服务端: [Download | VisualSVN Server](https://www.visualsvn.com/server/download/)

客户端: [Downloads · TortoiseSVN](https://tortoisesvn.net/downloads.html) (往下找还有个中文包)

服务端用于底层功能,客户端是用户界面和右键菜单.

SVN服务端安装

说明： 由于软件版本的不一致，安装界面上可能会有些许区别，这里使用的是 VisualSVN-Server-3.8.1-x64

双击安装文件，点击 Next

勾选协议，点击 Next

使用默认选项，点击 Next

单击 Standard Edition

选择软件安装位置、仓库位置、端口号以及备份文件位置

单击 Install

安装完成，点击 Finish

如果电脑安装了360或者类似的安全软件，那么在软件安装过程中可能会拦截一些东西，这时只要都选择允许即可

SVN服务端配置

创建一个项目（以 Shop 为例）

首先在 SVN 服务器端的安装根目录下创建一个公有目录 WebApp 作为项目目录

在 WebApp 目录下创建 Shop 文件夹，作为 Shop 的版本仓库

注意：

在 SVN 中没有项目的概念，所有的文件夹都称之为版本仓库，一个项目就是一个仓库

现在的 Shop 文件夹还仅仅只能称之为一个普通文件夹，还不能称之为版本仓库，因为它还没有和 SVN 服务器产生任何联系，下面要做的就是将这个普通的文件夹转换为版本仓库

创建版本仓库，基本语法如下：

打开 cmd，输入如下命令（这里的路径名和最后的项目名应和自己电脑上的一致，下同）

svnadmin create D:\VisualSVN-Server-3.8.1-x64\WebApp\Shop

1

回车，如果 Shop 文件夹下出现了如下目录，则说明创建版本仓库成功，如果报错，则可以尝试重启电脑，以使指令生效

启动服务端监管

什么是服务端监管

对于Apache来说，我们通过在浏览器中输入 http://localhost(或ip地址) 来访问相关的文件，这个过程就叫做服务端监管

对于 SVN 来说，我们通过在浏览器中输入 svn://localhost(或ip地址) 来访问相关的数据仓库（如Shop仓库），这个过程就叫做服务端监管

如何启动服务端监管

打开 cmd，输入如下命令：

svnserve -d -r D:\VisualSVN-Server-3.8.1-x64\WebApp\Shop

1

说明：

-d 表示后台运行

-r 表示监管目录

回车，可以看到光标一直在闪烁，说明指令执行成功

特别注意：不要关闭当前的 cmd 窗口，如果关闭了，那么监管指令就强行终止了，也就没办法进行数据的连接了

通过以上指令，我们的 svn://localhost(或ip地址) 就可以直接指向 Shop 版本仓库，也就是说，当我们在浏览器中输入 svn://localhost(或ip地址) 的时候，就可以访问到 Shop 仓库下的相关数据，也即实现了监管

权限控制：

默认情况下，SVN服务器是不允许匿名用户上传文件到服务端的，所以必须更改项目的相关配置文件

如何更改相关权限

打开 Shop 文件夹中的 conf 文件夹，可以看到如下文件

用记事本打开 svnserve.conf 文件，找到 # anon-access = read ，将其更改为 anon-access = write ，然后保存，即将用户的权限由只可读改为可读可写

SVN客户端安装

说明： 由于软件版本的不一致，安装界面上可能会有些许区别，这里使用的是 TortoiseSVN-1.9.7.27907-x64-svn-1.9.7

双击安装文件，点击 Next

点击 Next

选择软件安装位置，然后点击 Next

点击 Install，直至软件安装完成，最后点击 Finish

如何验证SVN客户端是否安装成功

在桌面点击鼠标右键，如果结果如下图所示，则表示安装成功

SVN客户端汉化包的下载、安装

下载

点击这里

在打开的页面中往下拉，找到下图所在位置，并下载对应的汉化包（速度可能会比较慢，耐心等待一会）

安装

双击运行汉化包，点击 Next，等待安装完成，最后点击 Finish即可

安装完成后，在桌面点击鼠标右键，选择 TortoiseSVN --> Settings --> General，然后在右边的 Language 一栏选择简体中文，最后点击确定，再次打开设置界面时，就会发现是中文的了

如果想切换回英文，只需在上一步中将 Language 改为 English 即可

使用SVN客户端连接SVN服务器

相关说明

首先，我们设想这样一个使用场景：公司在开发项目时，需要把项目部署在 SVN 服务器上，假设这台 SVN 服务器的 IP 地址为 192.168.1.1

以之前的 Shop 为例，当要开发这个项目时，项目经理会把项目相关的配置和初始化工作做好

项目相关的配置也即上面所提到的 SVN 服务端的配置这部分工作

项目的初始化涉及到的有：编写核心框架、编写核心函数库等，项目经理在自己的电脑上完成这些文件的编写后，就会使用 commit 指令将这些文件上传到 SVN 服务器的 Shop 项目中（在使用 commit 命令之前，需要先使用 checkout 指令与服务器建立连接，下面会详细说明）

