

SVN 的使用

SVN 简介

工作流程

优缺点

SVN的目录结构

意义

SVN Server 服务端搭建（Window）

VisualSVN Server 安装

VisualSVN 创建仓库

建立用户组和用户

设置用户权限

SVN Server 服务端搭建（Ubuntu）

安装 Subversion

创建仓库

设置权限

客户端 TortoiseSVN

安装

使用

功能介绍

Get Lock 锁

更新与提交

Checkout 与 Exprot

Update to revision 与 Revest

分支

提交与合并冲突解决

其它参考资料

Elicpse 插件

SVN 的使用

SVN 简介

SVN 是 Subversion 的简称，是一个开放源代码的版本控制系统，相较于RCS、

CVS，它采用了分支管理系统，它的设计目标就是取代CVS。互联网上很多版本控制服务已从CVS迁移到Subversion。

SVN = 版本控制 + 备份服务器

工作流程

集中式代码管理的核心是服务器，所有开发者在开始新一天的工作之前必须从服务器获取代码，然后开发，最后解决冲突，提交。所有的版本信息都放在服务器上。如果脱离了服务器，开发者基本上可以说是无法工作的。

优缺点

所有的文档都显示 SVN 可以取代 CVS，同时SVN的问题和缺点都被隐藏了。不幸的是，我们并不认为SVN是CVS的替代品，尽管很多缺陷都被修改了。更有甚者，它甚至让人重回CVS。CVS和SVN的比较类似于比较C++和Java。很明显 CVS和SVN都远比 SourceSafe 强大的多，如同 C++ 和 Java 比 Basic 强大的多。CVS 代表了几乎代码控制系统的所有功能项，尽管有时他的实现并不很方便。SVN 修正并添加了一些 CVS 并不拥有功能。例如，创建标志和分支 dubious，你在编辑文件时其他人不会有任何通知。SVN 并不是 CVS 的替代品，只是个不同的系统，类似于 CVS。它有些特有的功能，足以作为采用它的理由。这些功能使他更适合于开发环境，例如对PowerBuilder。下面你可以找到两者的相对优势、劣势。

1. 存储类型格式

CVS 是个基于 RCS 文件的版本控制系统。每个 CVS 文件都不过是普通的文件，加上一些额外信息。这些文件会简单的重复本地文件的树结构。因此，不必担心有什么数据损失，如果必要的话可以手工修改 RCS 文件。

SVN 是基于关系数据库的(BerkleyDB)或一系列二进制文件的(FS_FS)。一方面这解决了许多问题 (例如，并行读写共享文件)以及添加了许多新功能(例如运行时的事务特性。)。然而另一方面，数据存储由此变得不透明。

2. 速度

CVS 比较慢。

整体而言,由于架构实现的不同，SVN 的确比 CVS 快很多。在网络上它只传输很少的信息并支持更多的离线模式的功能。但这也是有代价的。速度的代价就是巨大的存储（完全备份所有的工作文件）。

3. 标志&分支

SVN 采用标志和分支而抛弃了其他三件东西，实际上这意味着他们把这个概念替换为在档案库内部复制文件或目录以便保存日志。这样一来，无论标志创建还是分支创建都只是仓库内部的文件复制了。对分支而言：分支不过是在仓库内部的一个单独的目录而已了，不像早期还做些什么交错。对标志而言：已经不能对代码加标志了。在某种程度上说，SVN 全文件编号补足了这个缺陷，SVN 里整个仓库都有版本号，但不是针对单个文件。

4. 元数据

CVS 只允许存储文件。

SVN 允许一个文件有任意多的可命名属性，功能十分完全。

5. 文件类型

CVS 最初是为文本文件存储而设计的。因此其他文件类型（二进制，统一码）文件的支持几乎没有，如需要的话则要有其他信息，并且客户端服务器端都要调整。

SVN 会关心所有的文件类型，不需要你来手工操作。

6. 回滚

CVS 允许任意的回滚，在任意一个已递交的版本上，尽管这要花些时间（所有的文件都要分别处理）。

SVN 不允许递交后回滚。建议把版本库里好的状态版本加到末尾，覆盖掉损坏的版本。而损坏的版本无论如何也是会存在数据库里的。（SVN 的滚回操作实际上是 merge 操作）

7. 事务

CVS 中的“零或一”事务原则根本没有实现。如果检入几个文件的话（加到服务器上），很有可能部分文件完成了，而另几个没有。作为一个潜规则，手工纠正这些并且对余下的文件（而不是所有文件）一一重复检入。这样这些文件将在两阶段中被检入。SVN 的确支持“零或一”事务原则，这是 SVN 的一大优势。

SVN的目录结构

SVN 的标准目录结构：trunk、branches、tags

trunk 是主分支，是日常开发进行的地方。

branches 是分支。一些阶段性的 release 版本，这些版本是可以继续进行开发和维护的，则放在 branches 目录中。又比如为不同用户定制化的版本，也可以放在分支中进行开发。

tags 目录一般是只读的，这里存储阶段性的发布版本，只是作为一个里程碑的版本进行存档。

意义

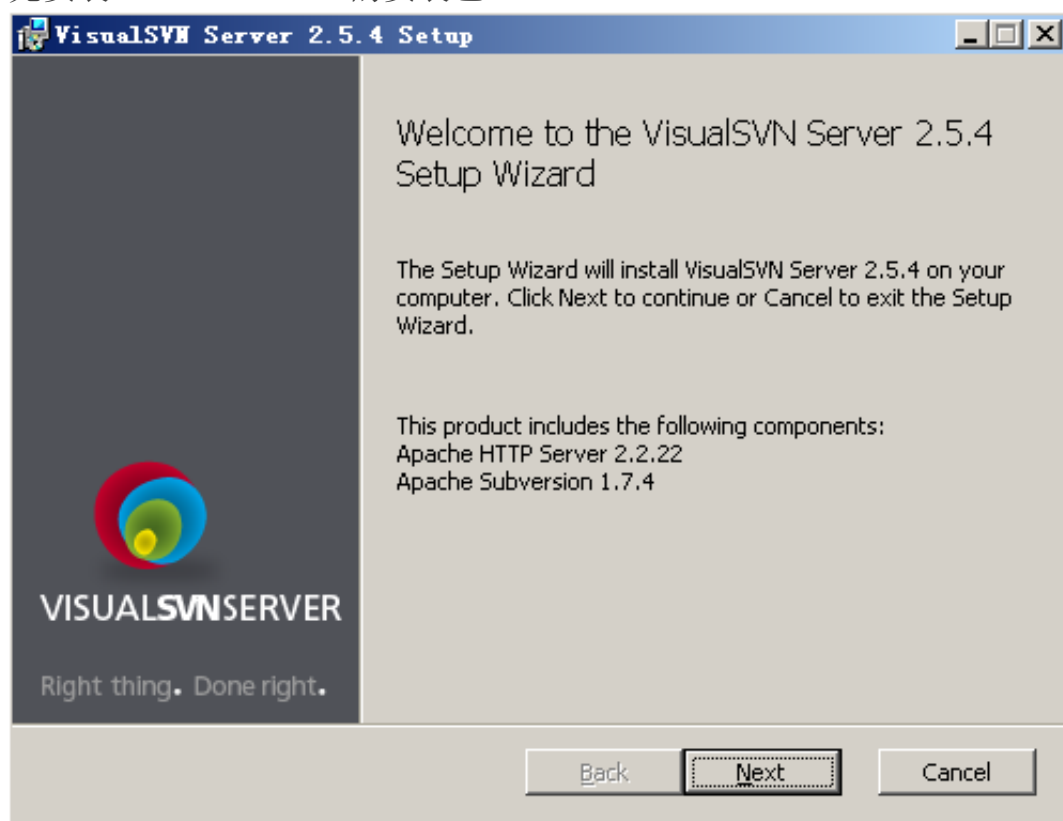
SVN 解决了我们数据备份、版本控制、数据同步等面对的问题，因而深受很多人喜爱和使用，它是一个不错的管理知识的软件工具。

