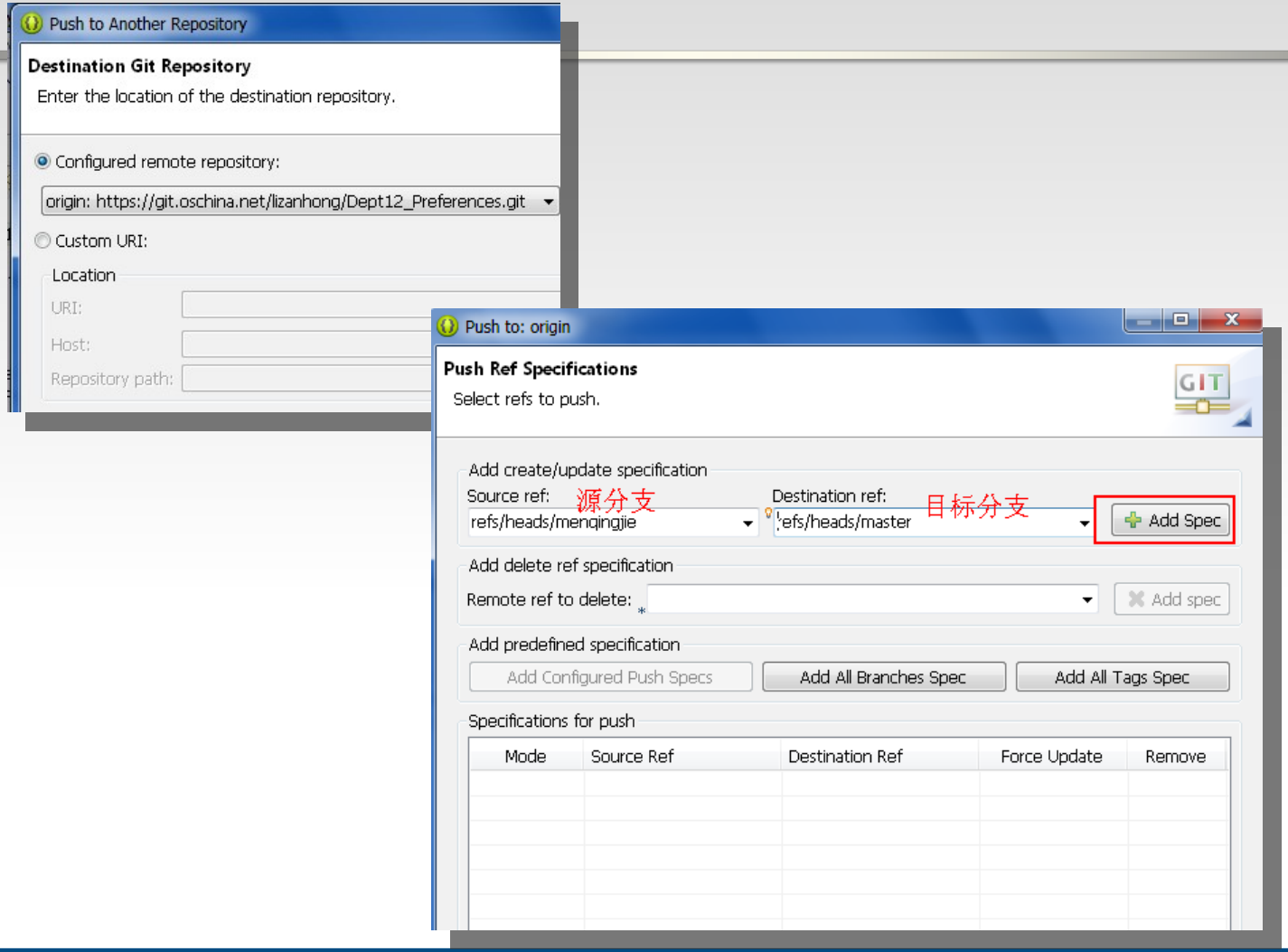

Edit
Refactor
Source
Packag
Type Hi
Dept12_Handler
Dept12_Preferences [Dep
src
com.trkj.dept12_pr
x.y.z
MainActivity.java
MyActivity.java
SetupActivity.ja
SetupPreferenc
gen [Generated Java F
Android 4.4
Android Private Librarie
bin
libs
res
AndroidManifest.xml
AndroidManifest.xml
ic_launcher-web.png
proguard-project.txt
project.properties
IZhuZhou [iZhuZhouX dep
New
Go Into
Open in New Window
Open Type Hierarchy F4
Show In Alt+Shift+W
Copy Ctrl+C
Copy Qualified Name
Paste Ctrl+V
Delete Delete
Build Path
Source Alt+Shift+S
Refactor Alt+Shift+T
Import...
Export...
Refresh F5
Close Project
Close Unrelated Projects
Assign Working Sets...
Run As
Debug As
Profile As
Validate
Team
Commit...
Fetch from Upstream
Push to Upstream
Remote
Switch To
Advanced
Pull
Synchronize Workspace
Merge Tool
Merge...
Reset...
Rebase...
Create Patch...
Apply Patch...
Ignore
Add to Index
Remove from Index
Untrack
Show in Repositories View
Show in History
Disconnect

合并分支


Branches
Local
menqingjie 9
Remote Tracking
origin/lizanhon
origin/master
origin/menqing
Tags
References
Remotes
Working Directory
Dept12_Preferences_Ite
iZhuZhouX [dep_lzh]
Push Branch...
Create Branch...
Rename Branch...
Configure Branch...
Synchronize with Workspace
Merge...
合并到其他分支
Rebase...
Reset...
Merge 'menqingjie'
Merge 'menqingjie'
Select a branch or tag to merge into the 'menqingjie' branch
type filter text
Local
menqingjie
Remote Tracking
origin/lizanhong
origin/master
origin/menqingjie
Tags
选择要合并的分支

合并分支的另一种方法


Packag
Type Hi
Dept12_Handler
Dept12_Preferences [D
src
com.trkj.dept12_
x.y.z
MainActivity.ja
MyActivity.jav
SetupActivity
SetupPrefere
