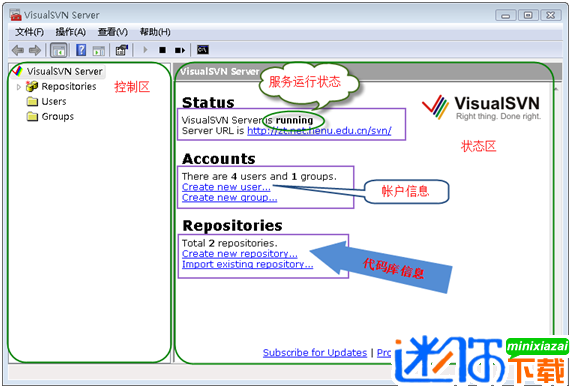
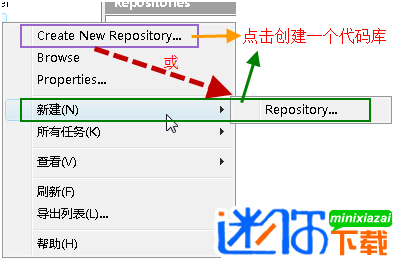


**一、VisualSVN Server的配置和使用方法【服务器端】**

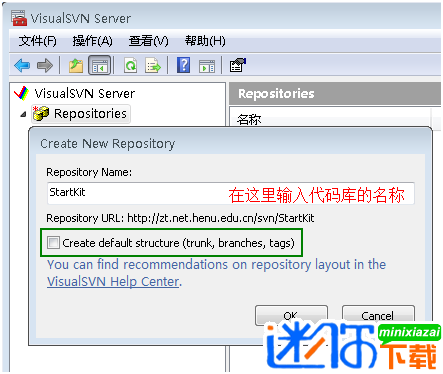
安装好VisualSVN Server后【安装过程看这里】，运行VisualSVN Server Manger，下面是启动界面：



好的，下面我来添加一个代码库【Repository】，如下图：

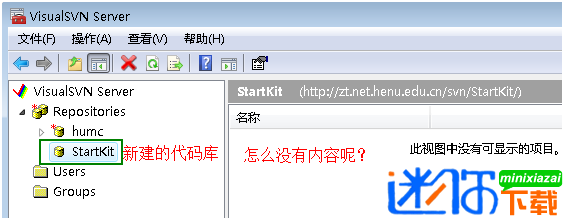


按上图所示，创建新的代码库，在下图所示的文本框中输入代码库名称：



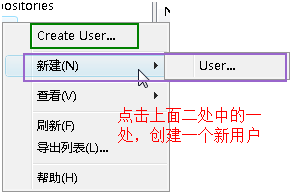
注意：上图中的CheckBox如果选中，则在代码库StartKit下面会创建trunk、branches、tags三个子目录;不选中，则只创建空的代码库StartKit。

点击OK按钮，代码库就创建成功了。



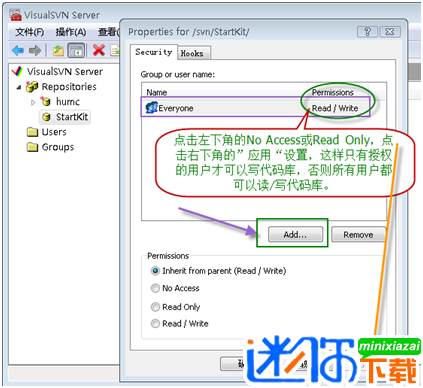
创建完代码库后，没有任何内容在里面。我会在这个教程的第二部分说明如何迁入源代码。

下面，我们开始安全性设置，在左侧的Users上点击右键：



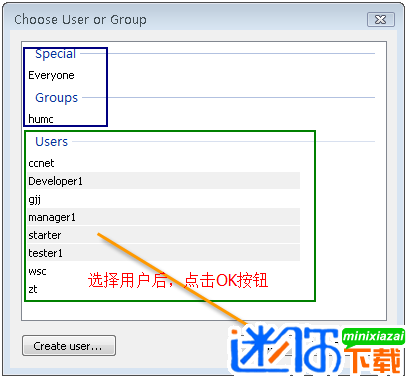


输入上面的信息，点击OK，我们就创建一个用户了。按照上面的过程，分别添加用户Developer1、tester1、manager1，好了，我们开始添加这些用户到我们刚才创建的项目里：

