SVN使用参考

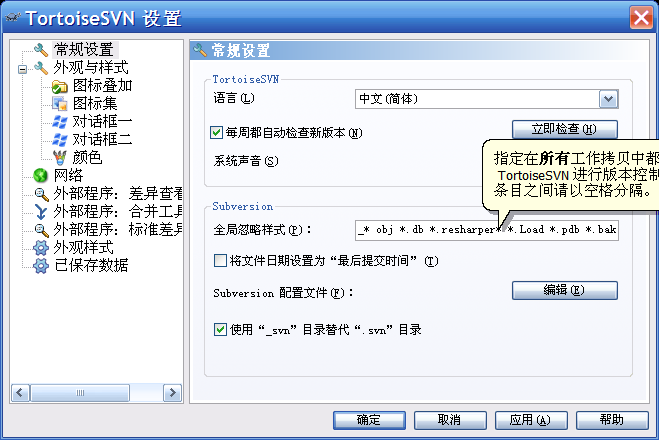
# 安装后的常规设置

### 1.1忽略文件

为防止在提交操作时将不必要的文件上传到SVN服务器上，可将项目中不需要受SVN控制的文件和文件夹通过设置忽略选项的方式排除掉。常被忽略的文件有 \_\* obj \*.db \*.resharper\* \*.Load \*.pdb \*.bak。

可以根据实际需要来修改忽略参数， 每个需要忽略的文件（文件夹）名称之间用空格作分隔符，需要忽略的文件夹只要写文件夹名称就行。例如 obj，忽略obj文件夹及所包含文件；\_\*忽略所有\_（下划线）开头的文件夹及所包含文件。

注意，这个设置仅对本机有效，而且对本机的所有项目的本地工作区有效，对其他项目组成员没有影响。被忽略的文件仍可以通过“添加”命令使其受SVN控制。操作界面如下图，只需将忽略参数填写在“全局忽略模式”后的文本框中即可。



### 1.2 配置强制注释

每次提交时添加注释是一个非常好的习惯。项目负责人通过设置项目的TortoiseSVN:logminisize设置为5，以便强制注释。设置方法：在项目的本地工作区文件夹上点击鼠标右键选择“属性”，进入 Subversion标签，选中TortoiseSVN: logminisize，确保复选框“递归应用该属性”选中，取值设为5，其意思是指提交的注释最短长度为五个字。注意，这个设置对其他项目成员的也有效，但是对别的SVN客户端工具无效。



# 版本控制原则

SVN只是一个版本控制的辅助工具，项目成员在项目进程中要尽量通过优化流程来解决，而不是将希望寄托于软件工具来自动解决一切问题。建议的开发过程组织：

### 2.1项目刚开始阶段，单独开发；项目稳定阶段，完整开发。

项目开发初期，各个项目成员负责自己的文件夹（或者模块），与SVN服务器间的更新、提交等操作只需要针对自己负责的文件夹（或者模块）；

项目稳定阶段，也就是每天的变更量很小了，所有项目成员与SVN服务器的更新、提交等操作需要针对项目的所有文件夹（或者模块），各个项目成员在其本地编译时本地工作区的全部项目程序（或者资料）均为最新的版本，保证项目作为整体能够顺利运行。

### 2.2尽量保证一份文件只有一个项目成员在编辑。

举例说明：程序员A负责文件 DBAccess.php的编写，如果程序员B的工作要求他为DBAccess.php增加两个方法，程序员B应该通知程序员A来增加而不是自己增加；

如果此时A的确需要B帮助修改，B先更新（update）本地的DBAccess.php，然后开始修改，修改完成后立即提交并立即通知A更新其本地的DBAccess.php文件，这样就可以通过缩短提交间隔来减少冲突。

