## 是什么

Subversion 是一个集中式的版本控制系统。版本库是 Subversion 的核心部分，是数据的中央仓库,版本库以典型的文件和目录结构形式文件系统树来保存信息。

## 解决什么问题

版本控制系统的核心任务是实现协作编辑和数据共享。

## 如何解决的

使用“拷贝*-*修改*-*合并模型”；

一个好的实践是：创建一个trunk目录来保存开发的“主线”，一个 branches 目录存放分支拷贝，tags 目录保存标签拷贝。

Subversion的修订号是针对整个**目录树**的，而不是单个文件。每一个修订号代表了一次提交后版本库整个目录树的特定状态，另一种理解是修订号N代表版本库已经经过了N次提交。

### 工作拷贝怎样追踪版本库

对于工作拷贝的每一个文件，Subversion在管理区域.svn/记录两项关键的信息：

* 工作文件所作为基准的修订版本（叫做文件的工作修订版本）和
* 一个本地拷贝最后更新的时间戳。

给定这些信息，通过与版本库通讯，Subversion可以告诉我们工作文件是处与如下四种状态的那一种：

|  |  |
| --- | --- |
| 未修改且是当前的 | 文件在工作目录里没有修改，在工作修订版本之后没有修改提交到版本库。**svn commit**操作不做任何事情，**svn update**不做任何事情。 |
| 本地已修改且是当前的 | 在工作目录已经修改，从基本修订版本之后没有修改提交到版本库。本地修改没有提交，因此**svn commit**会成功的提交，**svn update**不做任何事情。 |
| 未修改且不是当前的了 | 这个文件在工作目录没有修改，但在版本库中已经修改了。这个文件最终将更新到最新版本，成为当时的公共修订版本。**svn commit**不做任何事情，**svn update**将会取得最新的版本到工作拷贝。 |
| 本地已修改且不是最新的 | 这个文件在工作目录和版本库都得到修改。一个**svn commit**将会失败，这个文件必须首先更新，**svn update**命令会合并公共和本地修改，如果Subversion不可以自动完成，将会让用户解决冲突。 |

## 混合版本限制

Subversion能够在工作拷贝中混合有不同的修订版本。使用混合版本的限制是：

* 你不可以提交一个不是完全最新的文件或目录，如果有个新的版本存在于版本库，你的删除操作会被拒绝，防止你不小心破坏你没有见到的东西。
* 如果目录已经不是最新的了，你不能提交一个目录的元数据更改。你将会在第6章学习附加“属性”，一个目录的工作修订版本定义了许多条目和属性，因而对一个过期的版本提交属性会破坏一些你没有见到的属性。

## 未掌握内容

## 学习新意点

## 与git的不同之处

1. 集中式版本控制
2. 分支概念。Subversion并没有内在的分支概念而只有拷贝，当你拷贝一个目录，这个结果目录就是一个“分支”，只是因为你给了它这样一个含义而已。你可以换一种角度考虑，或者特别对待，但是对于Subversion它只是一个普通的拷贝， 只不过碰巧包含了一些额外的历史信息。第二，因为拷贝机制，Subversion的分支是以普通文件系统目录存在的，这与其他版本控制系统不同，它们都为分支定义了另一维度的“标签”。

## 常用指令：

**Svn import：**是将未版本化文件导入版本库的最快方法，文件会直接提交到版本库。

**svn help：**

svn checkout：下载工作副本

模式 访问方法  
file:/// 直接版本库访问(本地磁盘)  
http:// 通过配置Subversion的Apache服务器的WebDAV协议  
https:// 与 http:// 类似，但是包括 SSL 加密。  
svn:// 通过定制的协议访问 svnserve 服务器  
svn+ssh:// 与 svn:// 类似，但通过 SSH 隧道。

**svn update：更新本地版本；**

**svn commit：提交当前版本**

**svn commit** 操作可以作为一个原子事务，发布任意数量文件和目录的修改。在你的工作副本里，你可以改变文件内容, 删除, 改名，以及复制文件和目录，然后作为一个原子事务提交。

**svn status：**检查混合工作修订版本，可以使用。

你希望拷贝或移动工作副本的一个文件，你应该使用 **svn copy** 或者 **svn move，而不是操作系统的命令。**