## Svn常用命令及使用方法

### 检出(co)

让svn从版本库的一个目录中取出东西并创建一个工作拷贝。   
命令1：svn co URL 导出SVN库中的最新版本    
   
备注：   
如果省略路径参数，则 URL 最末尾的目录名作为目标目录名。如果指定多个 URL，则依次将其签出到 PATH 的子目录中，子目录名就是 URL 最末尾的目录名。

### Up更新(up)

如果你不是唯一一个在项目上工作的人，在你工作的时候很有可能其他人已经更新过项目仓库了。频繁地更新你的工作拷贝，不断合并他人的改动是一个好习惯。   
命令1：svn up 没有指定版本，则将工作副本更新到 HEAD 版本    
命令2：svn up -r 版本号 更新到-r 选项所指定的版本    
备注：在更新过程中，svn会显示每一个文件的状态以及对应的重要活动。添加到项目仓库中的文件和目录会被添加到工作副本，从项目仓库中移除的文件和目录也会从工作副本中移除。   
每更新一项就输出一行信息，使用首字符来报告执行的动作。这些字符的含义是:   
A 已添加   
D 已删除   
U 已更新   
C 合并冲突   
G 合并成功

### 添加(add)

svn只记住了你想要添加到项目仓库中的文件的名字；他并没有实际添加文件或者把改动让团队的其他人也能看见，所以在add之后，你需要commit。   
命令1：   
svn add 目录名称 目录下的所有文件一同纳入版本控制   
svn ci -m “ ”    
命令2：   
svn add 文件名称 仅将此文件纳入版本控制   
svn ci -m “ ”   
    
备注：Svn自动添加了目录中的所有文件，以及其子目录，除非你指定--non-recursive选项

### 新增目录 (mkdir)

命令：svn mkdir -m “ “ URL(新增目录)   
主要参数：--parents：创建中间目录，如果中间目录不存在使用此参数

### 删除 (delete, remove, rm)

命令1：   
svn delete 目录名称/文件名称 删除本地目录/文件，下次ci时从库中删除   
svn ci -m “ ”    
   
命令2：   
svn delete URL -m “ ” 直接从库中删除文件   
   
备注： 不能删除模块及模块以上级别的目录 ，因为命令2中直接对库操作，所以需要加注释

### 重命名 (mv, rename, ren)

命令1：   
svn rename wc(本地原来目录名/文件名) wc（重命名后的目录名/文件名）   
svn ci -m “ ” 将重命名提交到版本库   
命令2：   
svn rename URL(版本库里原目录名/文件名) URL（重命名后库里的目录名/文件名） -m “ ”   
  
注：我们规定： 不能修改模块及模块以上级别的目录名，禁止添加新的中文等非英文文件名的文件，在SVN代码库SCM已经对此进行了限制。

### 提交(commit)

命令：   
svn ci -m “ ” 指定日志信息ARG   
svn ci –F “filename” 编辑器会被启动，从指定文件（filename）读取日志信息。   
注：后台进行钩子控制，要求必须提交注释，并且超过5个字节

### 分支/标签 (cp)

分支是把整个目录拷贝到一个新的位置。标签是为一些文件的拷贝提供了版本库在某个特定时刻的快照。

* 复制：   
  命令：svn cp jchuff.h test.h
* 创建分支：   
  命令：1，确定基线：将基线版本co到本地工作副本   
  2，创建分支：svn cp . URL（branches）-m “create a branch ”   
  例如：想要在app/search/iknow /ac模块的主干末端版本创建分支ac\_1-0-5\_BRANCH,通过svn info 查询到本地所在工作副本即为主干，那么直接执行如下命令即可创建分支：
* 标记tag：   
  命令：在本地工作副本完成修改并提交   
  svn cp . URL（tags） -m ”create a tag”   
  输出：

备注：   
1， svn cp对于原来文件和拷贝的历史会跟踪到同一个来源。svn记住了文件的共享历史，跟踪历史的功能也知道这些操作的存在；   
2， 可以复制多个源到一个目的节点，此时该节点必须是目录。

### 切换(sw)

将工作副本切换到同一版本库中某个分支，主干或者标签 ，和cvs up -dr很像，因为它没有丢弃你在本地做的修改，在工作副本里当你进行切换的时候任何没有提交过的修改都会被混合。   
命令：svn switch URL

### 合并（merge）

#### 合并：

当分支版本测试通过或者上线完成后，要及时将此版本合并到其它进行中的分支或主干上。   
合并命令：

* **1，将已经上线的版本合并到其它并行开发的版本中（在svn中操作）：**   
  （1）svn co （合并目标）URL   
  （2）cd ~/wc   
  （3）svn merge （基线版本）URL （上线）URL 或者    svn merge （基线版本）URL  
    
