一个大项目在开发中可能会拆分成几个小项目，分别分去，同时共通的部分再由人做，做完后再统一合并。同时，在开发中，共通的部分修改后，其它人要保持同步。

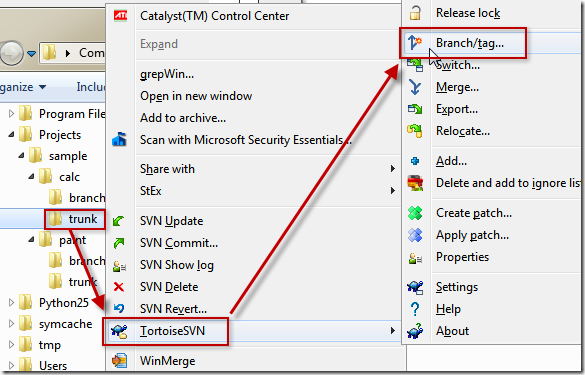
这种情况反应到SVN的分支/合并功能上，再贴切不过了。

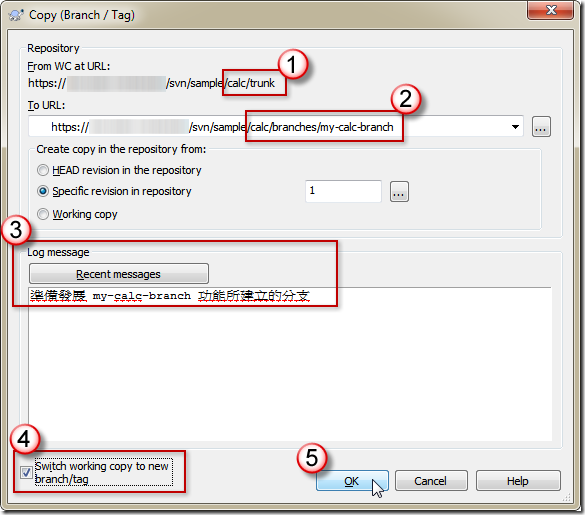
SVN可以为一个版本库中的内容(主干)建立一个分支.分支和主干完全独立，就相当于把代码再复制一份，重新添加到版本库中。但SVN提供另一个功能，就是把主干做出的修改合并到分支中，以及把分支修改的内容合并到主干中。当然，我们也可以把主干的版本库的路径切换到分支上，然后更新，来实现把分支的修改更新到主干；以及修改分支路径来同步主干的修改。但过程复杂，还容易出错。

SVN 的分支功能还提供一些合并的测试，可以在不改动版本路径的情况下完成上面的需求。

示例:

1.将一个整项目建一个分支.





建立时要注意：

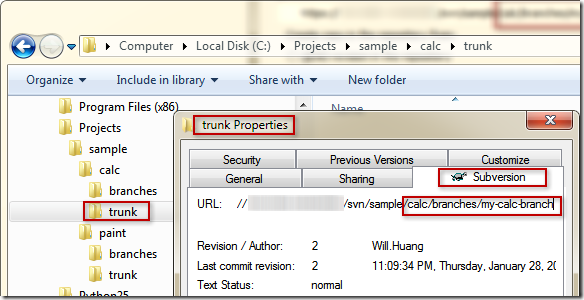
1.当前复制源,即专业术语中的 "主干(truck)"

2.分支存放的位置. 当然，分支也是在SVN版本库中.

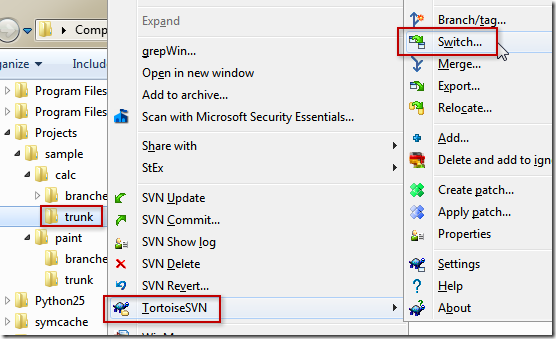
3.写上日志.这个大家应该懂的.

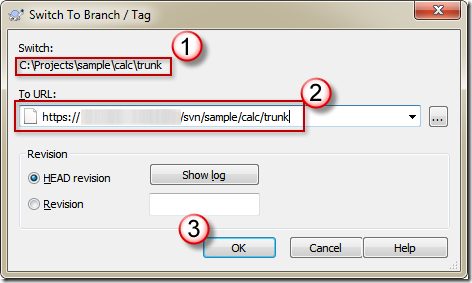
4.是否把主干的路径切换到分支.如果勾选了，建立分支后，在主干里做出的修改并提交后，更新会提交到分支上。主干的版本源内容不会变.