项目经理的 checkout（检出）过程

重要说明：

当项目经理在 SVN 服务器上完成了项目的配置之后，接下来要做的就是在自己的电脑上通过 SVN 客户端与 SVN 服务器建立连接

无论是项目经理还是程序员，如果想建立与 SVN 服务器的连接并更新数据到本地，第一次的操作都是 checkout

当项目经理使用 checkout 指令与服务器建立连接后，会从服务器下载一个名为 .svn 的隐藏文件夹到本地，这个文件夹是否出现也是检出成功与否的标志

在第一次之后，如果还需要下载服务器端的数据到本地，则使用的是 update 指令，而不再是 checkout 指令

我们把第一次与 SVN 服务器进行连接操作并更新 Shop 项目中的相关数据到本地的过程称之为检出

项目经理 checkout 的步骤：

首先要启动服务端监管，方法在上面已经介绍过

假设自己现在是项目经理，在自己电脑中选择一个位置新建一个文件夹，在这里我选择的是 E:\Shop ，然后在 Shop 文件夹下新建一个名为 jingli 的文件夹

进入 jingli 这个文件夹，在空白处点击鼠标右键，选择 TortoiseSVN --> 版本库浏览器（汉化之后的名称，下同）

在弹出的窗口中输入如下图所示 URL，这里因为是在本机测试，所以写的是 localhost，如果不是在本机，就写 SVN 服务器的 IP 地址，比如上面提到的 192.168.1.1

点击确定之后，会弹出如下界面，选中左边栏的条目，用鼠标右键单击，然后在弹出的菜单中单击检出

在接着弹出的如下两个窗口中都点击确定即可，这时在 jingli 这个文件夹下就会生成一个名为 .svn 的隐藏文件夹，将其显示出来即可看到，这也是我们与 SVN 服务器成功建立连接的标志。至此，项目经理的 checkout 的过程已全部完成

SVN三大指令

检出（checkout）

前提：必须先启动服务端监管

作用：链接到 SVN 服务器、更新服务端数据到本地

注意：checkout 只在第一次链接时使用一次，以后如果进行更新操作请使用 update 指令，这点在前面已经提到过，这里再强调一次

提交（commit）

前提：必须先启动服务端监管

作用：提交本地数据到服务器

如何使用 commit 指令

以项目经理为例，假设项目经理在自己的电脑上（也就是在上面提到的 jingli 这个文件夹下）编写好了项目的需求文档等文件，现在需要把这些文件上传到 SVN 服务器，那么只需在进入 jingli 这个文件夹后，在空白位置单击鼠标右键，然后单击 SVN提交 即可，如下图所示：

点击 SVN提交 之后，会出现如下窗口，勾选需要上传的文件和目录/文件夹，然后点击确定，若出现下图2所示界面，则代表提交成功

特别注意：备注信息必须填写

更新（update）

前提：必须先启动服务端监管

作用：将服务器上的数据下载到本地

使用方法：进入项目文件所在文件夹，在空白处点击鼠标右键，然后点击 SVN更新 ，如果出现如下窗口，则表明更新成功

SVN常用图标解释

如何查看所有图标

在任一空白处点击鼠标右键 ——> TortoiseSVN ——> 设置 ——> 图标覆盖 ——> 图标集，即可查看所有图标

图标含义解释

常规图标

含义：当客户端文件与服务端文件完全同步时，系统显示这个图标

冲突图标

含义：当客户端提交的文件与服务端已存在的文件有冲突时，系统显示这个图标

删除图标

含义：当服务端数据已删除，那么在客户端的该文件就会显示这个图标

增加图标

含义：当我们编写的文件已添加到提交队列，但还没有提交成功时，系统就会显示这个图标

无版本控制图标

含义：当我们编写的文件没有添加到上传队列时，系统就会显示这个图标

修改图标

含义：当客户端文件有修改但还未提交时，系统就会显示这个图标

只读图标

含义：当客户端文件以只读形式存在时，系统就会显示这个图标

锁定图标

含义：当服务端数据已锁定，那么客户端文件就会显示这个图标

忽略图标

含义：当客户端文件已忽略，不需要进行提交时，就会显示这个图标

SVN是什么

SVN 是 Subversion 的简称，是一个开源的版本控制系统，CVS的接班人，目前绝大多数公司都是用 SVN 作为版本控制软件

相关概念

SCM： 全称是 Software Configuration Management，即软件配置管理，意思就是对软件的源代码进行控制与管理

为什么需要SVN

SVN 可以用于协作开发、远程开发和版本回退

SVN的特点

操作简单，入门容易，支持跨平台（Windows、Linux、MacOS）

支持版本回退功能（时间机器）

属于 C/S 结构的软件，因此使用 SVN 需要分别安装客户端和服务端

TortoiseSVN VisualSVN Subversion 以及 VisualSVN Server的区别

客户端软件： TortoiseSVN 和 VisualSVN

服务端软件： Subversion 和 VisualSVN Server

从根本上说，TortoiseSVN（内核使用的是 Subversion） 和 VisualSVN 都是 Subversion 的衍生物，是

为 Subversion 服务的。

TortoiseSVN 可以很方便地向服务器提交代码、从服务器下载代码以及查看自己以前提交的代码

VisualSVN 是 SVN 的客户端，和 Visual Studio 集成在一起，VisualSVN 是收费的

Subversion 是 SVN 的源码库，是一个集成了客户端和服务端的“内核”，相关配置需要手动进行

VisualSVN Server 的内核使用的也是 Subversion，它包括 apache、Subversion 和 WMI（用于实现操作界面等），而且其中的很多配置都已经配置好了，比较容易上手，VisualSVN Server 标准版是免费的，企业版是收费的，在安装时可自行选择

为什么不直接使用Subversion

因为 Subversion 比较偏向于底层，我们需要用到的很多东西都没有集成，对于新手来说，使用难度很大