# Git使用手册

1. **服务器信息**

Gitlab服务器地址：<http://gitlab.thunisoft.com>

用户名：Cocall用户ID（英文ID）

密码：初始密码为6789@jkl

1. **Web端使用**

网页打开服务器地址后，使用用户名、密码登录。登录后完善个人信息，如已在项目中，可通过项目页签查看对应的项目版本配置库信息，同时还可以看到相关的项目数据，服务器配置等。

1. **客户端使用**

Git客户端有很多，推荐使用TortoiseGit，因为TortoiseGit与SVN的TortoiseSVN类似，操作简单容易上手，手册主要以TortoiseGit使用介绍为主。另外推荐的客户端为SourceTree。

1. **客户端下载**

Git基础环境下载地址：

[\\172.16.11.40\软件\02开发工具\Git\Git-2.8.4-64-bit.exe](file:///\\172.16.11.40\软件\02开发工具\Git\Git-2.8.4-64-bit.exe)

TortoiseGit下载地址：

[\\172.16.11.40\软件\02开发工具\Git\TortoiseGit-2.1.0.0-64bit.msi](file:///\\172.16.11.40\软件\02开发工具\Git\TortoiseGit-2.1.0.0-64bit.msi)

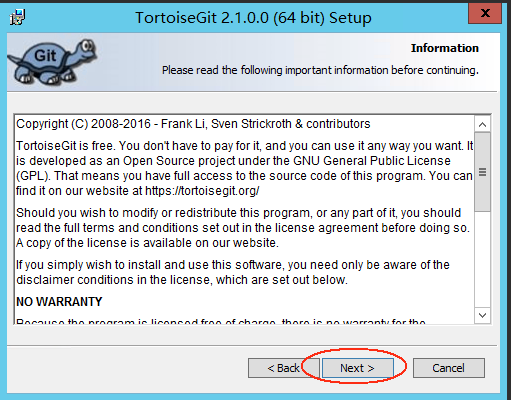
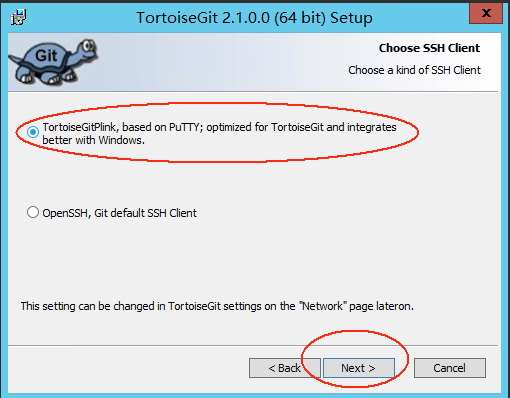
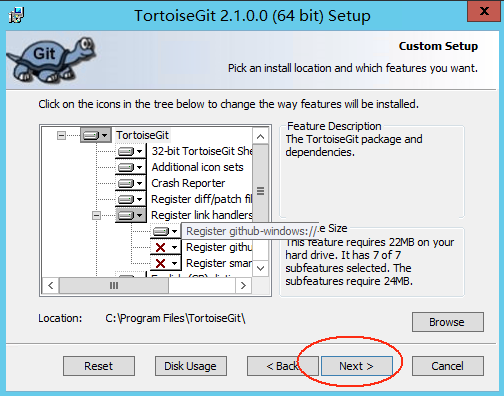
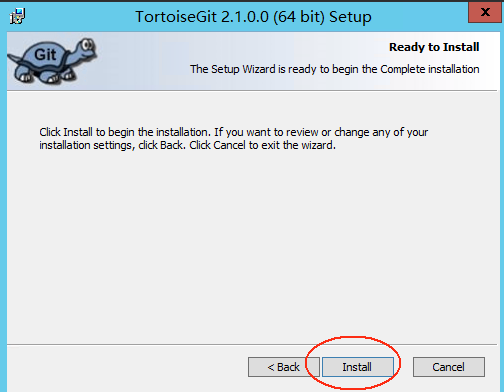
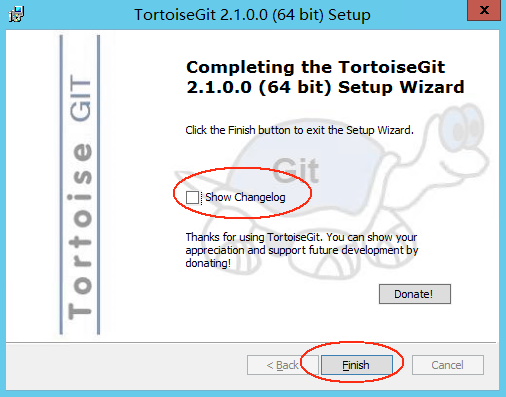
TortoiseGit汉化包下载地址：