我们自己也可以用SVN管理我们的知识。

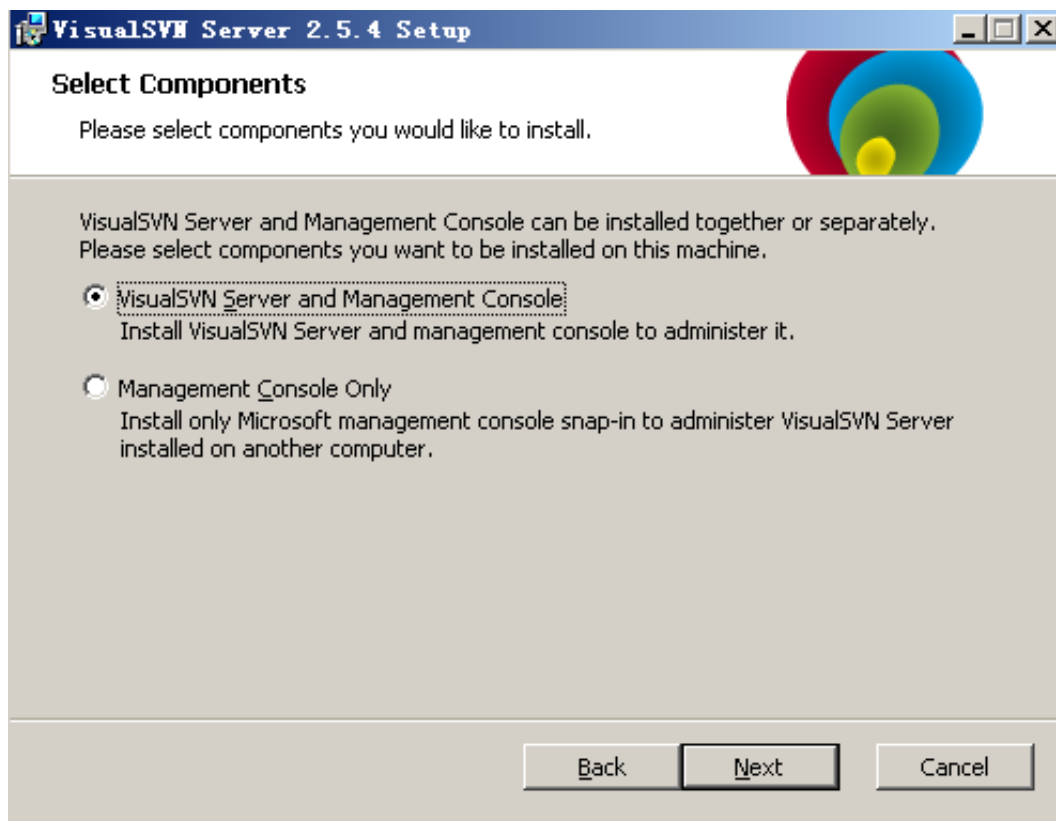
SVN Server 服务端搭建（Window）

VisualSVN Server 安装

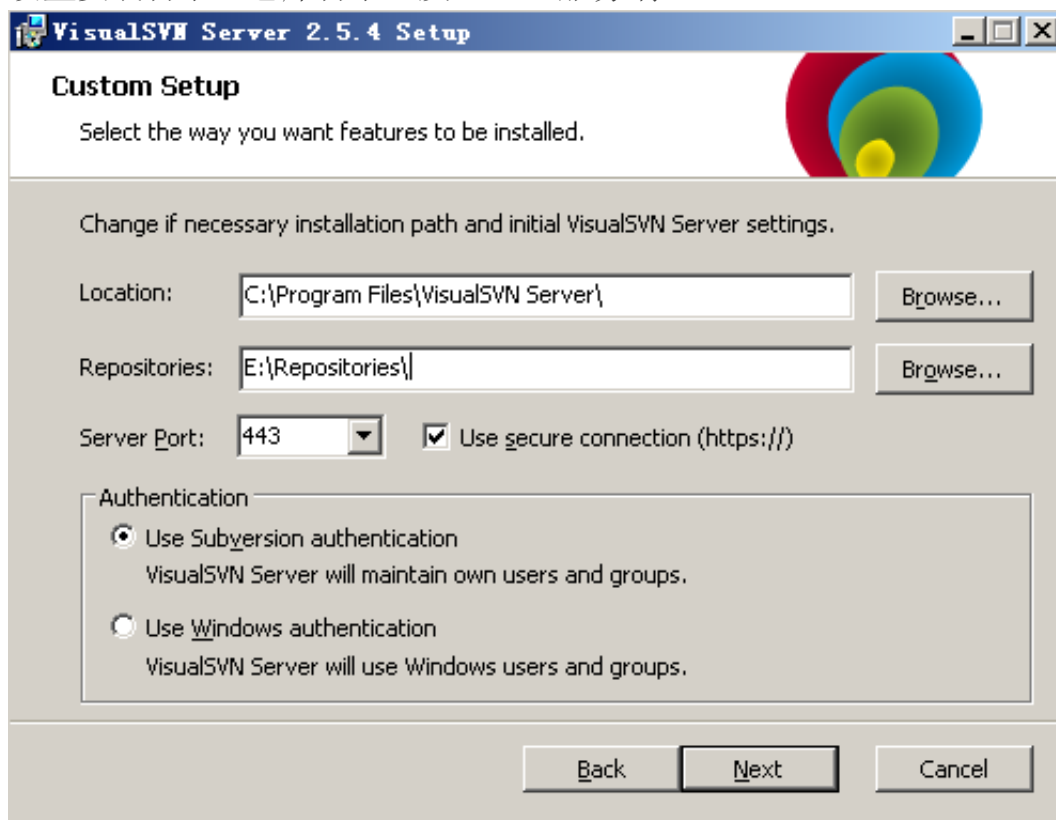
先安装VisualSVN server的安装包



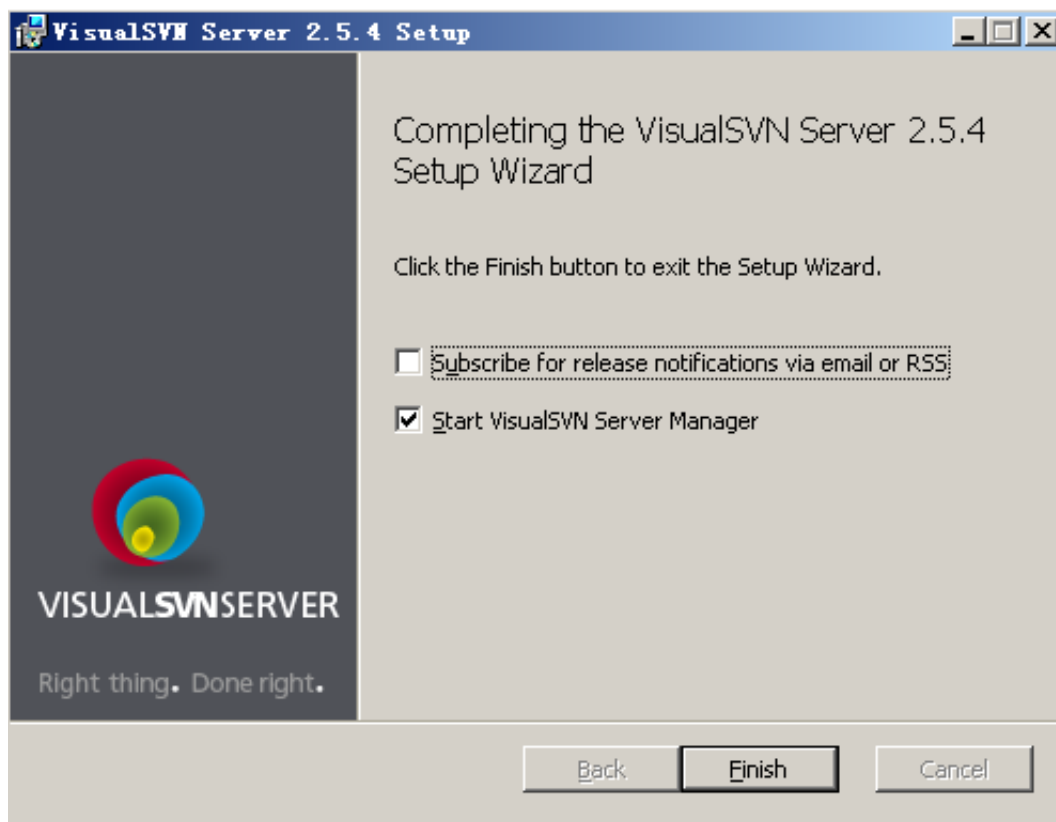
选择安装 Server 服务，并安装控制台管理



设置安装目录、仓库目录，及 Server 服务端口



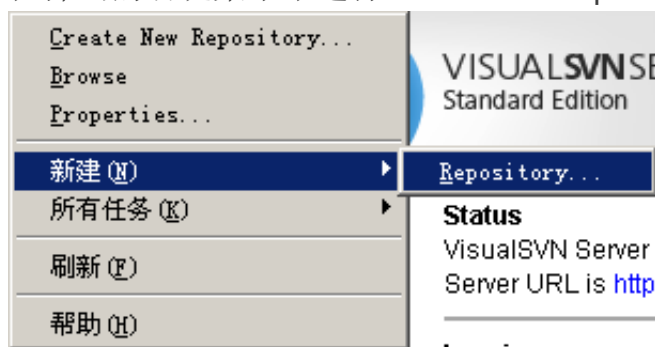
安装完成



VisualSVN 创建仓库

可以在窗口的右边看到版本库的一些信息,比如状态,日志,用户认证,版本库等.要建立版本库,需要右键单击左边窗口的Repository

在弹出的右键菜单中选择Create New Repository或者新建->Repository



输入版本库名称,勾上Create default structure复选框(推荐这么做).点击OK,版本库就创建好了,版本库中会默认建立trunk,branches,tags三个文件夹。

