**[TortoiseSVN菜单项功能说明](http://www.cnblogs.com/guohu/p/3965815.html)**

TortoiseSVN是windows下其中一个非常优秀的SVN客户端工具。通过使用它，我们可以可视化的管理我们的版本库。不过由于它只是一个客户端，所以它不能对版本库进行权限管理。

TortoiseSVN不是一个独立的窗口程序，而是集成在windows右键菜单中，使用起来比较方便。

TortoiseSVN每个菜单项都表示什么意思

01、SVN Checkout(SVN取出)  
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，即可检出服务器上的配置库。

02、SVN 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

03、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。

04、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

05、Resolving Conflicts(解决冲突)  
   有时你从档案库更新文件会有冲突。冲突产生于两人都修改文件的某一部分。解决冲突只能靠人而不是机器。当产生冲突时，你应该打开冲突的文件，查找以<<<<<<<开始的行。冲突部分被标记：  
<<<<<<< filename  
your changes  
=======  
code merged from repository  
>>>>>>> revision  
Subversion为每个冲突文件产生三个附加文件：  
filename.ext.mine  
更新前的本地文件。  
filename.ext.rOLDREV  
你作改动的基础版本。  
filename.ext.rNEWREV  
更新时从档案库得到的最新版本。  
使用快捷菜单的编辑冲突Edit Conflict命令来解决冲突。然后从快捷菜单中执行已解决Resolved命令，将改动送交到档案库。请注意，解决命令并不解决冲突，而仅仅是删除filename.ext.mineandfilename.ext.r\*文件并允许你送交。

06、Check for Modifications（检查更新）  
点击Check for Modifications,系统列表所以待更新的文件及文件夹的状态.

07、Revision Graph(版本分支图)  
查看文件的分支,版本结构,可以点击Revision Graph,系统以图形化形式显示版本分支.

08、Rename(改名)  
   SVN支持文件改名,点击Rename,弹出文件名称输入框,输入新的文件名称,点击确定,再把修改提交,即可完成文件改名

09、Delete(删除)  
   SVN支持文件删除,而且操作简单,方便,选择要删除的文件,点击Delete,再把删除操作提交到服务器

10、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

11、Revert(还原)  
   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

12、Branch/Tag(分支/标记)  
   当需要创建分支，点击Branch/Tag，在弹出的提示框中，输入分支文件名，输入日志信息，点击确定，分支创建成功，然后可查看文件的版本分支情况

13、Switch(切换)  
   文件创建分支后，你可以选择在主干工作，还是在分支工作，这时候你可以通过Switch来切换。

14、Merge(合并)  
   主干和分支的版本进行合并，在源和目的各输入文件的路径，版本号，点击确定。系统即对文件进行合并，如果存在冲突，请参考冲突解决。

15、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。

16、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

17、Add to Ignore List(添加到忽略列表)  
   大多数项目会有一些文件（夹）不需要版本控制，如编译产生的\*.obj, \*.lst,等。每次送交，TortoiseSVN提示那些文件不需要控制，挺烦的。这时候可以把这些文件加入忽略列表。

18、SVN其它相关功能  
   客户端修改用户密码:  
   打开浏览器,在地址栏内输入<http://192.168.1.250/cgi-bin/ChangePasswd>,启动客户端修改用户密码的界面,输入正确的用户名,旧密码,新密码(注意密码的位数应该不小于6,尽量使用安全的密码),点击修改即可.

19、SVN Commit（版本提交）  
把自己工作拷贝所做的修改提交到版本库中，这样别人在获取最新版本(Update)的时候就可以看到你的修改了。

20、Show log（显示日志）  
显示当前文件(夹)的所有修改历史。SVN支持文件以及文件夹独立的版本追溯。

21、Repo-Browser（查看当前版本库）  
查看当前版本库，这是TortoiseSVN查看版本库的入口，通过这个菜单项，我们就可以进入配置库的资源管理器，然后就可以对配置库的文件夹进行各种管理，相当于我们打开我的电脑进行文件管理一样。

22、Revision Graph（版本图形）  
查看当前项目或文件的修订历史图示。如果项目比较大型的话，一般会建多个分支，并且多个里程碑（稳定版本发布），通过这里，我们就可以看到项目的全貌。

23、Resolved（解决冲突）  
如果当前工作拷贝和版本库上的有冲突，不能自动合并到一起，那么当你提交修改的时候，tortoisesvn就会提示你存在冲突，这时候你就可以通过这个菜单项来解决冲突。冲突的解决有两种，一种是保留某一份拷贝，例如使用配置库覆盖当前工作拷贝，或者反过来。还有一种是手动解决冲突，对于文本文件，可以使用tortoiseSVN自带的工具，它会列出存在冲突的地方，然后你就可以和提交者讨论怎么解决这个冲突。同时它也对Word有很好的支持

24、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。注意：获取下来之后，对这个文件不建议进行任何操作。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。

25、Revert（回滚）  
如果你对工作拷贝做了一些修改，但是你又不想要了，那么你可以使用这个选项把所做的修改撤销

26、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。

27、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

28、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。

29、Export（导出）  
这个功能是方便我们部署用。当我们需要发布一个稳定版本时，就可以使用这个功能将整个工程导出到某个文件夹，新的文件夹将不会包含任何版本信息了。

30、Relocate（版本库转移）  
当我们版本库发生转移的时候就需要用到这个功能了。例如我原先的版本库是建在U盘上的，现在转移到（复制整个配置库文件夹）开发服务器上，使用https代替文件系统的访问。因此就需要将原来的工作拷贝的目标版本库重新定位到开发服务器上。

31、create patch（创建补丁）  
创建补丁。如果管理员不想让任何人都随便提交修改，而是都要经过审核才能做出修改，那么其他人就可以通过创建补丁的方式，把修改信息（补丁文件）发送给管理员，管理员审核通过之后就可以使用apply patch提交这次修改了。