[\\172.16.11.40\软件\02开发工具\Git\TortoiseGit-LanguagePack-2.1.0.0-64bit-zh\_CN.msi](file:///\\172.16.11.40\软件\02开发工具\Git\TortoiseGit-LanguagePack-2.1.0.0-64bit-zh_CN.msi)

1. **客户端安装**

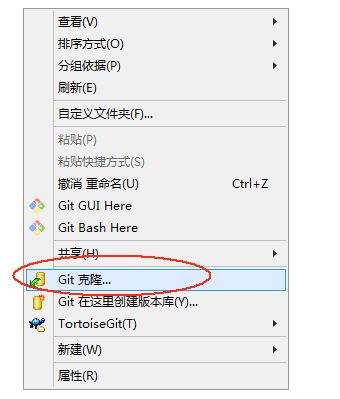
TortoiseGit使用需要依赖Git基础环境，所以首先安装Git基础环境。双击Git-2.8.4-64-bit.exe，同意安装协议后，下一步默认安装即可。

安装TortoiseGit及可选汉化步骤，如图：

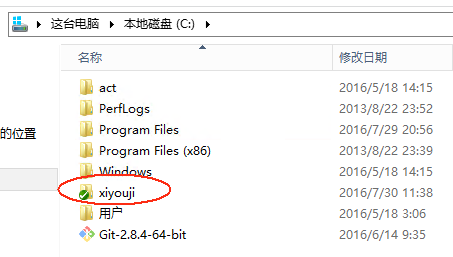
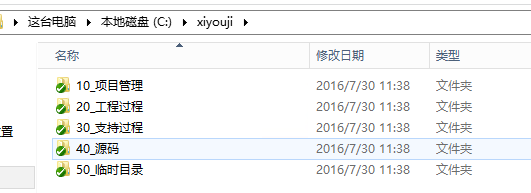
       

