# SVN操作手册

SVN（全称Subversion），中文名可以理解为版本控制。所谓版本控制就是对程序，文档等进行版本的控制。

SVN服务分成两个部分：一个是服务端（版本库），一个是客户端。服务端提供版本控制、文件存储、文件比对等功能，客户端可以跟服务端进行“沟通”，完成文件提交、修改提交、文件检出等功能。

简述SVN服务端版本控制概念

Subversion是一种集中分享信息的系统，它存储所有的数据，版本按照文件树形式存储数据——包括文件和目录。任何数量的客户端可以连接到版本库，读写这些数据。

在版本库中，每当版本库中接受了一个提交，文件系统进入一个新的状态，叫做一次修订，每一个修订版本被赋予一个独一无二的自然数，一个比一个大。

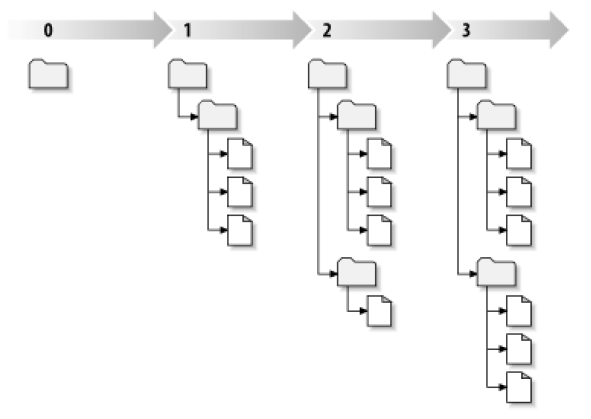


图 1 版本树

操作

操作，首先安装一个可视化操作界面，该软件为：TortoiseSVN（开源）。可从网上下载使用，然后安装，安装完成后进行下面的操作。

安装完软件后，可以首先进行观察下版本库上的情况（根据自己帐号的权限有关）。首先在桌面上右击（软件安装完后会在右击快捷菜单上多出关于TortoiseSVN的几个操作），选择TortoiseSVN->Repo-browser。