# 初始化版本库

初始化版本库有两种常用的方式

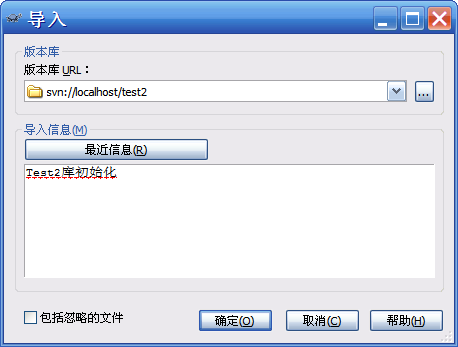
### 3.1方法一：直接导入“Import…”

对要执行导入操作得项目文件夹进行如下清理：

1.把项目中不需要的文件删除。（临时文件、编译器创建的文件，比如\*.obj、二进制文件等。）

2.把文件夹和子文件夹中的所有文件整理一遍。虽然你可以在导入之后再来进行重命名或删除等操作，但是还是推荐你在导入之前把你的项目整理好。

在资源管理器（windows explorer）中选择项目(本地硬盘上)的根文件夹，单击鼠标右键，选择**导入Import…**命令，跳出一个对话框：

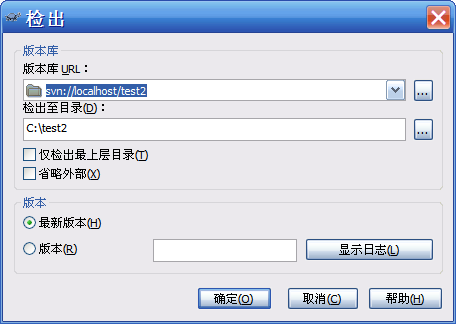


在这个对话框中你需要填写你要将项目导入仓库的URL地址（svn必须小写）。导入信息（Important Message）是用来记录日志信息的。注意：与“忽略样式exclude pattern”匹配的文件或文件夹不会被导入，除非选择了“包含忽略的文件”选项。

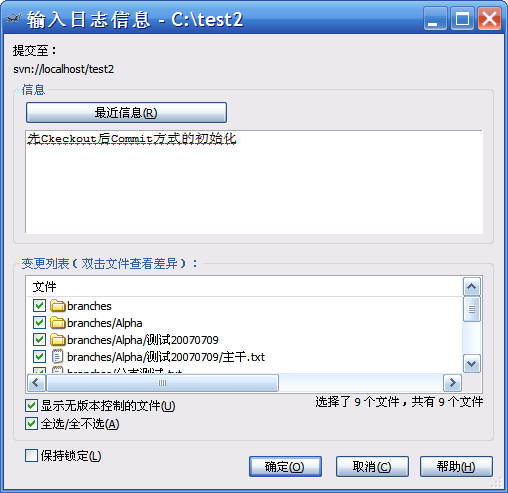
按下“确定”按钮后，TortoiseSVN就开始把整个文件夹树（包括所有文件）导入到仓库。用来做“导入Import”操作的这个文件夹的名字不会出现在仓库中，只有文件夹中的内容会出现。注意：刚才用来做“导入Import”操作的这个文件夹并没有处于版本控制下！要获取一份处于版本控制之下的[本地工作区]副本，你需要对刚导入的版本做一次“检出Checkout”操作。

### 3.2方法二：检出“Checkout”—>“提交Commit”

新建一个空文件夹作为[本地工作区]的存放文件夹（建议与项目同名），在文件夹上（或者文件夹里）单击鼠标右键后在命令菜单中选择“检出Checkout…”，出现操作窗口。注意：只能检出Checkout到一个空文件夹。