这时我们看一下 trunk 目录的属性，可以看到它的路径已经变成: /calc/branches/my-calc-branch 了。



为了避免产生困惑。以及失误。在建立的时候不要勾上 "切换到分支" 的选项。如果勾上了，我们还是切换回去:





注意:

1.主干的目录

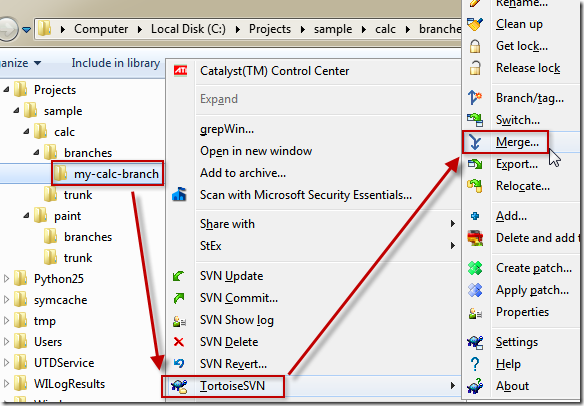
2.版本库源路径

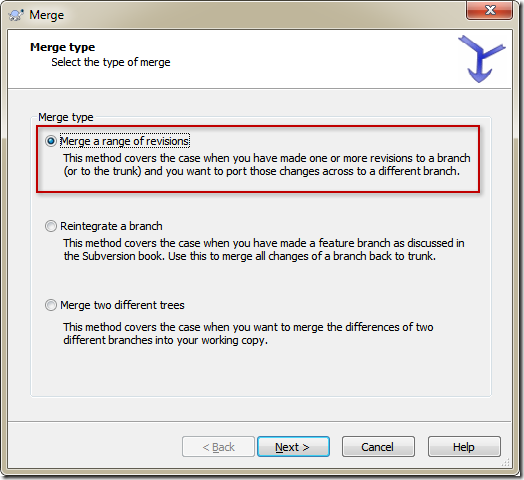
这时你便可以在 /calc/branches/my-calc-branch 分支上开发新的功能，且不会影响到其他成员开发或维护主干的内容。

2.合并主干的变更

也许过了一段时间，原本的 /calc/trunk 主干可能已经有其他成员陆续修正了一些 Bugs，但这时你的分支 /calc/branches/my-calc-branch 就可以直接套用主干 ( /calc/trunk ) 的更新，除了避免重复的工作外 ，也可以避免版本的冲突，因为多人改同样的文件可能发生冲突。

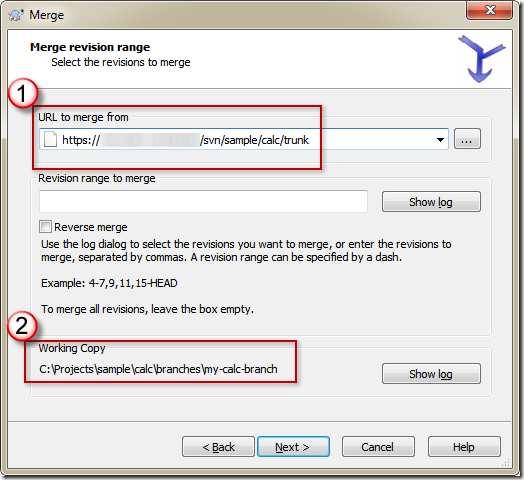
经常將 开发主干 ( /calc/trunk ) 的变更透过 svn merge 合并至 分支 ( /calc/branches/my-calc-branch ) 是一个非常好的习惯，这样才不会让你脱离主干(trunk) 过久而导致将分支 ( /calc/branches/my-calc-branch ) 合并回 主干 ( /calc/trunk ) 时发生许多冲突。





从主干 ( /calc/trunk ) 合并至分支 ( /calc/branches/my-calc-branch ) 通常选第 1 个，也就是 [Merge a range of revisions]