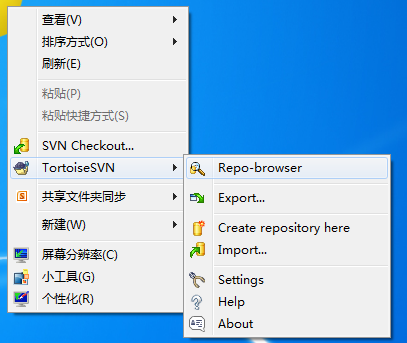


图 2

弹出一个对话框，输入URL，需要具体到帐号所能进入的目录为止。

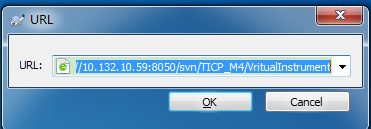


图 3

点击OK，进入帐号和密码输入界面。

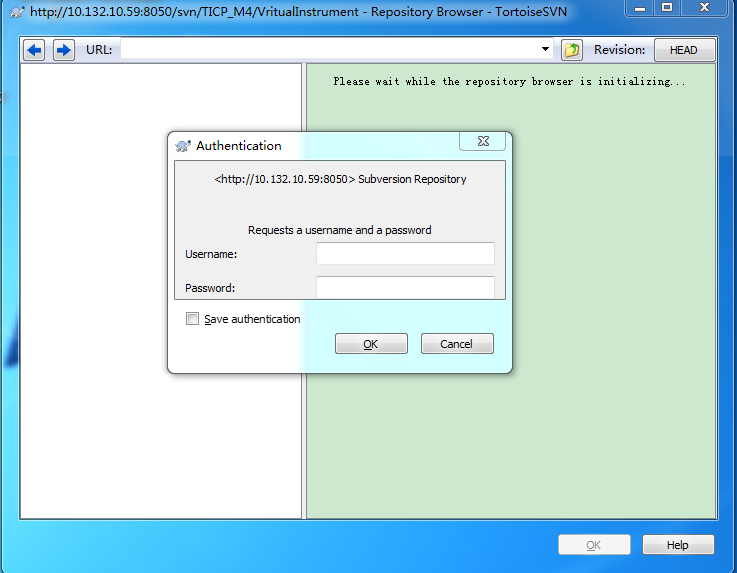


图 4

输入自己的帐号和密码，进入SVN版本库界面。

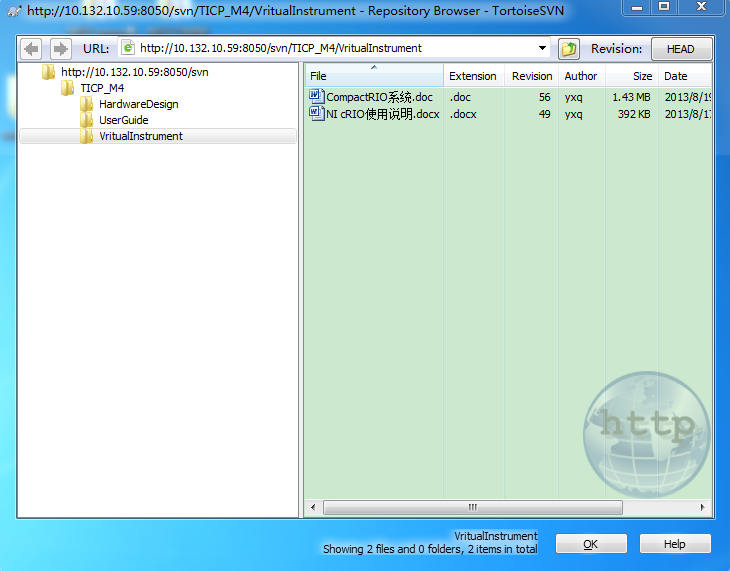


图 5

在这个界面中右击有许多功能，此处可以自己熟悉，一般在这个界面上直接操作比较少。

Import操作

要在SVN版本库中存有自己的文件首先需要进行提交操作（import）。

第一步：右击选中需要提交到版本库中的文件，如图6.然后点击Import，见文件提交到版本库中。

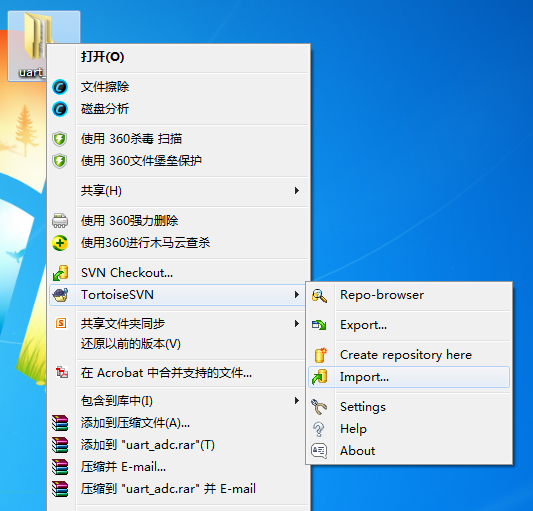


图 6

第二步：弹出如下对话框，可以选择文件需要提交到版本库那个文件夹中。

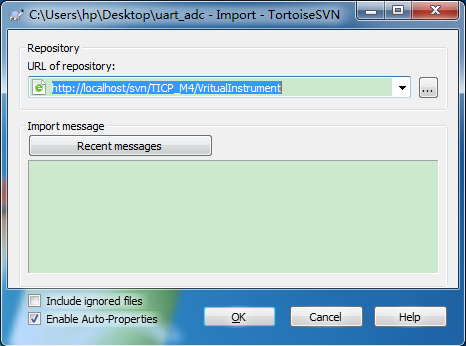
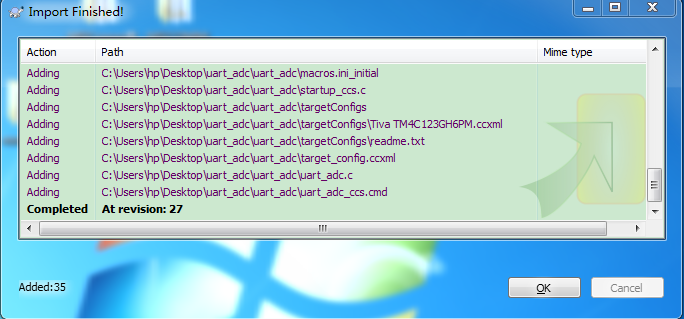