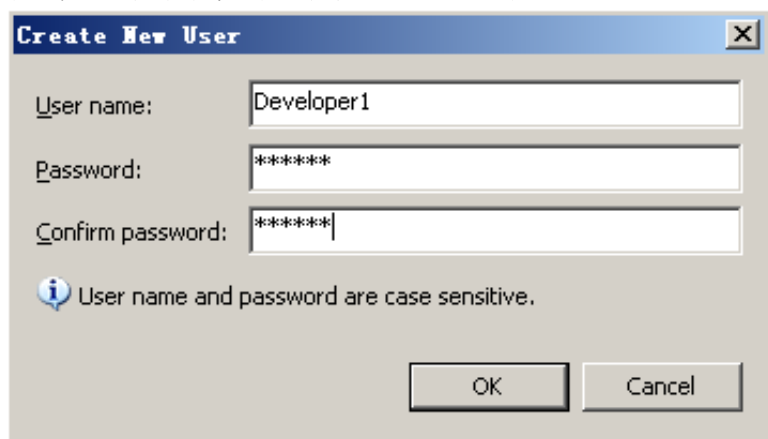


建立用户组和用户

在VisualSVN Server Manager窗口的左侧右键单击用户，选择 Create User或者 新建->User，创建用户。



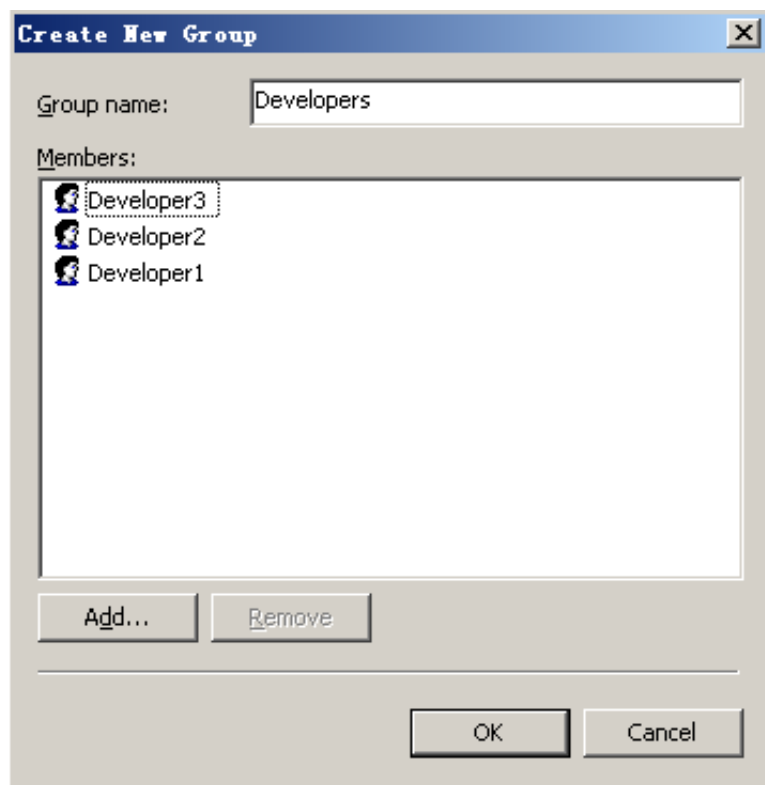
在弹出的对话框中填写User name和Password。



在VisualSVN Server Manager窗口的左侧右键单击用户组，选择 Create Group或者 新建->Group，创建用户组。



Add按钮,在弹出的窗口中添加用户。



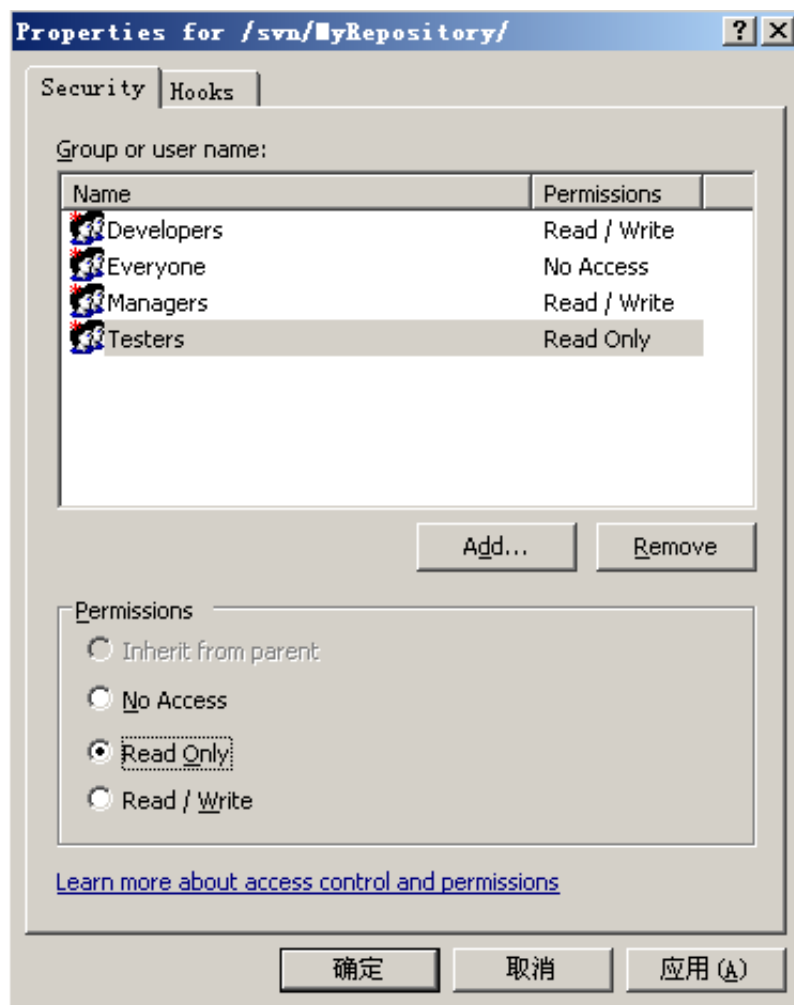
设置用户权限

接下来我们给用户组设置权限

在已创建的仓库名上单击右键



选择Security选项卡，点击 Add 按钮,选中某一用户组,然后添加进来，并给用户组设置权限。



SVN Server 服务端搭建（Ubuntu）

安装 Subversion

首先 subversion 这个软件

```
apt-get install subversion
```

创建仓库

在 home 目录下创建一个名为 svn 的文件夹（文件夹的名字随便起）

```
mkdir /home/svn
```

创建数据仓库（可以根据需要创建多个）

```
svnadmin create /home/svn/test
```

启动svn网络服务

```
svnserve -d -r /home/svn
```

其中 -d 参数让 svnserve 运行在后台，-r 参数限定了数据仓库，在网络上可以访问的地址。

/home/svn 指定 svn 数据仓库存放的目录位置

设置权限

设置 svn 的访问权限

每个数据仓库目录下都有一堆目录，进入 conf 并打开 svnserve.conf 这个文件，找到以下几行，
并把前面的注解符号‘#’去掉，注意千万在每行的前面别留任何空格。

```
anon-access = none  
auth-access = write  
password-db = passwd  
authz-db = authz
```