1. **客户端使用配置**

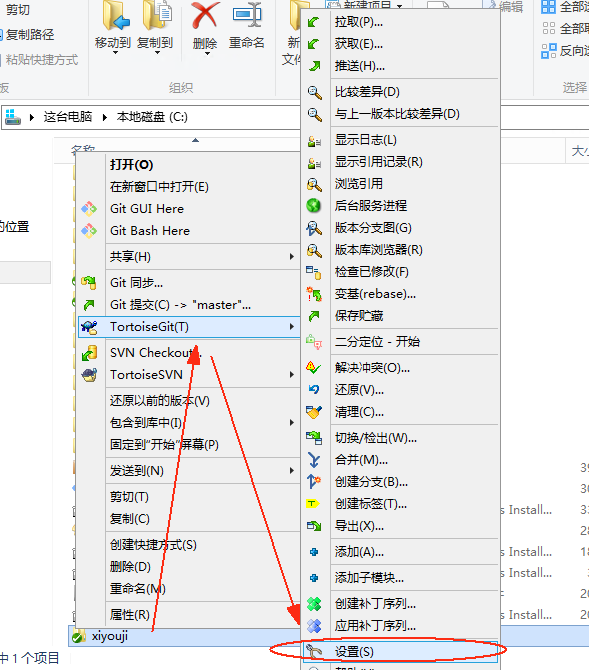
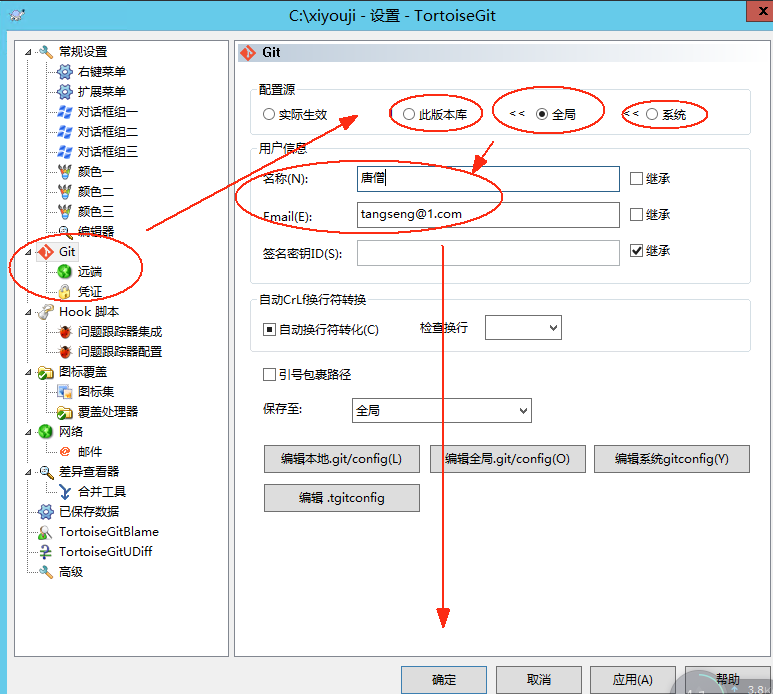
安装完成后，右键会自动生成TortoiseGit功能选项，可以通过这些功能键进行使用配置。在本地磁盘（或任何文件夹）点击右键，点击克隆，在URL中输入项目地址（注意项目git名称及链接方式），在目录中输入存放项目的位置，点击确定即完成了对项目的克隆（克隆类似SVN的检出checkout功能，但克隆是同服务器一样的整个版本库，可以离线查看、提交）。



克隆完成后，本地磁盘里的文件夹与SVN类似

下载到服务器上的项目内容后，在提交文件前，需要对客户端进行设置，告诉服务器你是谁，你的联系方式。在所选项目上点击右键，选中TortoiseGit设置，在远端中输入个人信息（请与Web端信息保持一致），如下图：

这样你在提交时，服务器会匹配邮箱，从而知道你是谁。做完以上设置就可以编写，上传文档了。

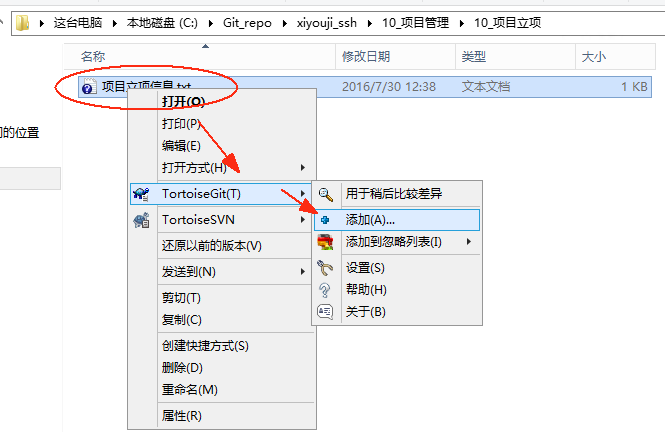
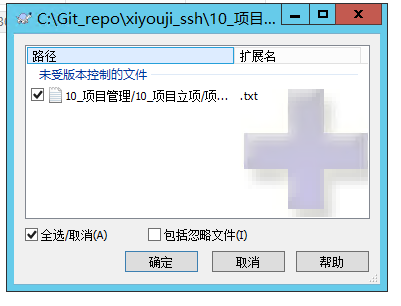
1. **使用场景**