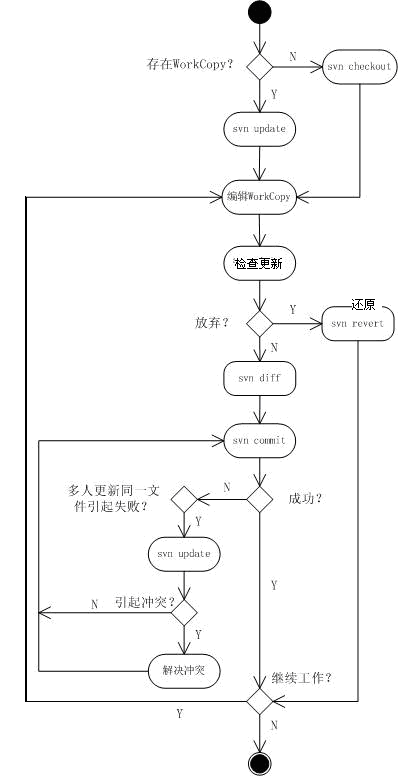


把本地硬盘上的项目文件夹中的内容（不要复制项目的根文件夹）全部复制到[本地工作区] 文件夹中，并对文件夹中进行清理操作（同“导入Import”操作前的清理操作）。然后选择[本地工作区] 文件夹单击鼠标右键跳出菜单，选择命令“提交Commit…”：



提交对话框会显示每一个有修改的文件，包括新增的，删除的，还有没有版本控制的。如果不想“提交Commit”某个有变更的文件，只需取消勾选那个文件即可。相反，如果要提交某个未受版本控制的文件，只需勾选它就可以然后“提交Commit”即可。

# 日常开发中的SVN操作



### 4.1 版本号的解释

SVN系统中的版本号准确应该叫做“修订版本号”，每当版本库接受了一个提交Commit，文件系统进入了一个新的状态，叫做一次修订（revision），每一个修订版本被赋予一个独一无二的、递增的自然数，这个自然数就是修订版本号。

受控项目中任何一个受控文件/文件夹进行了修改提交Commit操作，项目的版本号就会提升一次，没有进行修改的文件/文件夹其版本号不会发生变化。受控项目每一个装态都包含项目的所有文件/文件夹，引起该次版本号增加的即修改了的文件/文件夹保存新副本，没有发生修改的文件/文件夹保存引用，因此项目中的文件/文件夹可能与项目的版本号不同。

### 4.2更新(Update)本地工作副本



把他人做的修改融合到自己的本地副本当中，这个把改动从服务器拿到本地的过程就是“更新Update”。“更新Update”操作可以针对一个文件，或几个被选择的文件，或某个文件夹中的所有文件。选择想要进行“更新Update”操作的文件和（或）文件夹，单击鼠标右键，在菜单中选择“更新Update”， 执行窗口显示正在“更新Update”的进度。他人做的修改会合并到自己的文件中，而自己所做的修改会被保留。如果不能合并，将会出现冲突（红色文字显示），冲突的文件图标将出现*标志。

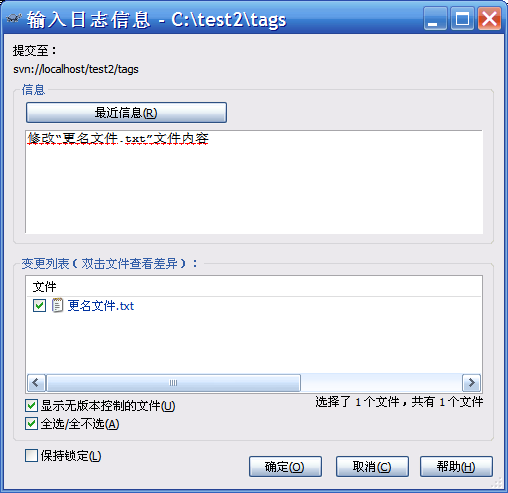
命令 “更新至版本Update to Revision...” 可以使[本地工作区]更新到自己选择的一个特定版本。

