

分布式版本控制系统

1、下载安装git服务

2、安装完成 改相应的仓库目录及访问端口

目录： \gitblit-1.7.1\data

修改： 安装目录中的gitblit.properties文件

增加修改内容：

#资料库路径

git.repositoriesFolder = E:\Git\gitRepository

#http协议的端口号

server.httpPort = 10101

#设定服务器的IP地址,这里就设定你的服务器IP,不能配成127.0.0.1 本机IP地址

server.httpBindInterface = 192.168.154.32

server.httpsBindInterface = localhost

3、启动服务

运行 gitblit.cmd文件

4、用浏览器打开服务器默认密码admin

ip+端口号

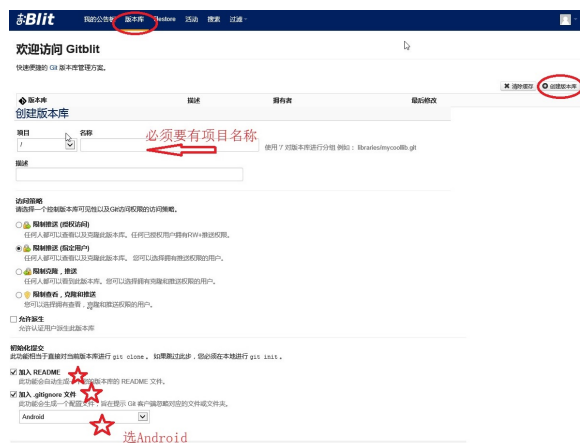
浏览器访问地址： 计算机名： 10101

IP:10101

如： 192.168.154.32:10101

5、创建仓库

版本库 -----> 创建版本库 -----> 填写**项目名称**，选中初始化提交的 **加入 README** 及 **加入 .gitignore 文件** 两项 并 选择项目类型



6、创建用户及分配权限

用户 -----> 添加用户 -----> 确定用户名和密码

分配权限：版本库 -----> 选择 创建的项目 -----> 编辑 -----> 选择permissions -----> 选择用户 -----> 添加相应的权限

添加用户

常规 属性 访问权限

用户名

密码

确认密码

显示名称

邮箱

管理权限 ☐ Gitblit 服务器管理员

允许创建 ☐ 允许创建私人版本库

允许派生 ☐ 允许派生认证版本库到私人版本库

exclude from federation ☐ 禁止已 federated 的 Gitblit 实例从本账户拉取

禁止用户 ☐ 禁止当前账户进行认证



7、初始化一个android项目,关联android项目到git服务器

在版本库中选择项目 -----> 复制管理员的地址 -----> 在电脑上通过地址下载仓库（通过GIT
用户端环境下载，或者通过GIT乌龟 需要密码 在某个文件夹下右键 Git-Bash-here 或者 Git
Clone 输入地址 下载仓库）-----> 在下载的仓库中复制 .git文件 然后将文件粘贴到 AS 已经创
建好的项目中去

