# 常见版本控制软件

* ***CVS***
* ***SVN***
* ***ClearCase***
* ***VSS***
* ***GIT***

# 环境搭建

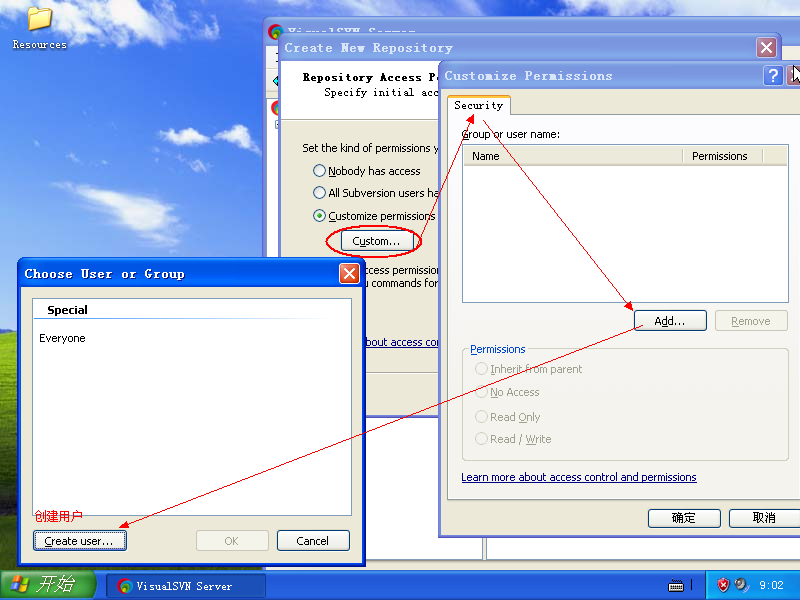
* 安装Oracle VM VirtualBox，虚拟机软件
* 在里面装两个虚拟系统，相当于两台电脑，一个是svn服务器所在的电脑，一个是开发者的电脑
* 虚拟系统搭建完毕后，在服务器端的电脑上，安装VisualSVN-Server-2.7.3.msi，安装步骤ppt中有
* 安装完VisualSVN-Server-2.7.3.msi之后，为服务器端的电脑，设置ip地址(网络连接中设置), 然后:https://ip就可以访问这个服务器了
* 这里，我们把服务器端电脑的ip设置为：192.168.1.250
* 创建仓库，放代码用