  （4）svn ci -m "commit log"
* **2，提交前若需要回滚，可执行命令：**   
  svn st   
  svn revert . -R
* **3，提交后回滚**   
  svn info   
  svn log   
  svn merge -r head:N current\_URL   
  N为要回滚后的版本，current\_URL为info中给出的本地wc对应的url
* **4，提交后马上回滚**   
  svn merge -c -M current\_URL   
  M为当前版本号
* **5，出现冲突如何解决**   
  见下文

#### 同步：

当分支上的版本上线后，如果主干上没有开发，则需要把分支上的内容同步到主干；（对于scmpf中pre-released的处理）；同步即覆盖的意思，将某个分支或者tag的内容完全覆盖到另外一个分支或者主干上。同步命令如下：

* **1，确定同步的目标位置，将本地更新到目标位置的末端**   
  svn co （主干）URL   
  cd ~/wc
* **2，执行覆盖操作**   
  svn merge （主干）URL （待同步tag）URL  or   svn merge （待同步tag）URL   
  svn ci -m "commit log"

### 比较差异 (diff)

查看自从上次在项目仓库更新了工作副本后都改动了什么。   
命令：   
svn diff base与当前工作副本   
svn diff -r reversion1:reversion2 两个版本之间对比   
svn diff -r head 比较库中最新代码与当前工作副本   
svn diff (tag1)URL (tag2)URL 版本库中任意两个tag做对比 

### 还原(revert)

命令：svn revert PATH...   
说明：本子命令不会访问网络，是本地操作，它能解除任何冲突的状态，但是，它不恢复被删除的目录。恢复被删除的目录，需要使用up命令。   
当执行完merge操作，在执行完svn revert 合并后的文件后，还要在相应的目录下执行 svn revert . 操作，具体见还原merge操作。

### 显示信息(info)

在工作副本中会有大量不同的工作拷贝，要搞清楚工作拷贝是从哪里来的，可使用以下   
命令：   
svn info URL 显示该URL当前版本信息   
svn info –r revision URL 显示该URL第xx版本的信息   
svn info –r revision –R URL 递归显示该URL下每一个文件及目录的信息   
备注：这里的重点是第二行的URL，SVN会告诉你本地工作副本目录的原始来源。

### 日志(log)

在一棵大的目录树的顶端执行这项操作可能会产生大量的输出，建议可以通过使用|more管道来分页svn的输出.   
命令：   
svn log 显示本地工作区当前目录下所有文件版本信息   
svn log URL 显示该URL所有版本的基本信息   
svn log –r 显示你所关注的指定版本   
svn log –v 显示所有相关路径的日志信息   
svn log --stop-on-copy -v 显示当前tag/branch版本详情

### 解决冲突 (resolve)

resolve: 解决工作副本中目录或文件的冲突。   
命令: svn resolve --accept=ARG [PATH...]   
主要参数： --accept ARG : 指定自动解决冲突动作的源   
('base', 'working', 'mine-conflict‘, 'theirs-conflict', 'mine-full', 'theirs-full')

### 查看状态 (stat, st)

命令： svn st 说明：输出的前七列各占一个字符宽度，其中第一列最为常用，表示一个项目是增加、删除，还是修改 ，输出结果常出现的几种状态：   
“ ” 无修改   
“A” 增加   
“C” 冲突   
“D” 删除   
“I” 忽略   
“M” 改变   
“R” 替换   
“X” 未纳入版本控制的目录，被外部引用的目录所创建   
“?” 未纳入版本控制   
“!” 该项目已遗失(被非 svn 命令删除)或不完整   
“~” 版本控制下的项目与其它类型的项目重名

### 列表 (ls)

命令： svn list (ls) 列出版本库中的目录内容。

### 帮助 (?, h)

命令： svn help 子命令 描述本程序或其子命令的用法。

## 锁定/解锁(lock/unlock)

对工作副本或URL进行锁定/解除锁定   
用法: lock/unlock TARGET...   
注：只能对文件进行锁定，不能对目录进行操作

### 查看文件内容

blame (praise, annotate, ann)：输出指定文件或URL的追溯内容，包含版本和作者信息   
cat：输出指定文件或URL的内容

### 清理 (clean up)

cleanup，因某些原因工作副本滞留在一个不一致的状态，可以用cleanup来解决

### 文件集 (changelist)

changelist (cl)，可以定义一个或多个修改文件列表，当提交该文件列表时只涉及到该表中的文件。这个功能可以扩展到需求管理、bug管理等方面。

### 显示合并的相关信息 (mergeinfo)

mergeinfo

### 属性设置

propdel (pdel, pd)：删除目录、文件或版本的属性   
propedit (pedit, pe)：使用外部编辑器编辑属性   
propget (pget, pg)：显示目录、文件或版本属性的值   
proplist (plist, pl)：列出目录、文件或版本的属性   
propset (pset, ps)：设置目录、文件或版本的属性   
svn:keywords - 关键字设置，类似于CVS，却比CVS更灵活   
svn:executable - 如果存在此属性则设置文件为可执行的。   
svn:mime-type - 文件的媒体类型。   
svn:needs-lock - 此属性表明文件在修改前应该加锁。