

# 版本管理工具—SVN

## 一、SVN 基础

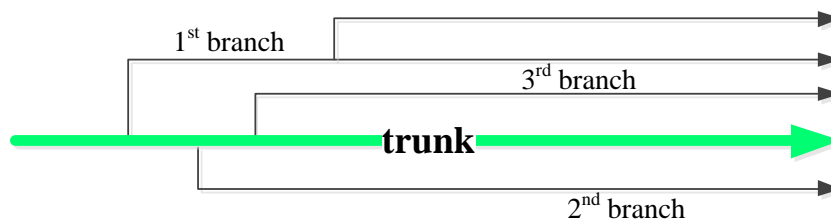
### 1、SVN 定义及作用

svn(subversion)是版本控制工具，是通过差异算法来记录和存储每次提交的文件差异来，可以实现回滚到某个时间点上的代码，或对比最新的和本机代码看有哪些变化等，对排除程序bug,代码审查，保存稳定版本程序是作用很大的。

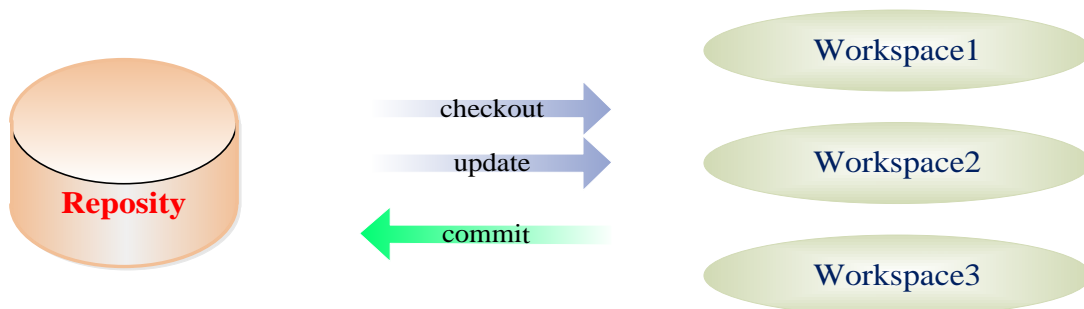
### 2、SVN 的基本概念

svn 的基本概念有：

**Repository**（资源库）运行于 svn 服务器端存储所有的文件和变化的数据文件；**Workspace**（工作空间）运行与客户端，用户的工作空间，从资源库(repository)取出的文件都在 workspace 中；**Checkout**（取出）从 repository 取文件到工作空间；**Commit**（提交）将工作空间内的文件提交到 svn 服务器；**Update**（更新）从资源库更新工作空间的文件；**Revision**（修订版本）；**Tag**（标签）；**Branch**（分支）；**Conflict**（冲突），主干和分支之间的关系如下所示：



客户端在本地 checkout(取出)或者 update(更新)远程 repository(资源库)的文件到本地，经过修改等各种操作 commit(提交)到远程资源库：