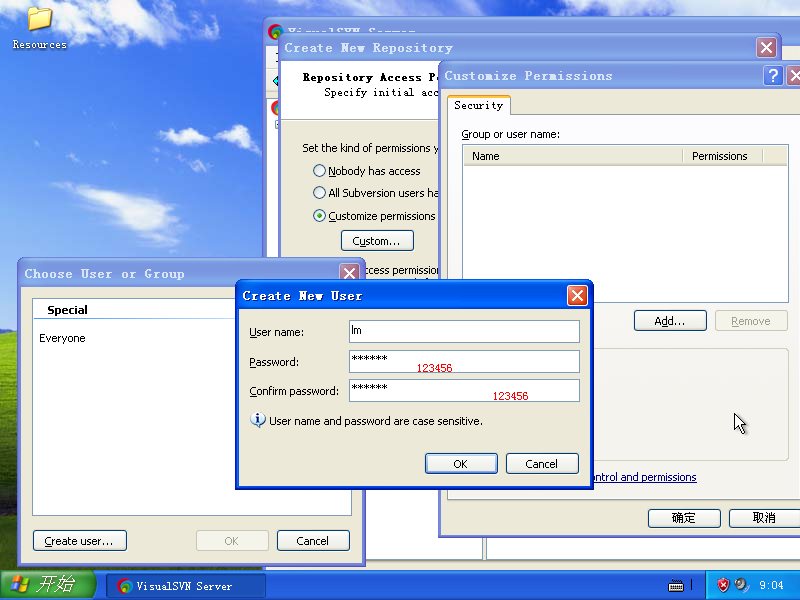


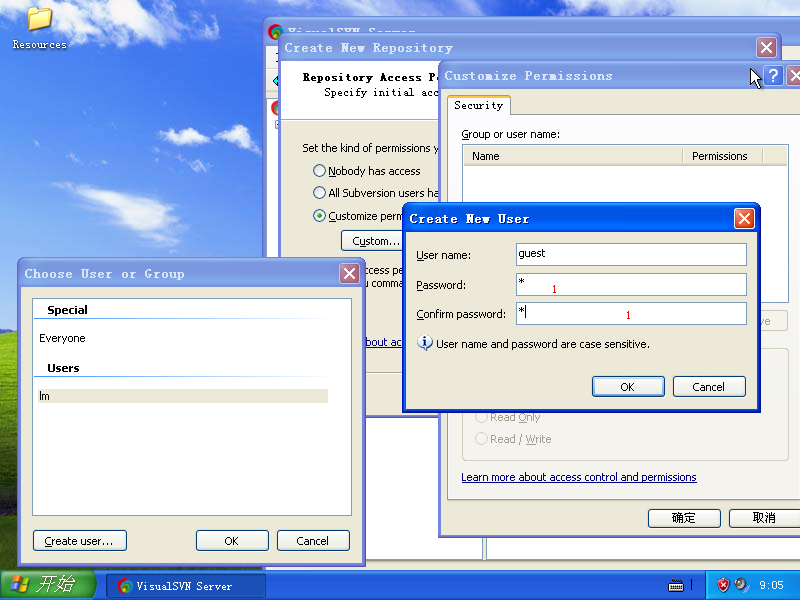


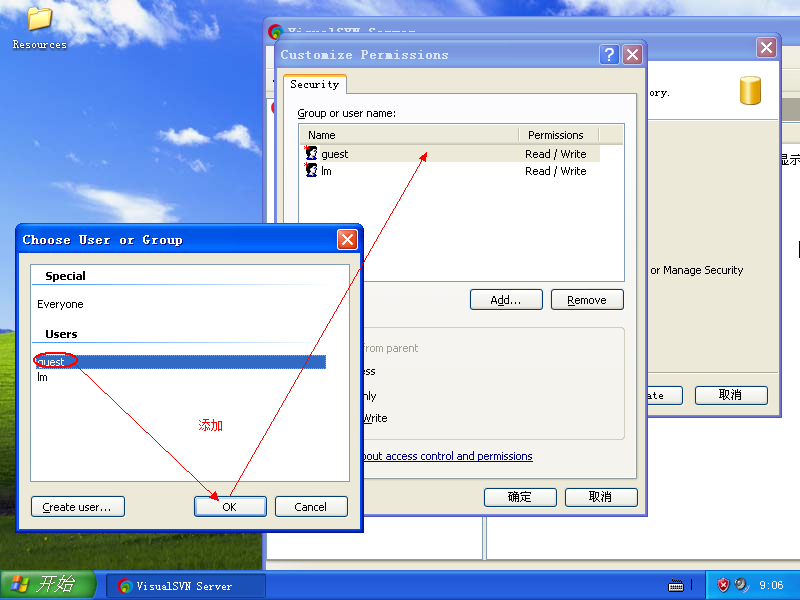




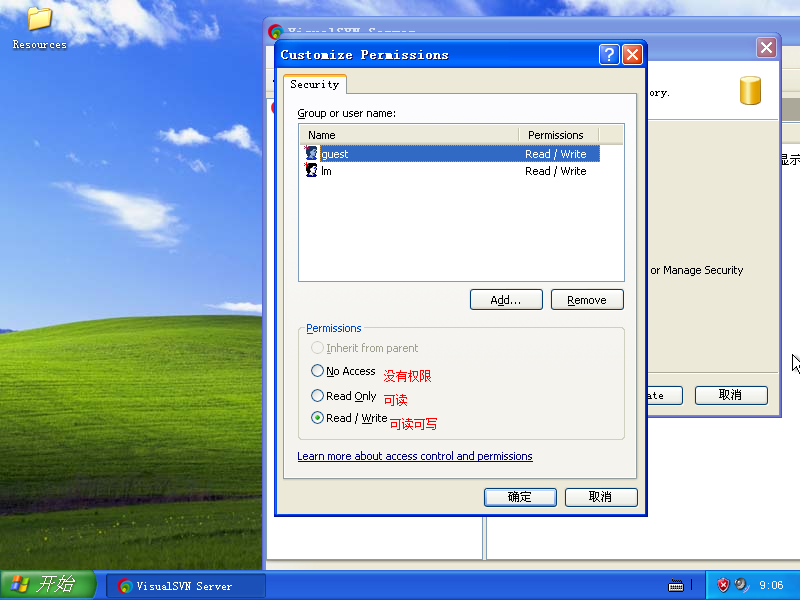
创建两个用户，一个是管理人员的，一个是来宾人员的



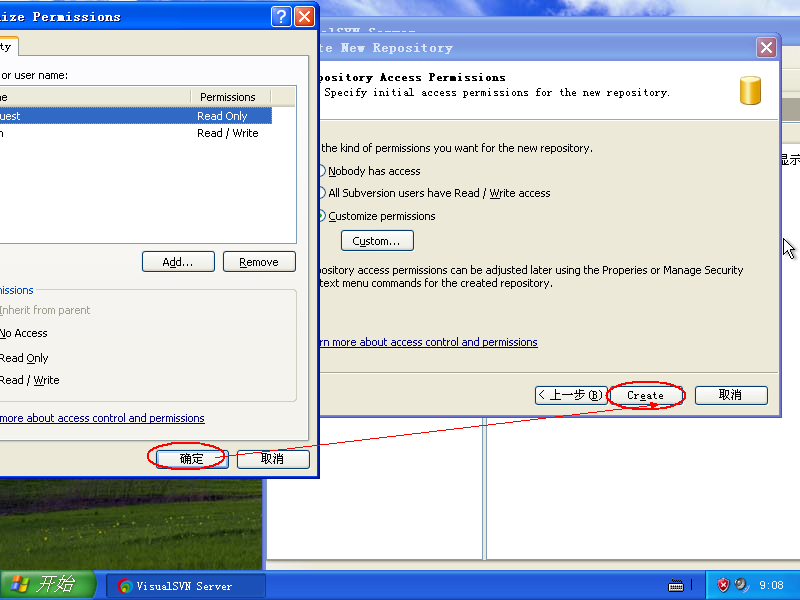


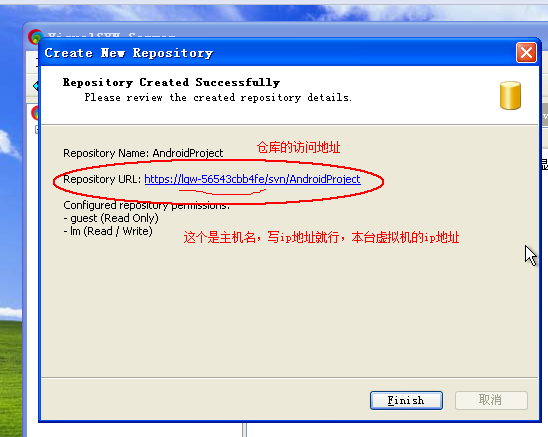


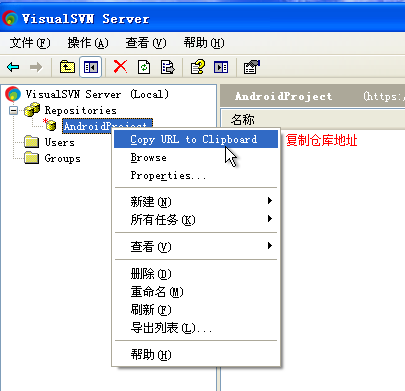