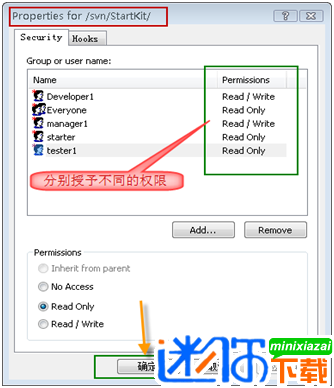


点击上图中的"Add..."按钮，在下图中选择我们刚才添加的用户,点击OK按钮：

说明：大家可能注意到了下图中的Groups，是的，你也可以先创建组，把用户添加到各个组中，然后对组进行授权，操作比较简单，在此略过。



按照下图所示，分别对用户【或组】进行授权：



点击"确定"按钮，上面的用户就具有了访问StartKit代码库的不同权限。

因为用户starter在团队中是新来者，不希望他向代码库中提交新代码，所以他只能读取代码库中的代码，不能提交代码。tester1是测试人员，不负责代码编写，所以也是只读权限。而Developer1和manager1是开发人员和项目经理，自然具有读、写的权限。

在实际的项目开发过程中，Developer和tester往往不可能只有一个人，这时候使用组来授权更加方便，这个大家可以自己练习一下。

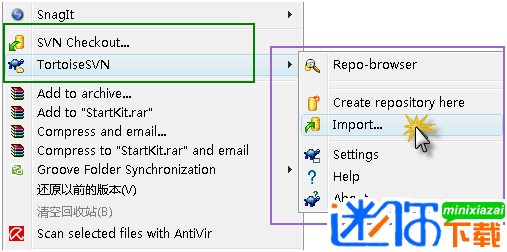
二、TotoiseSVN的基本使用方法

在 项目管理实践教程一、工欲善其事，必先利其器【Basic Tools】中，我已经讲解了怎样安装TortoiseSVN。在上面的讲解中已经讲了怎么使用VisualSVN Server了，今天我要讲的是，TortoiseSVN的简单使用方法。

一、签入源代码到SVN服务器

假如我们使用Visual Studio在文件夹StartKit中创建了一个项目，我们要把这个项目的源代码签入到SVN Server上的代码库中里，首先右键点击StartKit文件夹，这时候的右键菜单如下图所示：

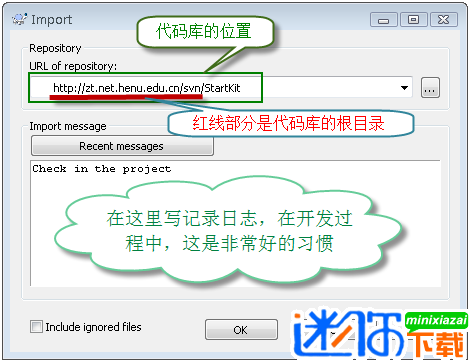
图2-2-1



点击Import，弹出下面的窗体，其中http://zt.net.henu.edu.cn 是服务器名，svn是代码仓库的根目录，StartKit是我们在上个教程中添加的一个代码库：