### 4.3 解决冲突

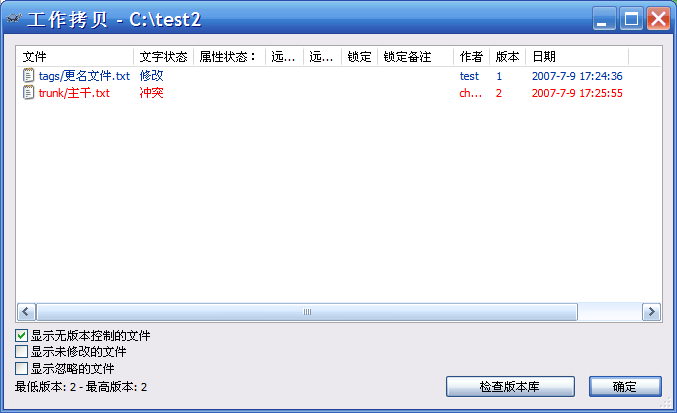
选择*的文件，点击鼠标右键菜单中选择“编辑冲突Edit Conflict”来打开合并工具或冲突编辑器，做一些必要的修改，然后保存文件。然后选择菜单中的“已解决的Resolved”命令执行，接着提交commit修改到仓库。

注意：命令“已解决的Resolved”并没有真正的解决冲突，它只不过是把“文件名称.扩展名.mine” 和 “文件名称.扩展名.r\*”（r\*中星号代表任意位数字，即版本号）删除，并允许你提交修改而已。因此，建议在有冲突发生时要先执行“编辑冲突Edit Conflict” 再执行“已解决的Resolved”；如果确信自己的修改是正确的，不需要保留他人的修改，可将文件名称.扩展名.mine去掉“.mine”后缀，并把“.r\*”后缀的文件和“*”的文件删除，然后正常“提交Commit”，版本库中将保存自己的版本。

### 4.4 提交修改到仓库

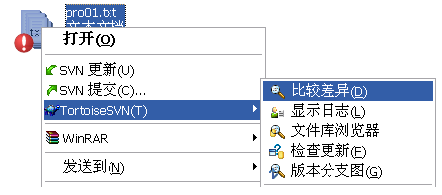


强烈建议在提交之前，应该确认[本地工作区] 是最新的。可以直接作一次“更新Update”操作，或者先“检查更新Check for Modifications”看看在本地或在服务器上哪些文件修改过。如果没有这样的操作直接提交的话，很有可能出现冲突现象。“检查更新Check for Modifications”操作后的结果界面如下：



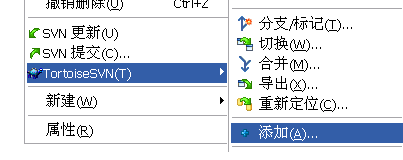
如果[本地工作区] 是最新的，并且没有冲突，就可以提交你的修改了。选中你想要提交的任意文件或文件夹，然后选择在菜单中选择“提交Commit…”在提交对话框中双击一个有修改的文件，可以启动外挂的比较工具来显示修改细节。 在按下“确定OK”按钮之后，会显示提交的进程情况。

### 4.4 比较差异(Diff)



[本地工作区] 修改后，被修改的文件出现*标志，选择 “比较差异Diff” 命令，可以查看该文件与版本仓库中文件具体差异，选择命令后出现对应的差异察看窗口。

### 4.5加入(Add)新文件/新文件夹



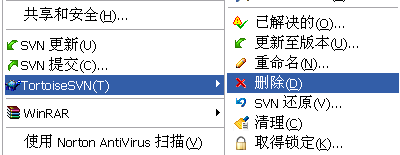
想把在开发过程中创建的新文件/文件夹加入到SVN控制之下，可以这样做：选中文件/文件夹（在新文件/文件夹所在父文件夹点击右键），然后在菜单中选择“添加Add”命令。不需要受SVN控制的文件请取消钩选。



在把文件或文件夹加入到控制之下后，这些文件或文件夹就会显示为一个Added覆盖图标*的样子，接下来必须对[本地工作区] 执行提交commit操作，以便其他团队成员能够看到这些文件或文件夹。