有了两个用户之后，给两个用户分配权限，管理者可读可写，来宾可读



自定义权限添加好后，创建仓库







在浏览器打开，就可以访问这个服务器的仓库了。

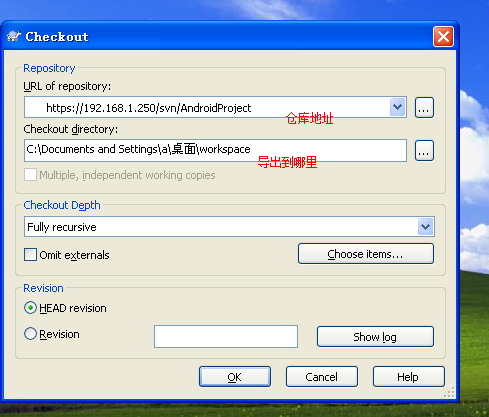
也可以使用客户端访问这个服务器的仓库

# 使用客户端访问svn服务器的仓库

* 有了客户端，就不用再命令行写命令来操作服务器的仓库了
* **TortoiseSvn**
* 在开发中虚拟系统中，安装这个客户端（步骤ppt有）
* 在xp中安装的时候，可能会出现错误，只要安装上补丁就行了WindowsXP-KB942288-v3-x86
* 为开发者虚拟系统，设置ip为：192.168.1.251
* 安装完客户端之后，桌面右键
* 