说明：左下角的CheckBox，在第一次签入源代码时没有用，但是，在以后你提交代码的时候是非常有用的。

图2-2-2



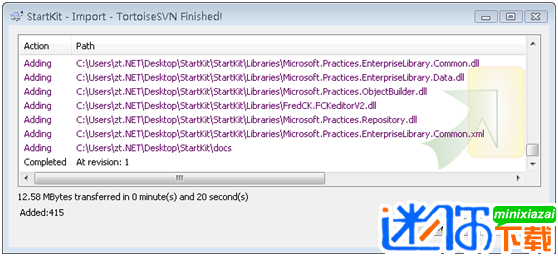
点击OK按钮，会弹出下面的窗体，要求输入凭据：

图2-2-3



在上面的窗体中输入用户名和密码，点击OK按钮：

图2-2-4



如上图所示，好了，源代码已经成功签入SVN服务器了。这时候团队成员就可以迁出SVN服务器上的源代码到自己的机器了。

二、签出源代码到本机

在本机创建文件夹StartKit，右键点击Checkout，弹出如下图的窗体：

图2-2-5



在上图中URL of Repository：下的文本框中输入svn server中的代码库的地址，其他默认，点击OK按钮,就开始签出源代码了。

说明：上图中的Checkout Depth，有4个选项，分别是迁出全部、只签出下一级子目录和文件、只签出文件、只签出空项目，默认的是第一项。上面的例子中，我们也可以使用web的方式访问代码库，在浏览器中输入http://zt.net.henu.edu.cn/svn/StartKit/

这时候也会弹出对话框，要求输入用户名和密码，通过验证后即可浏览代码库中的内容。

搞定!源代码已经成功签出到刚才新建的StartKit目录中。

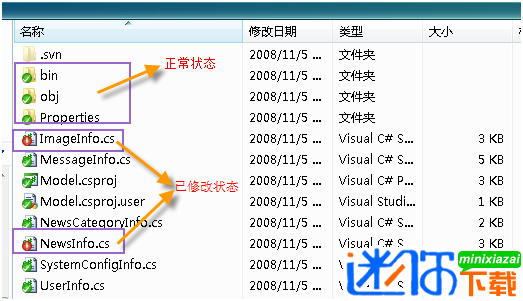
打开StartKit目录，可以看到如下图的文件夹结构：

图2-2-5



一旦你对文件或文件夹做了任何修改，那么文件或文件夹的显示图片机会发生变化。下图中我修改了其中的二个文件：

图2-2-7



大家看一下不同状态所对应的图片：

图2-2-8



我们已经知道怎么将源代码签入到SVN服务器，怎么从服务器签出代码到本机，也简单了解了不同状态所对应的图案啦。

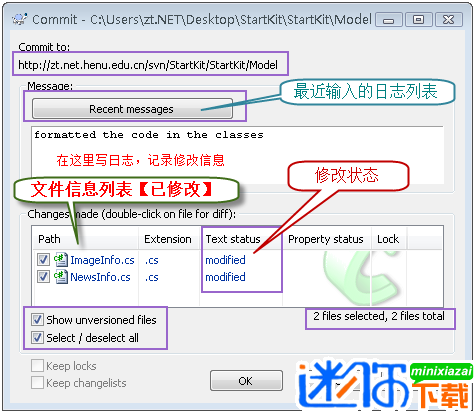
三、提交修改过的文件到SVN服务器

上面的图2-2-7中，我修改了位于Model文件中的二个文件ImageInfo.cs和NewsInfo.cs，下面演示如何提交到SVN服务器。

注意：提交源代码到服务器时，一定确保本机的代码是最新版本，否则可能提交失败，或者造成版本冲突。

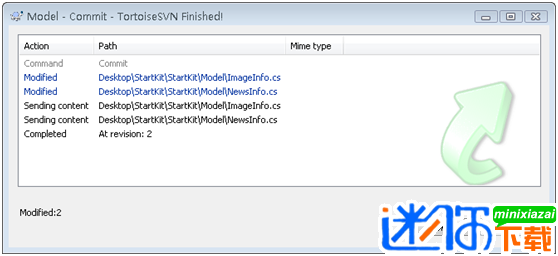
在Model文件夹上点击右键或在Model文件下的空白处点击右键，点击SVN Commit…弹出下面的窗体：