图 7

第三步：点击OK后出现如图8所示界面。其表示文件已经提交至版本库中。

  
图 8

当然可以进行查看确认，操作过程和图2到图5过程类似。

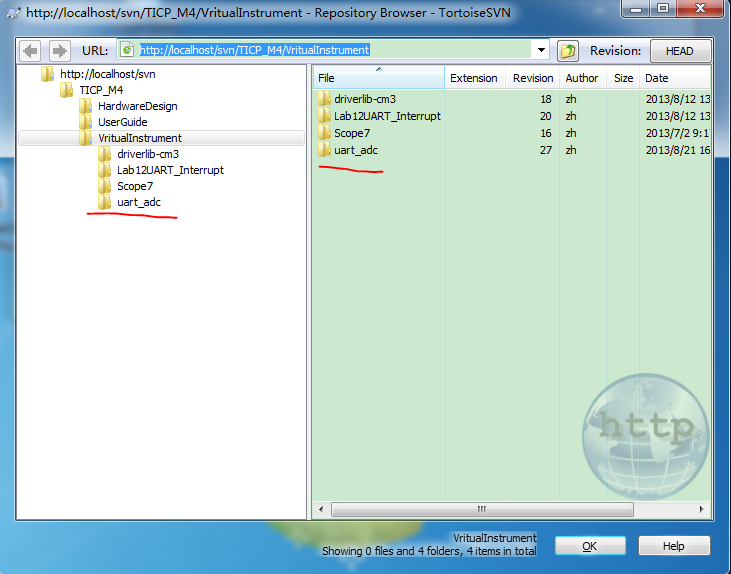


图 9

Checkout操作

在Import操作之后，版本库中已经存在你所提交的文件，如图9所示，但是你所使用的本地计算机上的对应的文件还没有和版本库进行交互——也就是你本地计算机上没有相应版本库的工作拷贝。所以此时需要进行Checkout操作，获得一个工作拷贝。

第一步：在桌面或者任何空白区域右击鼠标，如图10。选择SVN Checkout。

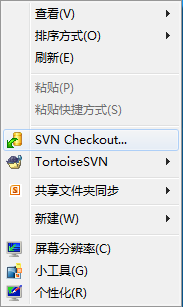


图 10

第二步：点击SVN Checkout，弹出Checkout界面，选择正确的URL of repository(就是你需要Checkout出的文件路径)，以及Checkout directory（就是Checkout出的文件在本地计算机中保存的路径）。如图11.界面右下角的Revision可以根据需要选择不同的版本。可以通过Show log进行选择。

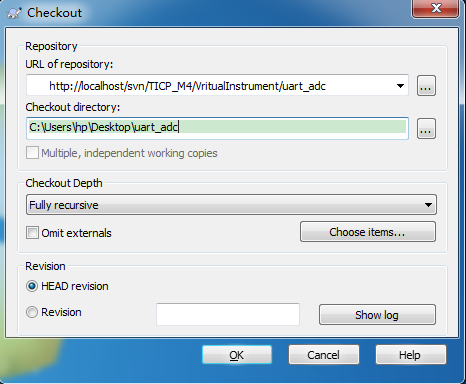


图 11

第三步：设置好后，点击OK进入图12界面。显示Checkout Finished，表示Checkout成功了。

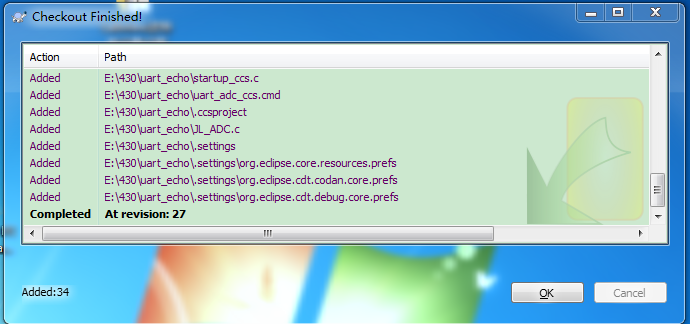


图 12

下面观察下Checkout出的文件跟原始文件有什么不同。可以发现Checkout出的文件在文件上多了一个绿色的对号，打开文件夹后发现所有的文件都多了一个对号并且在显示隐藏文件的时候可以发现多了一个.svn的文件夹。这就表示该文件已经是一个工作拷贝了。