-----> **在AS中激活(VCS----->Enable Version Control
Integration -----> 选择Git)**

Git-Bash-here输入地址：git clone 地址



在某个文件夹下右键 Git-Bash-here 或者 Git Clone 输入地址 下载仓库

8、添加忽略

- 1、在AS中修改.gitignore来忽略不关联的文件
- 2、确定文件类型：

*.iml
.gradle
local.properties
.idea
build
app/app.iml
app/build

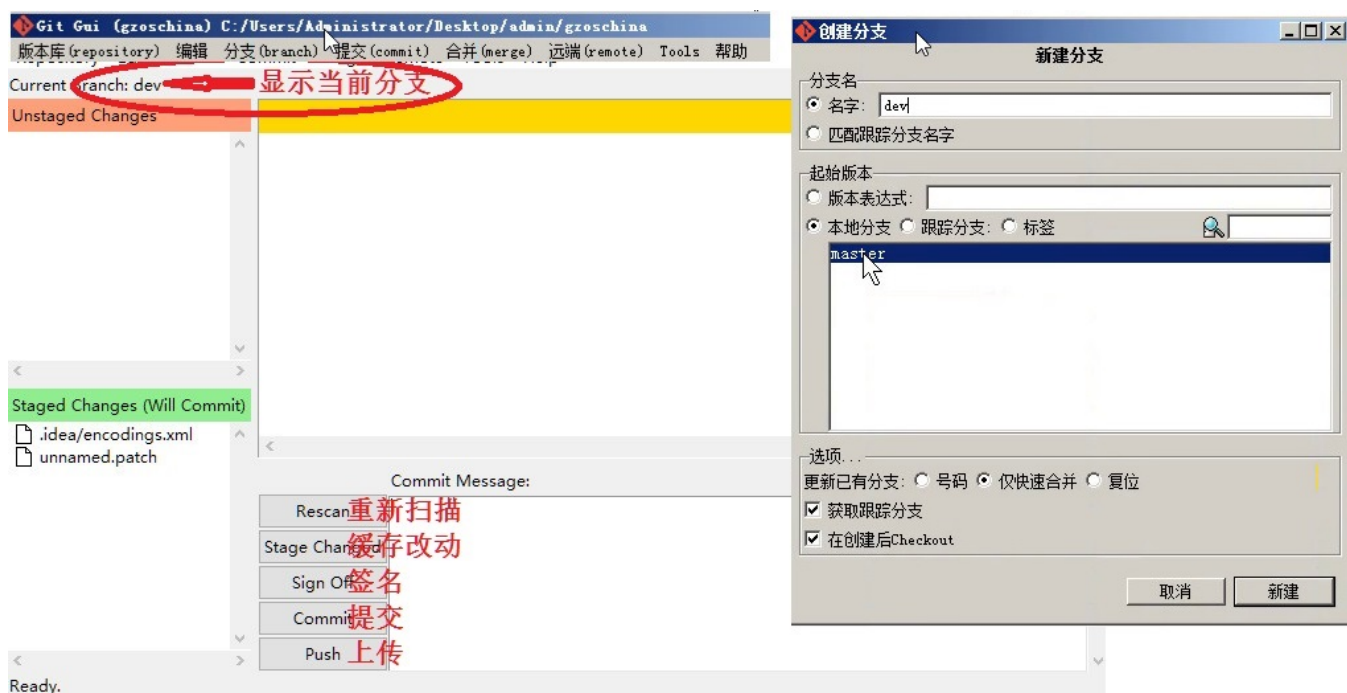
9、add项目到版本管理，初次提交

项目右键---->Git ----> Add ----> VCS 提交（提交界面右侧，去掉Before Commit下的所有选中项）

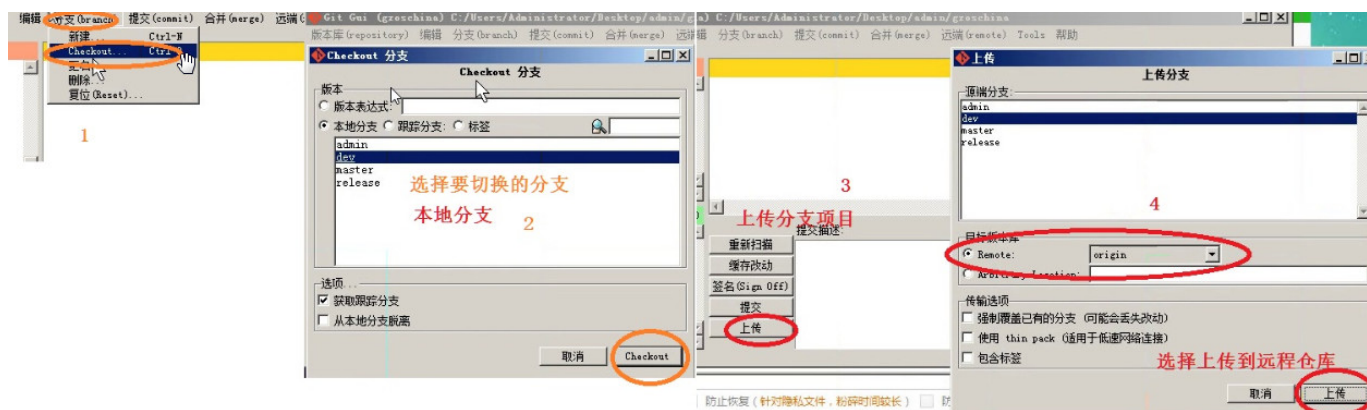
10、分支创建及管理

- 1、首先管理员先将仓库中的代码检出，然后进入项目

目录 右键 Git Gui ----> 分支 -----> 创建分支 develop分支 release分支 hotfix分支



2、管理员切换上传分支项目的方法



git分支介绍：

git的分支管理-->难

联想svn的分支管理

trunk 主分支-->放开发过程中的代码

branches -->放置一些重要的代码

tags v1 v2 v3-->标记分支

Develop -->可以缩写为dev-->主开发分支-->类似svn里面trunk

release branches-->发布分支-->类似svn里面的tags

Hotfix-->修复分支-->可以创建特定的分支去修复bug

Master分支-->默认创建的分支-->Production 分支-->这个分支只能从其他分支合并，不能在这个分支直接修改