手册主要以简单的上传、同步、查看日志、回退历史版本为主要场景，辅助带以分支、冲突的介绍，目的是使大家可以尽量简单的上手Git，对于复杂的使用场景，及更详细的场景举例，后续会有《Git中级、高级使用手册》以供参考。

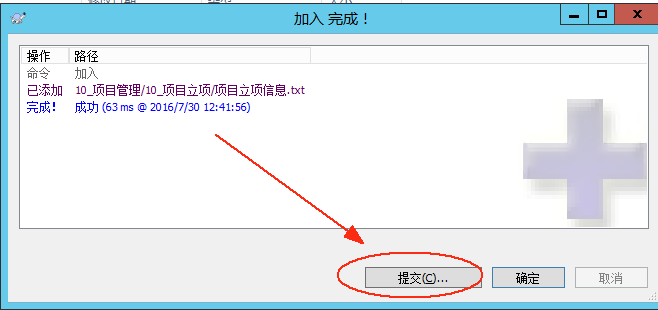
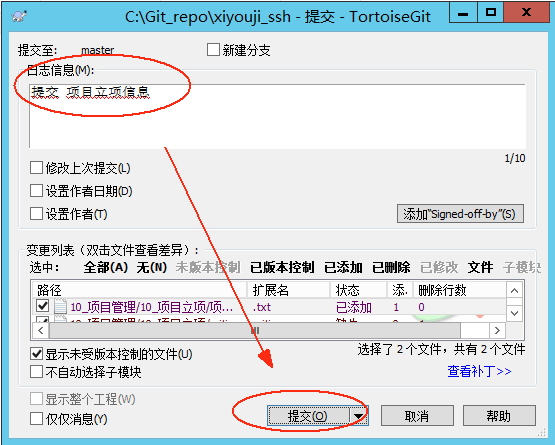
1. **上传提交文件**

Git上传提交文件比SVN上传文件多了一个步骤，这是与SVN最大的不同，因为SVN是集中管理的，不存在本地分布式的管理，所以提交时直接提交给服务器就好了；Git是分布式的，在提交给服务器前，首先要提交给自己版本库，之后再由自己的版本库提交到服务器。这个功能也是分布式最强大的地方，这里只是通过最简单、通俗的方式去介绍，实际使用并非这么简单，往往会因为场景复杂而展示出强大功能，有兴趣的可以自行百度下Git的优点及原理。

上传提交文件，在修改或新建的文件或文件夹上点击右键——添加——确定——输入提交信息——提交，提交后可选择推送把文件上传提交至服务器。如下图过程：

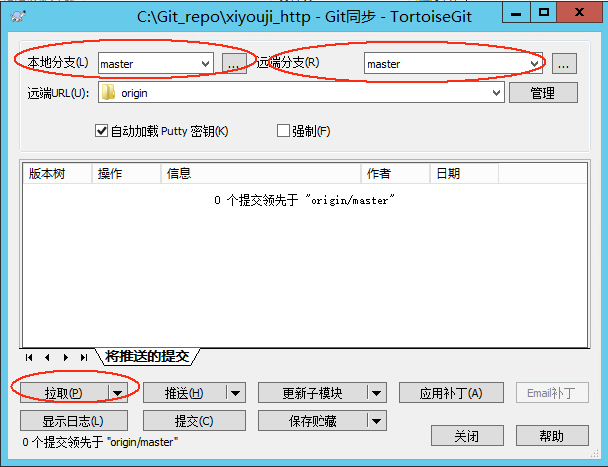
确定完成后，可以点击提交（注意，这个提交是提交到本地，并非是服务器），输入提交信息