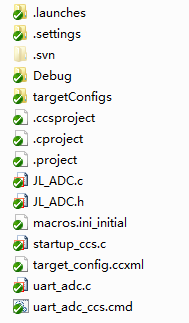


图 13

当文件发现变化时，文件上的对号就会变成红色的叹号，表明此时的文件跟Checkout出的原始版本文件已经发生了变化。

Commit操作

Commit：将修改提交到版本库。

第一步：右键点击需要将修改提交到版本库的文件，选择SVN Commit。

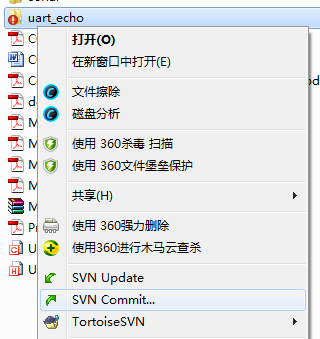


图 14

第二步：弹出Commit界面，如图15.会在Changes made（double-click on file for diff）中列出被修改过的文件。

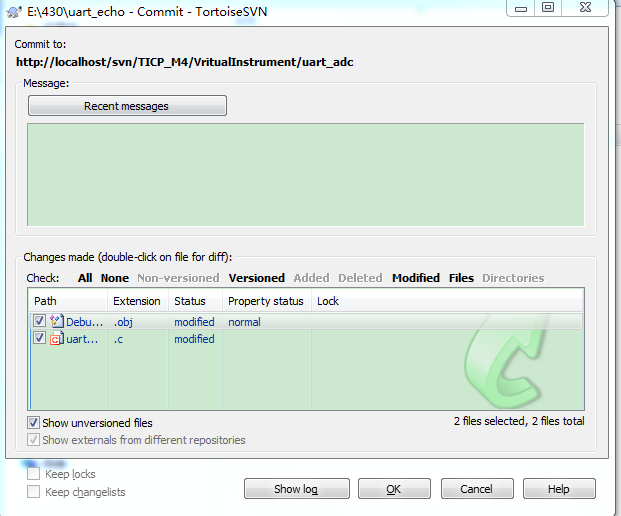


图 15

同时此时可以查看原版本库文件跟现在做出修改后的文件的不同之后（修改处），双击文件即可。将会出现图16界面。其中修改处会高亮显示，并且会在修改处前出现一些标记，比如图中的“+”标记。此次功能在团队合作开发项目时非常有用，比如A和B在同时修改一个文件，A先于B提交了自己的文件到版本库，到B提交时就可以使用diff这个功能来查看不同之处，并且做出相应的调整。