图2-2-9



点击OK按钮后，弹出如下图的窗体：

图2-2-10

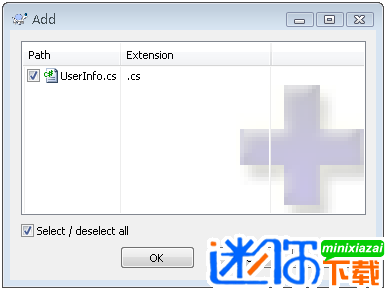


四、添加新文件到SVN服务器

我们在Model文件下添加一个新的类文件UserInfo.cs，在Model文件下的空白处点击右键，点击SVN Commit…，和上面讲的提交修改过的文件到SVN服务器一样，就可以了。

另外也可以在文件UserInfo.cs上点击右键，点击TortoiseSVN=>>Add，弹出如下图的窗体：

图2-2-11



选中UserInfo.cs文件，点击OK按钮，这样并没有将这个文件提交到SVN服务器，只是将这个文件标记为源代码库库中的文件，并将其状态置为修改状态。之后，我们要再SVN Commit这个文件一次，才可以将其真正提交到SVN服务器上的代码库中。

上面讲是添加文件，实际上，添加文件夹的步骤也是一样的，这里就不说了。

五、更新本机代码与SVN服务器上最新的版本一致

这个也很简单，只要在需要更新的文件夹上点击右键或在该文件下的空白处点击右键，点击SVN Update，就可以了。

注意：更新操作可能会因为版本冲突而失败，这是可以使用合并【Merge】或其他方法解决;也可能因为锁定【Get Lock】而失败，这是需要先解锁【Release Lock】。

六、重命名文件或文件夹，并将修改提交到SVN服务器

只要在需要重命名的文件或文件夹上点击右键，点击TortiseSVN=>>Rename…，在弹出的窗体中输入新名称，点击OK按钮，就可以了。此方法也不是直接重命名，而是将该文件或文件夹的名称标记为重命名后名称，也需要我们使用SVN Commit提交到SVN服务器后才真正重命名。

七、删除文件或文件夹，并将修改提交到SVN服务器

最简单就是，你直接删除文件或文件夹，然后使用SVN Commit提交更新到SVN服务器。另外一种方法是在你要删除的文件或文件夹上点击右键=>>TortoiseSVN=>>Delete删除,此方法也不是直接删除，而是将该文件或文件夹的状态置为删除，也需要我们使用SVN Commit提交到SVN服务器后才真正删除。

说明：实际上，从你把源代码迁签入SVN服务器开始，每一个版本的数据和文件，就算是你已经删除了的，也都可以随时迁出。

以上只是TortoiseSVN最简单的几个功能，其实他的功能远不止这些，其他的功能大家可以在使用的过程中慢慢体会，有些功能我会在下面的教程中使用到，到时候会和大家讲清楚用法。

注意：向SVN服务器提交源代码的时候，一定不要提交bin、obj等文件夹，否则会很麻烦。但是web项目的bin目录除外，但是web项目的bin目录中的引用其他项目而生成的dll不需要提交。

一个好习惯：如果项目中引用了其他的第三方的程序集，比如EnterpriseLibrary、FCKEditor等，这时候不要简单从他们的安装位置引用，而是在你的解决方案下，添加一个Library的目录，把需要的程序集复制到这里，然后从Library目录引用，这样有什么好处，自己想一想吧!

Tortoise SVN 客户端 基本用法

1. export 和check out

export 下载源代码

用法：

1、新建一个空的文件夹，右键点击它，可以看到TortoiseSVN菜单以及上面的SVN Checkout。

2、不用管这个Checkout，我们选择TortoiseSVN菜单下的Export...，接着它会让你输入url。

3、比如输入【迷宫探宝】的SVN地址是：http://game-rts-framework.googlecode.com/svn/trunk/

4、其他选项不需要更改，Omit externals不要勾选，HEAD Revision选中表示最新的代码版本，接着点击OK即可将代码导出到这个目录中：)

check out 意思签出，虽然和Export的效果一样是把代码从服务器下载到本地，但是Checkout有验证的功能，Checkout到某处的代码，将会被TortoiseSVN监视，里面的文件可以享受各种SVN的服务。