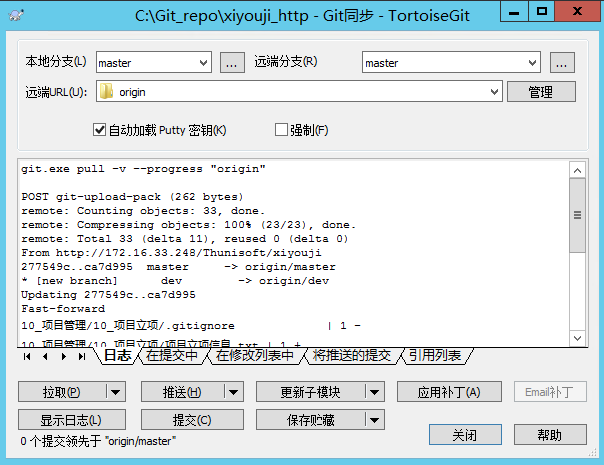
提交完成后，会显示如下页面，其中推送就是把刚刚提交的内容推送至服务器，点击推送（push）。只有推送后，才能在服务器上看到提交的信息，如只是提交，则只存储在本地存放该提交信息。注意提交、推送的区别，注意提交页面中的各个选项，一般默认即可，在复杂应用时，会用到这些选项。注：Git不允许提交空文件夹。

