

# 智游3G实训之SVN

---

BOX



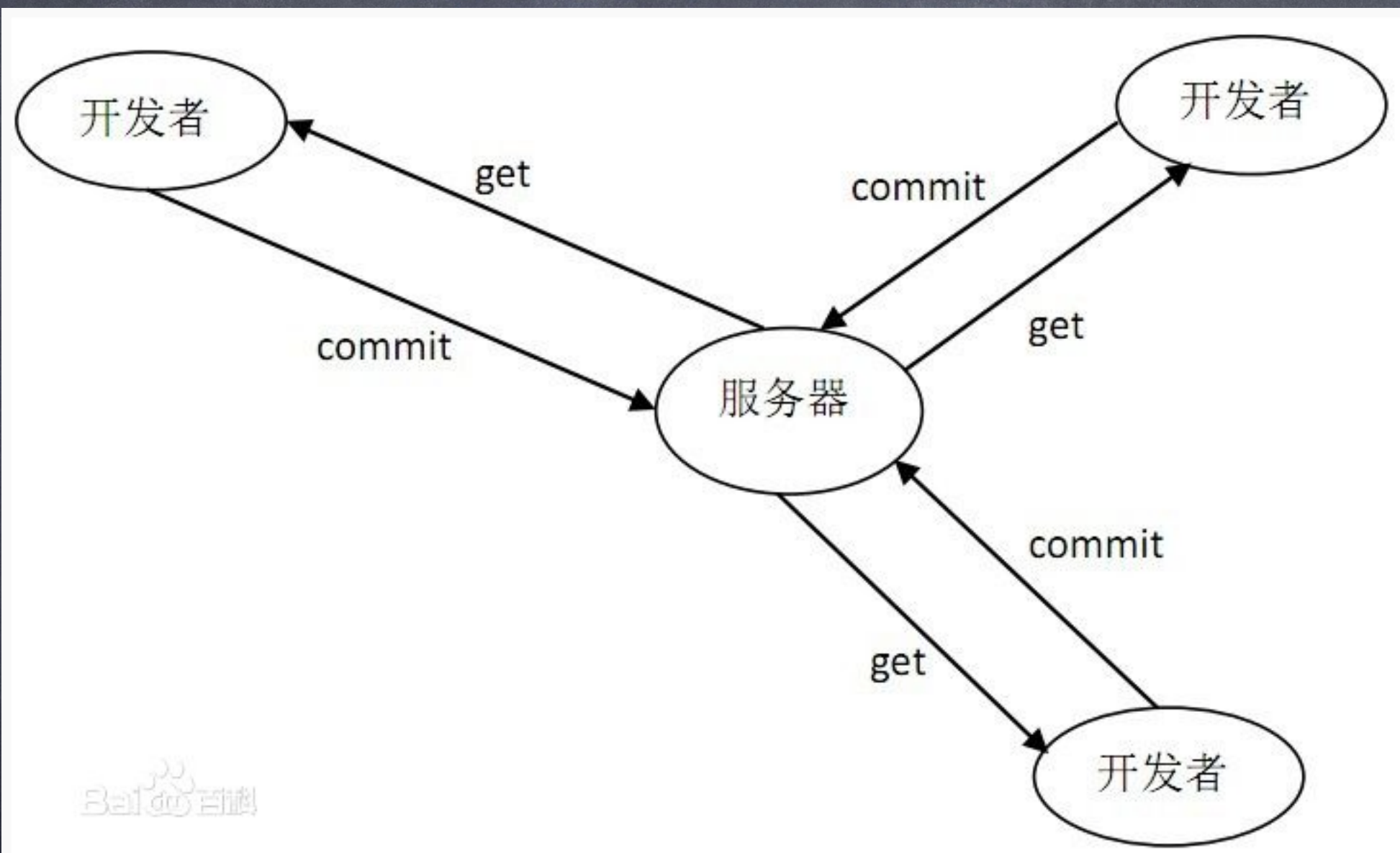
# 什么是SVN (Subversion) ?

简单说 SVN = 版本控制 + 备份服务器

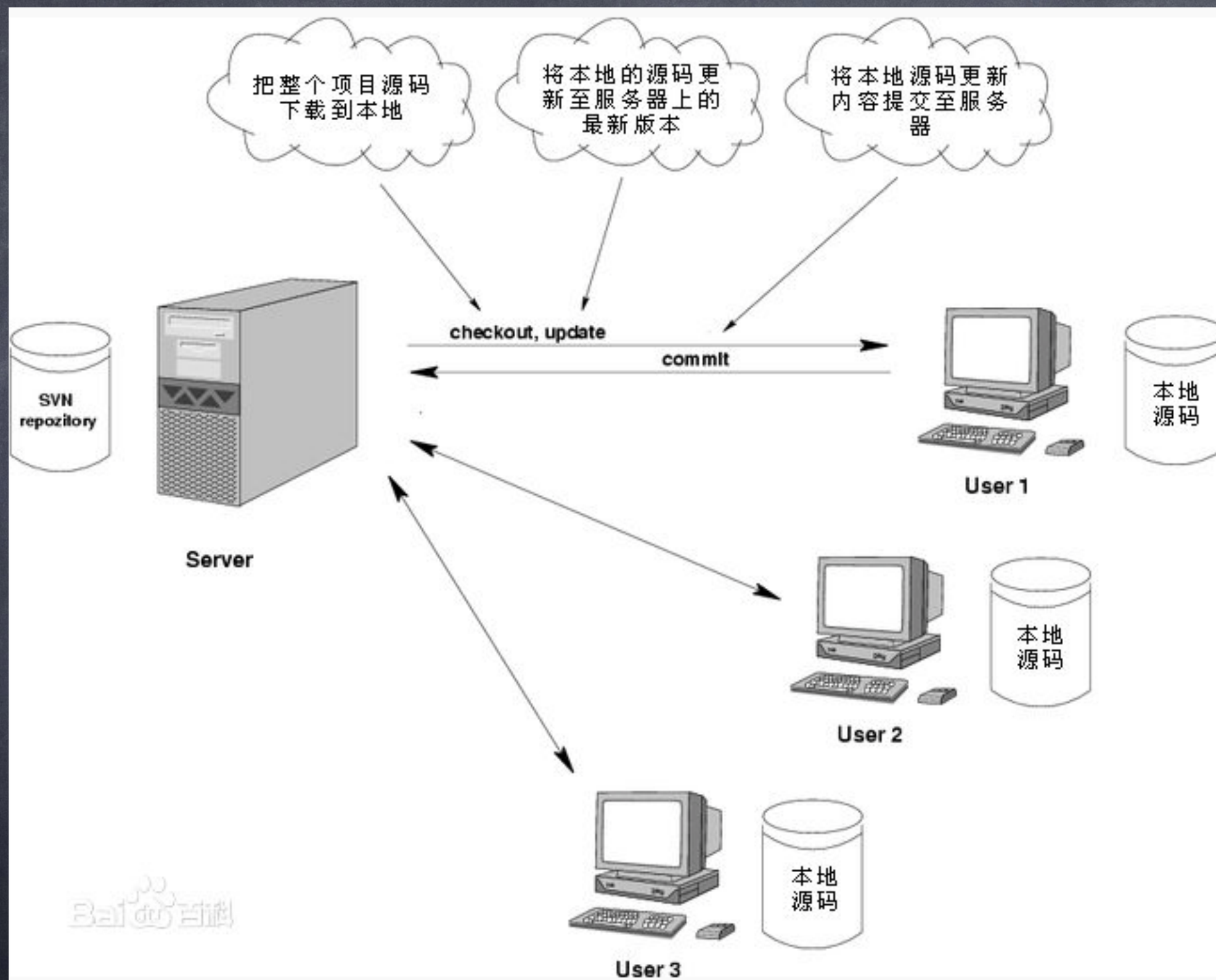
简单的说，可以把SVN当成您的备份服务器，更好的是，它可以帮您记住每次上传到这个服务器的档案内容。并且自动的赋予每次的变更一个版本。

通常，我们称用来存放上传档案的地方就做Repository。用中文来说，有点像是档案仓库的意思。不过，通常我们还是使用Repository这个名词。基本上，第一次我们需要有一个新增(add)档案的动作，将想要备份的档案放到Repository上面。日后，当您有任何修改时，都可以上传到Repository上面，上传已经存在且修改过的档案就叫做commit，也就是提交修改给SVN server的意思。针对每次的commit，SVN server都会赋予他一个新的版本。同时，也会把每次上传的时间记录下来。日后，因为某些因素，如果您需要从Repository下载曾经提交的档案。您可以直接选择取得最新的版本，也可以取得任何一个之前的版本。如果忘记了版本，还是可以靠记忆尝试取得某个日期的版本。











# 为什么要用SVN?

1. 备份工作档案的重要性。您永远不知道计算机上的硬盘何时会坏掉。根据经验法则再加上墨菲定理，坏事情往往都发生在最重要的时刻。例如，要**release**东西的前一刻，硬盘完全坏掉，无法修复。所以，常常备份工作数据是非常重要的。大部分公司的**server**都会有专门的备份机制，甚至是异地备援。绝对比放在自己的计算机里头，或是随身碟上面来的安全。
2. 版本控管的重要重要性。您无法保证手头上最新版本永远都是对的。很多时候，在经过数天努力工作后，您才发现走错方向。需要将所有的修改回复到数天前版本。没有几个人能够完全记住自己修改过什么东西。如果没有做好版本控管，那么，最差的情况就是要全部重来。
3. 伙伴间的数据同步的重要性。很多时候，除了您个人外，还有其它的伙伴也需要您的档案。怎么把最新的档案提供给伙伴呢？用**e-mail**？根据经验，用**e-mail**是一个很差的办法。到了最后，根本就不知道哪一封**e-mail**才是新的。因为可能您最新的**e-mail**已经被当成垃圾信给丢了。别忘了，现在很多**e-mail**软件都有**anti-spam**(反垃圾邮件)的功能，说不定这些信件都被当成广告信给处理掉了。而且，您也应该知道一般档案如果放在**e-mail**内，档案大小会变得比较大（通常是原来的**4/3**倍以上）。如果每天更新一次，恐怕几天没收信的话，信箱就爆了！



