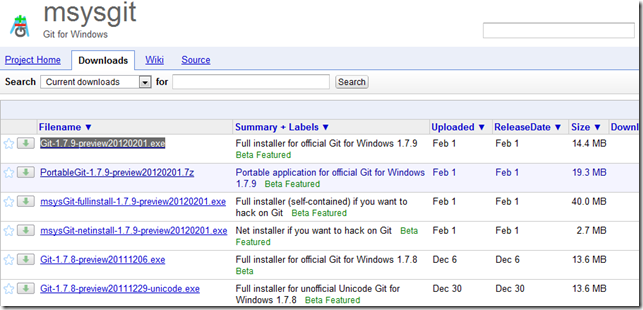
**[TortoiseGit日常使用指南](http://www.cnblogs.com/zhaojin/archive/2012/03/29/2423248.html)**

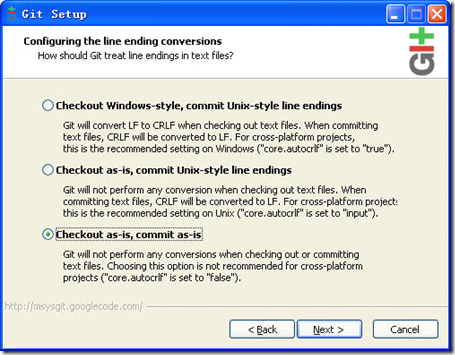
本文在介绍了软件安装和设置后, 写了TortoiseGit 常用的一些功能, 包括:  
创建新库  
添加文件及文件夹  
创建分支  
看分支情况及修改log  
比较版本差异  
合并分支  
其他操作: Stash; 忽略文件  
本文不包括:  
Git 服务器设置  
Push 版本到服务器上  
从其他机器上Pull 版本  
解决中文字符问题  
《Pro Git》和TortoiseGit 的帮助文档是两篇很不错的参考文档. 如果时间比  
较紧, 可以看看TortoiseGit 的帮助文档, 对Git 的用法, 文件存储等都会有比较好  
的了解.  
对比常用的CVS, Git 至少有以下几个优点:  
不需要连接服务器就能实现版本管理;  
一个库不管有多少层目录, 只有一个管理目录  
在commit 时, 可以将整个项目commit, 这样可以实现不需要添加Tag 就  
保存整个版本信息.  
在做Tag 时可以添加说明

**2. 安装**

(1) 安装msysgit  
到http://code.google.com/p/msysgit/下载msysgit, 目前能下载到的最新版本是  
[Git-1.7.9-preview20120201.exe](http://code.google.com/p/msysgit/downloads/detail?name=Git-1.7.9-preview20120201.exe&can=2&q=)

[](http://images.cnblogs.com/cnblogs_com/zhaojin/201203/201203291343089908.png)

安装过程中要注意, 在设置行结束转换时, 选择Checkout as-is, commit as-is,  
这样Git 就不会修改换行风格了. 其他用缺省设置即可.

[](http://images.cnblogs.com/cnblogs_com/zhaojin/201203/201203291343186867.png)

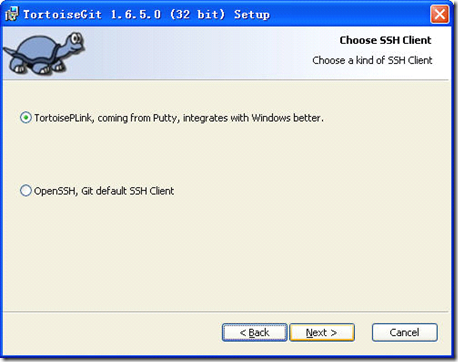
**(2) 安装TortoiseGit**

到http://code.google.com/p/tortoisegit/下载TortoiseGit, 目前的最新版本是  
[TortoiseGit-1.7.7.0-64bit.msi](http://code.google.com/p/tortoisegit/downloads/detail?name=TortoiseGit-1.7.7.0-64bit.msi&can=2&q=)

[TortoiseGit-1.7.7.0-32bit.msi](http://code.google.com/p/tortoisegit/downloads/detail?name=TortoiseGit-1.7.7.0-32bit.msi&can=2&q=)

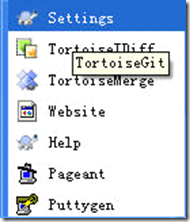
[](http://images.cnblogs.com/cnblogs_com/zhaojin/201203/201203291343276814.png)