其中 anon-access 和 auth-access 分别为匿名和有权限用户的权限，默认给匿名用户只读的权限。

password-db 用户设置文件

authz-db 权限设置文件

编辑 authz 文件，添加用户组

编辑authz 制定管理员组 即admin组的用户为tone admin组有 rw（读写权限），所有人有r（读权限）。

```
[groups]  
admin = tone
```

```
[/]
```

```
@admin = rw  
*=r
```

编制 passwd 文件，添加用户及密码

```
[users]  
admin = admin  
zhangsan = 123
```

客户端 TortoiseSVN

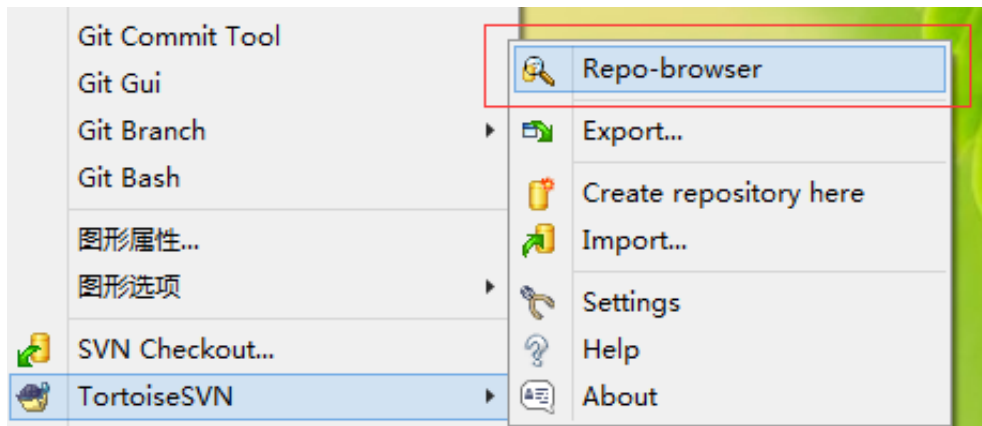
安装



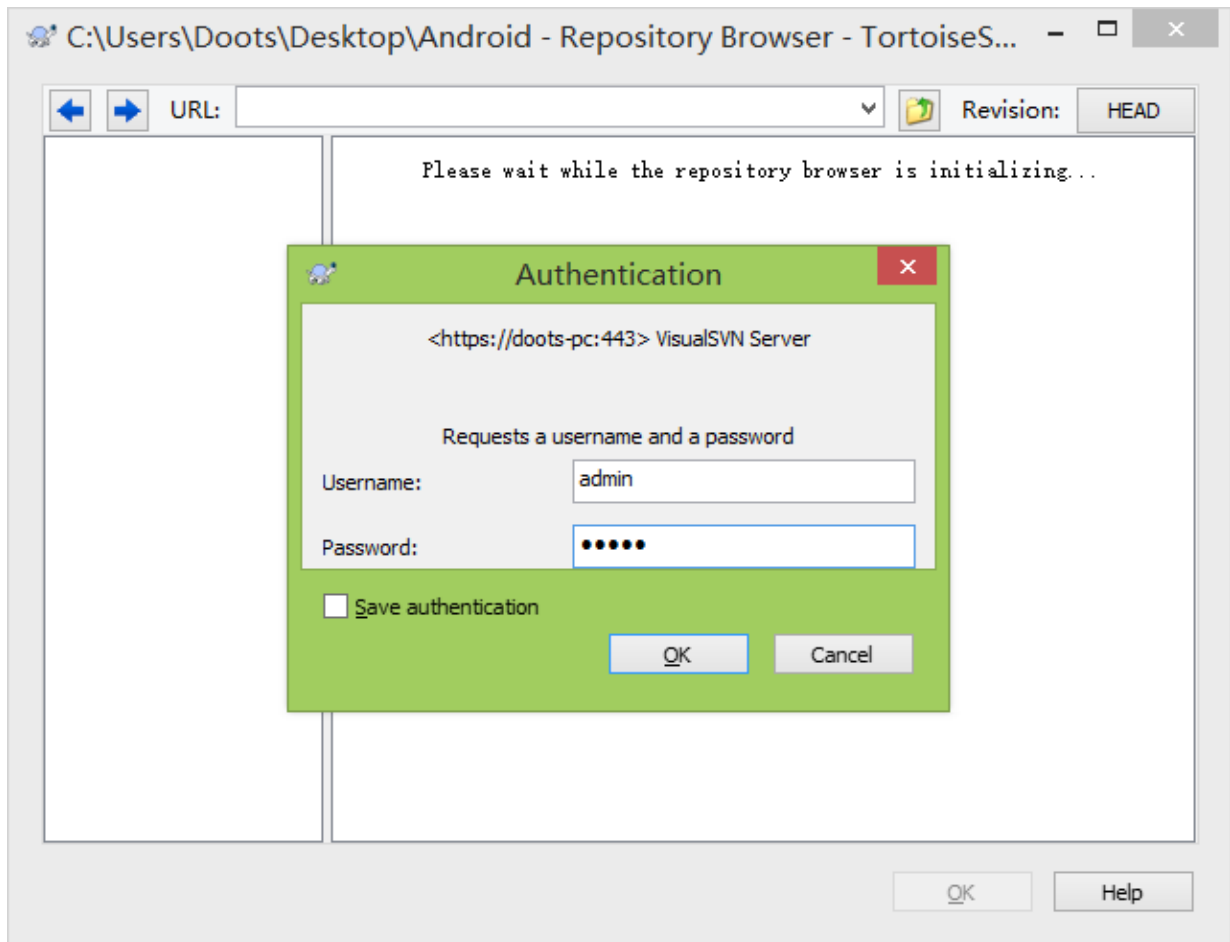
之后一路顺风点下一步即可。

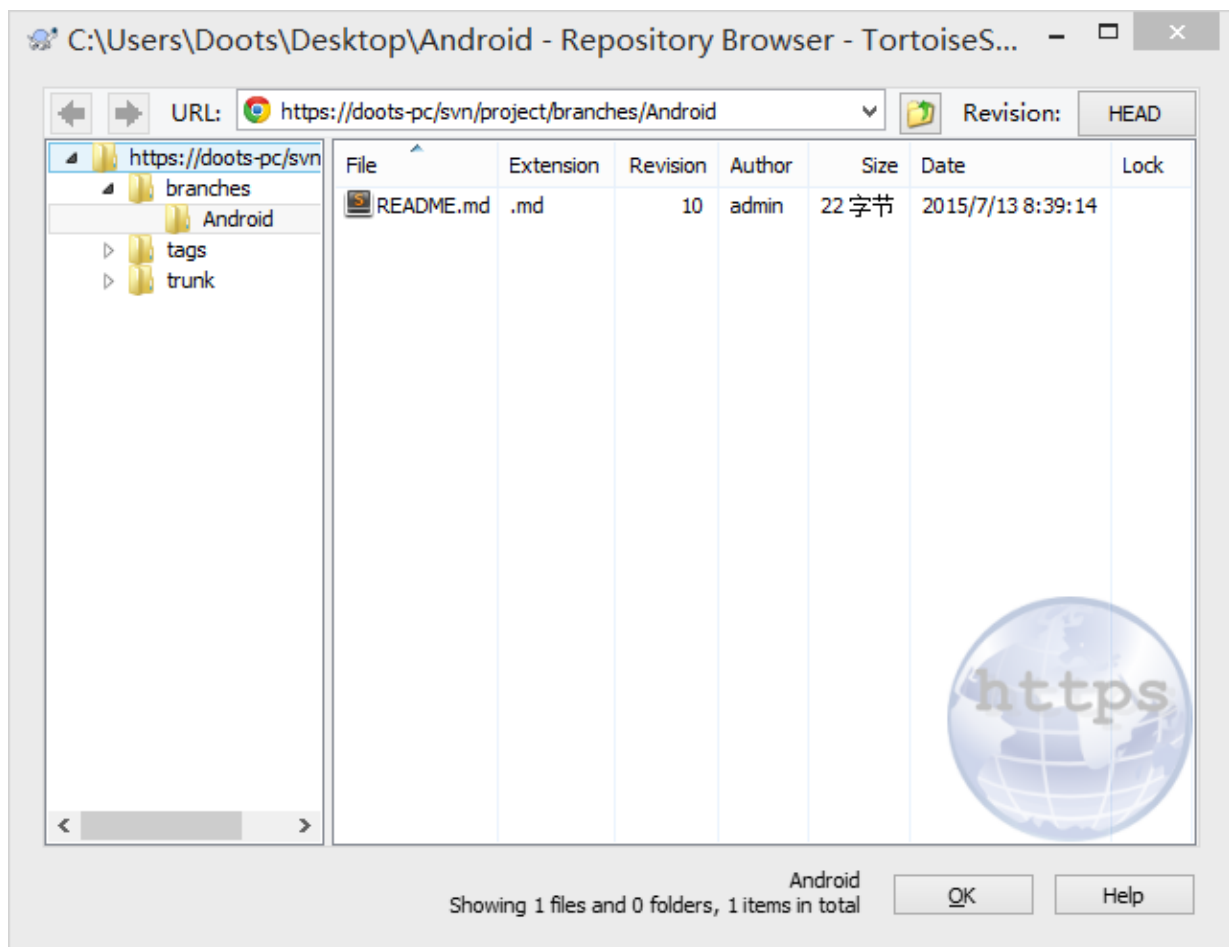
使用

在桌面任意位置右键依次选择 TortoiseSVN -> Repo-browser 打开 TortoiseSVN 浏览我们的版本库