1. **同步更新**

在实际使用中，由于时间关系，需要同步更新本地Git项目，以便使本地副本保持与服务器一致。操作如下，在项目上点击右键——Git同步——拉取——，如下图：

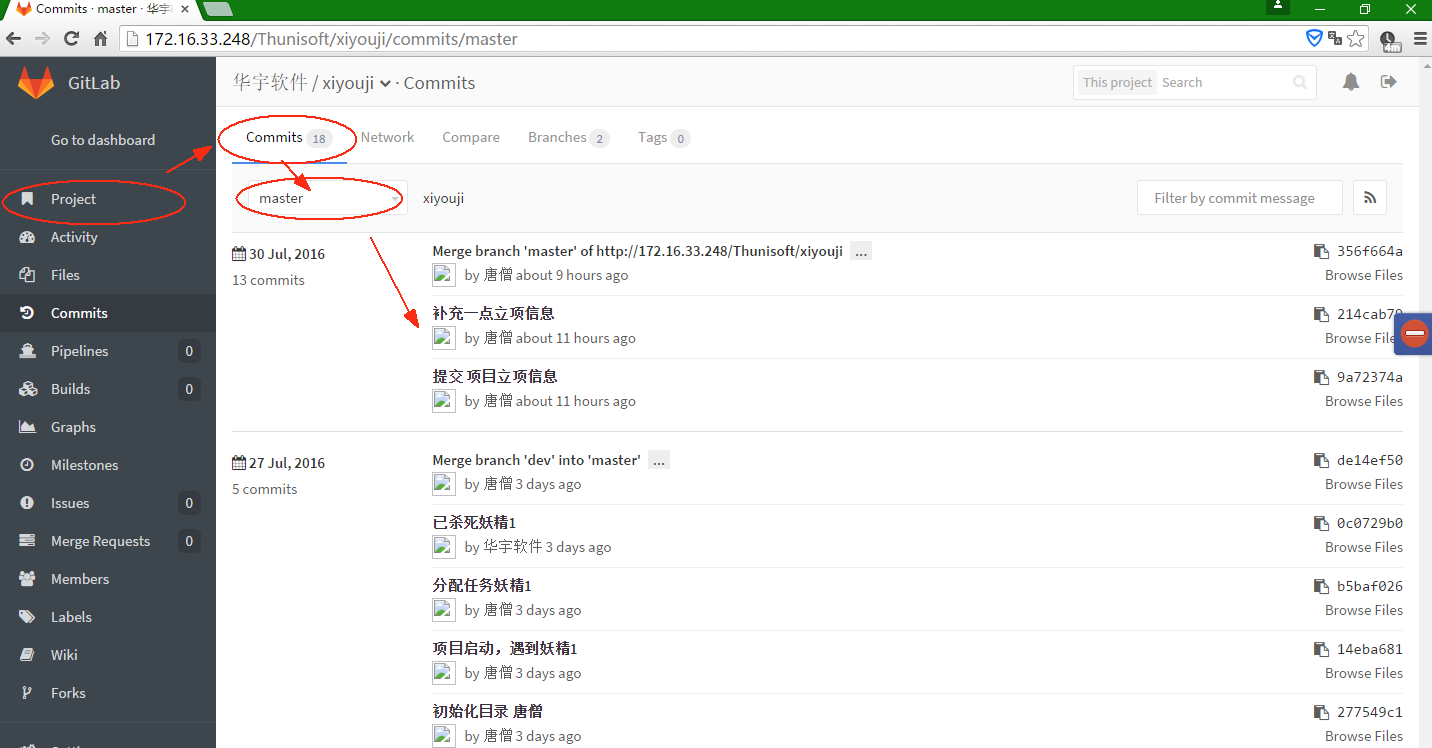


拉取后，可以通过同步窗口查看变化，可以观看日志、提交的记录、修改的列表等

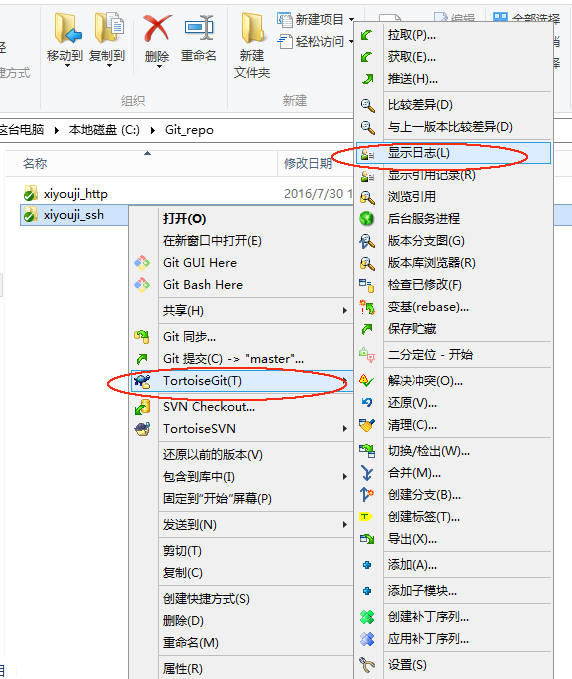
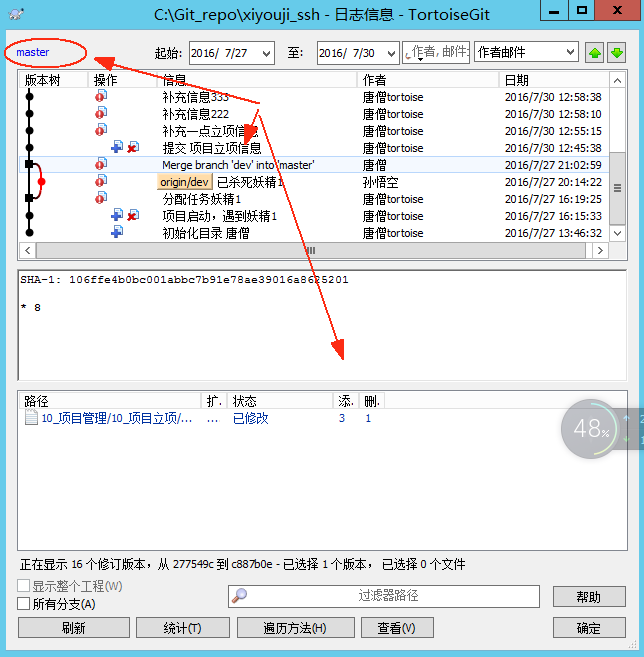


