**SVN Checkout(SVN取出)**

1 建立一个目录，如d:\dir。要空目录  
2 进入目录中, 点击SVN Checkout，弹出检出提示框，在URL of repository输入框中输入服务器仓库地址，在Checkout directory输入框中输入本地工作拷贝的路径，点击确定。

SVN会在您的工作目录下，以及其子目录下建立这个.svn的子目录。您不应该更动这个目录下面的任何内容。否则会很可能会造成SVN无法正常运作。

由于，原来的repository是空的，所以我们现在的working目录也是空的。如果您现在checkout的是一个已经有内容的repository，您将会看到working目录下面现在多了许多目录及档案。

如果您要在一个已经存在的SVN Server上面checkout出上面的档案，您只需要给定正确的URL以及working目录的名称。就可以取得指定的档案及目录了。

**Export(导出)**

把整个工作拷贝导出到本地目录下,导出的文件将不带svn文件标志,文件及文件夹没有绿色的”√”符号标志。导出文件夹下没有.svn目录。

**Update(SVN更新)**  
如果配置库在本地已有工作拷贝，则取得最新版本只是执行SVN Update即可，点击SVN Update，系统弹出更新提示框，点击确定，则把服务器是最新版本更新下来

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 更新本机代码与SVN服务器上最新的版本一致

**Update to revision(更新至版本)**  
从版本库中获取某一个历史版本。这个功能主要是方便查看历史版本用，而不是回滚版本。

如果你做了修改，那么当你提交的时候SVN会提示你，当前版本已失效（即不是最新版本），无法提交，需要先update一下。这样你所做的修改也就白费了。

注意

在要Update to revision的文件(夹)右键点击，Update的是右键的文件(夹)

在该文件下的空白处点击右键Update to revision，Update 的当前所在的文件夹

**Import（导入）**  
选择要提交到服务器的目录，右键选择TortoiseSVN----Import，系统弹出导入提示框，在URL of repository输入框中输入服务器仓库地址，在Import Message输入框中输入导入日志信息，点击确定，则文件导入到服务器仓库中。

Rename(改名) 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。

Delete(删除) 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。

**Moving(移动)**  
选择待移动的文件和文件夹；按住右键拖动right-drag文件（夹）到跟踪拷贝内的新地方；松开左键；在弹出菜单中选择move files in Subversion to here

**Revert(还原) 放弃修改**  
还原操作,如刚才对文件做了删除操作,现在把它还原回来,点击删除后,再点击提交,会出现如上的提示框,点击删除后,再点击Revert,即已撤销删除操作,如果这时候点击提交,则系统弹出提示框:没有文件被修改或增加,不能提交

当你添加、修改、删除文件后，决定放弃修改，你可以单击右键，TortoiseSVN->Revert，本地的“Working Copy”中的文件和目录会恢复到你修改前的状态。

Commit修改文件

 用文本编辑器或IDE对文件修改后，文件的状态图标会变化，然后单击右键，SVN Commit...提交修改，只有当执行Commit提交修改后，你所作的修改才会反映到服务器端的Repository中。

**Add(加入)**  
如果有多个文件及文件夹要提交到服务器，我们可以先把这些要提交的文件加入到提交列表中，要执行提交操作，一次性把所有文件提交，如图，可以选择要提交的文件，然后点击执行提交（SVN Commit）,即可把所有文件一次性提交到服务器上

Relocate(重新定位)  
当服务器上的文件库目录已经改变，我们可以把工作拷贝重新定位，在To URL输入框中输入新的地址

Cleanup（清除状态）  
如果当前工作拷贝有任何问题的话，可以使用这个选项进行修正。例如，有些文件原来是版本控制的，但是你没有通过tortoiseSVN就直接删除了，但是tortoiseSVN还是保留着原来的信息（每个文件夹下都有一个.svn的隐藏文件夹，存放着当前文件夹下所有文件夹的版本信息）

所以这就会产生一些冲突。可以使用cleanup来清理一下。

GetLock/ReleaseLock（加锁/解锁）  
如果你不想别人修改某个文件的话，那么你就可以把这个文件进行加锁，这样可以保证只有你对这个文件有修改权。除非你释放了锁，否则别人不可能提交任何修改到配置库中

**Branch/tag（分支/标签）**  
Branch是分支的意思。例如当在设计一个东西的时候，不同的人有不同的实现，但是没有经过实践检验，谁也不想直接覆盖掉其他人的设计，所以可以引出不同的分支。将来如果需要，可以将这些分支进行合并。  
tag是打标签的意思。通常当项目开发到一定程度，已经可以稳定运行的时候，可以对其打上一个标签，作为稳定版。将来可以方便的找到某个特定的版本（当然我们也可以使用版本号来查找，但是数字毕竟不方便）  
SVN对于分支和标签都是采用类似Linux下硬链接的方式（同一个文件可以存在两个地方，删除一个不会影响另一个，所做修改会影响另一个），来管理文件的，而不是简单的复制一份文件的拷贝，所以不会有浪费存储空间的问题存在。

**[SVN中检出(check out) 和 导出(export) 的区别](http://blog.csdn.net/yedezhanghao/article/details/7463180)**

check out跟check in对应，export跟import对应。

check out导出获得文件后，导出的文件仍处于SVN版本控制中，与版本库保持关联，比如你可以进行Svn Update或者Svn Commit操作。同时导出文件夹下有一个.svn的隐藏文件夹，存储着一些版本的元数据信息。

export 简单导出一个版本的数据，导出的文件脱离SVN版本控制，修改后无进行Update和Commit操作。导出文件夹下没有.svn目录。

checkout（检出）：相当于从服务器获取到本地。获取的文件是受控的

export（导出）：也是将文件获取到本地。但获取的文件是不受版本控制的。导出文件夹下没有.svn目录。

**[TortoiseSVN Switch与Relocate的区别](http://blog.csdn.net/cdh1213/article/details/8128696)**

switch用于在同一个版本库内不同分支之间的切换  
relocate用于版本库访问地址变更时，重新定位版本库

先明确两个概念，工作副本就是本地的代码的目录，工作目录就是SVN服务器端的与本地工作副本相对应的目录。

Switch是转换当前工作副本对应的工作目录，一般是从trunk工作目录转向tag工作目录，或者从tag转回来，switch的类似update，将switch的目标工作目录的文件更新到本地，一般会产生很多冲突。

Relocate是当代码仓库的访问路径URL变更，而此时已检出修改的工作副本（working copy）没有变更，若此时直接提交（commit），肯定不能成功，因为此提交地址对应的svn服务器不存在了。

http://www.cnblogs.com/tugenhua0707/p/3969558.html

http://blog.csdn.net/windone0109/article/details/4030274

TortoiseSVN是windows平台下Subversion的免费开源客户端。

一般我们都是先讲讲服务器的配置，然后再讲客户端的使用，但是在TortoiseSVN上，却可以反过来。因为，如果你的要求不高，只是想在本机，或者是可信任的局域网络中使用SVN版本控制，可以不需要安装SVN，而只使用TortoiseSVN即可。

安装地址

<https://sourceforge.net/projects/tortoisesvn/files/1.8.12/Application/>