4。如果没有一个好的办法，备份不同版本是很耗费硬盘空间的。例如：您有一个总共10MB的目录，使用单纯的档案备份。如果有10个版本就会变成100MB。20个版本就有200MB。如果，不想备份这么多版本，势必就是要减少备份的次数。拉长备份周期将会导致数据遗失的危险性增高。把旧的不需要的备份删除？根据经验，只要时间一拉长，您跟本就不知道一个月以前的版本是重要或者是不重要。

SVN能帮我们解决上面的问题吗？答案是肯定的：

SVN Repository可以是自己计算机上的一个目录，或者是随身碟（不建议这样用）。当然也可以是公司的服务器。

SVN有很棒的版本控管机制。所有上传的版本都会帮您记录下来。日后您可以随时取得某一个时刻的版本。而且，也有版本分支及合并等好用的功能。

SVN可以让不同的开发者存取同样的档案，并且利用SVN Server作为档案同步的机制。也就是说，您有档案更新时，无须将档案寄给您的开发成员。只需要告诉他新的版本已经在SVN Server上面，请他自己去SVN Server上面就可以取得最新版本。而且，SVN Server也可以做到当您上传新版本后，自动发信给相关的成员。

SVN的存放档案方式是采用差异备份的方式。也就是说，他只会备份有不同的地方。所以很省硬盘空间。此外，他也可以针对所谓的非文字文件进行差异备份。



# 如何建立一个SVN Server?

- 参考 学习文档“建立svn服务器.rtf”



1

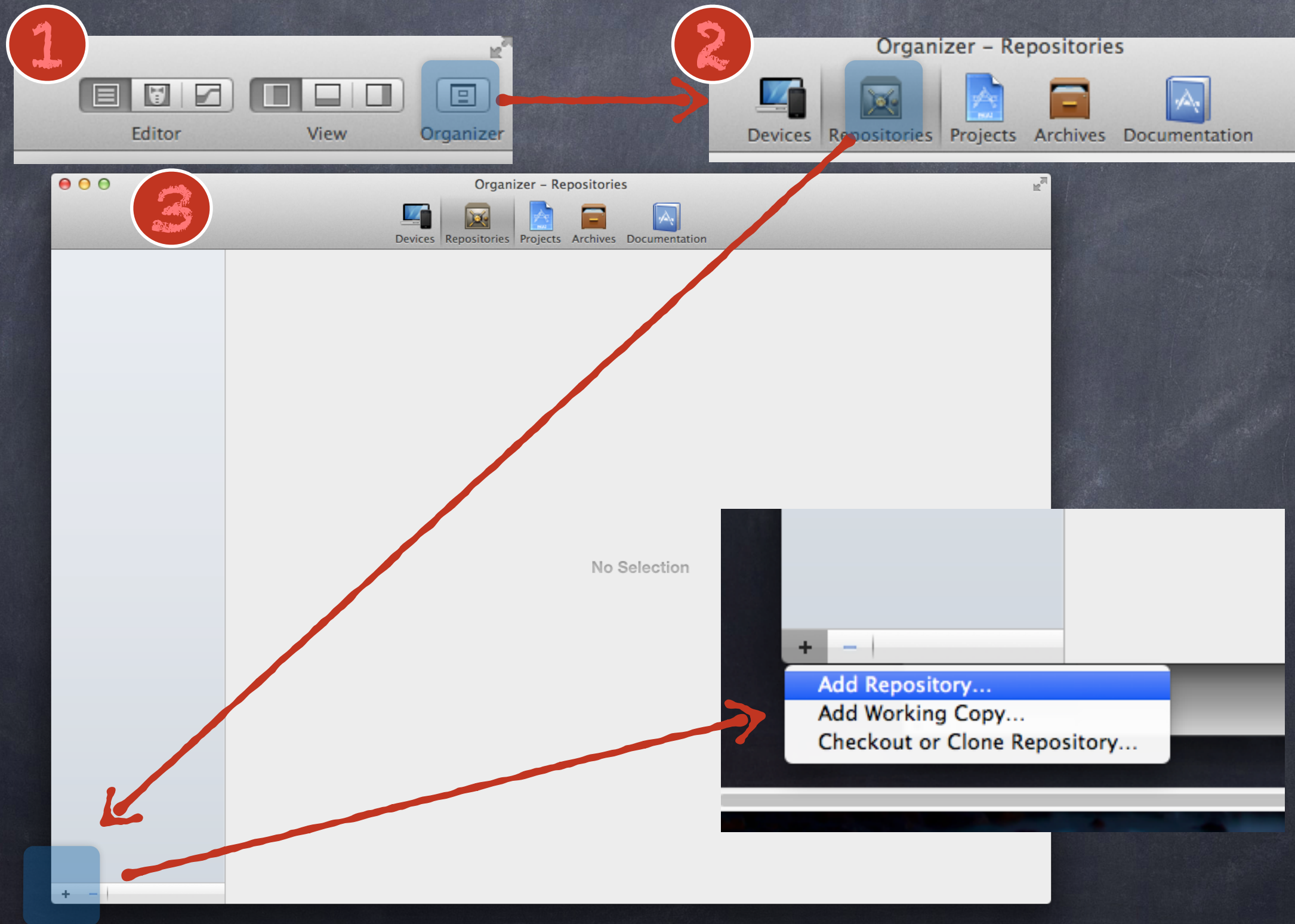
```
Wei — bash — 80x24
Last login: Tue Jan 7 20:46:16 on console
bogon:~ Wei$ svnadmin create /Users/Wei/Desktop/svn/projectName
```

2





# 如何在客户端使用SVN服务





创建仓库的名字也就是要存放本地代码仓库的名字建议用你的项目名称作为仓库名字

### Add a Repository

You may specify the location using either a URL or local file path.

Name

Location

Type

这里是服务协议类型

这个是你的工程项目在 `svn` 服务器上的地址，因是本地作为服务器，`so` 是这样的 `svn`：  
`//127.0.0.1/projectname`，  
要跟你建立在 `svn` 服务器上路径保持一致

如果没有问题，填完之后此处会有提示 黄色为 连接上 `svn` 服务器，也就是说没有问题 `next` 一步，红色则为有问题（这个是 `Xcode` 的 `bug`），如何解决看下一页