### 4.6删除、重命名、移动

选择要删除（或重命名）的文件/文件夹，点击鼠标右键选择相应命令：



如果使用TortoiseSVN的“删除Delete”操作删除了一个文件/文件夹，文件就已经从[本地工作区]中删除了，而被“删除Delete”的文件夹将显示为未命名覆盖图标。要恢复被“删除Delete”文件/文件夹，只需对他的上级文件夹进行“SVN还原revert”操作即可。

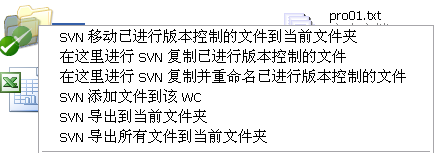
如果要在[本地工作区]内移动文件，可以使用鼠标来拖拽：

1．  选中要移动的文件或文件夹。

2．  用鼠标右键拖拽他们到[本地工作区]中一个新的文件夹。

3．  松开鼠标右键。

4．  在跳出的菜单中选择相应命令，如下图：



如果一个删除操作不是使用TortoiseSVN的“删除Delete”操作完成的，而是就像平常删除文件那样删除的。提交时，对话框窗口还是会显示这些删除的文件，并提示自己把它们从版本控制下删除。所以如果忘记使用TortoiseSVN“删除Delete”操作来删除这些文件，仍可在这里补上。

### 4.7取消改变(revert)

如果要取消对一个文件/文件夹所做的修改，只需用鼠标右键单击该文件/文件夹，选择TortoiseSVN子菜单中的“SVN还原revert”命令，在操作窗口中会显示需要还原的文件/文件夹，勾选文件/文件夹后按下“确定OK”按钮即可。注意这里所做的取消修改只能返回到该文件/文件夹上次“更新Update”后的状态。



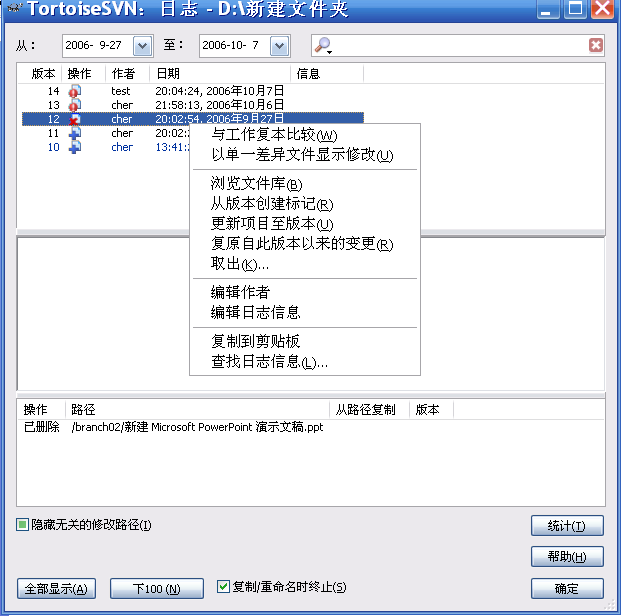
### 4.8修订日志窗口

每一次修改和提交，都应该做好日志记录。这样开发过程中就有了一个详细的记录，以后便可找出每一个修改是如何修改以及为什么这样修改的。

日志对话框会显示所有日志信息。显示分成3个部分：

1. 最上面部分显示的是所有被提交的版本列表。显示有日期时间、提交者、修订版本号和日志信息的前面一部分。用蓝色显示的行表示有某些文件被复制到了这个版本。（也许是从一个分支来的）
2. 中间部分显示的是选中版本的所有日志信息。
3. 底部显示的是选中版本所做修改的文件和文件夹列表。

还不仅仅是这些，对话框中还提供了很多的菜单命令可以使用。



### 4.9文件库浏览器

选择“文件库浏览器”菜单，就可以打开Subversion服务器上版本库的结构，并可以对版本库中的文件/文件夹进行操作，如下图：