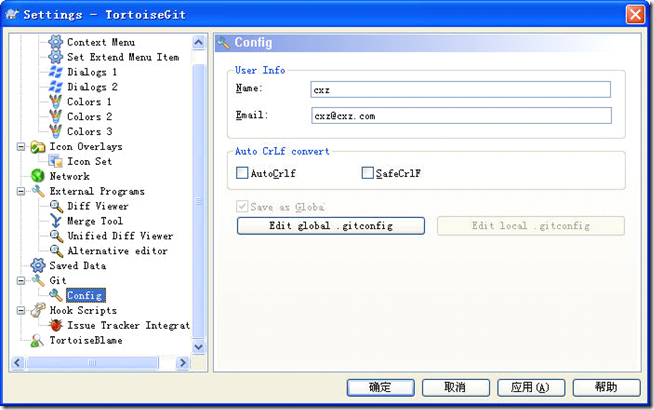
安装的过程中, 选择TortoisePLink  
这两个软件安装完成后, 就可以开始使用了.

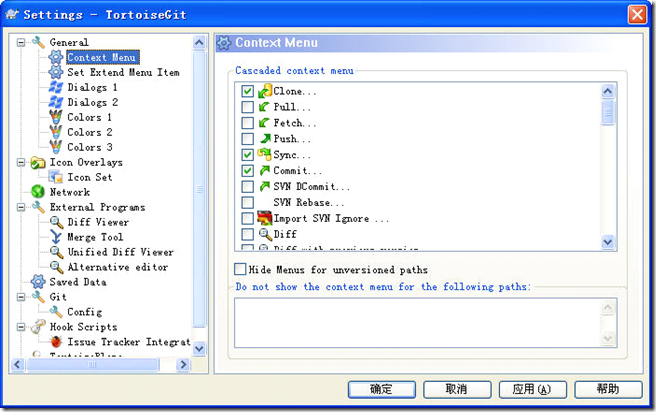
[](http://images.cnblogs.com/cnblogs_com/zhaojin/201203/20120329134331779.png)

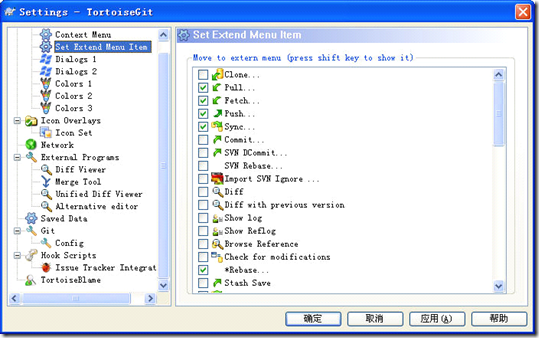
**3. 设置**

先到TortoiseGit 程序组中调用Settings 进行设置.

[](http://images.cnblogs.com/cnblogs_com/zhaojin/201203/201203291343377361.png)  
首先需要设置的是用户信息, 没有用户信息, 无法完成其他操作.

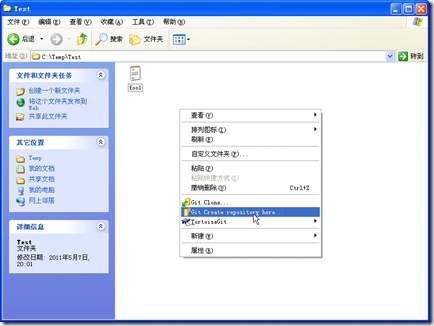
[](http://images.cnblogs.com/cnblogs_com/zhaojin/201203/201203291343417247.png)  
接着可以定制上下文菜单.

[](http://images.cnblogs.com/cnblogs_com/zhaojin/201203/201203291343511731.png)  
在定制时要注意, 在Set Extend Menu Item 时, 勾选的选项是不显示的选项.  
隐藏的菜单在文件夹中按右键时同时按下Shift 就能显示出来.

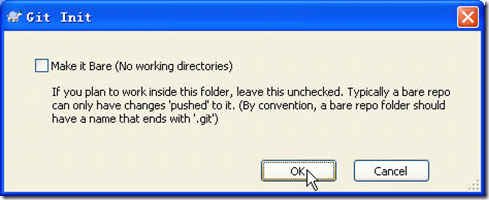
[](http://images.cnblogs.com/cnblogs_com/zhaojin/201203/20120329134409339.png)  
在External Programs 中, 还可以设置比较工具等, 我设置了WinMerge 为比较  
工具.