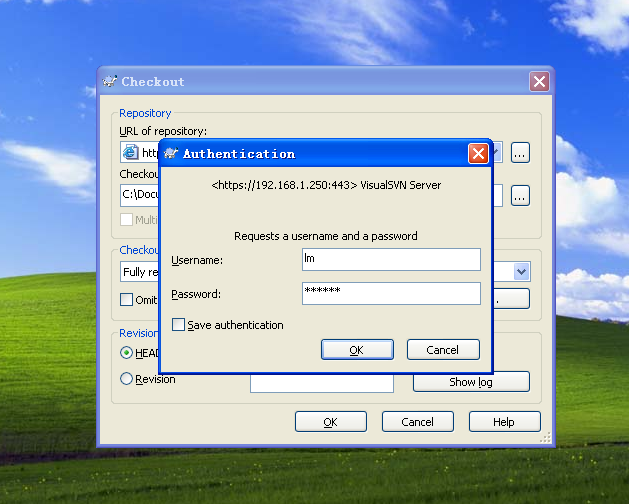
# 单用户的代码控制

* 桌面右键，svn的检出(svn Checkout)，把仓库的文件导出

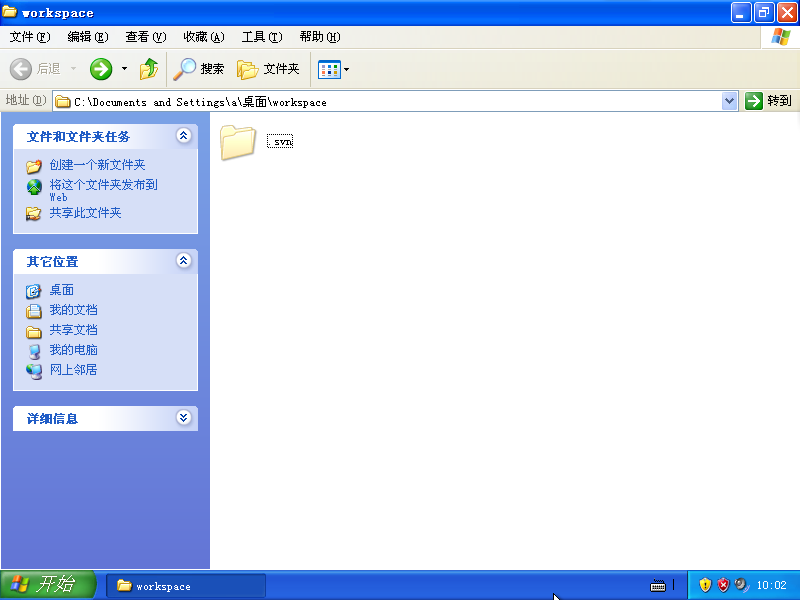




* 输入用户名/密码

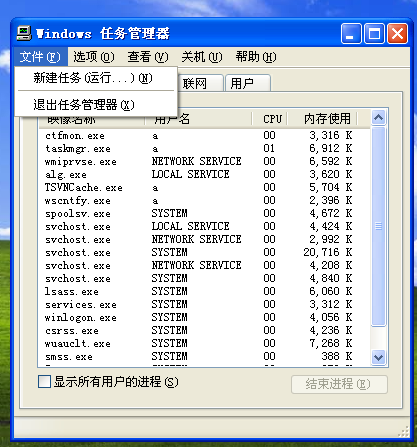


* 检出后，我们导出到的那个文件夹里，会有一个”.svn”的隐藏文件夹，这是配置信息，不要动他

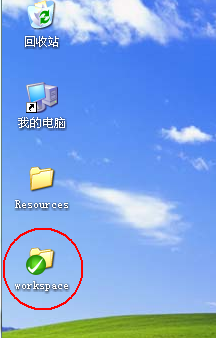


* 检出到的文件夹，会有一个绿勾。如果没有（因为没刷新），就结束一下窗体管理器进程，然后再新建一个窗体管理器进程





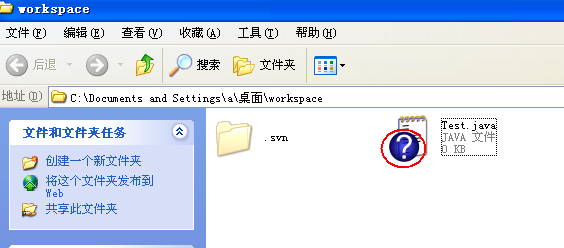




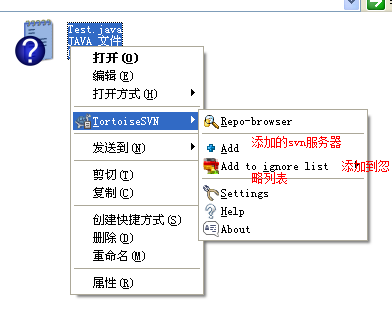
* 绿勾表示，一个新检出的工作复本使用绿色的对勾重载。表示svn状态正常。
* 如果里面的.svn文件夹消失。绿勾就会消失。
* 我们把它删除看看（会刷新不及时，把workspace文件夹拖到其他文件夹中，在拖回来就行了）



* 在wrokspace文件夹中，创建文件，有一个问号。代表版本控制服务器（svn服务器）不知道这个文件的信息，版本控制服务器不知道这个文件的信息，因为其没有被纳入到版本管理系统，因为是新建出来的。此时该文件可以被添加到版本管理或添加到ignore(忽略)列表中。



* 把它添加到版本控制服务器里，这时候版本控制服务器就会有它的记录了

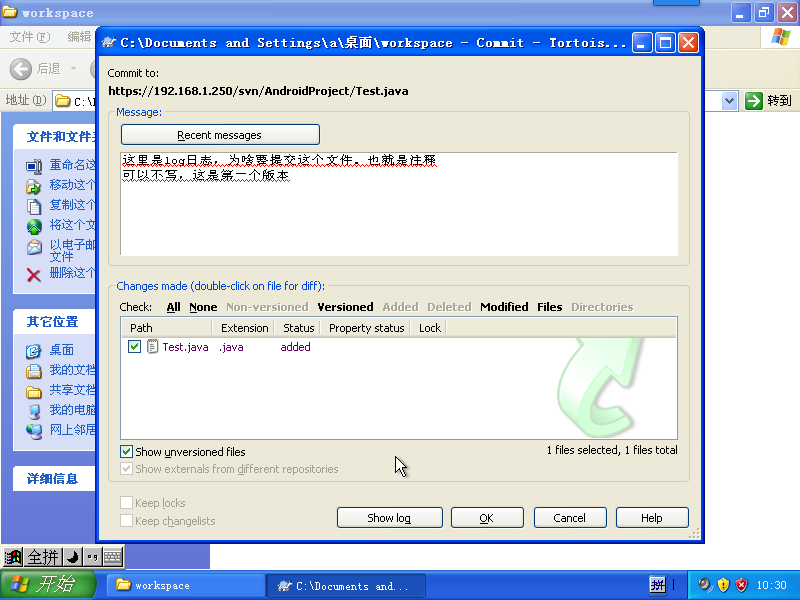


* 这时文件就变成一个”+”号了，表示：加号告诉你有一个文件或是目录已经被计划加入版本控制。
* 计划加入版本控制：下次提交的时候，会把这个文件提交到服务器，因为它现在还是个没有上传的文件，现在服务器仓库是没有这个文件的