Name

Location

☒ Authentication required

Type

Name

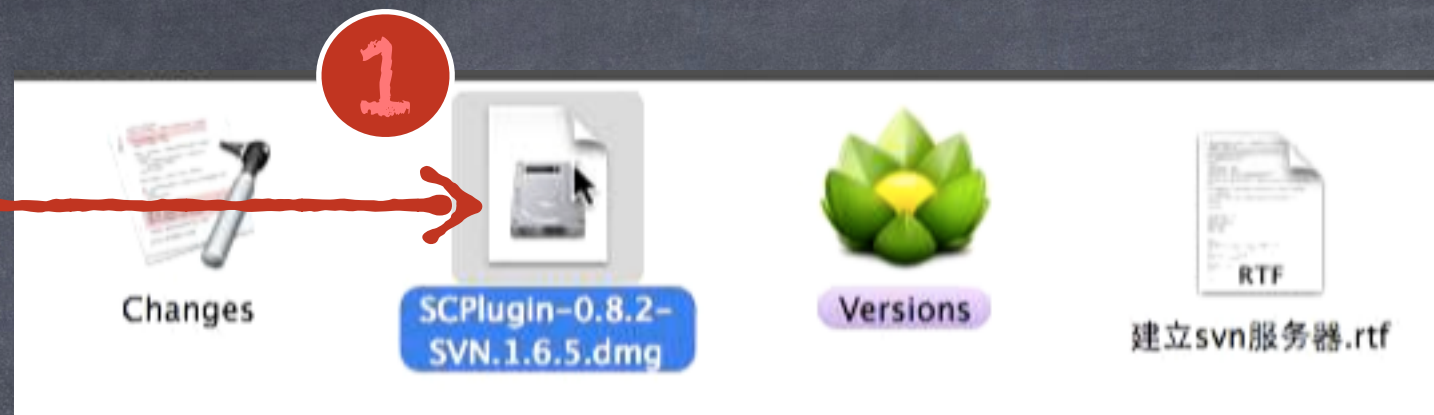
Location

☒ Host is unreachable

Type



安装

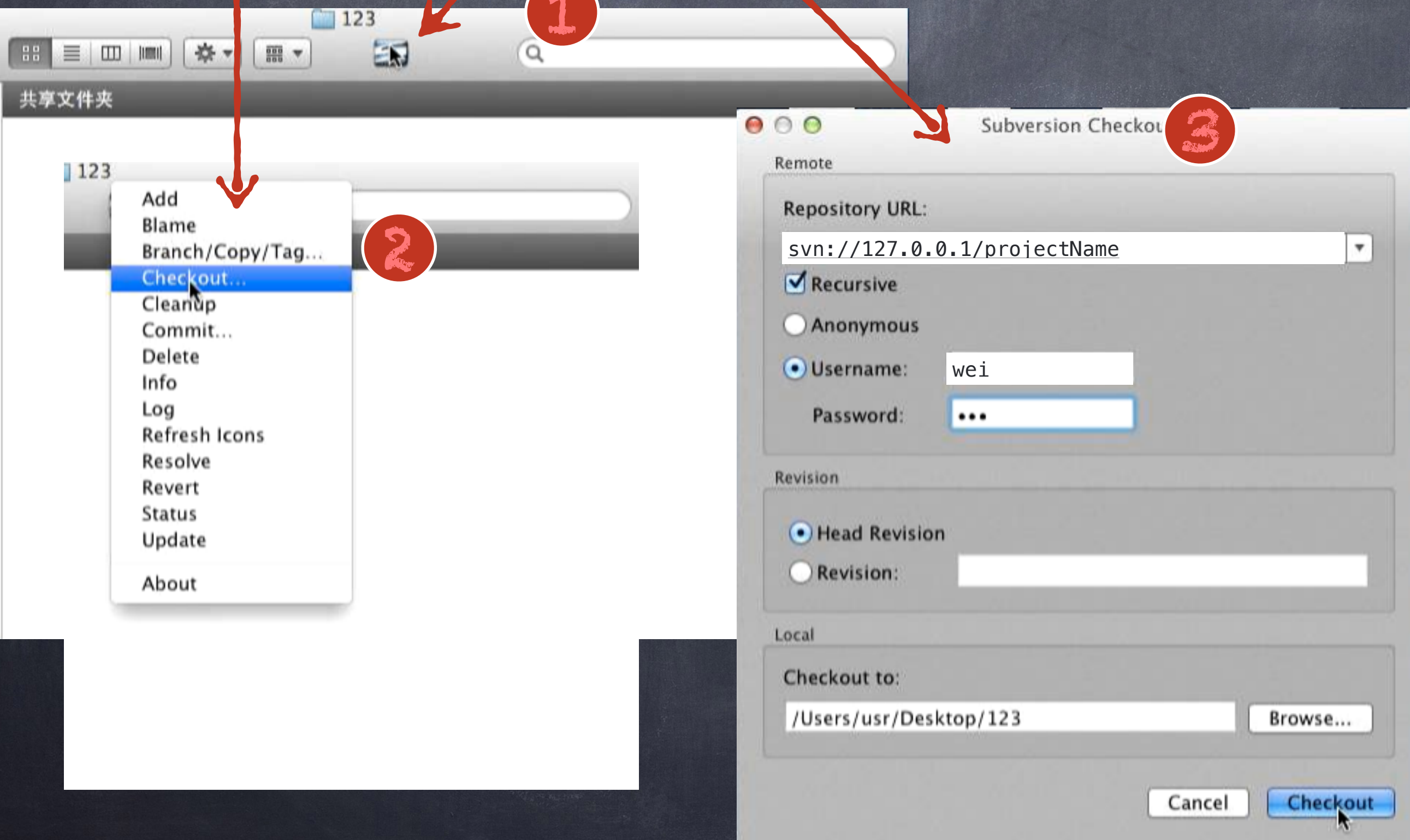


OK后在应用程序中，做如图所示操作





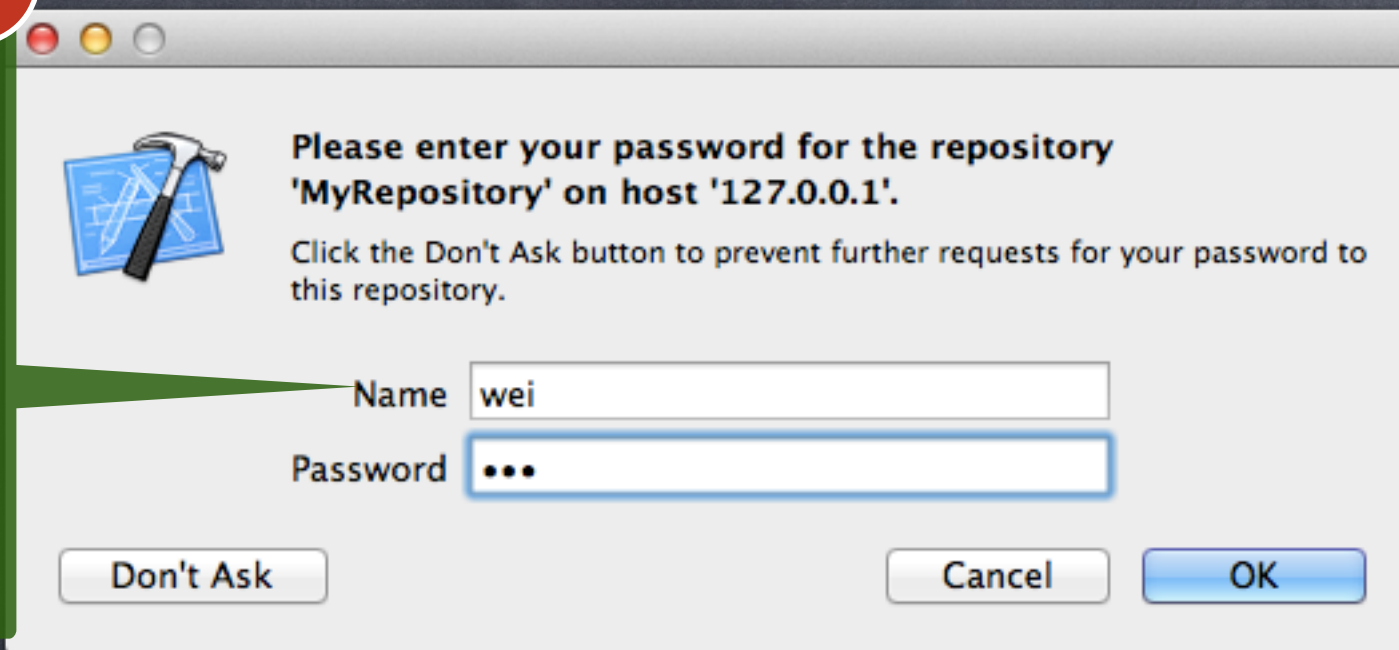
然后在桌面上随意建立一个文件夹如123文件夹，鼠标移动到如图所示位置，点击弹出对话框，选择checkout，有如图所示





1

填你在配置里面填写的用户名和密码，填完点OK



Please enter your password for the repository 'MyRepository' on host '127.0.0.1'.

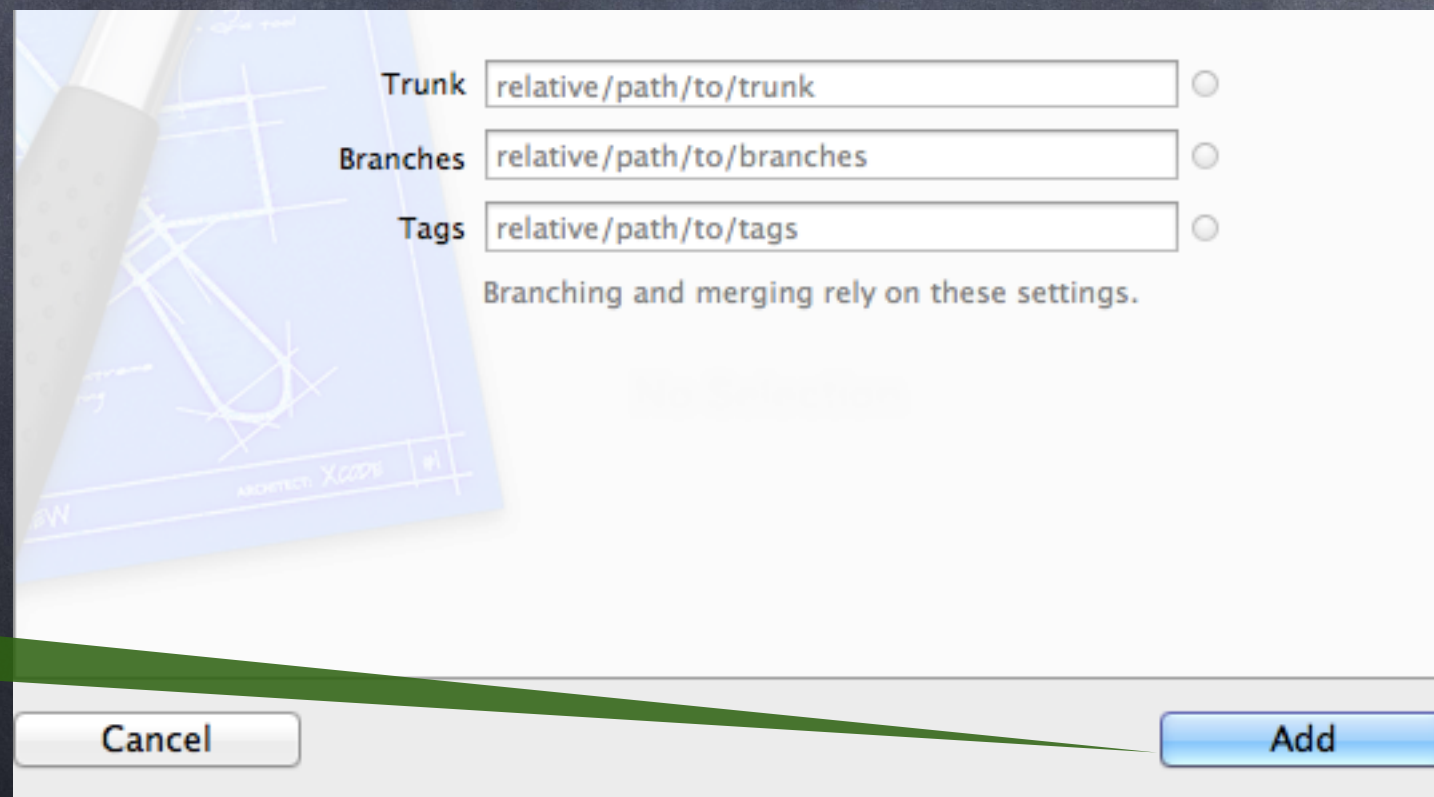
Click the Don't Ask button to prevent further requests for your password to this repository.