Feature 分支-->开发一个独立的功能

管理应该持有的分支

release branches

Master分支

开发人员持有的分支

Develop

Hotfix

Feature

开发者要做的事：

1、检出项目

创建空文件夹 -----> 右键 **Git-Bash-here** 或者 **Git Clone** 输入地址（相应用户的地址由管理员提供） -----> 基于某个分支拉取数据 -----> 创建自己的分支 -----> 打开AS选择要打开的项目文件（1） -----> 进入后激活（2） -----> 选择切换自己的开发分支（3） -----> 开始开发项目 -----> 更新并提交代码

如果使用gitFlow, 开发人员需要做些啥

1. git clone的时候, 基于dev分支进行clone

git clone --branch dev ssh://user10@billypc:29418/gzoschina.git

2. 创建了自己的feature分支-->user10

开发人员使用gitFlow操作

日常操作-->commit push

当开发人员完成了一个具体的功能的时候, dev需要合并相应的代码, 步骤如下

1. 开发人员需要完成代码的commit, 或者commit+push

2. 切换到dev

0. 更新dev, 保证本地的dev是最新的代码, 因为其他开发人员可能已经合并了自己的功能模块到dev分支中

1. fetch-->origin====>得知远程服务器上是否有更新

没有更新==>本地的dev就是最新的

有更新==>合并远程的dev到本地

1. 本地合并admin/user10代码

可能存在冲突, 冲突解决有3种方式

1. 用别人的

2. 用自己的

3. 自行解决冲突(综合代码)