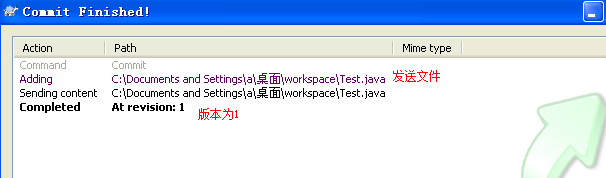


* 提交这个文件，提交之后，版本库（svn服务器仓库）里才有这个文件

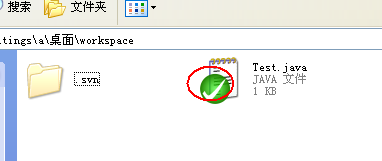




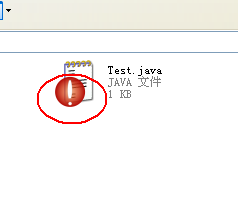
添加文件，把内容发过去，仓库版本为1



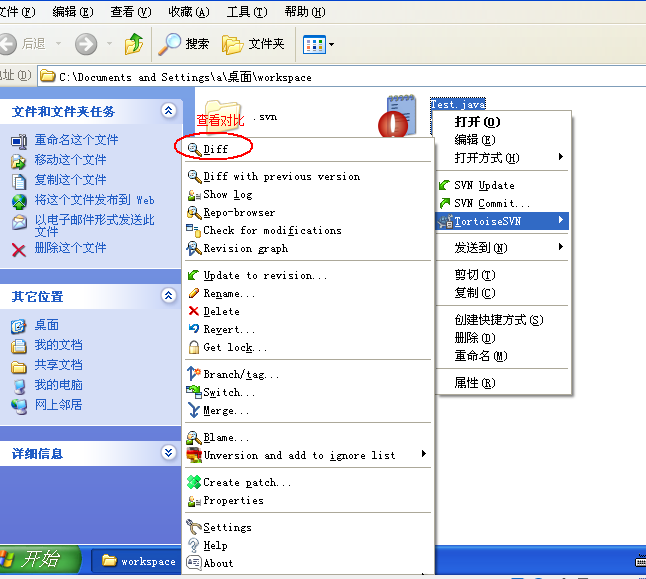
* 这时候文件变成绿勾，文件已经正常

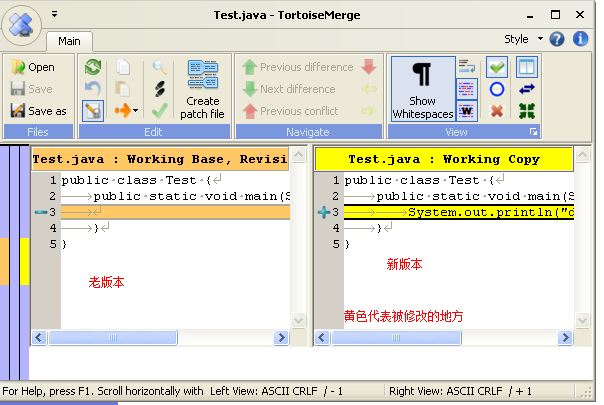


* 再次对文件进行修改，就是第二个版本，修改后文件会变成红色感叹号，表示已修改，在你开始编辑一个文件后，状态就变成了已修改，而图标重载变成了红色感叹号。通过这种方式，你可以很容易地看出哪些文件从你上次更新工作复本后被修改过，需要被提交。