Name

Password

2

不需要填写之需要  
点击 Add



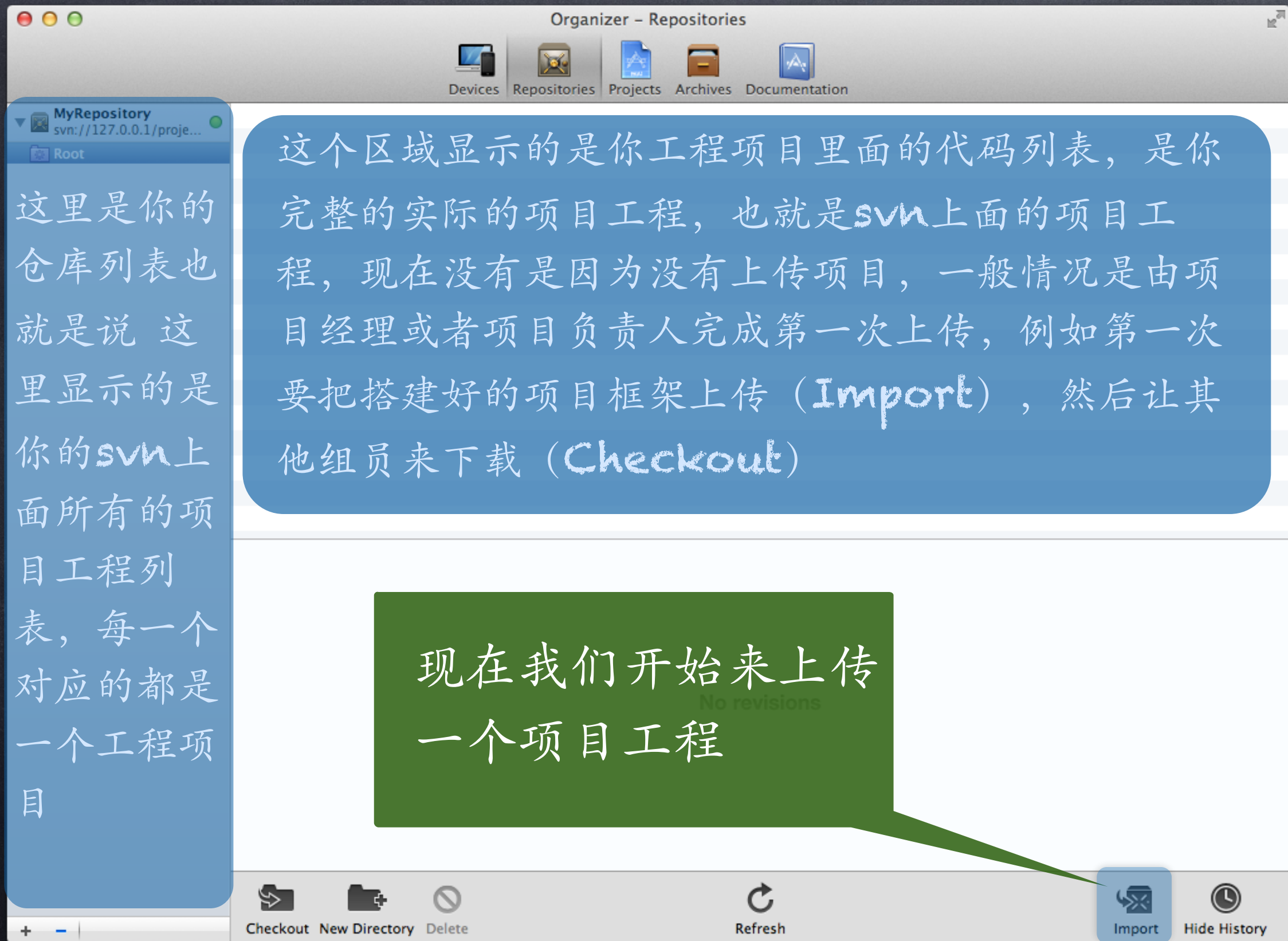
Trunk  ☐

Branches  ☐

Tags  ☐

Branching and merging rely on these settings.





这里是你的仓库列表也就是说 这里显示的是你的SVN上面所有的项目工程列表，每一个对应的都是一个工程项目

这个区域显示的是你工程项目里面的代码列表，是你完整的实际的项目工程，也就是SVN上面的项目工程，现在没有是因为没有上传项目，一般情况是由项目经理或者项目负责人完成第一次上传，例如第一次要把搭建好的项目框架上传（Import），然后让其他组员来下载（Checkout）