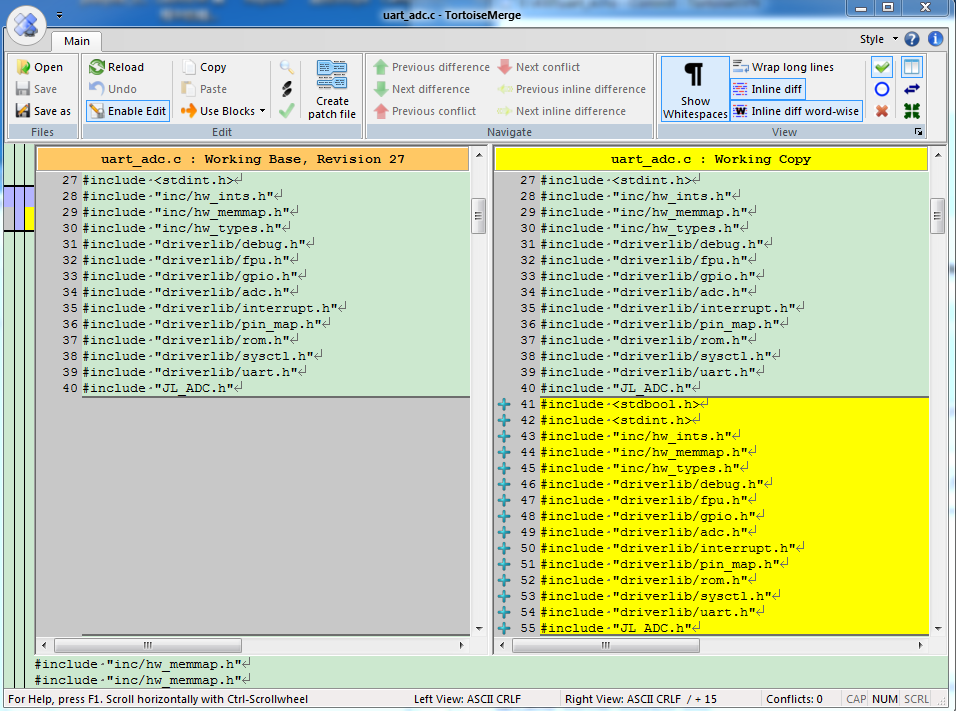


图 16

第三步：点击OK后，文件的修改将会被提交到版本库中。同时会给予一个新的版本号。

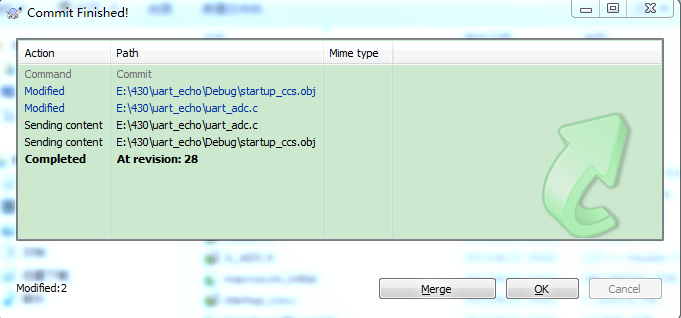


图 17

add操作

Add操作是将新添加的文件加入到版本库中，但是此时只是标记需要添加到版本库中而已，此时在版本库中仍是不存在需要add的文件的，只有最后进行Commit之后，文件才会自动提交上去。如你没有将新添加的文件标记问add状态的话，你在Commit的时就需要自己人为选定需要提交的新文件。

第一步：右击文件，选择TortoiseSVN->Add。之后文件会多出一个“+”标记

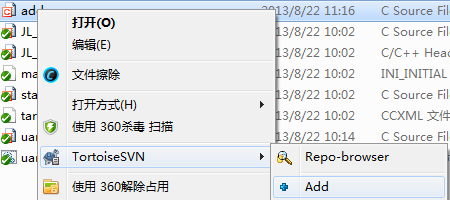


图 18

此时我们登录查看版本库中的文件，将会发现add.c不存在版本库中。（此处不在展示图片，你可自行查看）

然后我们在新建一个文本文档，此时不进行add操作，来查看进行add操作和不进行add操作的区别。

第二步：完成第一步后需要进行Commit操作，将修改提交到版本库中。右击文件选择SVN Commit，弹出Commit界面，观察Changes made（double-click on file for diff）。可以发现add.c文件的Status为added状态，而新建文本文档的status为non-versioned。并且两者之前的选择框默认情况下add.c为选中，而新建文本文档为不选中。这个就是其中的区别。当你选中新建文档文档时，在点击OK其可也可以提交到版本库中。（可以自行登录查看文件变化）