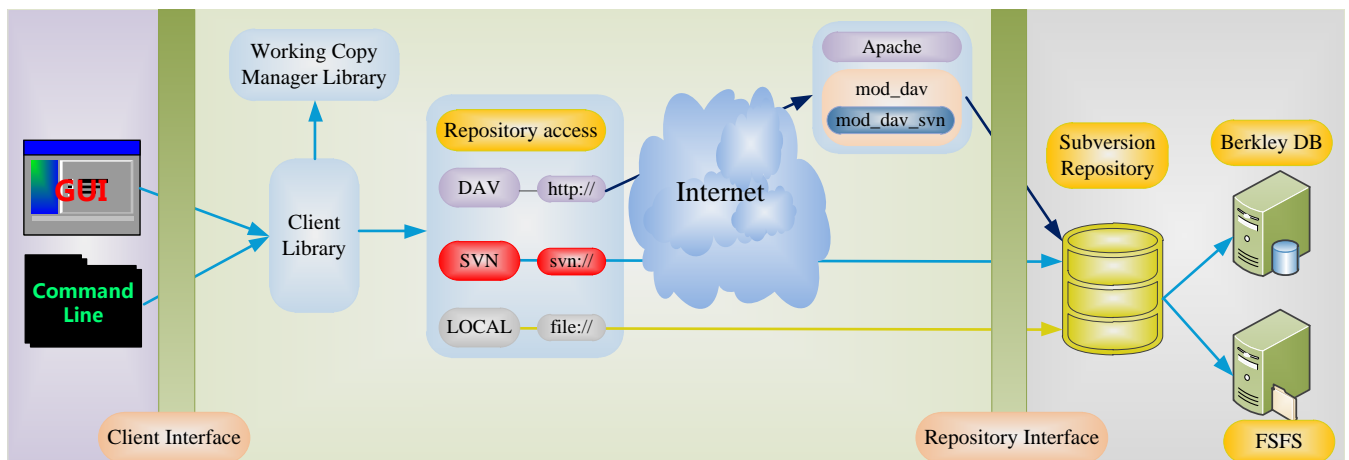


### 3、SVN 的体系结构

SVN 的体系结构如下图所示，是基于关系型数据库的(BerkleyDB)或一系列二进制文件的(FSFS)。一解决了并行读写共享文件以及运行时的事务特性。连接 Repository 有三种不同的方式：http/svn/file，其中 file 是利用 subversion 的 C/S 架构，从本地连接至本地的 svn 服务端，

使用较多的是第二种方式。

Subversion 体系结构图



## 二、SVN 服务器客户端的安装使用

### 1、SVN 服务器端的安装和配置

#### (1)SVN 的安装：

在官网 <http://subversion.apache.org/packages.html> 下载所需版本的安装包，一下以 CentOS 为平台演示安装过程（CentOS6.4+自带有 subversion 的安装包）：

```
##安装 subversion:
[root@vs1 ~]#yum install subversion -y
[root@vs1 ~]#mkdir -pv /data/svn/                #创建 svn 数据存放目录
[root@vs1 ~]#svnserver -d -r /data/svn/          #指定 svn 数据存放目录
[root@vs1 ~]#svnadmin create /data/repos         #创建 svn 的版本库
```

#### (2)SVN 的配置

SVN 的配置文件主要有三个，均默认在资源库目录下的 conf 目录下，以下是对配置文件的介绍，主配置文件(svnserver.conf)、用户文件(passwd)、权限配置文件(authz)

```
[root@vs1 repo]#cat conf/svnserver.conf          #编辑 repo 资源库下主配置文件 svnserver.conf
anon-access = none                               #是否允许匿名访问，允许是 read，拒绝是 none
auth-access = write                              #访问权限为写入
password-db = passwd                             #用户及密码数据存放位置，默认是 conf/passwd 文件
authz-db = authz                                 #权限配置文件，默认是 conf/authz 文件，可加绝对路径
[root@vs1 conf]#cat conf/passwd                  #添加访问资源库的用户及对应的密码
[users]
USERNAME1 = PASSWORD1                           #用户名和对应的密码
```

```

USERNAME2 = PASSWORD2

[root@vs1 conf]#cat conf/authz          #设置用户访问权限
[PROJECT_NAME]                          #项目名称
group_name1 = USER_NAME1,USER_NAME2... #组名称和对应的成员名
group_name2 = USER_NAME3,USER_NAME4... #组名称和对应的成员名
[PROJECT_NAME:/]                        #项目名称和对应的根目录
USER_NAME = rwx                         #给用户授权，空表示没有权限
@group_name1 = rwx                      #给用户组授权
@group_name2 = rwx                      #给用户组授权

```

配置文件举例：

```

[root@vs1 repo]#cat conf/svnserver.conf
[general]
anon-access = read
auth-access = write
password-db = /root/test/svndata/repos-1/conf/passwd
authz-db = /root/test/svndata/repos-1/conf/authz
[root@vs1 repo]#cat conf/passwd
[users]
maohua = maohua
[root@vs1 repo]#cat conf/authz
[groups]
admin = maohua
[/]
@admin = rw
##启动 svnsere 并连接：
[root@vs1 ~]#svnserve -d -r /root/trest/svndata
[root@vs1 ~]#svn checkout svn://192.168.88.11/repos-1 --username maohua --password maohua

```

## 2、SVN 客户端工具

SVN 的客户端工具有，Windows<——> TortoiseSVN Repository Browser (changy)，以及通用的 svn 命令行工具。

### (1) TortoiseSVN

TortoiseSVN 下载网址：<https://tortoisesvn.net/downloads.html>。SVN 客户端工具 TortoiseSVN 的安装使用可参考：<http://blog.csdn.net/liuyuelinjiayou/article/details/52210514>。

### (2) SVN 命令行工具

```

[root@vs1 ~]# svn help          #查看一系列的命令
1、将文件 checkout 到本地目录

```

`svn checkout path` (path 是服务器上的目录)

例如: `svn checkout svn://192.168.1.1/pro/domain`

简写: `svn co`

## 2、往版本库中添加新的文件

`svn add file`

例如: `svn add test.php` (添加 test.php)

`svn add *.php` (添加当前目录下所有的 php 文件)

## 3、将改动的文件提交到版本库

`svn commit -m "LogMessage" [-N] [--no-unlock] PATH` (如果选择了保持锁, 就使用 `-no-unlock` 开关)