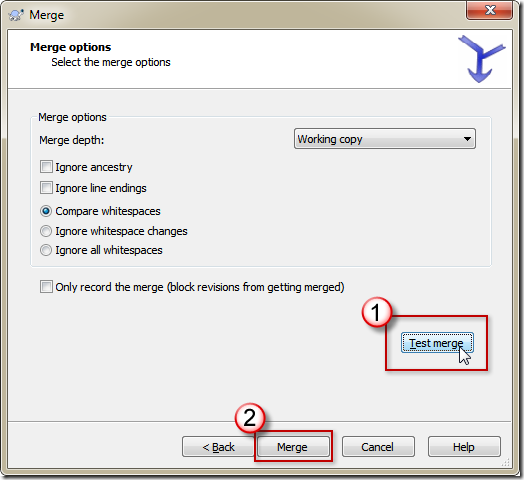
注意.我们是在分支上使用的 Merge 功能.因为是要在分支上应用主干的更新.



在 Merge 的窗口有以下注意事項：

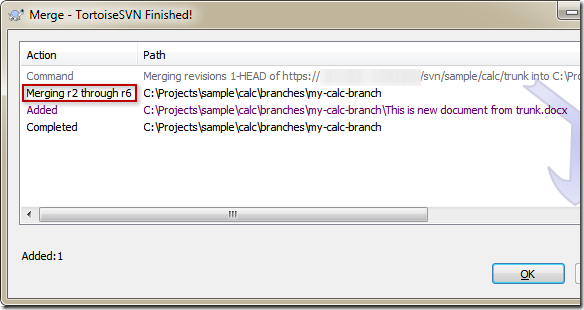
1.合并的来源，由于我们打算从主干 ( /calc/trunk ) 合并至分支 ( /calc/branches/my-calc-branch )，所以合并的來源要选 /calc/trunk 才对！

2.合并的结果会直接与目前「工作目录」(Working Copy) 做比对，并修改目前工作目录中的所有文件。因此建议在做合并之前可以将所有尚未 commit 的档案先 commit 到版本库，避免不必要的冲突事件发生。



在正式进行合并(Merge)之前，建议先执行 Test merge 看看是否会发生什么事!

若无异状则可直接按下 [Merge] 按钮进行合并动作，这时从 主干 ( /calc/trunk ) 分支出来的到目前工作目录的版本就会做个比较，然后直接套用变更到你现有的文件、目录或属性里。

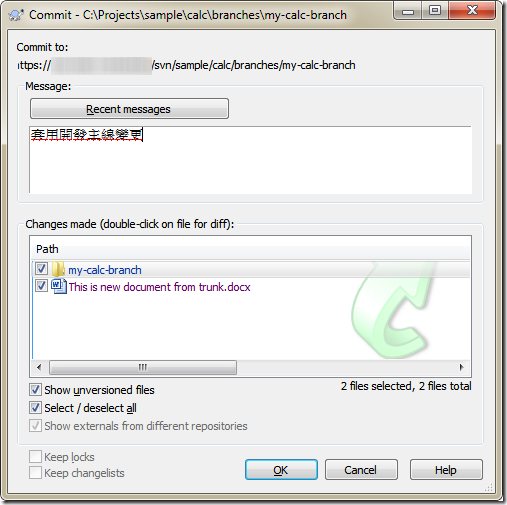


在合并之后如果没有发生冲突，不代表真的没冲突，所以必须再次对原始码做出验证后才能 commit 进版本库，建议可参考以下流程：

1.将项目进行建置(Build)

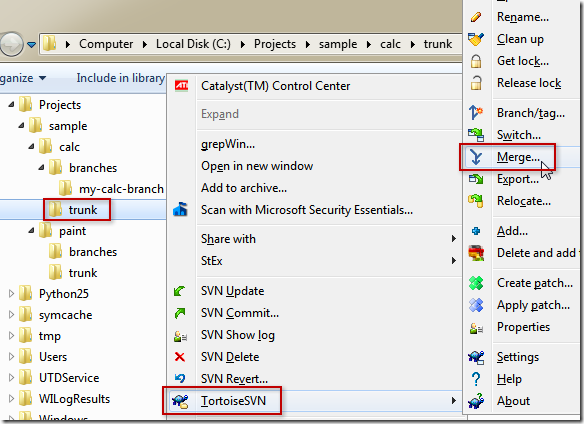
2.如果没问题再对项目进行单元测试(Unit Testing)或手动测试(Manual Testing)