1. 使用冲突解决工具merge tool

2. 手动解决-->推荐方式

1. 删除不满足语法要求的箭头

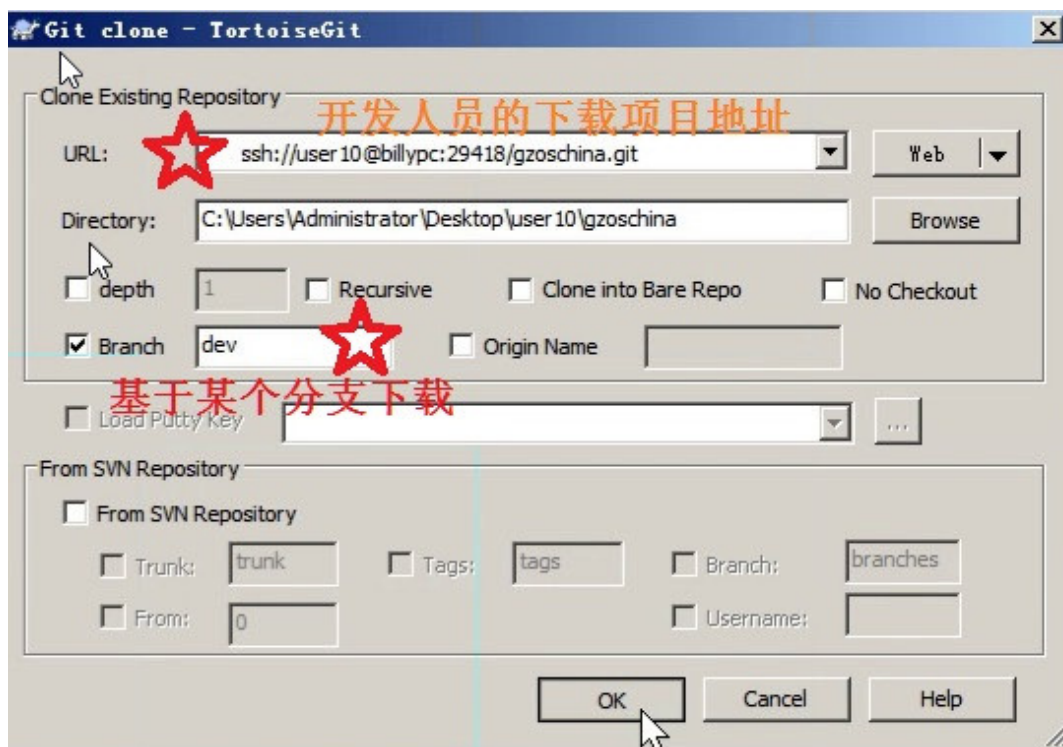
2. 根据自己的情况, 删除代码

2. push dev最新代码到远程的dev

3. 切换admin/user10, 继续开发

应该保证admin的代码也是最新的, 然后就基于最新的代码去开发即可-->合并dev上面的代码

基于某个分支拉取数据：

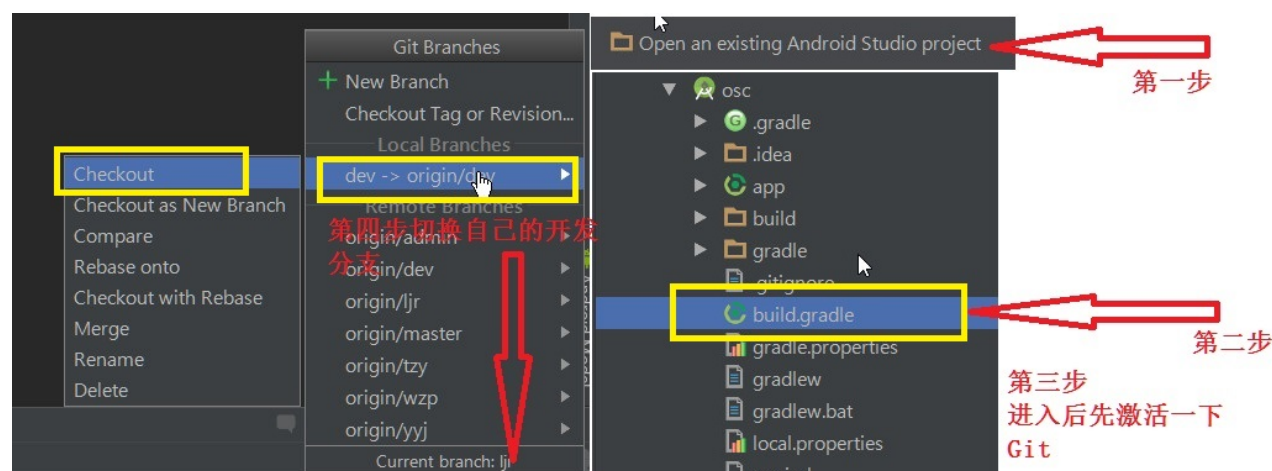


```
Administrator@BILLYPC /D:/Users/Administrator/Desktop/user10
$ git clone --branch dev ssh://user10@billypc:29418/gzoschina.git
```