现在我们开始来上传一个项目工程

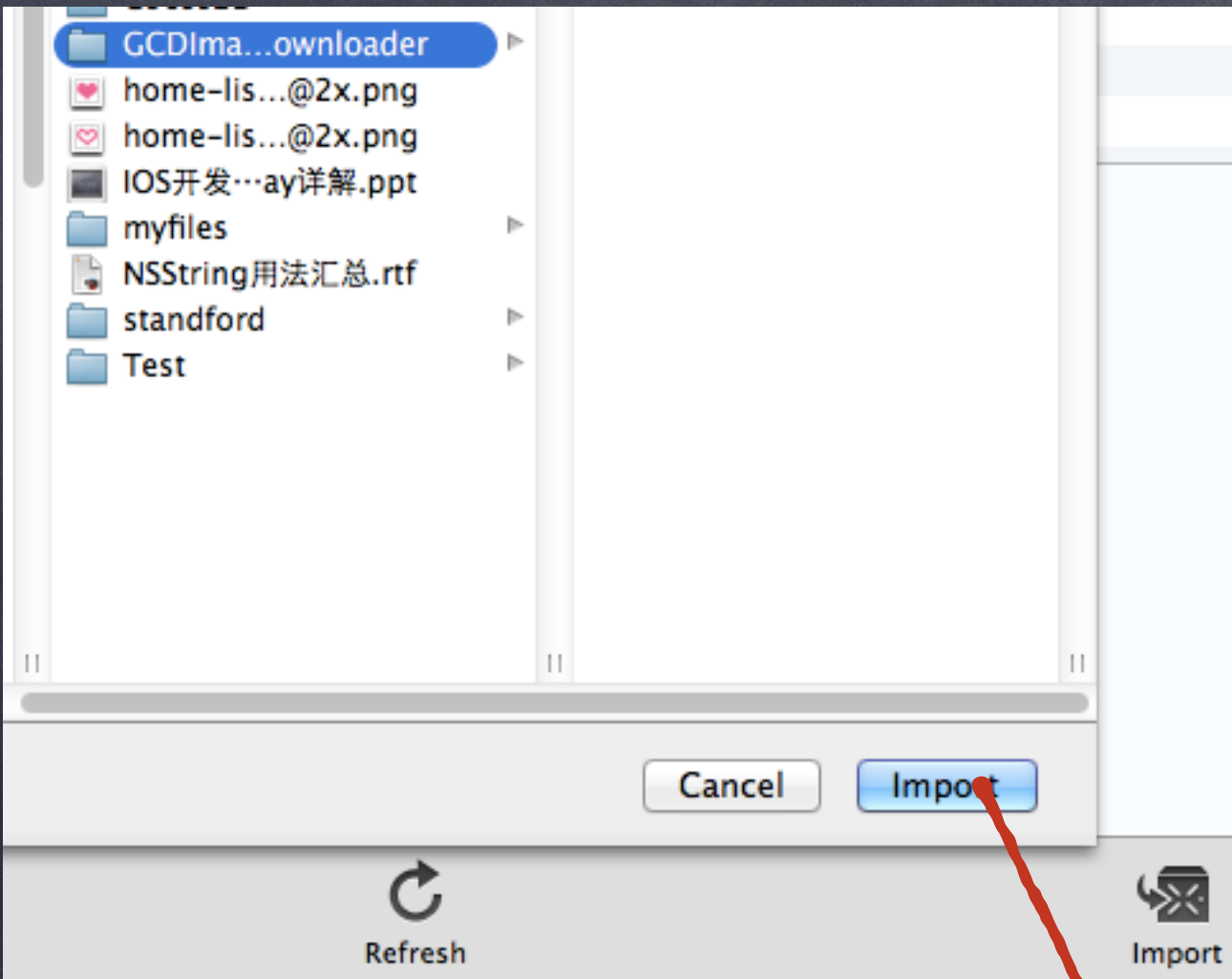
Checkout New Directory Delete

Refresh

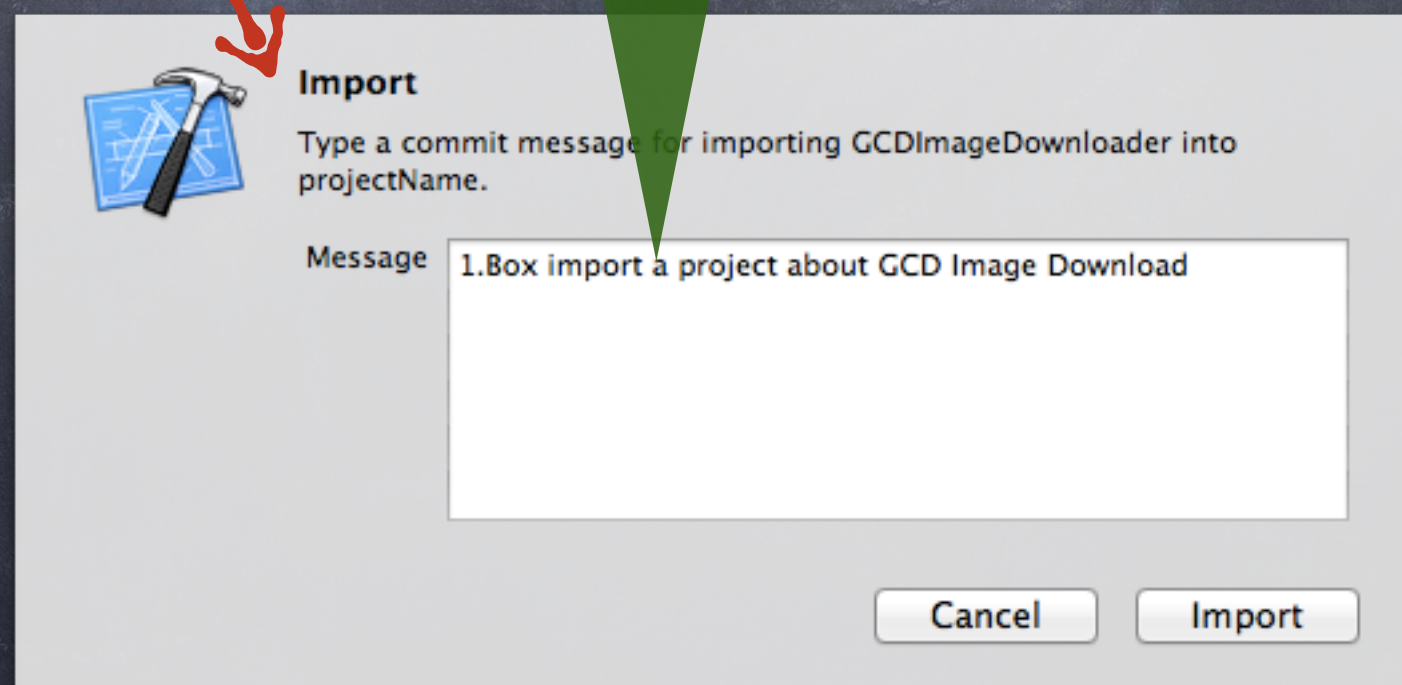
Import

Hide History



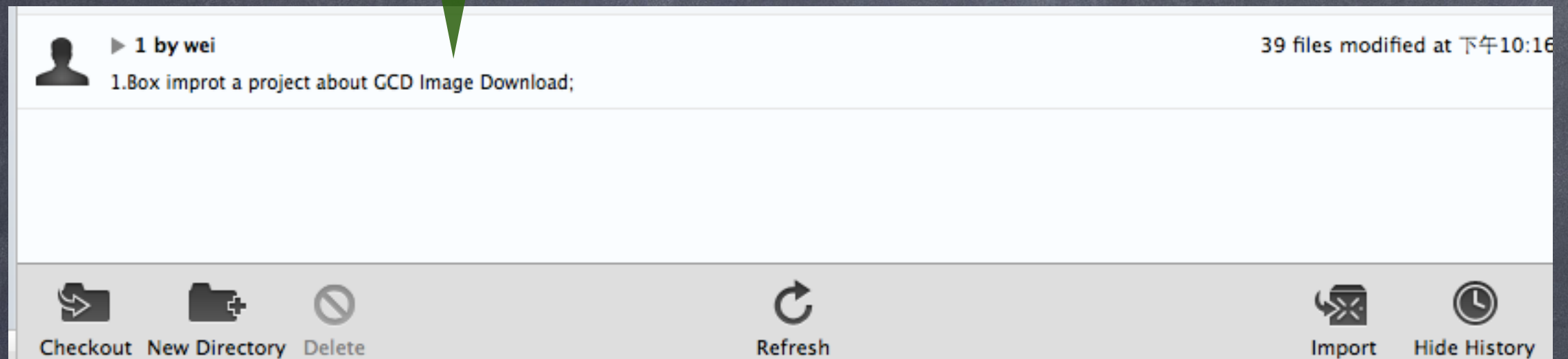


很重要：这里是需要填写日志，也就是你上传一个新的工程或者提交修改BUG之后的工程需要填写的一个修改笔记，此版本修改了啥，做个记录，让团队里面其他的队员知道你做了啥，而且最重要的是要让经理看到，你提交的修改BUG最多也就是说你的年终...大大的...:)





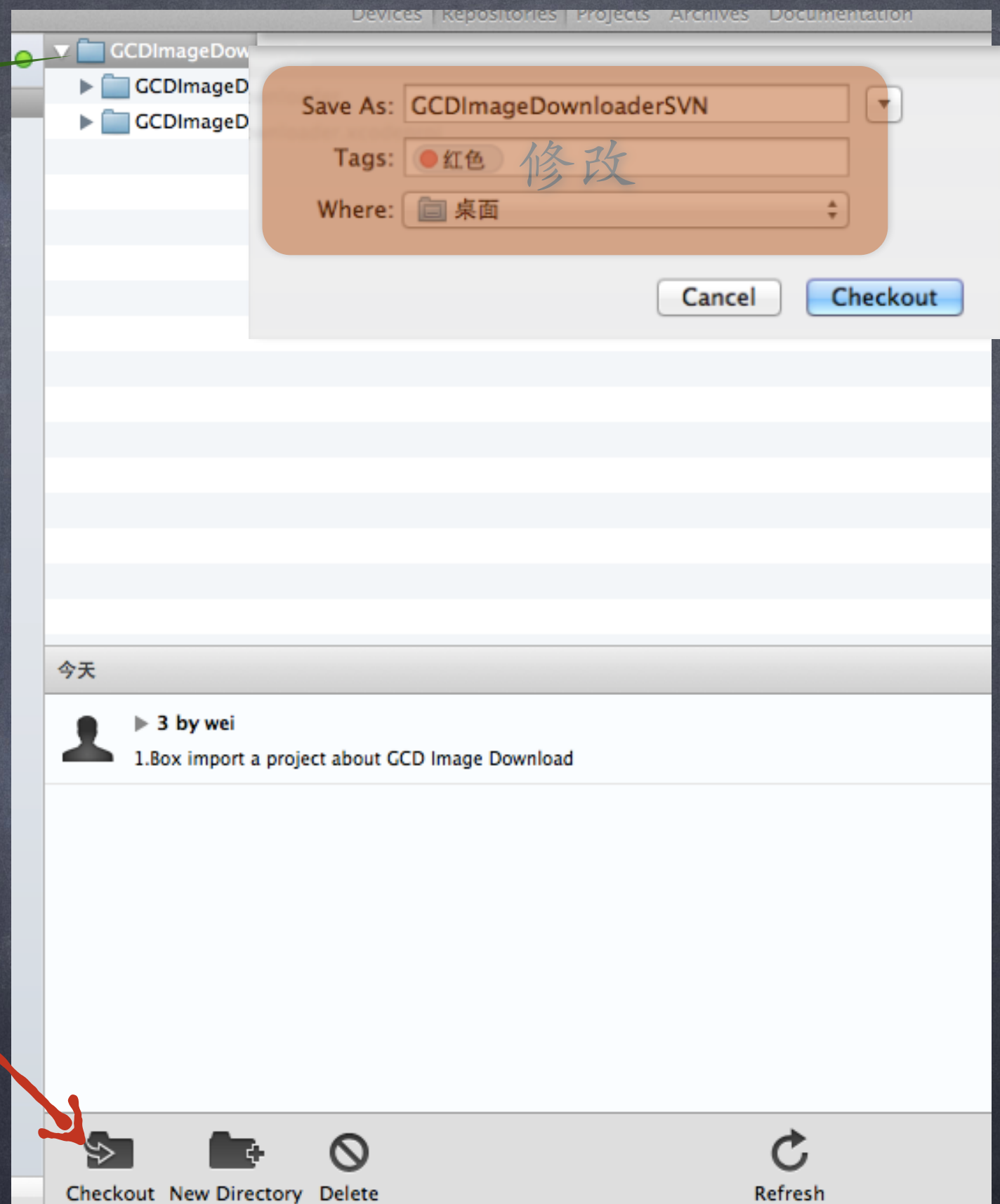
这里就是显示日志的区域，不光有你的还有你团队里面其他组员的，你上传了之后别忘了通知你的其他组员**CheckOut**新的代码



那如何来**checkout**?

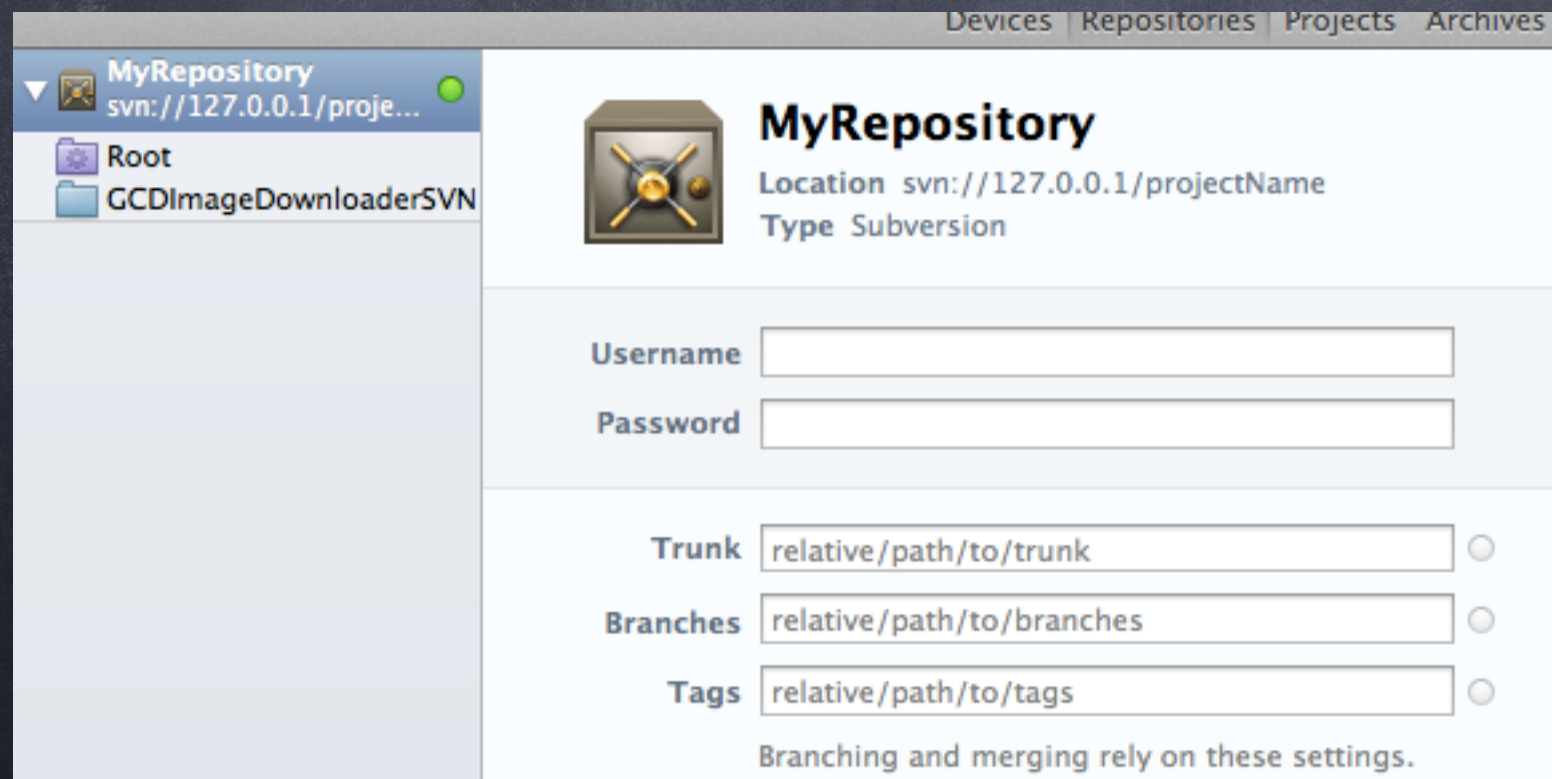
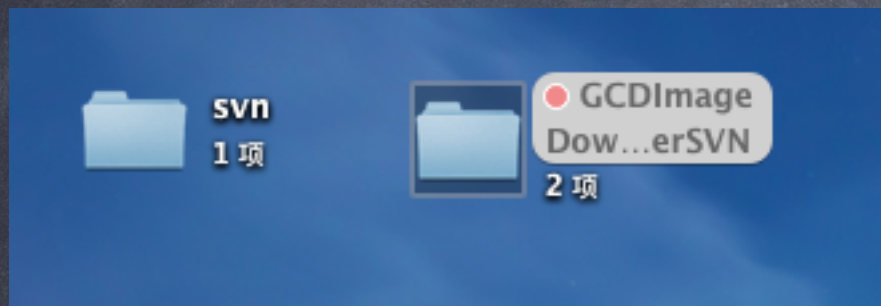
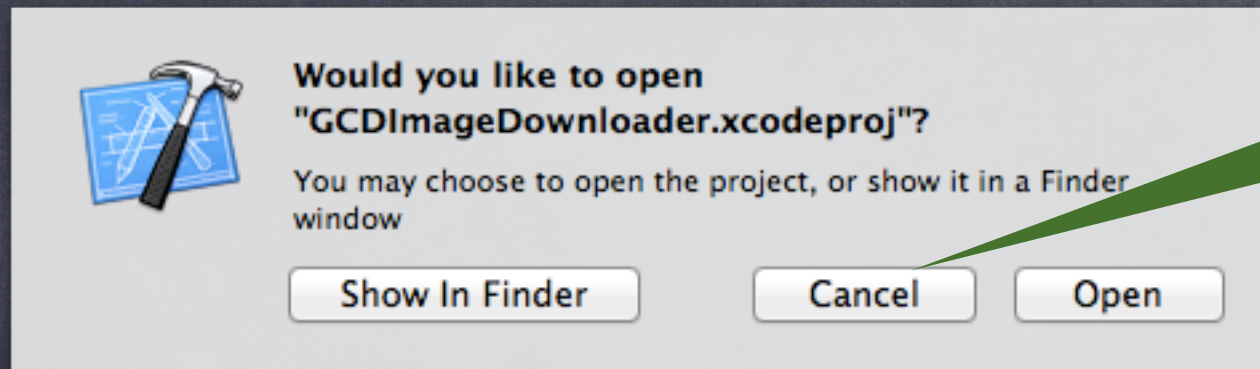