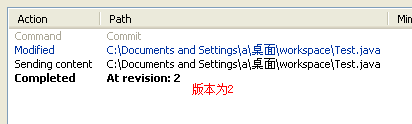


* 这时候，可以查看与之前版本的对比】

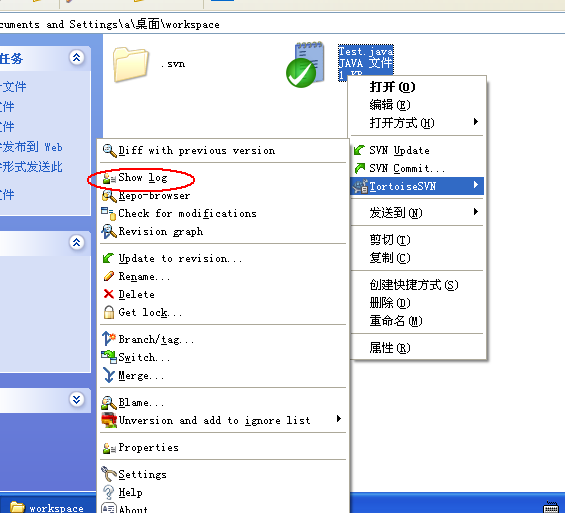


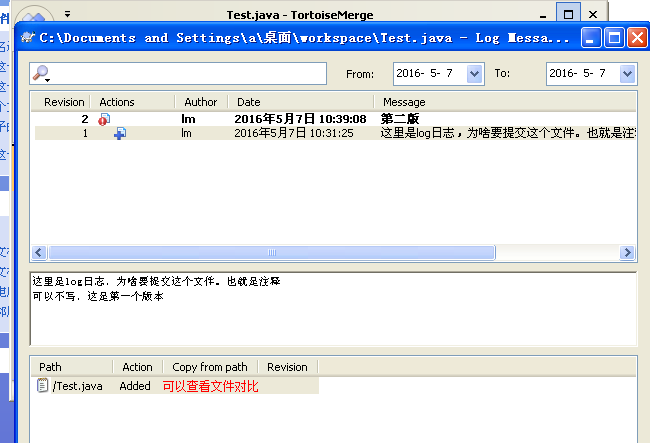


* 再次提交这个文件，修改文件，把文件发过去，仓库版本为2



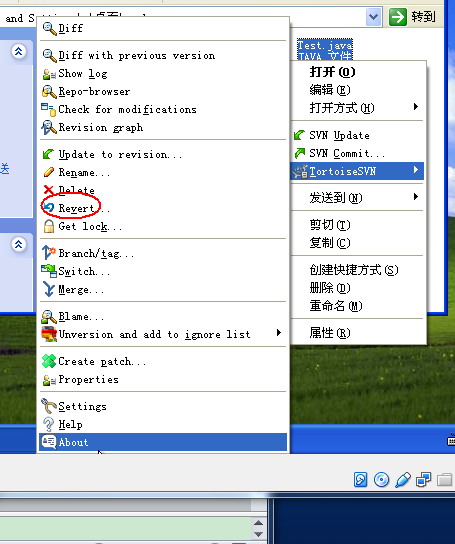
* 可以查看修改的历史记录

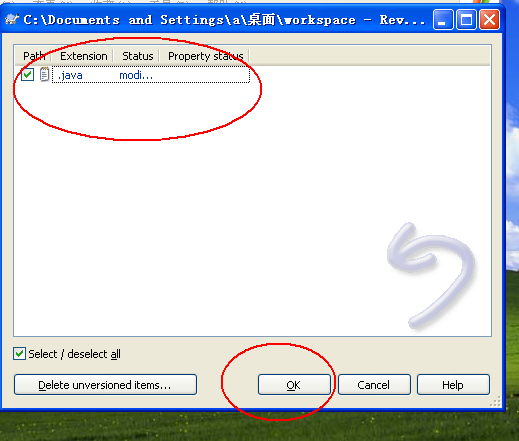




# 代码回滚

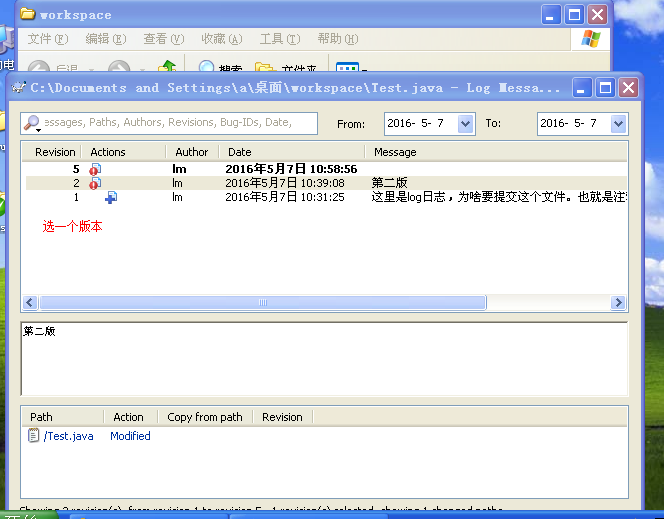
* 如果开发端的代码没了，重新检出就行了
* 代码打错了，恢复到版本原来的样子
* 先修改代码，然后文件变成了红色感叹号

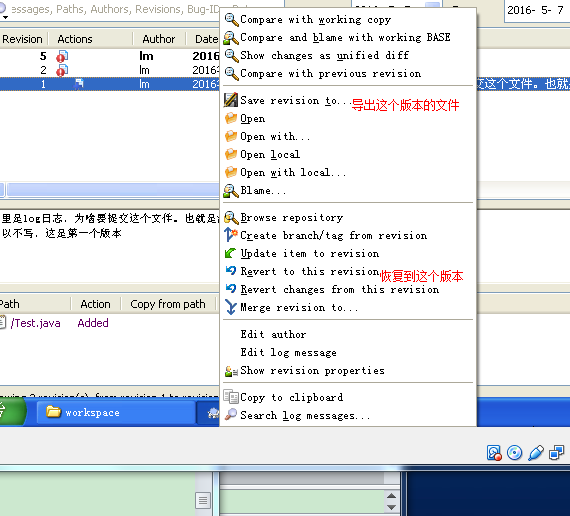




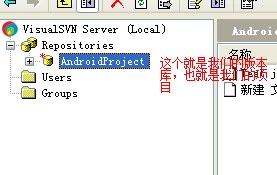
* 也可以恢复到其他版本

文件右键Show log

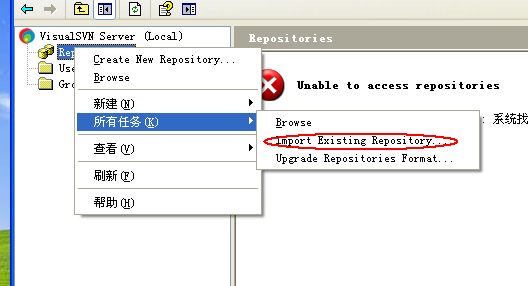


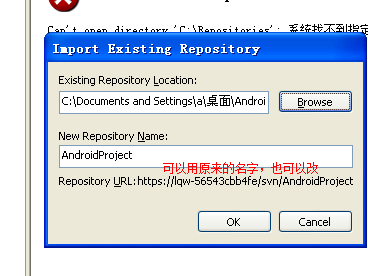


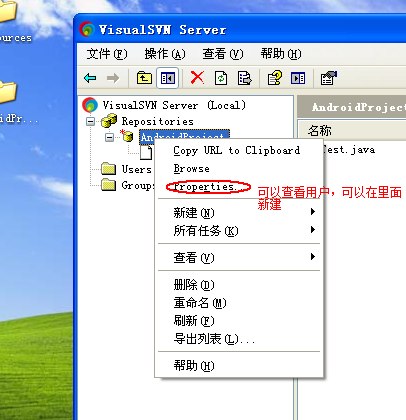
# 版本库的备份和还原



* 它存放在svn服务器所在盘的Repositories文件夹中：c:\ Repositories\AndroidProject
* 把AndroidProject备份到其他地方就可以了
* 还原：新装了一个svn服务器