命令模式

基于某个分支下载

打开AS选择要打开的项目文件：



提交代码的方法： 注 提交到本地其它人不可见也不会对其它代码产生影响

AS中提交的方法：



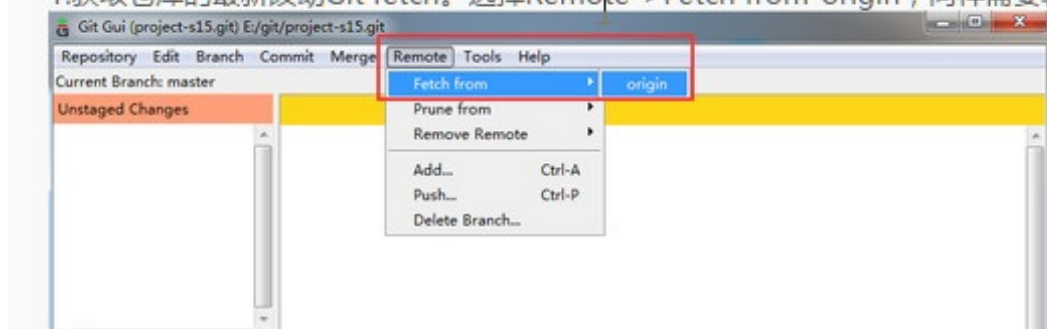
更新并合并远程代码 （每次获取远程代码都要合并一次）：

Remote----->fetch----->Merge----->合并到本地

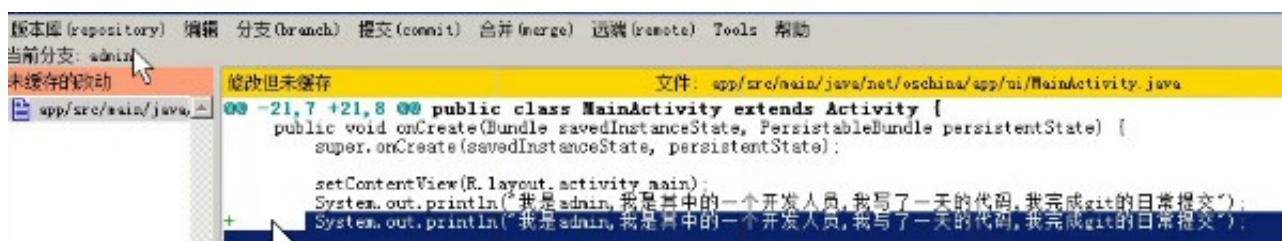
需先从远程仓库拉取一下文件，解决下合并冲突的问题再push

提交前要先更新一下远程的代码避免发生冲突

1.获取仓库的最新改动Git fetch。选择Remote->Fetch from-origin，同样需要输入项目的用户名和密码



Gui中提交方法：



Gui提交方法：

- 第一：重新扫描一下最新的改动项
- 第二：查看改动缓存，然后缓存改动
- 第三：签名 备注说明
- 第四：提交到本地
- 第五：上传到远程仓库

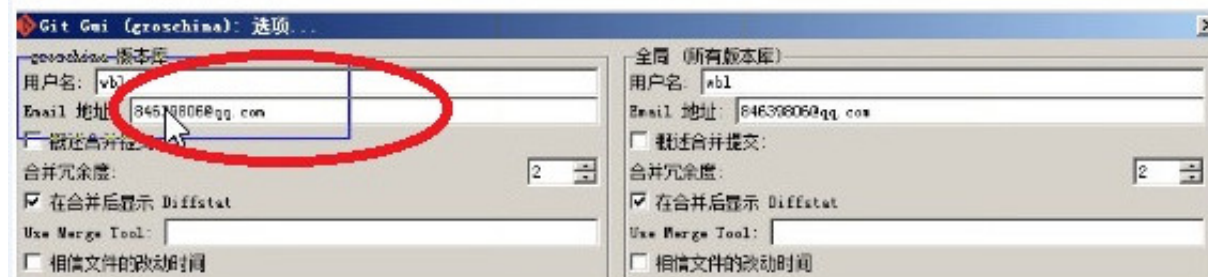
乱码处理：

改Gui的编码

注：第一次使用Gui要设置签名和邮箱（随便定义）



Gui使用注意事项： 改编码，设置用户名



在GUI的编辑 选项中设置 用户名和地址 改编码