目录

[一、从版本库获取信息 1](#_Toc322206702)

[二、从版本库到本地 2](#_Toc322206703)

[三、从本地到版本库 3](#_Toc322206704)

[四、高级应用: 5](#_Toc322206705)

[五、安装: 6](#_Toc322206706)

Subversion官方主页 <http://subversion.tigris.org/>

Subversion文档 <http://svnbook.red-bean.com/>

TortoiseSVN官方主页 <http://tortoisesvn.tigris.org/>

Svn新手教程

<http://sps.corp.alimama.com/scm/online/Wiki%20Pages/%E4%B8%BB%E9%A1%B5.aspx>

练习路径

<http://svn.simba.taobao.com/svn/QA/Scripts/SearchCenter/algorithm/test>

## 一、从版本库获取信息

svn help command

获取子命令说明

svn help：显示所有svn的命令。

svn help <subcommand>：显示子命令的说明

svn info $URL

查看工作空间信息

若是目录地址，查看本地目录信息，若无$DIR，默认为当前目录

若是文件地址，查看本地文件信息

svn list

显示给定目录在某一版本存在的文件

svn list 显示当前目录下svn记录文件列表，不访问版本库

svn list $URL 不下载到本地查看目录中的文件

检查历史:

svn log path

查看历史修改信息，展示每个版本附加在版本上的作者和日期信息和所有路径修改

svn log 5:19 查看版本5到19的历史信息

svn log 查看当前目录的历史修改信息

svn log filename 查看单个文件的历史修改信息

svn log $URL 查看$URL的历史信息

svn diff path 【di=diff】

将修改的文件与基础版本比较，查看所做的修改，即展示每一个文件改变的详细情况

svn diff 比较本地工作拷贝的修改

svn diff -r 3 filename 比较本地工作拷贝与版本库指定版本

svn diff -r 2:3 $URL比较版本库的两个版本

本地（工作拷贝）和本地修改前版本比较，精确查看修改内容，删除的行前一个-，添加的行前一个+

svn cat

在屏幕打印某个文件特定版本内容

svn cat -r 2 filename 显示文件指定版本内容

svn st [参数] 【st=status】

执行检查

【与svn diff 不同在于svn st只显示文件修改情况，而非详细内容】

svn st $DIR 查看本地已做的修改

在做svn revert/ci之前，请执行此操作，以免误操作

输出提示: M A D R K ? C \*

修改 增加 删除 提花 被锁定 未纳入版本控制 冲突 文件比版本库版本旧

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 例子 | 意义 | 是否联系版本库 |
| path | svn status | 查看当前目录下所有目录文件信息 | 否，与本地修改前比对 |
|  | svn status aa/a.txt | 查看指定目录下文件信息 |  |
| -v | svn status -v | 当前目录下所有目录文件信息，即使未改变过【显示文件和子目录状态】 | 同上 |
| -u | svn status -u | 与版本库最新版本比较，\*提示文件需要更新 | 是，与版本库最新版本比较 |

## 二、从版本库到本地

svn co $URL $DIR 【co=checkout】

将SVN库文件检出到本地工作空间

将目标文件检出到本地目录下，建立一个工作拷贝，每个目录都包含一个.svn管理目录

$URL与$DIR同级

若是不加本地目录地址，默认将远程文件夹整个拷到当前目录下

svn export $URL $DIR 【一般在打包时候使用】

从版本库导出一个干净的目录树，与svn co同，但不含.svn目录

非工作拷贝

svn up -r m path 【up=update】

更新本地工作空间，与SVN服务器保持同步

更新自己的工作拷贝，得到这对按时间他人的修改

如果后面没有目录，默认将当前目录以及子目录下的所有文件都更新到最新版本。

输出提示： U A D R G C

更新 增加 删除 替换 合并 冲突

svn up 更新本地工作空间，默认将当前目录及其子目录下所有文件都更新到最新版本

svn up filename 更新filename为最新

svn up –r 200 file.c –m “update the version to 200”将本地的file.c还原为200版本，并提交到服务器【本地是拿下来了，版本库端并没有被变】

## 三、从本地到版本库

svn import $DIR $URL -m "注释信息"【产生提交操作】

将$DIR下的目录导入到$URL下面去，而不会在$URL下新建目录

即，将DIR下的内容拷贝到URL下

例如：有一个文件夹test/code.java

svn import test $URL/test –m “import a new file”将code.java拷到版本库test文件夹下

此时本地的DIR无变化，本地与服务器并未建立管理，需要执行checkout取出服务器内容后才真正在本地建立了工作拷贝

相当于从本地拷贝到版本库，源文件与版本库文件无关

若想新建目录 svn import $DIR $URL/$DIR

svn add file 【本地操作】

往本地添加文件或者目录，svn ci时才进行提交

svn add test test.c 添加test目录及test.c文件

svn delete file -m “delete test fle“【**Delete\del\remove\rm**】

从本地工作或者URL里面删除目录，或移除文件

svn del test test.c 本地工作空间删除test目录和test.c ，执行svn ci时候才提交，版本增加【一旦svn del，本地文件夹或文件就被删除了，但本地svn list还可以查看到,只有提交并更新了，本地的svn文件才更新改变。未提交之前，版本库的还存在。】

svn del $URL –m “Del the dir for some reason” 产生提交，版本增加

svn mv $URL/old $URL/new -m "注释" 【mv=move 等价于svn cp 后svn delete】

【**Move\Mv/rename\ren 相同效果**】

移动或拷贝工作空间或者版本库的文件/目录 也可用于文件改名

svn mv foo.c f.c 工作空间文件改名，使用svn ci 则产生提交

svn mv –m “Move a file” $URL1 $URL2 移动文件，产生提交

svn rename SRCURL DSTURL

svn rm $URL/dir -m "注释"