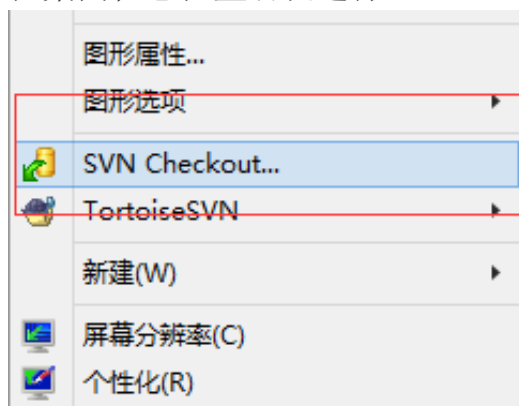


输入用户名密码

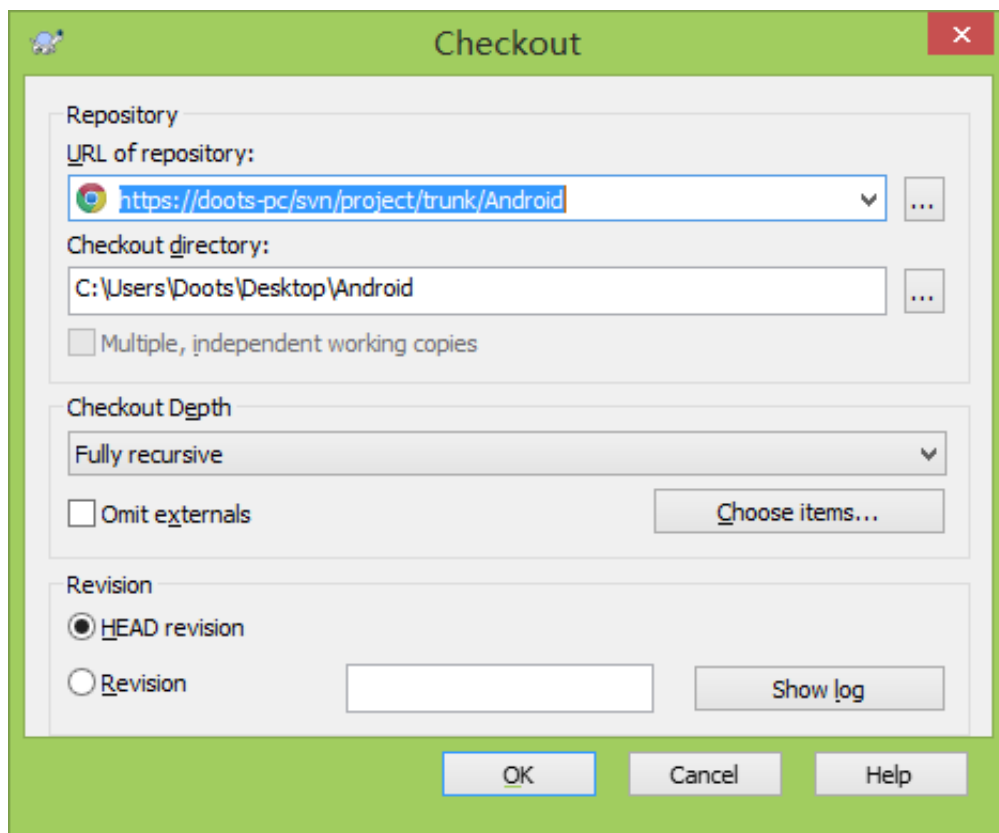




在桌面任意位置右键选择 SVN Checkout 将我们的版本库 copy 到本地



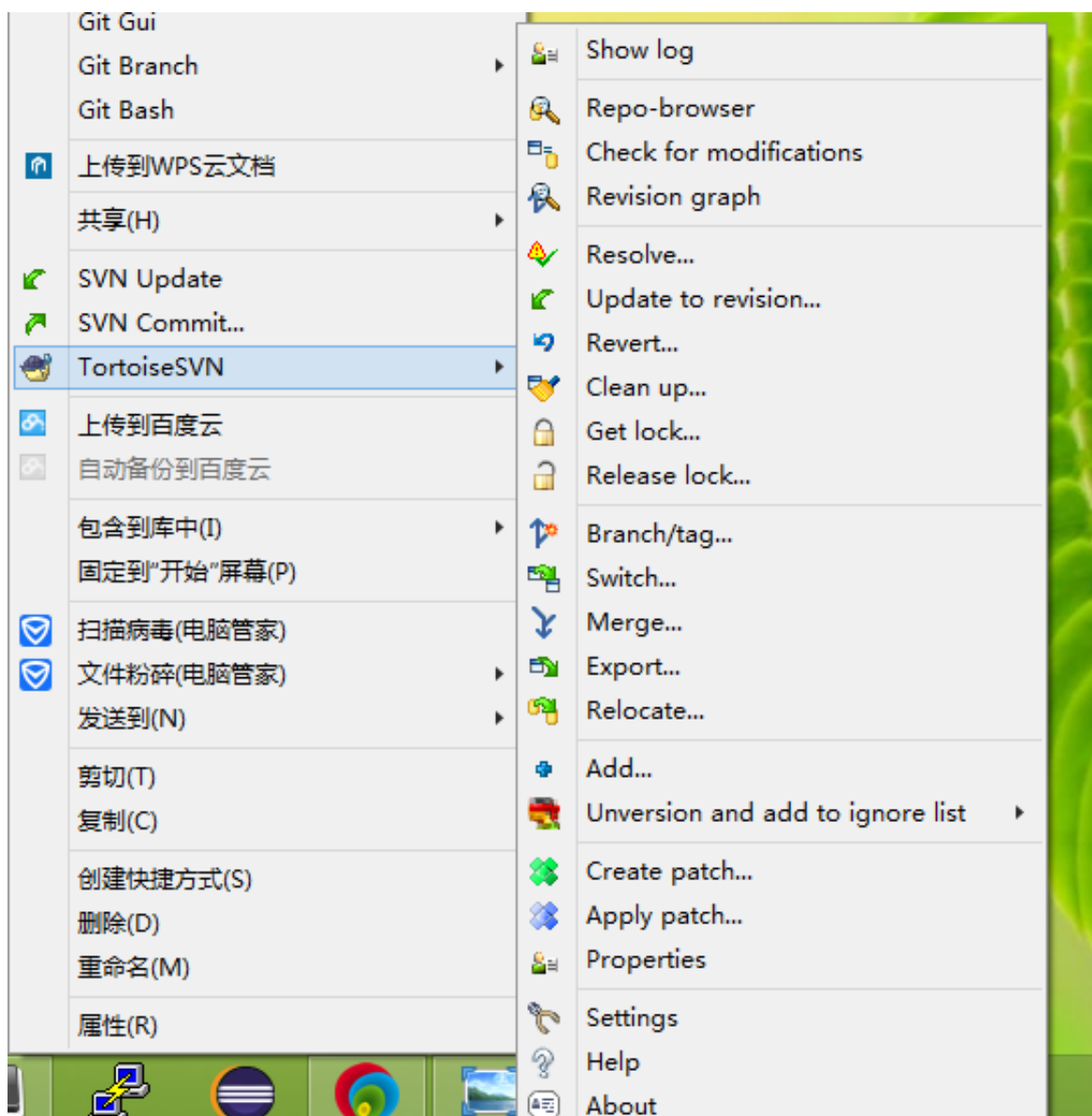
选择需要 copy 的版本库及保存到本地目录的路径。完成后，远程仓库里的文件就全部 copy 到了本地目录并且生成名为 .svn 的隐藏目录。



```
$ ls -al
total 9
drwxr-xr-x  4 Doots  Administ  4096 Jul 13 08:38 .
dr-xr-xr-x  1 Doots  Administ  8192 Jul 13 09:12 ..
drwxr-xr-x  7 Doots  Administ  4096 Jul 13 08:39 .svn
-rw-r--r--  1 Doots          22 Jul 13 08:38 README.md
```