# 多用户权限控制：看视频

# 多用户版本控制：看视频

# 多用户冲突合并：看视频

# Svn代码提交规范，ppt有文档

# SVN常见图标，看ppt和视频

# 常见ignore文件类型，视频

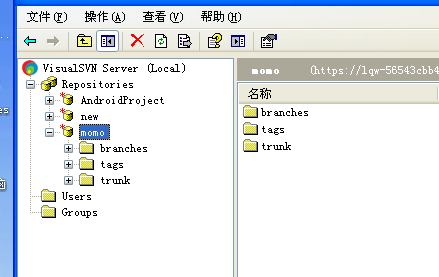
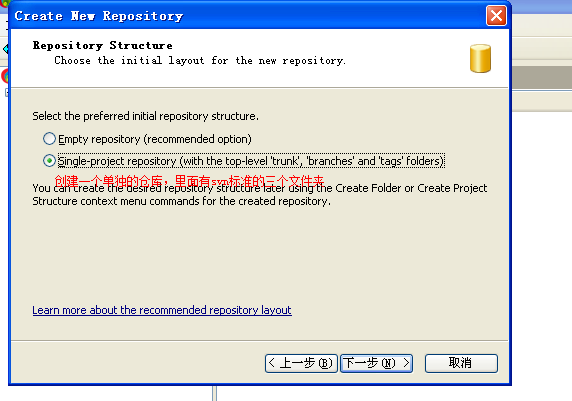
# 采用svn管理android项目，视频

# SVN标准目录结构

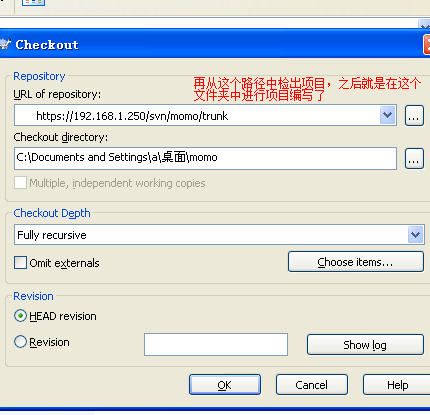
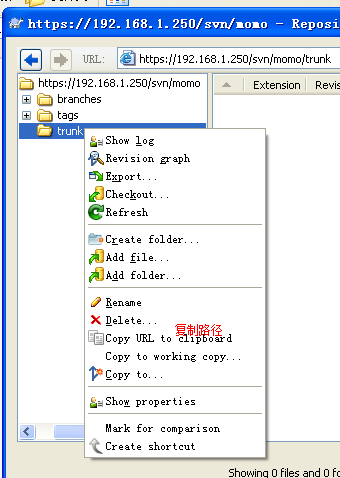
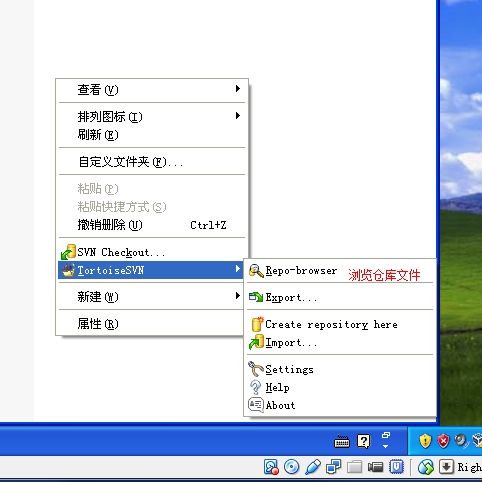
* Trunk
* 这是SVN目录的主分支（项目主版本），表示日常开发中的项目，任何时候Trunk里包含的都是最新的开发代码。 这里的代码将会工作到你的下一个主要发布版本。Trunk应该只被用来开发将会成为你的下一个重要版本的代码。在这个目录下，写代码
* Branches
  + Experimental branches：有时你想将某个新技术引进项目。这很好，但是你当然不想赌上你的整个项目。在这里写主版本的分支，新技术增加到这个版本中，对主版本没影响
  + Bug fix branches：分支也可以用于处理trunk或release branches里发现的严重的Bug。
* Tags：一般情况下，tag，是用来做一个milestone（里程碑）的，不管是不是release，都是一个可用的版本。Tags，一般是只读的
* 备份一些重要的可用版本