例如: `svn commit -m "add test file for my test" test.php`

简写: `svn ci`

## 4、加锁/解锁

`svn lock -m "LockMessage" [--force] PATH`

例如: `svn lock -m "lock test file" test.php`

`svn unlock PATH`

## 5、更新到某个版本

`svn update -r m path`

例如:

`svn update` 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。

`svn update -r 200 test.php` (将版本库中的文件 test.php 还原到版本 200)

`svn update test.php` (更新, 于版本库同步。如果在提交的时候提示过期的话, 是因为冲突, 需要先 update, 修改文件, 然后清除 `svn resolved`, 最后再提交 commit)

简写: `svn up`

## 6、查看文件或者目录状态

1) `svn status path` (目录下的文件和子目录的状态, 正常状态不显示)

【?: 不在 svn 的控制中; M: 内容被修改; C: 发生冲突; A: 预定加入到版本库; K: 被锁定】

2) `svn status -v path` (显示文件和子目录状态)

第一列保持相同, 第二列显示工作版本号, 第三和第四列显示最后一次修改的版本号和修改人。

注: `svn status`、`svn diff` 和 `svn revert` 这三条命令在没有网络的情况下也可以执行的, 原因是 svn 在本地的 .svn 中保留了本地版本的原始拷贝。

简写: `svn st`

## 7、删除文件

`svn delete path -m "delete test file"`

例如: `svn delete svn://192.168.1.1/pro/domain/test.php -m "delete test file"`

或者直接 `svn delete test.php` 然后再 `svn ci -m 'delete test file'`, 推荐使用这种

简写: `svn (del, remove, rm)`

## 8、查看日志

`svn log path`

例如: `svn log test.php` 显示这个文件的所有修改记录, 及其版本号的变化

## 9、查看文件详细信息

`svn info path`

例如: `svn info test.php`

## 10、比较差异

`svn diff path` (将修改的文件与基础版本比较)

例如: `svn diff test.php`

`svn diff -r m:n path` (对版本 m 和版本 n 比较差异)

例如: `svn diff -r 200:201 test.php`

简写: `svn di`

## 11、将两个版本之间的差异合并到当前文件

`svn merge -r m:n path`

例如: `svn merge -r 200:205 test.php` (将版本 200 与 205 之间的差异合并到当前文件, 但是一般都会产生冲突, 需要处理一下)

## 12、SVN 帮助

`svn help`

`svn help ci`

## 13、版本库下的文件和目录列表

`svn list path`

显示 path 目录下的所有属于版本库的文件和目录

简写: `svn ls`

## 14、创建纳入版本控制下的新目录

`svn mkdir`: 创建纳入版本控制下的新目录。

用法: 1、`mkdir PATH...`

2、`mkdir URL...`

创建版本控制的目录。

1、每一个以工作副本 `PATH` 指定的目录, 都会创建在本地端, 并且加入新增

调度，以待下一次的提交。

2、每个以 URL 指定的目录，都会透过立即提交于仓库中创建。

在这两个情况下，所有的中间目录都必须事先存在。

#### 15、恢复本地修改

`svn revert`: 恢复原始未改变的工作副本文件（恢复大部份的本地修改）。`revert`:

用法: `revert PATH...`

注意：本子命令不会存取网络，并且会解除冲突的状况。但是它不会恢复被删除的目录

#### 16、代码库 URL 变更

`svn switch (sw)`: 更新工作副本至不同的 URL。

用法: 1、`switch URL [PATH]`

2、`switch -relocate FROM TO [PATH...]`

1、更新你的工作副本，映射到一个新的 URL，其行为跟“`svn update`”很像，也会将服务器上文件与本地文件合并。这是将工作副本对应到同一仓库中某个分支或者标记的方法。

2、改写工作副本的 URL 元数据，以反映单纯的 URL 上的改变。当仓库的根 URL 变动（比如方案名或是主机名称变动），但是工作副本仍旧对映到同一仓库的同一目录时使用这个命令更新工作副本与仓库的对应关系。

#### 17、解决冲突

`svn resolved`: 移除工作副本的目录或文件的“冲突”状态。

用法: `resolved PATH...`

注意：本子命令不会依语法来解决冲突或是移除冲突标记；它只是移除冲突的相关文件，然后让 `PATH` 可以再次提交。

#### 18、输出指定文件或 URL 的内容。

`svn cat 目标[@版本]...`如果指定了版本，将从指定的版本开始查找。

`svn cat -r PREV filename > filename` (`PREV` 是上一版本,也可以写具体版本号,这样输出结果是可以提交的)

