

SVN的使用

“

为什么使用SVN? 在企业工作时，经常需要团队配合，这时一份工程需要多个人共同维护，业内称之为多人联编。联编的方式有很多种，最原始的方式就是A写完代码a,B写完代码b....M写完代码m，然后由N负责把a到_m的所有代码进行整合，而且N需要和所有人沟通，对所有代码都很清晰，整合时不能够出现差错，否则代码就有可能达不到预期效果。很显然这个工作是非常繁重而且效率非常低的。作为地球上最智慧的一群人，我们怎么能容忍这种低效率的工作呢。所以诞生了很多联编工具，帮助我们自己解决各种问题。其中应用最广泛的莫过于SVN了，当然目前git以其先进敏捷的特点，得到越来越多的人的认可和使用，著名程序猿交友网站[Github](#)就是提供git服务的网站。[git和SVN的区别](#)

[百度百科](#)给出SVN的定义：

“

SVN是Subversion的简称，是一个开放源代码的版本控制系统，相较于RCS、CVS，它采用了分支管理系统，它的设计目标就是取代CVS。互联网上很多版本控制服务已从CVS迁移到Subversion。

[百度百科](#)中还举了一个生动的例子：

开始新一天的工作：

- 1、从服务器下载项目组最新代码。
 - 2、进入自己的分支，进行工作，每隔一个小时向服务器自己的分支提交一次代码（很多人都有这个习惯。因为有时候自己对代码改来改去，最后又想还原到前一个小时的版本，或者看看前一个小时自己修改了哪些代码，就需要这样做了）。
 - 3、下班时间快到了，把自己的分支合并到服务器主分支上，一天的工作完成，并反映给服务器。
- 这就是经典的svn工作流程，从流程上看，有不少缺点，但也有优点。

以上是SVN的重要性的描述。下面我们一起来看看，SVN如何使用。

搭建SVN服务器

服务器的作用就如同引言中所说的工程师N的作用，负责整合代码，并且还能记录版本信息。

服务器的搭建过程非常简单，我们以Windows为例，使用基于Apache构建的VisualSVN，可以快速简单地搭建出SVN服务器。

具体请参考：[使用VisualSVN Server搭建服务器](#)

任何一台连接在局域网中的windows电脑都可以变成SVN服务器，如果把SVN托管到公网中，那么在世界任何角落你都可以开心的提交你的代码了，常见于跨国、跨地区合作。

使用SVN

在Mac OS X中，使用SVN可以使用命令行工具，也可以使用各种软件。我们先从命令行讲起，介绍几个简单的SVN命令。

SVN基本命令：

- 1、**Checkout**: 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、**Add**: 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、**Commit**: 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、**Lock:Checkout**: 加锁/解锁

```
svn lock -m "LockMessage" [--force] PATH
例如: svn lock -m "lock test file" test.php
svn unlock PATH
```

- 5、**Update**: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、**Status:**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、**Delete:**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、**Log:** 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、**Info:**查看文件详细信息

```
svn info path
```

例如: svn info test.php

- 10、**Diff:**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
```

SVN图形界面工具软件:

SVN图形界面工具软件有很多, 在Mac环境下可以使用Versions、Cornerstone等等, 这里我们只介绍一下Cornerstone的使用。

1. Checkout
2. Add
3. Delete
4. Revert
5. Commit
6. Update Update的重点在于如何解决冲突