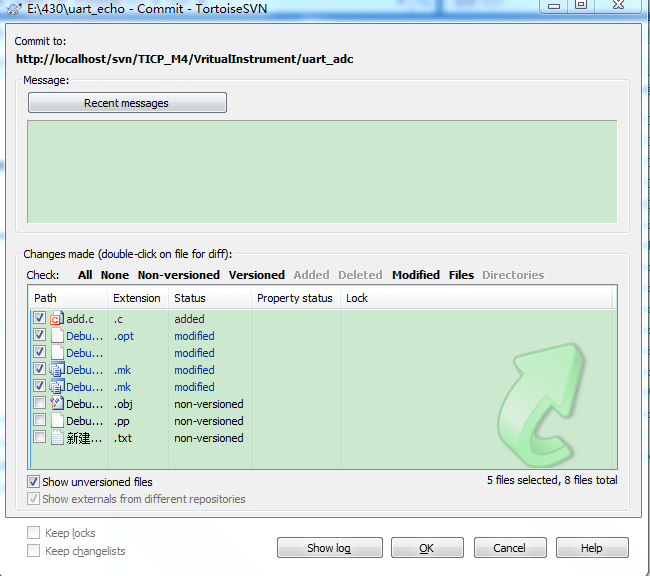


图 19

update操作

Update操作就是将版本库中的文件更新到本地工作拷贝中，将本地工作拷贝中的文件更新到版本库的中的最新版本。

从前面很多操作过来，本地工作拷贝中有些文件的版本号应该不是最新的，所以为保持与版本库一致就可以进行该操作，当然其最主要的是用在另一个情况下，当团队工作时，你将不清楚目前操作的版本和版本库上的版本的不同，在隔些天在继续工作时，为修改目前最新版本可以首先进行Update操作获得最新版本。

第一步：右击选择SVN Update。



图 20

第二步：弹出Update Finished界面。点击Ok完成操作。

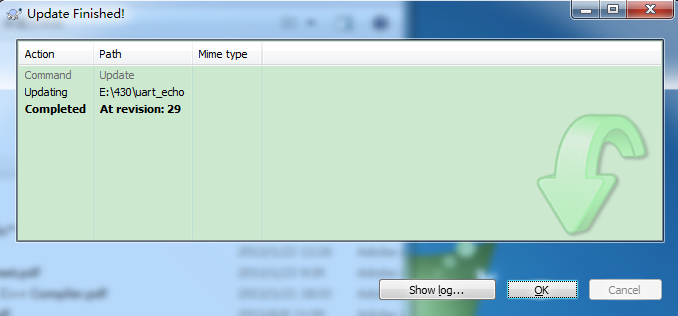


图 21

Diff操作

Diff操作在之前Commit操作中已有说明。当然也可以根据Commit操作中进行操作。下面描述另外的操作方式。

对于安装完成后，右击本地工作拷贝的文件（不是文件夹）时，右击菜单中是没有diff选项的，但是diff的功能有事必须需要的，所以此时可以根据以下操作完成。

第一步：右击任何位置，选择TortoiseSVN->Settings。进入设置界面，选择左项目栏中的Context Menu。在出来的右侧界面中选中Diff等操作即可。如图22.

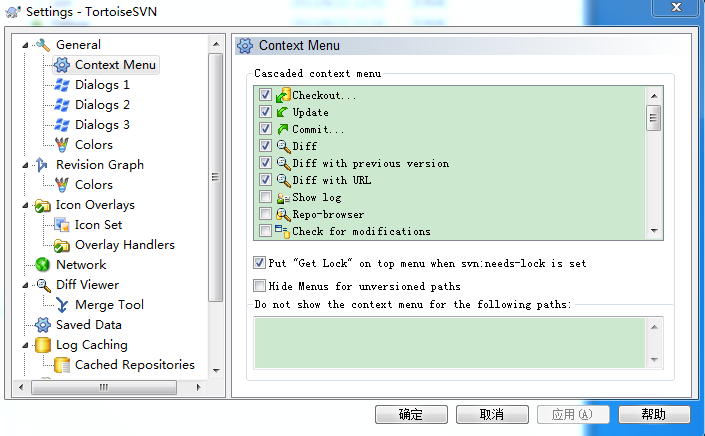


图 22

第二步：右击某个文件，菜单栏中将会出现Diff选项。



图 23

如果版本库中存在之前版本，则会出现图16界面，若不存在之前版本，则会提示不存在更早的版本。

至此已经可以进行SVN进行版本控制操作了。