目录删除

svn ci -m "注释信息" PATH 【ci=commit】

提交增删改操作(将本地修改的文件提交到svn)

注意，提交之后，本地的svn list依旧是旧的，需要手动svn up获取最新的

svn ci filename –m “comment” #提交并加注释，至少5个字符

svn revert filename【本地操作，会丢失修改，慎用】

恢复对文件或者目录的修改，用于未执行提交操作(ci)之前，撤销本地修改。

等价于执行svn rm filename,svn up -r BASE filename

svn revert –R $DIR 恢复工作空间$DIR目录下的所有修改

svn revert test.c 恢复test.c的修改

svn mkdir $URL/dir -m "注释"

工作空间或者版本库创建目录

svn mkdir newdir 本地工作空间新建目录newdir,通过svn ci后产生提交

svn mkdir –m “Making a dir” $URL/$NEW\_DIR 产生提交

svn cp $URL1 $URL2 -m "注释"【cp = copy这个操作成功但也可能造成后序有些操作失败掉，慎用，对新加入文件最好不用…】

工作拷贝或者版本库之间文件的相互拷贝

起源（SRC） 和 目的（DST） 可以是工作拷贝路径或地址(URL)：

工作拷贝 -> 工作拷贝 ： 复制和通过调度进行增加(包含历史)

工作拷贝 -> 地址(URL) ： 马上提交一个工作拷贝到地址(URL)

地址(URL) -> 工作拷贝 ： 签出地址(URL)到工作目录，通过调度进

行增加

地址(URL) -> 地址(URL) ： 完全服务器端复制；一般用于分支和标签

svn cp foo.txt bar.txt 本地文件拷贝

svn cp $URL $DIR 本地新增$URL目录，会把$URL目录放到$DIR下

svn cp $URL1 $URL2 –m “注释” 产生提交操作

svn cp $DIR $URL –m “注释” 产生提交操作 [DIR必须已在svn中存在，即，曾被提交过]

也可用于建立新的分支

之后 svn co $URL2，就可以在分支工作拷贝中进行操作，svn ci -m "fix bug 1031"

找回删除的项目

当前版本中无，上一版本中存在

-精确拷贝svn cp -r 19 $URL/filename filename

-检查结果 svn status

-提交 svn ci -m "resurrected c.txt from r19"

## 四、高级应用:

svn merge –r m:n path

将两个版本之间的差异合并到当前文件, 一般都会产生冲突，需要处理一下

比较两个版本树，将区别应用到本地拷贝 初始版本树 最终版本树 一个接收区别的工作拷贝

合并分支

-找到分支产生的版本

svn log -v --stop-on-copy $URL 查到分支产生的版本是r10

-使工作目录为主干的本地拷贝

$cd proj/trunk

$svn up

可以看到最新版本，假设r15

-执行合并

$svn merge -r 10:15 $URL

-检查合并结果，也许要手工解决冲突，最后提交

$svn ci -m "merged br\_1\_0 10:15 to trunk"

代码回滚

撤销一个已提交的版本,修改有误，不应该提交，需要回滚到上一个版本

-执行命令

$svn merge -r 20:19 $URL

-检查工作拷贝结果

$svn status

-提交拷贝结果

$svn ci -m "undo change commited in r20"

svn switch $URL

提供一种改变工作拷贝的快捷方式

比如原拷贝在trunk上，目前想切换到分支上工作

$svn switch $分支URL

运行svn info | grep URL 可看到档签的url是分支的路径

svn resolved filename

删除冲突标记，在svn up是，提示U（本地文件被更新）G（成功合并）没有必要处理

C本地和服务器修改冲突，需手工处理

当发生冲突 -C标记文件

-冲突文件中植入冲突标记 (<<... == ... >>)

-每一个冲突的文件，SVN在本地工作拷贝中生成三个未版本化的文件、

filename.mine本地修改后的版本

filename.rOLDREV上次更新后未作修改的本本

filename.rNEWREV服务器最新版本

此时执行svn ci失败

应 -手工解决冲突，打开冲突文件，根据冲突标记(<<...==...>>)修改文件，可参考三个未版本化文件，修改完成后删除这些标记

-运行svn resolved filename 或手工删除三个未版本化的文件

-运行svn ci -m "注释" 提交修改

svn lock -m “LockMessage” [-force] PATH

加锁, 锁定版本库的工作拷贝路径或URL，所以没有其他用户可以提交这些文件的修改。

改变工作拷贝，版本库【访问版本库了】

svn lock –m “lock test file” test.php

svn unlock PATH

解锁

## 五、安装:

1.在Windows下安装

安装Subversion命令行客户端

<http://subversion.tigris.org/>

安装TortoiseSVN资源管理器集成客户端

<http://tortoisesvn.tigris.org/>

2.在Linux下安装

安装Subversion命令行客户端<http://subversion.tigris.org/>

和Eclipse集成

安装Subclipse插件<http://subclipse.tigris.org/>