2 .每次提交代码需要注意哪些问题

如果你更新了目录中的文件，提交代码需要用到commit功能，commit的功能不仅仅是上传，他会和服务器上面的文件进行对比，假如你更新了某个文件而服务器上面也有人更新了这个文件，并且是在你checkout之后做的更新，那么它会尝试将你的更新和他人的更新进行融合(merge)，假如自动merge不成功，那么报告conflict，你必须自己来手动merge，也就是把你的更新和别人的更新无冲突的写在一起。

commit的时候，最好填写Log信息，这样保证别人可以看到你的更新究竟做了写什么。这就相当于上传文件并且说明自己做了那些修改，多人合作的时候log非常重要。

TortoiseSVN的commit只会上传原先checkout然后又被修改了的文件，假如你新加入了某些文件，需要右键点击文件选择Add，然后文件上面会出现一个加号，在下次commit的时候它就会被upload并且被标记为绿色对勾。没有绿色对勾的文件不会被commit。

假如你需要给带有绿色对勾文件改名或者移动它的位置，请不要使用windows的功能，右键点击它们，TortoiseSVN都有相应的操作。想象这些文件已经不在是你本地的东西，你的一举一动都必须让Tortoise知道。

假如修改了某个文件但是你后悔了，可以右键点击它选择Revert，它将变回上次checkout时候的情况。或者Revert整个工程到任意一个从前的版本.

下面描述在使用Commit时的几个注意点：

-------------如有多个文件需要同时提交，同时文件在不同的目录下，必须找到这些文件的最短目录上点击Commit，TortoiseSVN会搜索被点击目录以及该目录下所有的文件，并将修改变动的文件罗列在列表中。

-------------仔细查看列表中的文件，确定哪些文件时需要更新的，如果不需要更新某个已经变化了的文件，只需要在该文件上点击右键，选择还原操作;选择需要新增的文件，不要将临时文件添加到版本库中。

-------------如遇到文件冲突(冲突：要提交的文件已被其他人改动并提交到版本库中)要启用解决冲突功能。

3. 如何保持本地版本和服务器版本同步

使用update来同步本地和服务器上的代码。同样是右键选择SVN update，所有的更改就会从服务器端传到你的硬盘。注意，假如别人删除了某个文件，那么更新之后你在本地的也会被删除。

如果本地的代码已经被修改，和commit一样会先进行merge，不成功的话就会报告conflict

4 如何在同一个在一个工程的各个分支或者主干之间切换

使用tortoise SVN-->switch

在URL中输入branch或trunk的url地址

5.如何比较两个版本之间的差别

本地更改

如果你想看到你的本地副本有哪些更加，只用在资源管理器中右键菜单下选TortoiseSVN→ 比较差异。

与另外一个分支/标签之间的差异

如果你想查看主干程序(假如你在分支上开发)有哪些修改或者是某一分支(假如你在主干上开发)有哪些修改，你可以使用右键菜单。在你点击文件的同时按住Shift键，然后选择TortoiseSVN→ URL比较。在弹出的对话框中，将特别显示将与你本地版本做比较的版本的URL地址。

你还可以使用版本库浏览器，选择两个目录树比较，也许是两个标记，或者是分支/标记和最新版本。邮件菜单允许你使用比较版本来比较它们。阅读第 5.9.2 节 “比较文件夹”以便获得更多信息。

与历史版本的比较差异

如果你想查看某一特定版本与本地拷贝之间的差异，使用显示日志对话框，选择要比较的版本，然后选择在右键菜单中选与本地拷贝比较差异

两个历史版本的比较

如果你要查看任意已提交的两个历史版本之间的差异，在版本日志对话框中选择你要比较的两个版本(一般使用 Ctrl-更改)，然后在右键菜单中选比较版本差异

如果你在文件夹的版本日志中这样做，就会出现一个比较版本对话框，显示此文件夹的文件修改列表。阅读第 5.9.2 节 “比较文件夹”以便获得更多信息。

提交所有修改

如果你要在一个视窗中查看某一版本的所有更改，你可以使用统一显示所有比较 (GNU 片段整理)。它将显示所有修改中的部分内容。它很难显示一个全面清晰的比较，但是会将所有更改都集中显示出来。在版本日志对话框中选择某一版本，然后在右键菜单中选择统一显示所有比较。