功能介绍

在本地的版本库文件夹下右键选择 TortoiseSVN



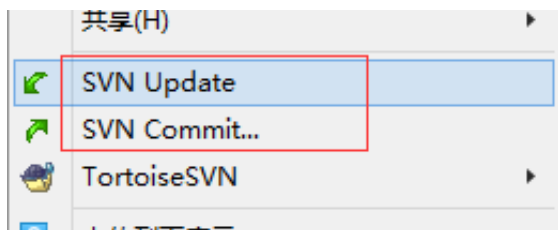
Get Lock 锁

某个开发人员对一个文件使用 svn 工具进行 get lock 操作后，其它人只有等这个人 release lock 之后才能进行编辑提交。但在某些特殊情况下，假使这位开发人员不在，就需要对已经锁定的文件进行强制解锁了。



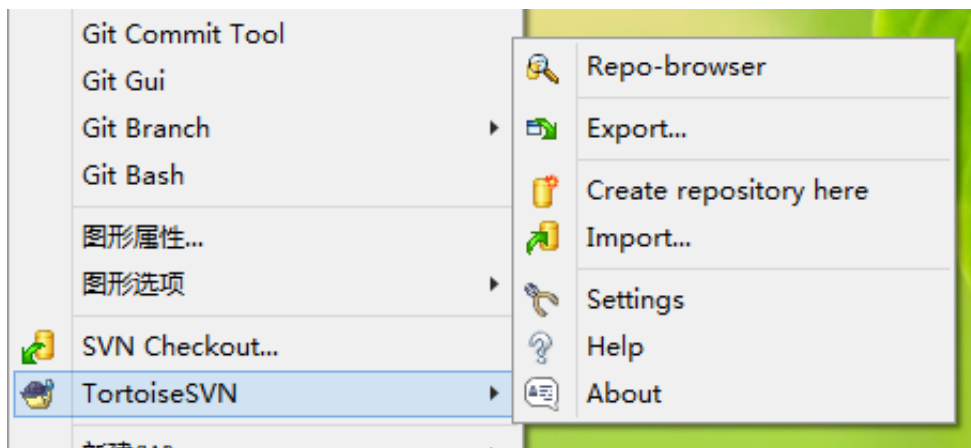
更新与提交

在本地仓库文件夹下右键，可以看到 SVN Update 和 SVN Commit，分别用于从服务器拉取最新版本，和将本地的版本提交到服务器。



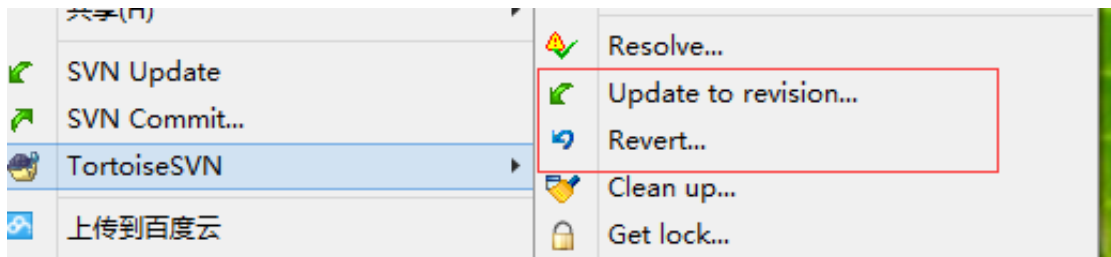
Checkout 与 Exprot

Checkout 与 Checkin 是一组，Exprot 与 Improt 是一组。Checkout 与 Exprot 都能从服务器上拉取最新的版本。它们的区别在于，Checkout 从服务器拉取的后续可以进行版本控制，而 Exprot 从拉取下来的文件是独立版本后续不受版本控制，也就是没有 .svn 目录。



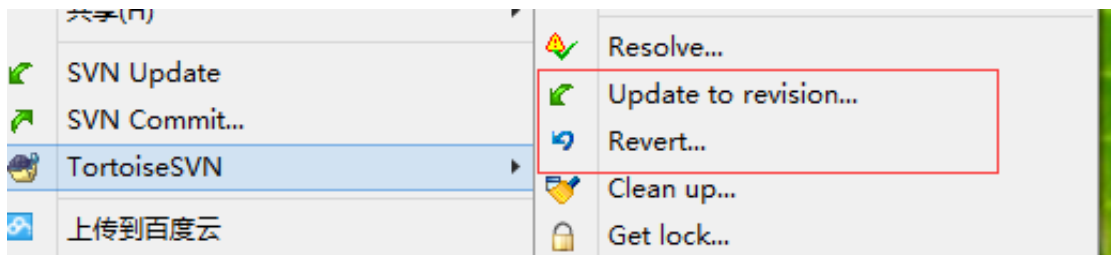
Update to revision 与 Revest

Update to revision 与 Revest 都可以让我们回到以前的版本。它们的区别在于 Update to revision 回滚后的版本不能提直接提交到服务器，而用 Revest 回滚后可以直接提交到服务器。