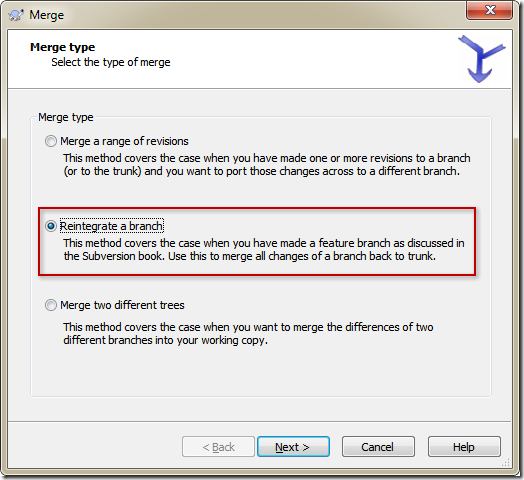
3.如果都没问题再 commit 目前合并无误的版本到版本库！



3.合并分支到主干

最后我们的 my-calc-branch 分支已经将新功能开发完成且测试无误，所以要将 分支 ( /calc/branches/my-calc-branch ) 的最终版本合并回 主干 ( /calc/trunk )，这时的手续如下：



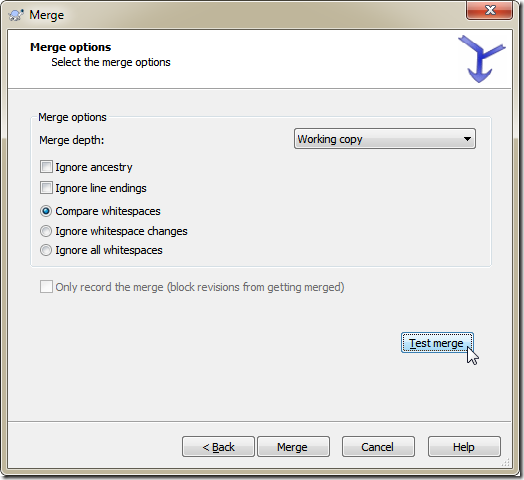


从 分支 ( /calc/branches/my-calc-branch ) 合并回 主干 ( /calc/trunk ) 通常选第 2 个，而特别选择 [Reintegrate a branch] 这个选项是很重要的，因为这有以下好处：

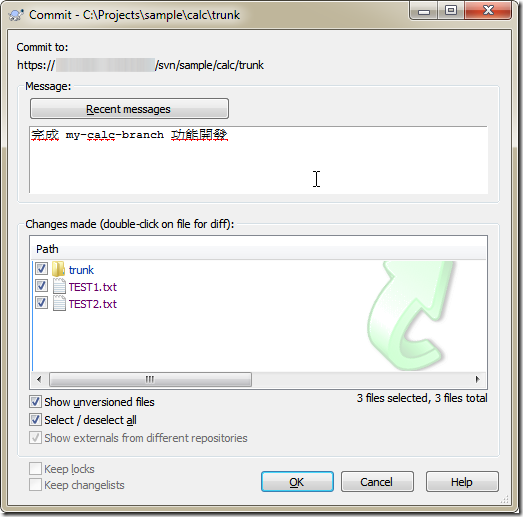
1.让 Subversion 能知道 主干 ( /calc/trunk ) 是从哪个分支、哪些版本合并进来的

2.有效节省 Subversion Repository (SVN储存库) 的空间，因为不用重复储存分支的所有变更信息

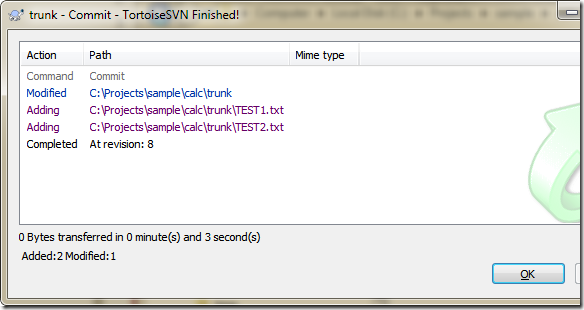
3.可以产生 Revision graph 得知项目开发的分支状况



一样可以先 测试合并(Test merge) 再正式进行 合并(Merge)



合并完后再将变更 commit 到版本库



4.删除使用完毕的分支

当 分支 ( /calc/branches/my-calc-branch ) 合并回 主干 ( /calc/trunk ) 并 commit 了之后，该分支就没用了，该分支如果未来不再更新或继续开发，Subversion 也不会继续追踪这个分支的变更 (因为之前已经 Reintegrate 过了)，建议将该分支删除。