文件差异

如果你要查看两个不同文件之间的差异，你可以直接在资源管理器中选择这两个文件(一般使用 Ctrl-modifier)，然后右键菜单中选TortoiseSVN→ 比较差异。

WC文件/文件夹与URL之间的比较差异

如果你要查看你本地拷贝中的任一文件与版本库中任一文件之间差异，

谴责信息之间的比较差异

如果你要查看的不仅是比较差异而且包括修改该版本的作者，版本号和日期，你可以在版本日志对话框中综合比较差异和谴责信息。这里有更多详细介绍第 5.20.2 节 “追溯不同点”。

比较文件夹差异

TortoiseSVN 自带的内置工具不支持查看多级目录之间的差异，但你可以使用支持该功能的外置工具来替代。在这里 第 5.9.4 节 “其他的比较/合并工具”我们可以介绍一些我们使用过的工具。

6.提交代码时怎样知道自己改了哪些文件，别人改了哪些文件

7. 如何知道某个文件的某一行是谁在哪个版本修改的

8. 如何为一个SVN主工程建立分支或tag

创建分支使用步骤：

1、选择你要产生分支的文件，点击鼠标右键，选择[分支/标记...]

2、在[至URL(T)]输入框中将文件重命名为你的分支文件名，输入便于区分的日志信息，点击确认。

3、在SVN仓库中会复制一个你所指定的文件，文件名称就是你所命名的，但是在你的本地目录上看不到新建的分支文件名，要使你的文件更新作用到你的分支上，你必须选择文件，点击鼠标右键，选择[切换...],选择你重命名的文件，点击确定即可。这样你的本地文件就和分支文件关联上了，不要奇怪，这时本地目录上看到的文件名仍然为旧的文件名。

经验小结：

1、如果操作的文件之前还未提交，而你又想把文件提交到新的分支上，记得一定要选择切换

2、SVN分支的管理实际上就是把不同的分支用不同的文件保存，因此你在取得新版本的时候会发现，不同分支的最新文件也会被获取下来。

创建tag操作，相当于把当前的代码版本复制一份到其他地方，然后以这个地方为出发点进行新的开发，与原来位置的版本互不干扰。

对于branches、tags、trunk这三个目录，并不是subversion必需的，而是被总结的一种良好的团队开发习惯，其使用方法为：

1、开发者提交所有的新特性到主干。 每日的修改提交到/trunk：新特性，bug修正和其他。

2、这个主干被拷贝到“发布”分支。 当小组认为软件已经做好发布的准备(如，版本1.0)然后/trunk会被拷贝到/branches/1.0。

3、项目组继续并行工作，一个小组开始对分支进行严酷的测试，同时另一个小组在/trunk继续新的工作(如，准备2.0)，如果一个bug在任何一个位置被发现，错误修正需要来回运送。然而这个过程有时候也会结束，例如分支已经为发布前的最终测试“停滞”了。

4、分支已经作了标签并且发布，当测试结束，/branches/1.0作为引用快照已经拷贝到/tags/1.0.0，这个标签被打包发布给客户。

5、分支多次维护。当继续在/trunk上为版本2.0工作，bug修正继续从/trunk运送到/branches/1.0，如果积累了足够的bug修正，管理部门决定发布1.0.1版本：拷贝/branches/1.0到/tags/1.0.1，标签被打包发布。

一般建立最初的repository时，就建好这三个目录，把所有代码放入/trunk中，如：要将project1目录下的代码导入repository，project1的结构就是：project1/branches，project1/tags，project1/trunk，project1/trunk/food.c，project1/trunk/egg.pc……，然后将project1目录导入repository，建立最初的资料库。然后export回project1，作为本地工作目录。