gen [Generated Java
Android 4.4
Android Private Libra
bin
libs
res
AndroidManifest.xml
AndroidManifest.xml
ic_launcher-web.png
proguard-project.txt
project.properties
IZhuZhou [iZhuZhouX d
New
Go Into
Open in New Window
Open Type Hierarchy F4
Show In Alt+Shift+W
Copy Ctrl+C
Copy Qualified Name
Paste Ctrl+V
Delete Delete
Build Path
Source Alt+Shift+S
Refactor Alt+Shift+T
Import...
Export...
Refresh F5
Close Project
Close Unrelated Projects
Assign Working Sets...
Run As
Debug As
Profile As
Validate
Team
Compare With
Commit... Ctrl+#
Fetch from Upstream
Push to Upstream
Remote
Switch To
Advanced
Pull
Synchronize Workspace
Merge Tool
Merge...
Reset...
Rebase...
Create Patch...
Apply Patch...
Ignore
Add to Index
Remove from Index
Untrack
Show in Repositories View
Show in History
Disconnect
uZhouX\.git
Quick Access
Fetch From...
Push...
Fetch from Gerrit...
Push to Gerrit...
Configure Fetch from
Configure Push to Up


Push to Another Repository
Destination Git Repository
Enter the location of the destination repository.
Configured remote repository:
origin: https://git.oschina.net/lizanhong/Dept12_Preferences.git
Custom URI:
Location
URI:
Host:
Repository path:
Push to: origin
Push Ref Specifications
Select refs to push.
GIT
Add create/update specification
Source ref:
源分支
refs/heads/menqingjie
Destination ref:
目标分支
refs/heads/master
Add Spec
Add delete ref specification
Remote ref to delete:
Add spec
Add predefined specification
Add Configured Push Specs
Add All Branches Spec
Add All Tags Spec
Specifications for push
Mode
Source Ref
Destination Ref
Force Update
Remove

济南大学

谢谢

济 南 大 学 信 息 科 学 与 工 程 学 院
School of Information Science and Engineering