**4. 日常用法**

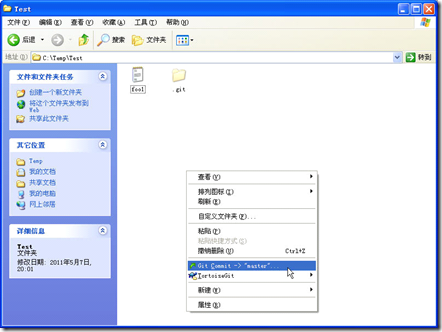
(1) 创建新库  
在文件夹中按右键, 选择Git Create repository here 就可以创建库了.

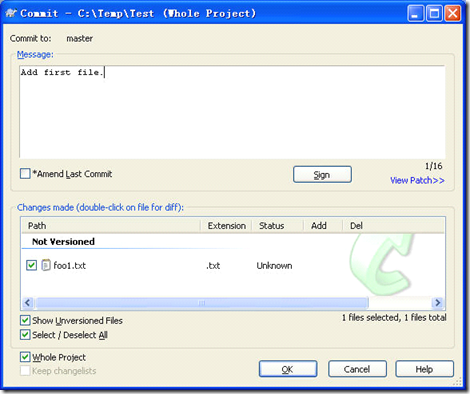
[](http://images.cnblogs.com/cnblogs_com/zhaojin/201203/201203291344218212.png)

在出现的窗口中, 不勾选选项, 直接按OK

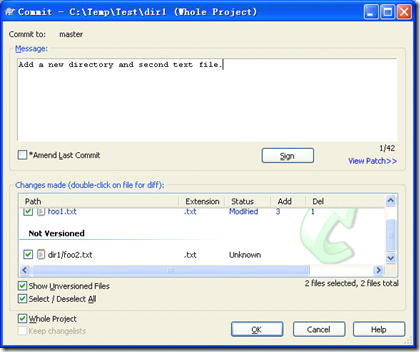
[](http://images.cnblogs.com/cnblogs_com/zhaojin/201203/201203291344274553.png)

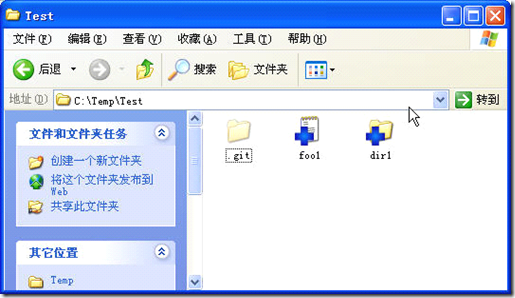
在目录中就会出现一个名为.git 的隐藏文件夹, 所有库的相关内容都会  
存在这个文件夹中. 以后不管这个项目添加多少个文件夹, 整个库只会有这  
一个管理文件夹, 这和CVS 和SVN 有较大差异.  
(2) 添加文件及文件夹

[](http://images.cnblogs.com/cnblogs_com/zhaojin/201203/201203291344349399.png)

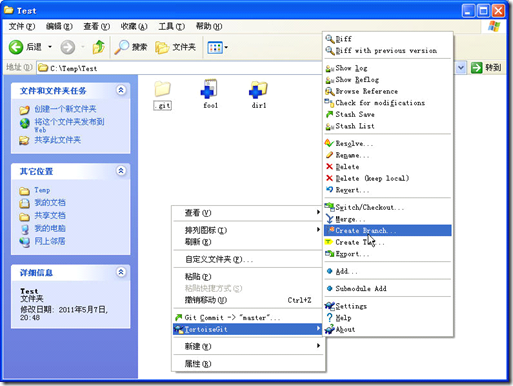
[](http://images.cnblogs.com/cnblogs_com/zhaojin/201203/201203291344418084.png)

在文件夹中按右键, 选Git Commit -> “master”…  
接着填写Message, 勾选Whole Project 选项, 这样Commit 的时候可以将整个  
项目的信息全部Commit 上去, 可以实现不需要打Tag 就能Checkout 出每次  
Commit 的内容.  
接着修改foo1.txt, 再创建一个文件夹dir1, 并且放置一个foo2.txt 在dir1 目  
录中, 再次commit 时, 就可以将dir1 和foo2.txt 一起加入了.