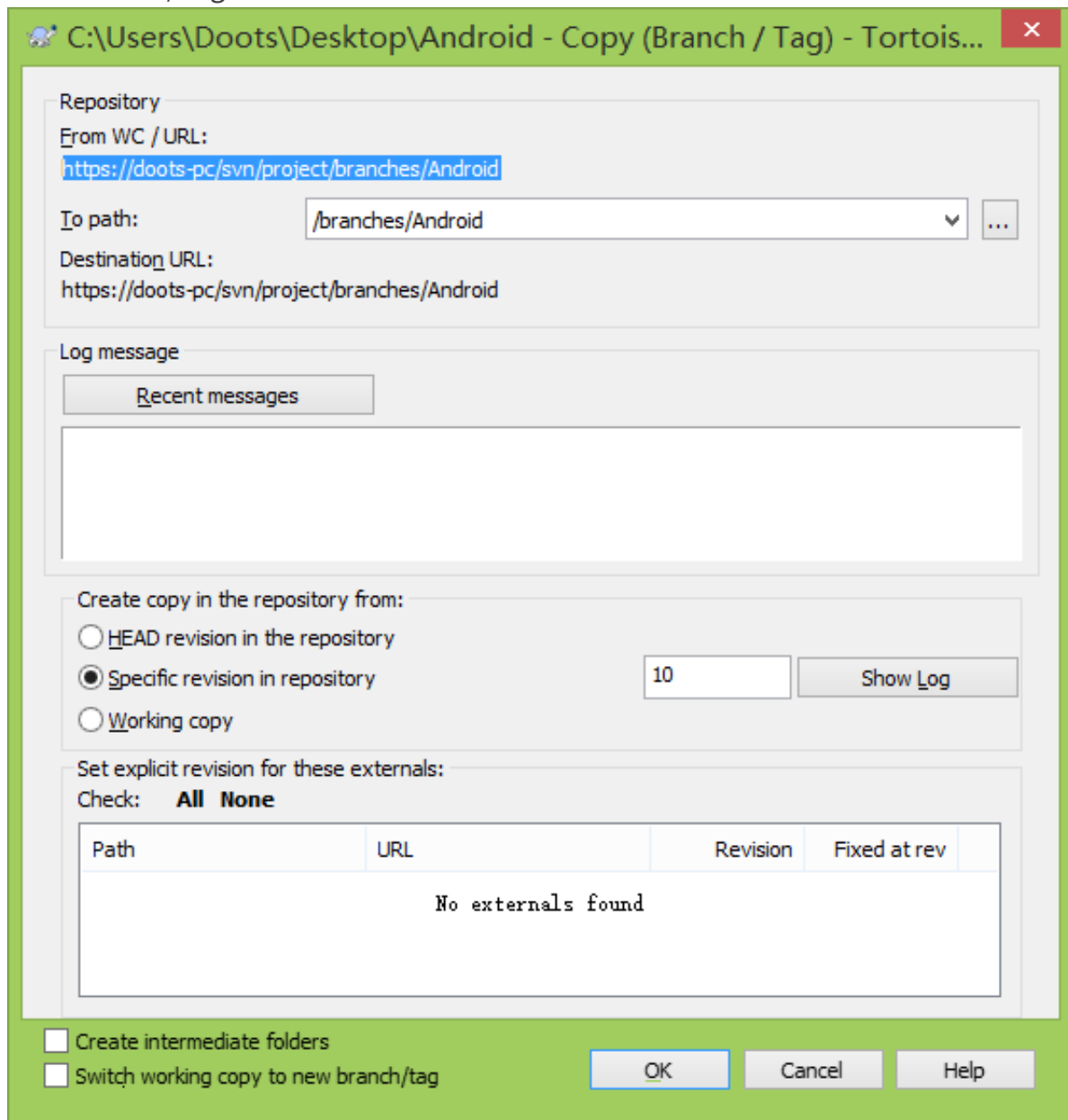


分支

分支的主要操作

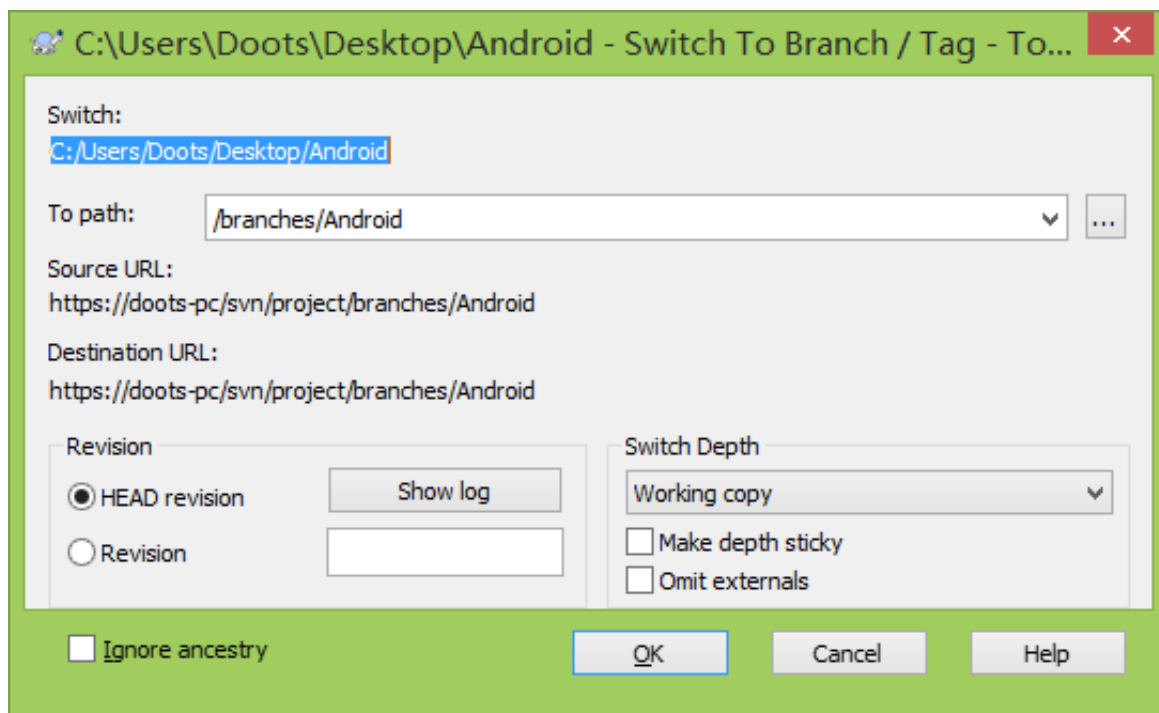


选择 Branch/Tag 新建分支

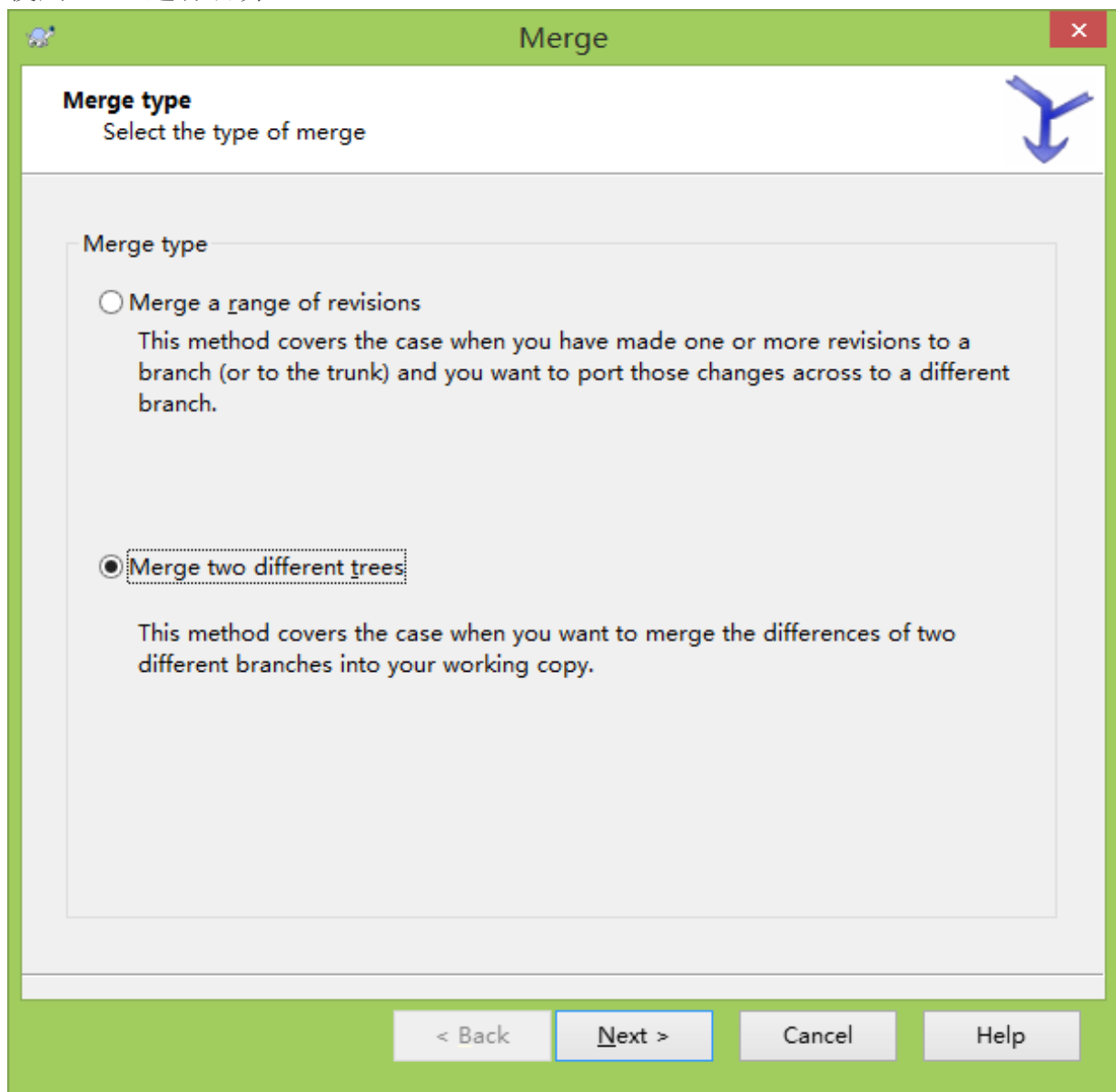


From WC/ URL: 来源目录，To Path 分支目录。

使用 Switch 可以切换分支

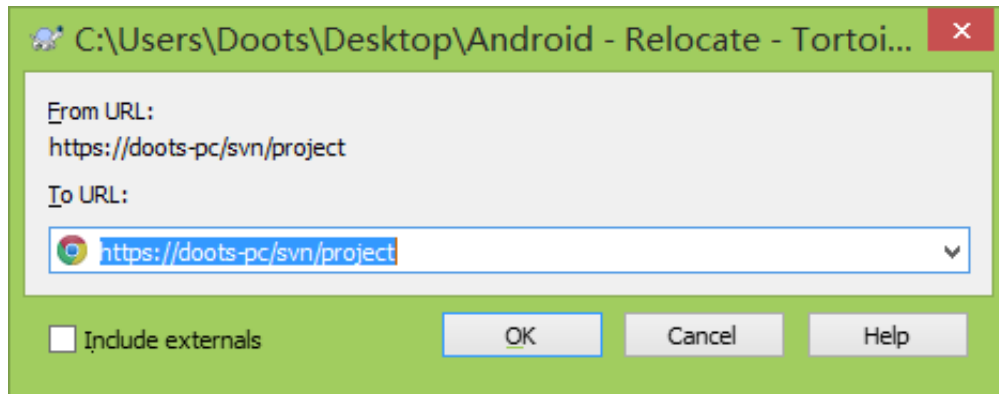


使用 Mere 进行合并

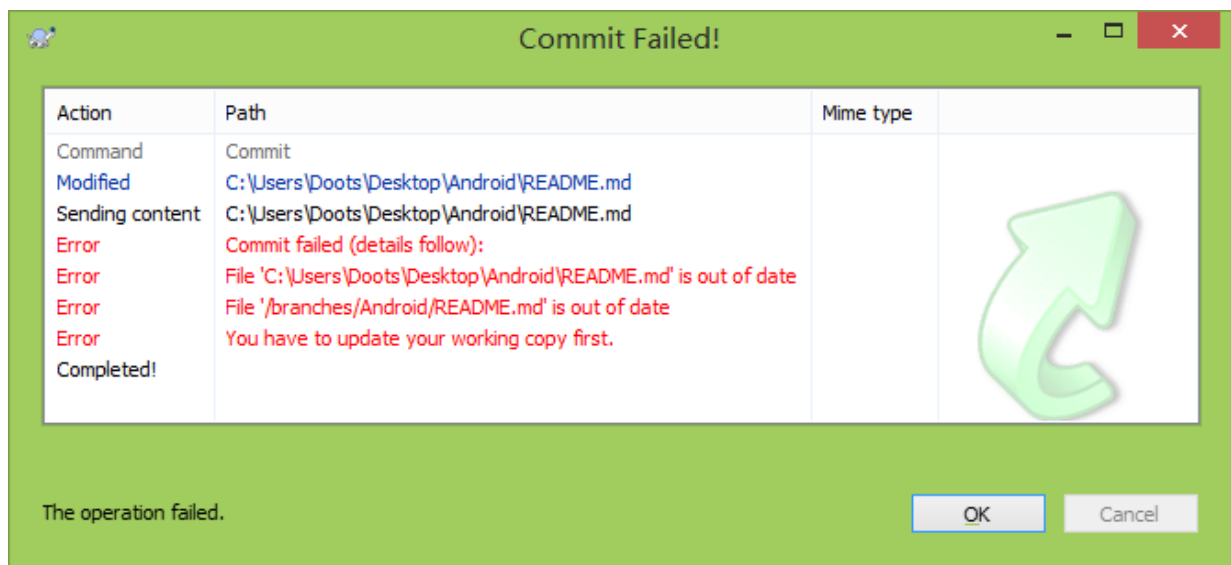


这里我们可以进行两个不同的版本合并，也可以进行两个不同的分支进行合并。

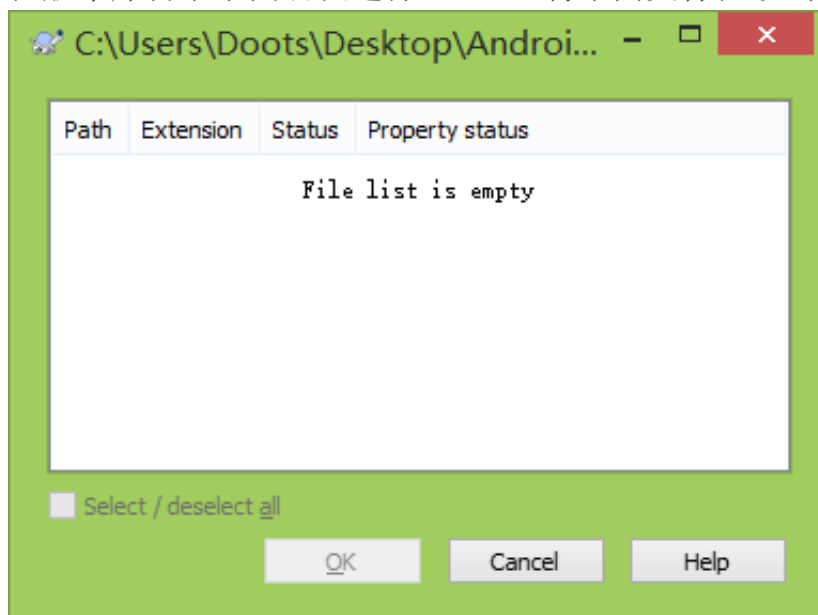
使用 Relocate 可以重定向版本库在服务器的路径。



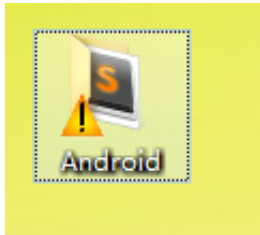
提交与合并冲突解决



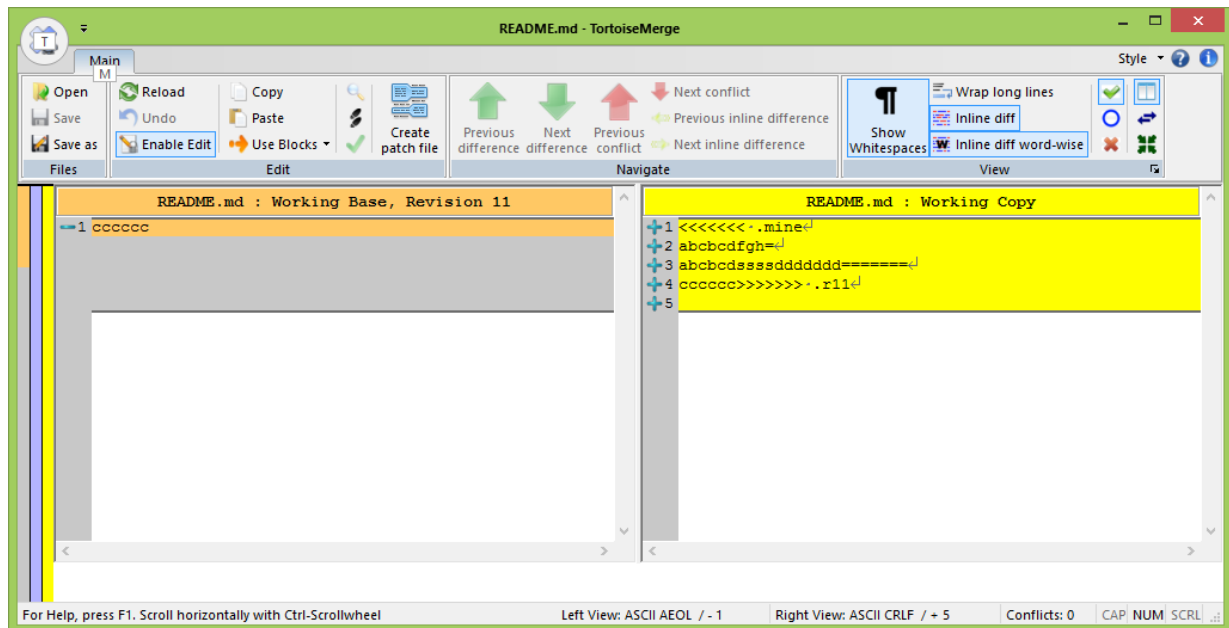
在版本库目录下面右键选择 Resolve 将不同文件拉取到本地。



在有冲突的文件目录下有一个感叹号表示冲突。



在有冲突的具体文件下右键选择 **diff** 查看冲突



最后进行修改并提交。

其它参考资料

Eclipse 插件

- **Links for 1.8.x Release:**

Eclipse update site URL

http://subclipse.tigris.org/update_1.8.x

svn插件包下载:

<http://subclipse.tigris.org/servlets/ProjectDocumentList?folderID=2240>

- **Links for 1.6.x Release:**

Eclipse update site URL

http://subclipse.tigris.org/update_1.6.x

svn插件包下载

<http://subclipse.tigris.org/servlets/ProjectDocumentList?folderID=2240>

- **Links for 1.4.x Release:**

Eclipse update site URL

http://subclipse.tigris.org/update_1.4.x

svn插件包下载:

<http://subclipse.tigris.org/servlets/ProjectDocumentList?folderID=2240>