选中在  
SVN的工  
程项目，  
点击  
checkout



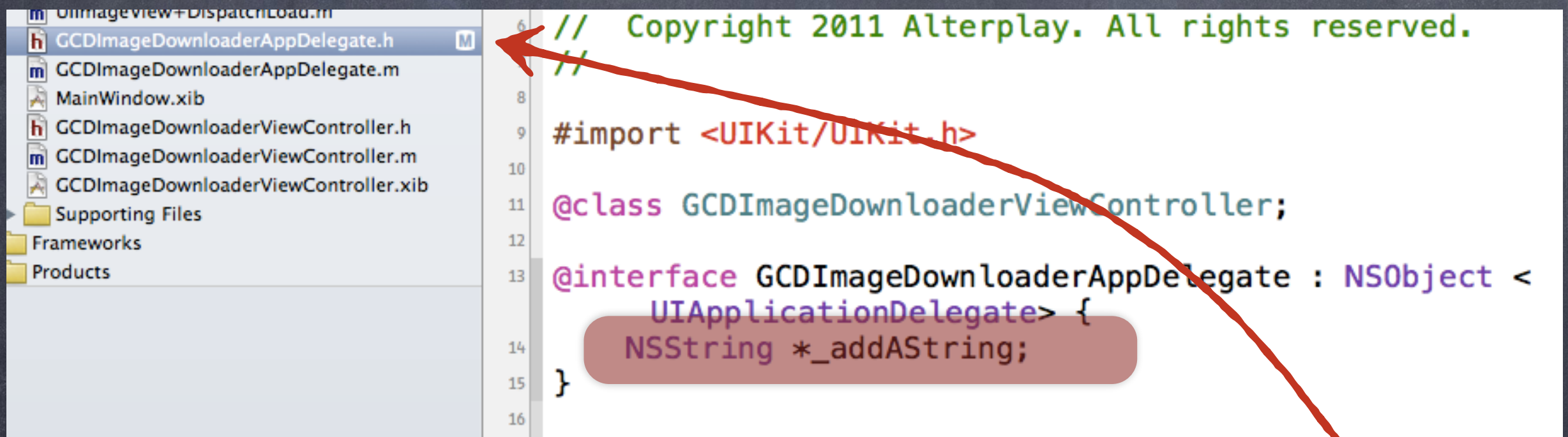


提示要不要打开工程





OK，此时所有队员都已经checkout最新的代码了，现在要做的事情就是如何来 修改代码和提交修改之后的代码，修改代码和之前的是是一样的现在你只需要知道如何提交修改之后的代码。。。打开checkout之后的工程，添加如图所示

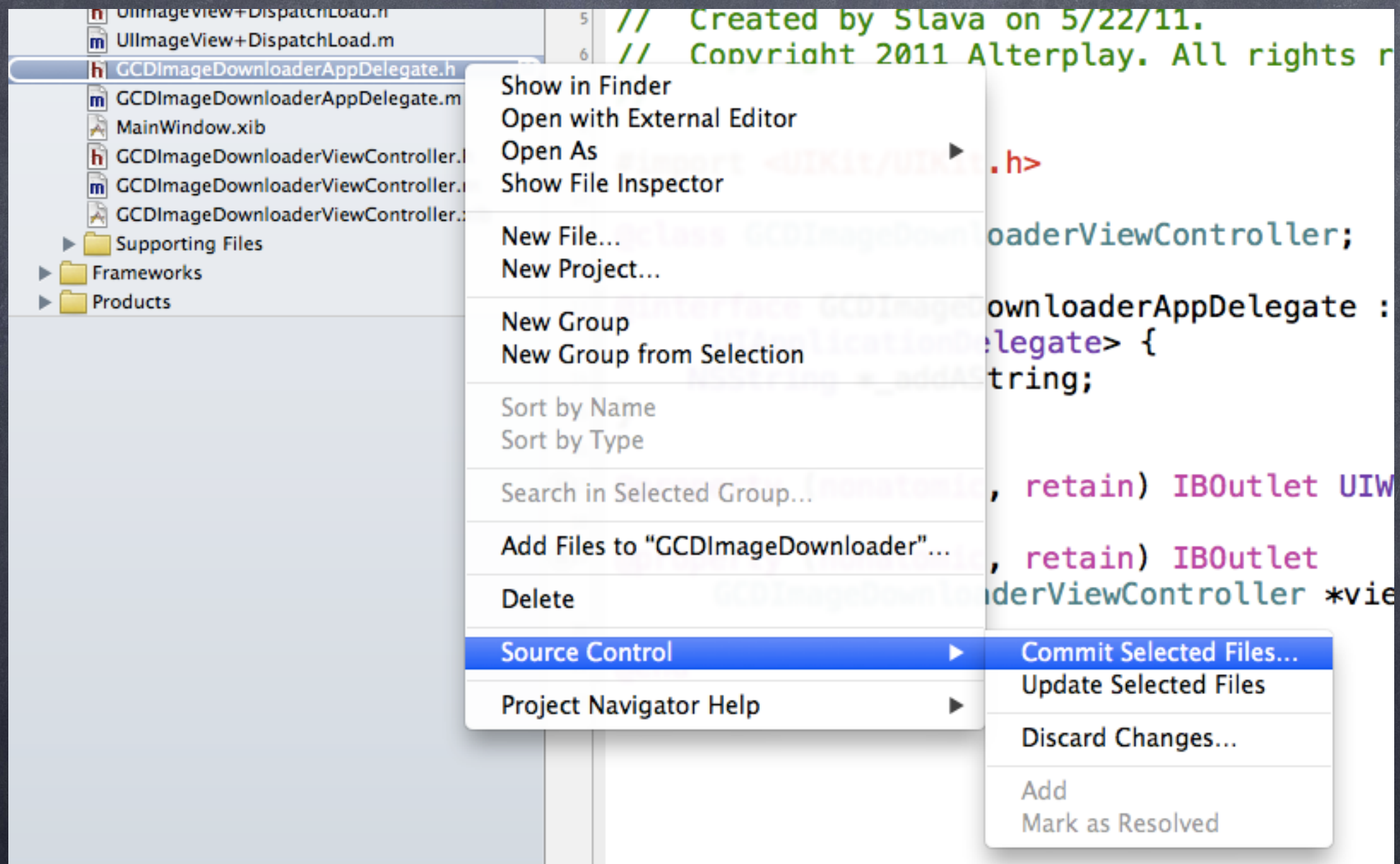


添加完之后保存，你会发现在修改文件的旁边多了一个M

M:Modify U:Update A:Add C:Conflict



# 提交修改





GCDImageDownloader  
1 file changed

GCDImageDownloader  
GCDImage...elegate.h

viewController;

@interface  
GCDImageDownloaderAppDelegate :  
NSObject <  
UIApplicationDelegate> {  
NSString \*  
\_addAString;  
}  
@property (nonatomic,  
retain) IBOutlet  
UIWindow \*window;  
@property (nonatomic,  
retain) IBOutlet  
GCDImageDownloaderVi  
ewController \*  
viewController;  
@end

viewController;

@interface  
GCDImageDownloaderAp  
pDelegate : NSObject  
<  
UIApplicationDelegate  
> {  
}  
@property (nonatomic,  
retain) IBOutlet  
UIWindow \*window;  
@property (nonatomic,  
retain) IBOutlet  
GCDImageDownloaderVi  
ewController \*  
viewController;  
@end

GCDImageDownloaderSVN > Local Revision

GCDImageDownloa... > 14-1-8 wei 3 (BASE, HEAD)

修改之后的  
文件

SVN服务器  
上的文件

对比

填写日志区

Commit to Branch...

Cancel

Commit 1 File



22

GCDImageDownloaderSVN > ⌚ Local Revision

◀ 1 ▶



GCDImageDownloa... > ⌚ 14-1-8 wei 3 (BASE, HEAD)

1.Add a String at the AppDelegate.h file;

Commit to Branch...