### 3、SVN + HTTP (HTTPS) 登录

安装 `svn+http` 需要装载 `mod_dav_svn`, 需安装软件: `mod_dav_svn`、`httpd-devel`、`openssl-devel`、`mode_ssl`(用于实现 `https+svn`)

#-----安装软件-----#

```
[root@vs1 ~]#yum install httpd httpd-devel mod_dav_svn mode_ssl -y
```

#-----SVN 的配置-----#

```
[root@vs1 ~]#useradd svn
```

```
[root@vs1 ~]#svnadmin create /home/svn/repos
##在/home/svn 目录下建立账号目录以及权限目录，统一管理各个资源库的用户
[root@vs1 ~]#cat /home/svn/svn_passwd
maohua = maohua
[root@vs1 ~]#cat /home/svn/svn_authz
[aliases]
[groups]                #svn 的用户组
admins = maohua          #admins 用户组下目前只有 hujie 一个用户，多个用英文逗号隔开
[repos:/]                #lfdcwtxt 版本库的根目录
@admins = rw              #用户组前面需要加上@符号，rw 表示读、写权限
* =                       #其他一切用户连读都不让读
[root@vs1 ~]#cat /home/svn/repos/conf/svnserve.conf
[general]
anon-access = none
auth-access = write
password-db = /home/svn/svn_passwd          #建立的统一用户文件
authz-db = /home/svn/svn_authz              #建立的统一的权限文件
[root@vs1 ~]#svnserve -d -r /home/svn        #启动 svnserve 服务
```

## #-----http 的配置-----#

```
##建立 web 访问的用户，切记和 svn 的用户不是同一用户
[root@vs1 ~]#htpasswd -cm /home/svn/svn_http_passwd maohua  #建立访问网页的 web 用户
[root@vs1 ~]#cat /etc/httpd/conf.d/subversion.conf
LoadModule dav_svn_module      modules/mod_dav_svn.so
LoadModule authz_svn_module    modules/mod_authz_svn.so
<Location /svn>
    DAV svn
    SVNListParentPath on
    SVNParentPath /home/svn      #资源库目录的上一级目录
    AuthType Basic
    AuthName "Login with your svn account"
    AuthUserFile /home/svn/svn_http_passwd      #指定 web 访问的用户，和 svn 用户文件不同！
    AuthzSVNAccessFile /home/svn/svn_authz      #指定用户权限，权限文件可与 svn 公用
    Satisfy all
    Require valid-user            #设定需要 web 认证
</Location>
```

至此 SVN+HTTPS 配置完成，在浏览器中输入 `http://192.168.88.11/svn` 之后输入 web 访问的账号密码进行登录。

#-----https 的配置-----#

## ## 先建立自签 CA 证书

```
[root@vs1 ~]# cd /etc/pki/CA/
[root@vs1 ~]# (umask 077;openssl genrsa -out private/cakey.pem 2048)
[root@vs1 ~]# openssl req -new -x509 -key private/cakey.pem -out cacert.pem -days 3650
##通过自签证书给 svn 服务颁发 CA 证书
[root@vs1 ~]# mkdir /etc/httpd/conf.d/ssl
[root@vs1 ~]# (umask 077;openssl genrsa -out svn.key 2048)           #生成 svn 的秘钥文件
[root@vs1 ~]# openssl req -new -key svn.key -out svn.csr             #生成证书
[root@vs1 ~]# openssl ca -in svn.csr -out svn.crt -days 3650       #签发证书
```

## ## subversion.conf 文件

```
[root@vs1 ~]# cat /etc/httpd/conf.d/ssl.conf
SSLCertificateFile /etc/httpd/conf.d/ssl/svn.crt                   #将证书目录修改为此
SSLCertificateKeyFile /etc/httpd/conf.d/ssl/svn.key                #将秘钥目录修改至此
[root@vs1 ~]# cat /etc/httpd/conf.d/subversion.conf
LoadModule dav_svn_module      modules/mod_dav_svn.so
LoadModule authz_svn_module    modules/mod_authz_svn.so
<Location /svn>
    DAV svn
    SVNListParentPath on
    SVNParentPath /home/svn                                         #资源库目录的上一级目录
    AuthType Basic
    AuthName "Login with your svn account"
    AuthUserFile /home/svn/svn_http_passwd                         #指定 web 访问的用户, 和 svn 用户文件不同!
    AuthzSVNAccessFile /home/svn/svn_authz                         #指定用户权限, 权限文件可与 svn 公用
    Satisfy all
    Require valid-user                                              #设定需要 web 认证
    SSLRequireSSL                                                    #在之前基础上追加证书认证过程, 可实现 https 功能
</Location>
```

至此 SVN+HTTPS 配置完成, 在浏览器中输入 `https://192.168.88.11/svn` 之后输入 web 访问的账号密码进行登录。