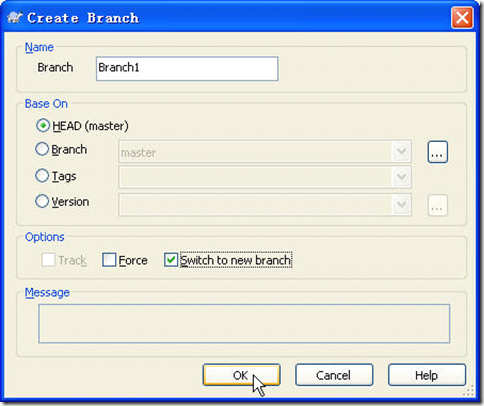
[](http://images.cnblogs.com/cnblogs_com/zhaojin/201203/201203291344485896.png)

[](http://images.cnblogs.com/cnblogs_com/zhaojin/201203/201203291345009723.png)

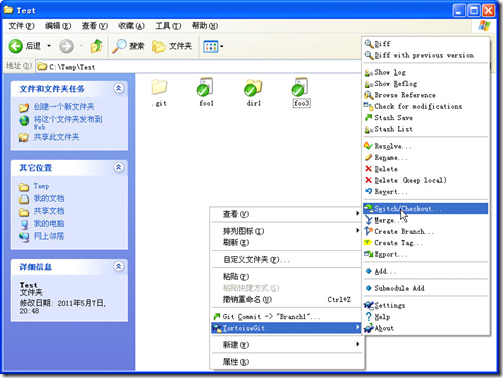
(3) 创建分支  
在目录中选择TortoiseGit 再选Create Branch…就可以创建分支了.

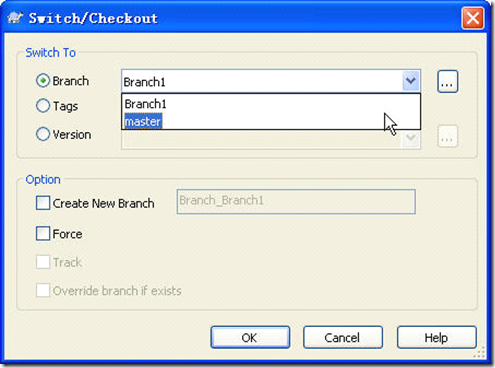
[](http://images.cnblogs.com/cnblogs_com/zhaojin/201203/201203291345105569.png)

勾选Switch to new branch, 就可以跳转到建立好的分支上.

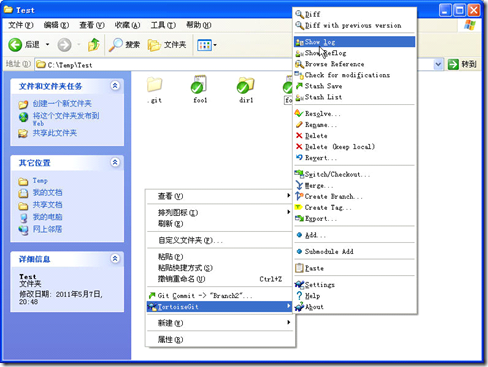
[](http://images.cnblogs.com/cnblogs_com/zhaojin/201203/201203291345167940.png)

添加一个foo3.txt 后, commit 修改.  
接着通过Switch/Checkout….可以切换回master 分支

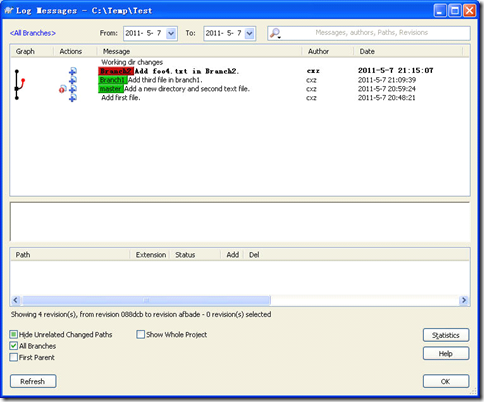
[](http://images.cnblogs.com/cnblogs_com/zhaojin/201203/201203291345264932.png)

[](http://images.cnblogs.com/cnblogs_com/zhaojin/201203/201203291345408463.png)

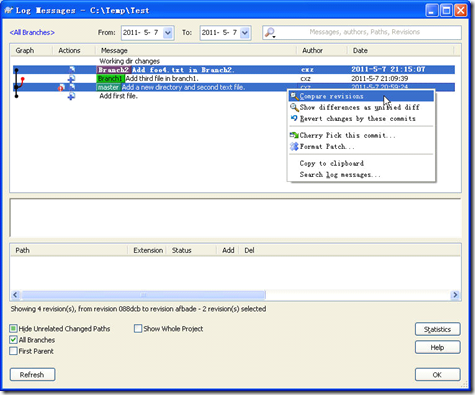
再创建一个Branch2, 在Branch2 中添加foo4.txt. 下面就可以看版本发展的  
情况了.  
(4) 看分支情况及修改log  
通过选择Show log, 可以看分支情况和修改log