Cancel

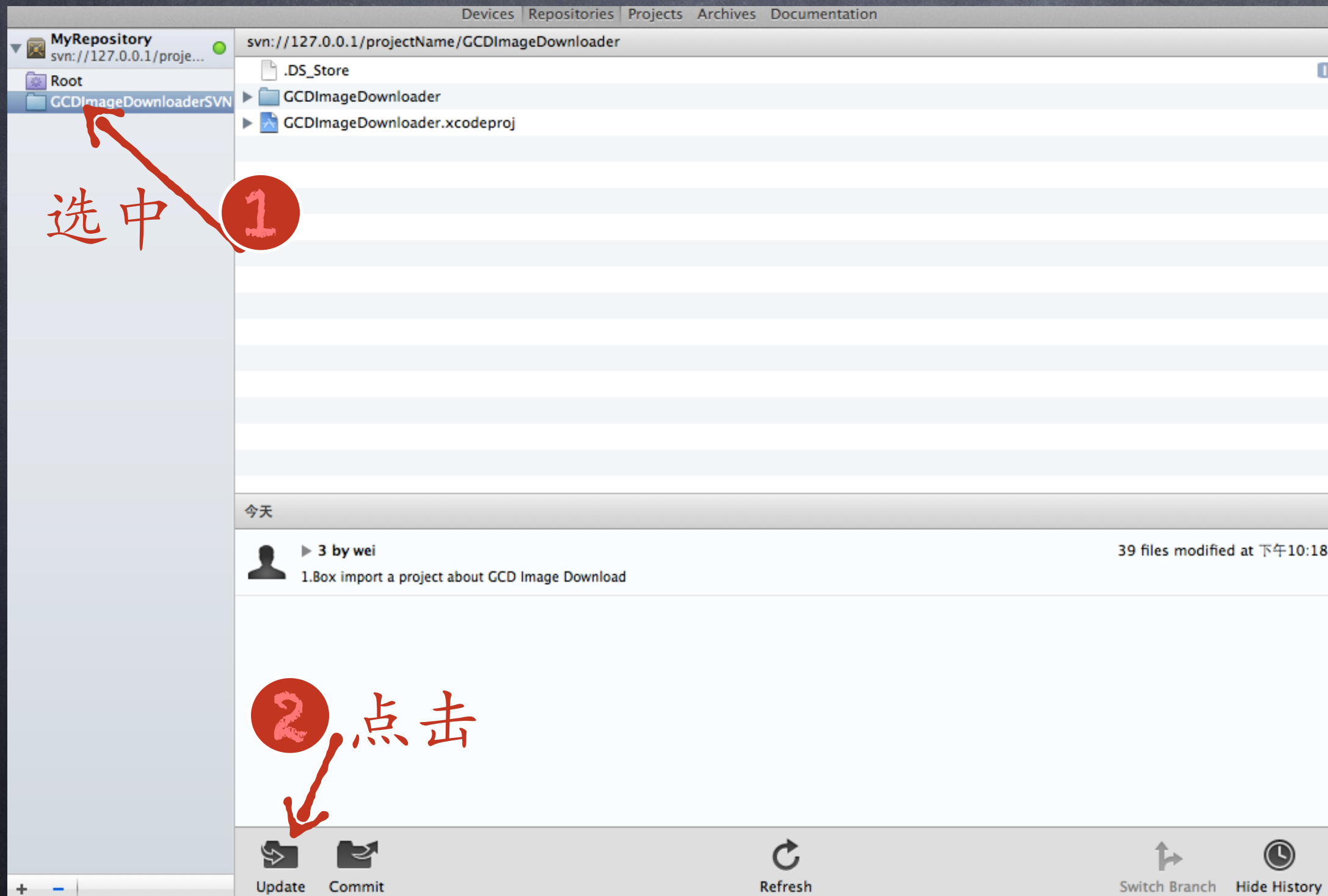
Commit 1 File

今天		
	▶ 4 by wei 1.Add a String at the AppDelegate.h file;	1 file modified at 下午10:36
	▶ 3 by wei 1.Box import a project about GCD Image Download	39 files modified at 下午10:18



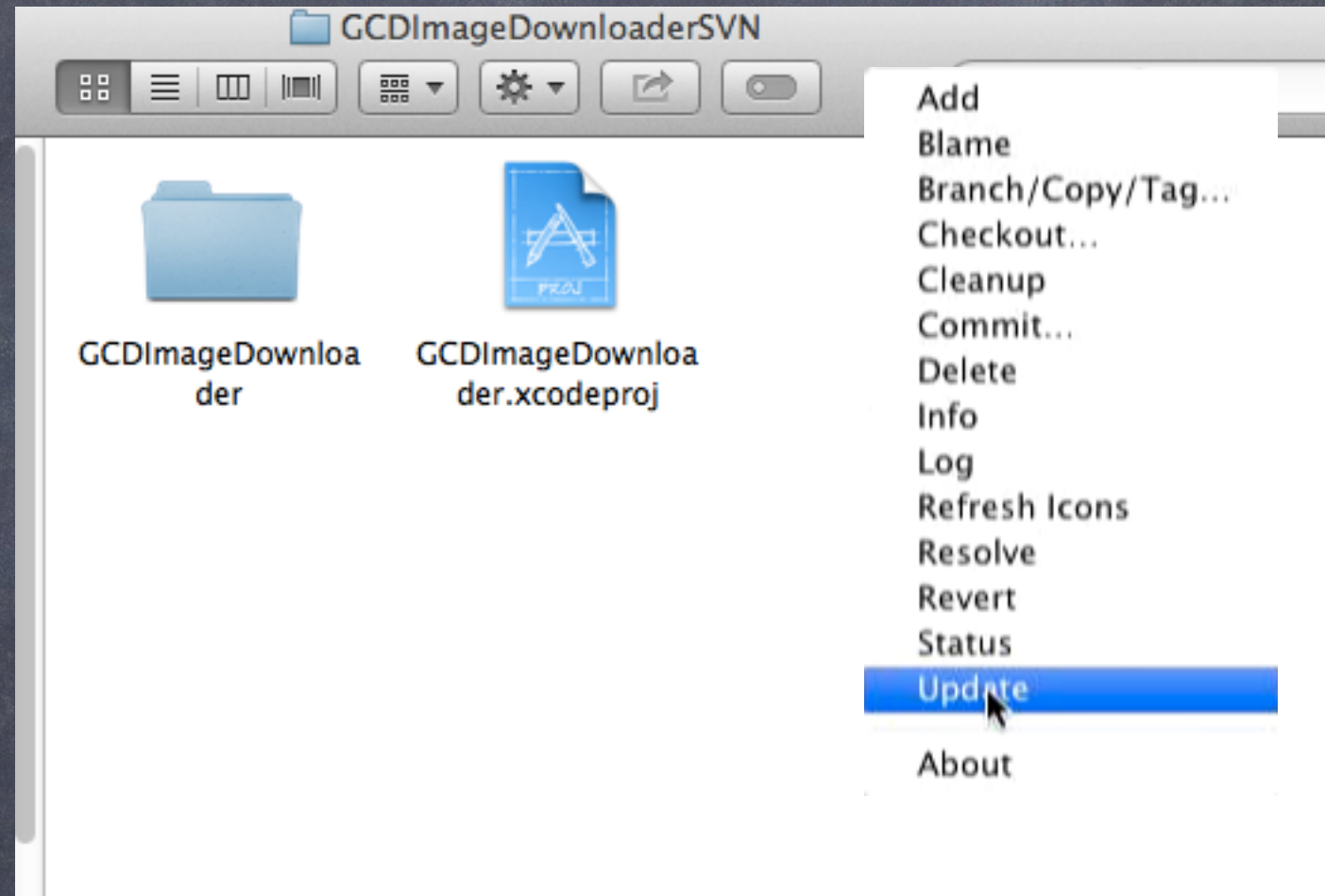
# 如何更新?

## 第一种方式:





第二种方式:



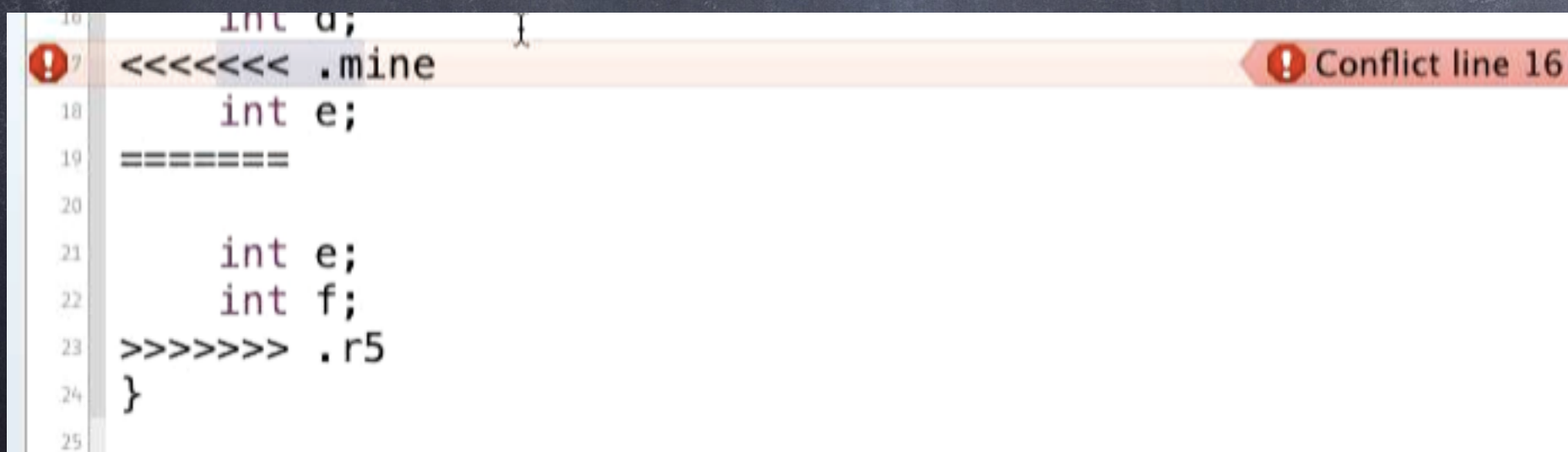


第三种方式：莲花



现在做这个操作：需要一个队员A协助

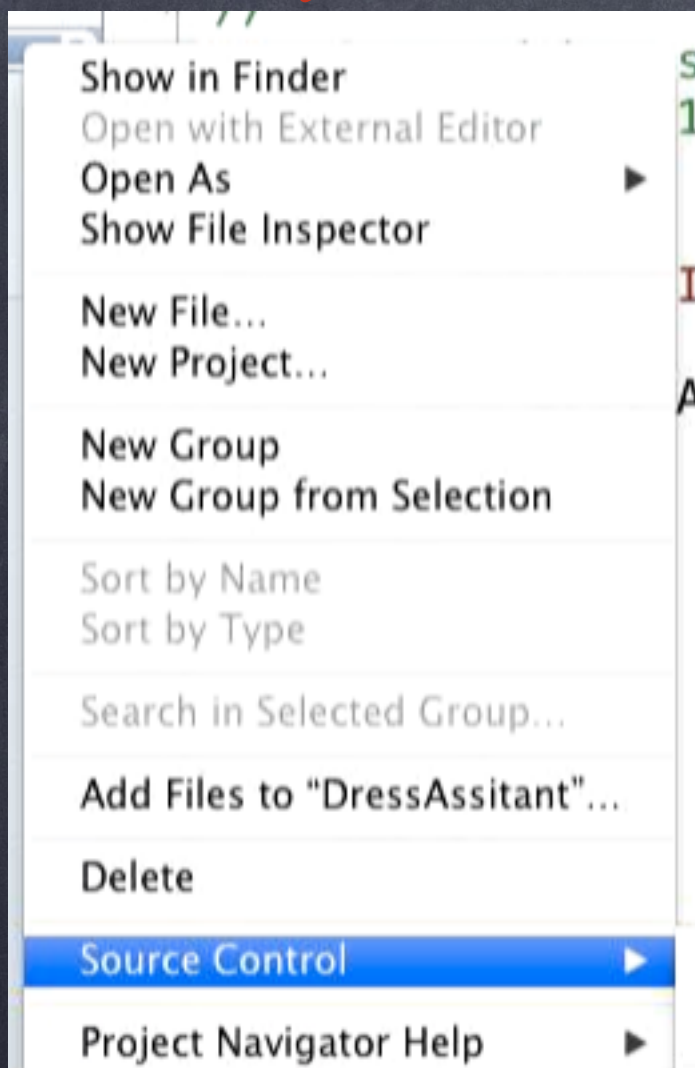
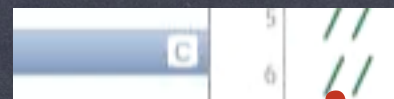
1.首先在xxx.h申明中添加 `int a; int b; int c;` 修改完之后 `commit`；完成之后通知队员A `update`；队员A做完之后，在他的工程里面在同样的xxx.h中添加`int d; commit`，然后你 `update`一下，完成后这个时候 你做这个操作 你在你更新完了之后的xxx.h中添加一个 `int e`；不 `commit`这个时候让队员A在xxx.h上添加一个 `int e; int f;` 之后 `commit`；这个时候你`update`一下 会出现这个情况：



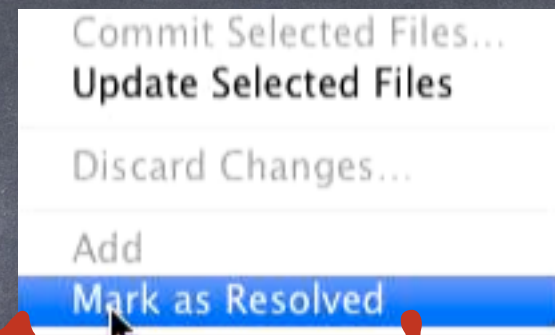
```
16      int d;
17  <<<<<<< .mine
18      int e;
19  =====
20
21      int e;
22      int f;
23  >>>>>>> .r5
24  }
```

Conflict line 16

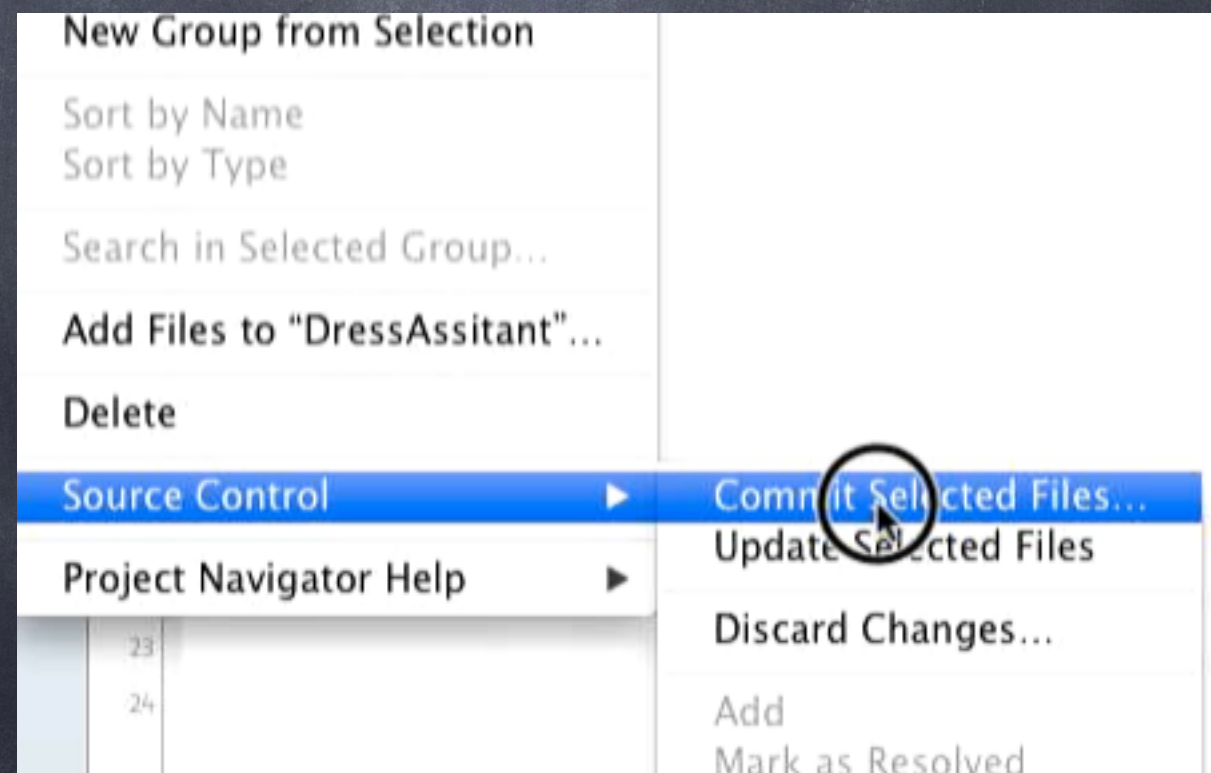




这个时候不能  
**commit**是因为这  
个时候是冲突的不  
能上传所以必须先  
解决冲突



再次上传





# 如何添加文件删除文件

1. 添加文件要注意的地方是你提交的时候要同时把你要增加的文件和工程文件一起提交，为啥？因为你的工程文件是你整个项目文件的索引所以必须都要一起提交；
2. 这种情况的出现：队员A提交了 增加的文件和工程文件，这个时候其他队员也都分别不同程度的 增加了文件或者修改了 自己的工程文件的索引，这个时候 其他队员 `update` 整个工程，会发现你的工程打不开了，这个时候莫急，没什么大不了的，这个时候解决的办法是去你 工程所在的文件夹目录下，把工程文件删除掉，重新 `update` 一下就OK了，不过这个时候是队员A的工程文件也就是他的工程文件的索引，so 你苦X了 只能重新把你之前增加的东西在增加一边，然后 OK后立马 `commit` 一下，记住是去 工程文件所在的文件夹下面去删除而不是在 `xcode` 里面删除，切记，为啥后续。



附：Xcode 5.0+使用SVN



Update your project with changes from the repository using the Source Control menu.

For a Git repository, choose Source Control > Pull.

For a Subversion repository, choose Source Control > Update.

For a project that contains both Git and Subversion repositories, choose Source Control > Update and Pull.

For projects with multiple repositories, select the ones you want to update.

Resolve differences by using the left and right buttons to specify which file's contents to use.

After reconciling all differences, click Pull (Git) or Update (SVN) to complete the operation.

XCODE5中怎么上传到SVN服务器，老外给出的结论是：两个办法，一个是用svn client，客户端软件；另一个方法是命令行。**One way is using an svn client. The one which is obviously available is the command line svn client.**



# 肿么办? 手敲最实惠

```
svn import /Users/Wei/Desktop/ZY1920140114/  
ZYCoreDataDemoSuper svn://127.0.0.1/MyProjectName -m  
"inital import"
```

```
bogon:~ Wei$ svnserve -d -r /Users/Wei/Desktop/svn  
bogon:~ Wei$ svn import /Users/Wei/Desktop/ZY1920140114/ZYCoreDataDemoSuper  
svn://127.0.0.1/MyProjectName -m "inital import"
```



创建本地代码仓库



### Repository

Create a new bookmark to a repository URL



### Working Copy

Create a new bookmark to an existing working copy



### Free Online Repository

Create a repository at Basecamp

Update Commit Checkout

Local Changes Compare Blame History Quick Look Inspector

Revert Add Delete

## BOOKMARKS



### New Repository Bookmark

这个是你本地仓库名字建议你用项目+版本作为标签

Name: MyProject\_V1.0

Location: svn://box@127.0.0.1/MyProjectName

这是SVN服务器的路径，写的时候格式为 svn://服务器IP/代码仓库名，就是项目的SVN的路径

Username: box

Password: ...

☐ Show links to issue tracker in timeline

URL Prefix https://server/tickets/%s

Comments in the timeline that match #123 will be linked to the url, %s will be replaced by 123.

Cancel

Create

Create a local repository