1. **查看日志**

查看日志鼓励使用服务器的Web端进行查看，这是查看服务器端日志的最好的方法，如下图，还可以点击进去查看详细内容。



客户端的日志查询可分为查询服务器日志、本地日志，因为Git是分布式的原因，本地也存在版本库，所以利用客户端去查询日志时需注意区分是服务器日志还是本地日志，以及所处在的分支。如下图:

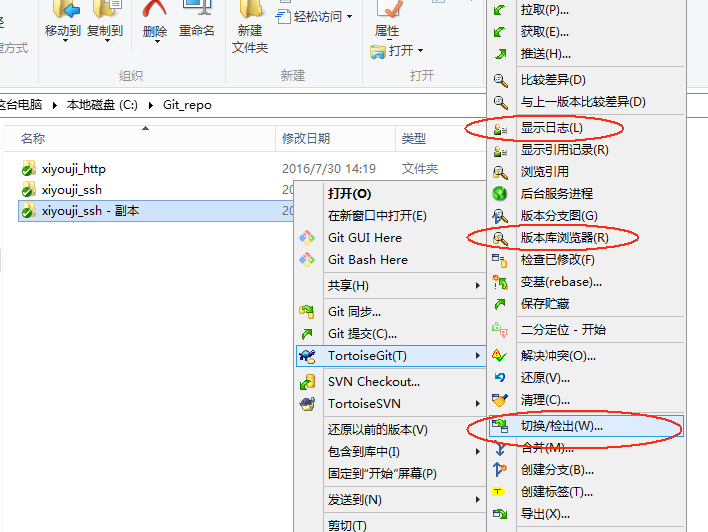
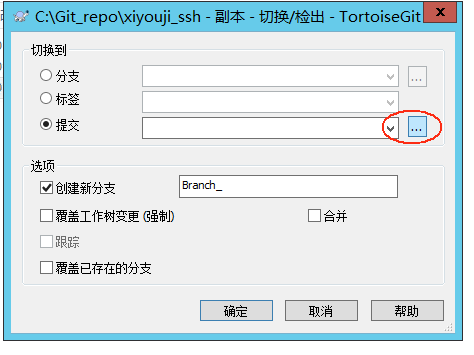
 

1. **查看、回退历史版本**

在使用过程中，会出现要回退到历史版本的情况，Git的回退要比SVN强大，因为Git的回退可以实现本地副本的轻松切换，回退历史版本有三种方法：

* 工作目录下查看日志，选中对应分支及对应历史版本，通过右键点击浏览版本库，或通过切换检出改历史版本
* 通过版本库浏览器，指定对应分支及历史版本，点击切换或检出该历史版本
* 直接点击切换／检出功能项，在弹出对话框里选中切换到提交页面，在提交后选择该分支及该历史版本

其实，3种方法的本质都是建立该历史版本的分支，并使工作区切换到该分支，如图：

1. **分支、冲突介绍**

分支功能是Git最好用的功能之一，因为Git有本地分支的概念，所以也要比SVN分支要很多，Git的分支能极大的满足开发人员的使用需求，特别是在实验、多线开发中发挥重要作用。分支最通俗的说法就是一个环境的副本，省去了人为的文件或文件夹复制。

在实际使用中，还会产生冲突，如两个人同时编辑了同一基础的文件，在第一个人提交该文件后，第二个提交人会产生冲突，需要解决该冲突之后再能提交。对于非文本文件的冲突，如word，需要人为更新、同步服务器的最新版本，在本地在新基础上解决冲突，再进行提交。对于文本文件类的冲突，可以使用Git的编辑冲突进行解决，并提交。发成冲突时，可暂存本地副本，以便进行解决冲突使用，具体操作方法参考中级、高级使用手册中的冲突及解决。