[](http://images.cnblogs.com/cnblogs_com/zhaojin/201203/201203291345569254.png)

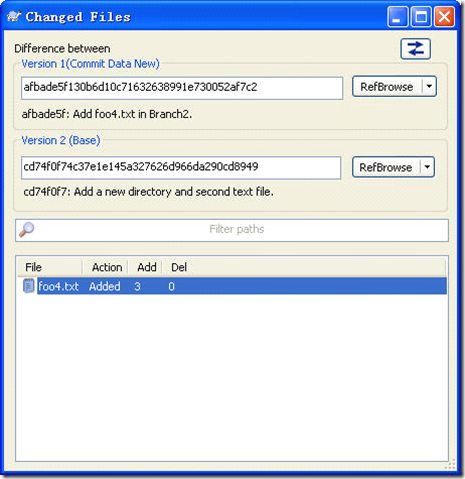
勾选All Branches 可以看到所有分支的情况.在Message 列中, 绿色的是分支, 红色的是当前工作分支.

[](http://images.cnblogs.com/cnblogs_com/zhaojin/201203/201203291346069343.png)

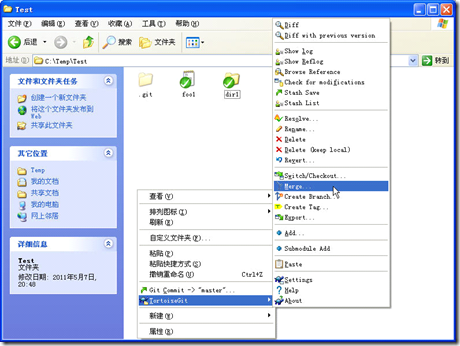
(5) 比较版本差异  
通过按Shift 和鼠标左键, 可以选中两个版本, 接着再按鼠标右键, 选中Compare revisions, 就可以比较两个revision 了.

[](http://images.cnblogs.com/cnblogs_com/zhaojin/201203/20120329134615828.png)

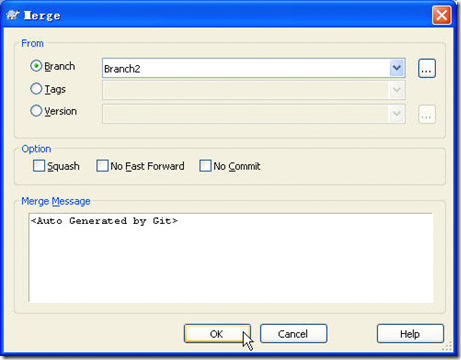
比较主版本和Branch2, 可以发现是添加了foo4.txt

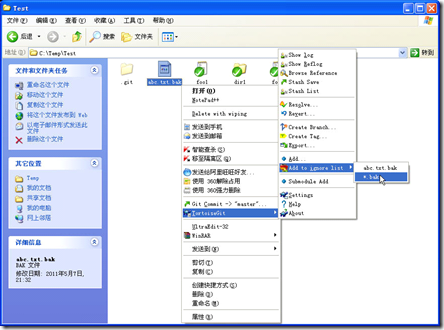
[](http://images.cnblogs.com/cnblogs_com/zhaojin/201203/201203291346228149.png)

(6) 合并分支  
首先切换到master 分支, 接着选Merge

[](http://images.cnblogs.com/cnblogs_com/zhaojin/201203/201203291346303146.png)

就可以实现将分支合并到主版本

[](http://images.cnblogs.com/cnblogs_com/zhaojin/201203/201203291346371548.png)  
(7) 其他操作  
a) Stash  
Git 提供了一个暂存修改的功能, 称为Stash, 在一些程序进行了修改, 但还不想commit 成revision 时, 就可以将这些修改Stash 起来, 等到后面需要时再将它  
们Pop 出来.  
b) 忽略文件  
一些编辑器在修改文件后会产生bak 文件, 一般不希望这些备份文件也加入库中, 可以选中一个.bak 文件, 然后选择Add to ignore list 中的\*.bak, 这样bak 文  
件以后就不会被commit 了.

[](http://images.cnblogs.com/cnblogs_com/zhaojin/201203/201203291346565946.png)

忽略的文件是通过在项目的目录中添加一个.gitignore 文件实现的, 因此也要将.gitignore 加入忽略文件列表中.