# 沫沫项目——版本控制实战

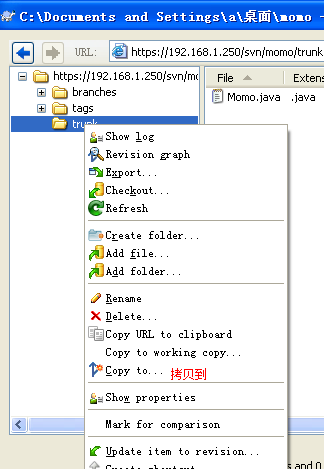
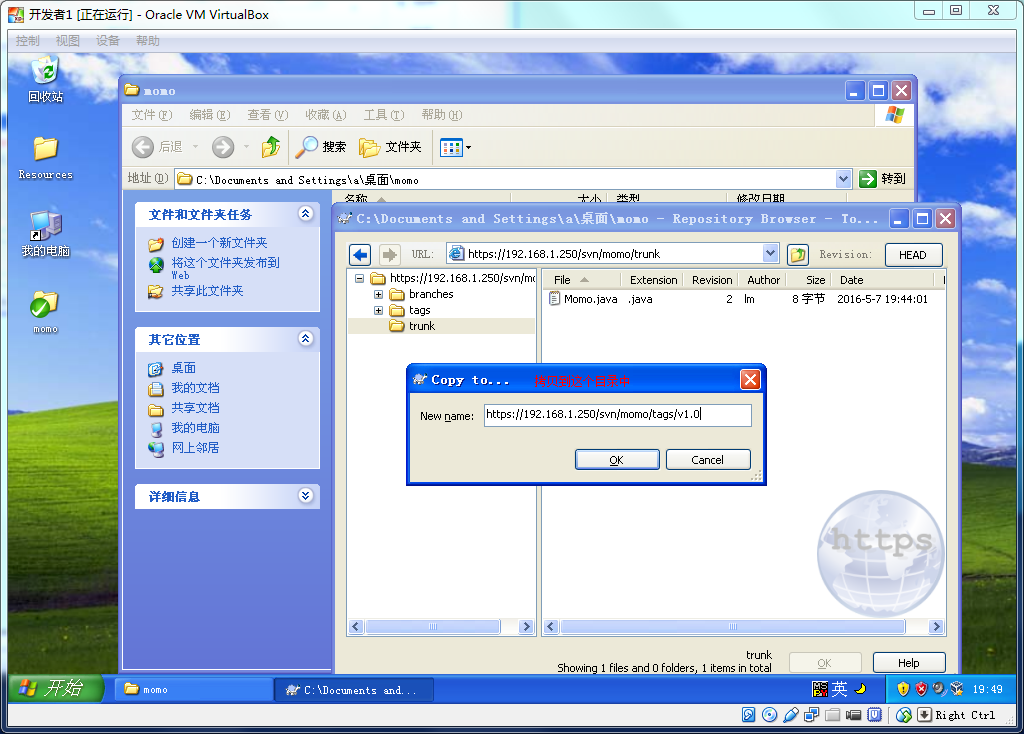
* 在svn服务器中，新建一个仓库（存放项目）



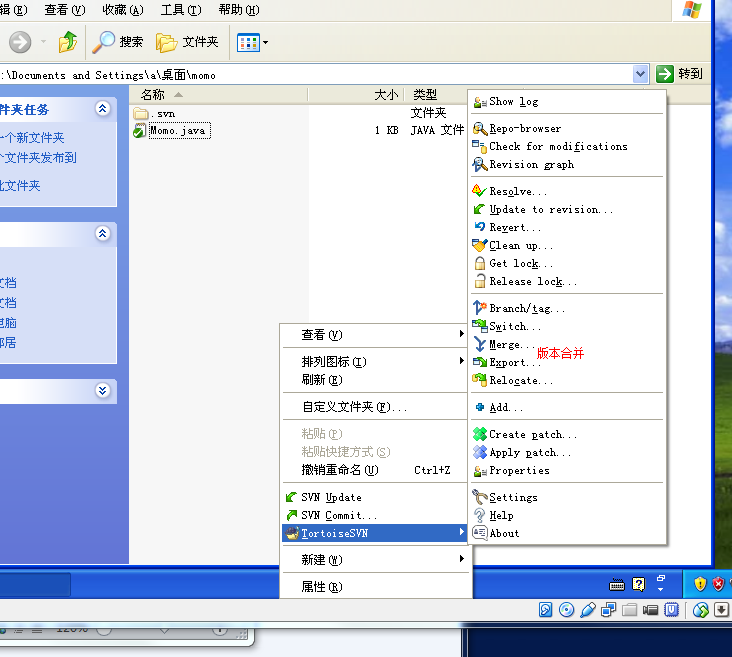
* 客户端检出项目，可以三个文件夹都检出，也可以只检出trunk文件夹（因为主要项目放在这里操作）



* 项目写完后，提交成功。再进行里程碑的备份，备份到Tags目录中
  + 浏览仓库文件

* 又增加了新功能，再次提交。备份到tags\v1.1目录中
* V1.0版本出现了bug，就在tags中把v1.0版本拷贝到branches目录中。把这个目录中的v.10项目 检出，修改bug，然后提交
* 在到v1.1版本中，把修复好的v1.0版本进行合并，变成新的v1.2版本，然后提交，备份到tags，版本合并看视频



# Git

* Git可以不联网，在本地创建一个仓库，提交也是提交到本地
* 具体看视频
* 先安装Git-1.9.5-preview20141217，再安装TortoiseGit-1.8.12.0，这是客户端。服务器端不用装。Xp中报错就先安装WindowsXP-KB942288-v3-x86
* 具体看视
* Github账号：android-lqw
* 邮箱：[957366764@qq.com](mailto:957366764@qq.com)
* 密码：2048910.86w