Mac 环境下svn的安装和使用

[TOC]

2017年06月19日 16:50:53 doubleface999 阅读数：63975
标签： macsvncornerstone 更多
个人分类： 杂项

在Windows环境中，我们一般使用TortoiseSVN来搭建svn环境。在Mac环境下，由于Mac自带了svn的服务器端和客户端功能，所以我们可以在不装任何第三方软件的前提下使用svn功能，不过还需做一下简单的配置。

我们首先来看下，如何在Mac环境下搭建svn服务器端环境。

一、创建代码仓库，用来存储客户端所上传的代码

我先在/User/apple目录下新建一个svn目录，以后可以在svn目录下创建多个仓库目录

打开终端，创建一个mycode仓库，输入指令：svnadmin create /Users/apple/svn/mycode

指令执行成功后，会发现硬盘上多了个/Users/apple/svn/mycode目录，目录结构如下：

注：这地方出现路径的错误可以通过

输入sudo xcode-select -switch /Applications/Xcode.app/Contents/Developer命令

password是你的登录密码。

二、配置svn的用户权限

主要是修改/svn/mycode/conf目录下的三个文件

1.打开svnserve.conf，将下列配置项前面的#和空格都去掉

C# code

?

```
`#`` anon-access = read``#`` auth-access = write` `#`` password-db = passwd` `#`` au
```

anon-access = read代表匿名访问的时候是只读的，若改为anon-access = none代表禁止匿名访问，需要帐号密码才能访问

2.打开passwd，在[users]下面添加帐号和密码

[users]

mj=123

jj=456

帐号是mj，密码是123

3.打开authz，配置用户组和权限

我们可以将在passwd里添加的用户分配到不同的用户组里，以后的话，就可以对不同用户组设置不同的权限，没有必要对每个用户进行单独设置权限。

在[groups]下面添加组名和用户名，多个用户之间用逗号(,)隔开

[groups]

topgroup=mj,jj

说明mj和jj都是属于topgroup这个组的，接下来再进行权限配置。

使用[/]代表svn服务器中的所有资源库

[/]

@topgroup=rw上面的配置说明topgroup这个组中的所有用户对所有资源库都有读写(rw)权限，组名前面要用@

如果是用户名，不用加@，比如mj这个用户有读写权限

[/]

mj=rw

至于其他精细的权限控制，可以参考authz文件中的其他内容

4.启动svn服务器

前面配置了这么多，最关键还是看能否正常启动服务器，若启动不来，前面做再多工作也是徒劳。

在终端输入下列指令：svnserve -d -r /Users/apple/svn

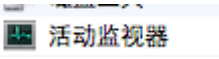
或者输入：svnserve -d -r /Users/apple/svn/mycode

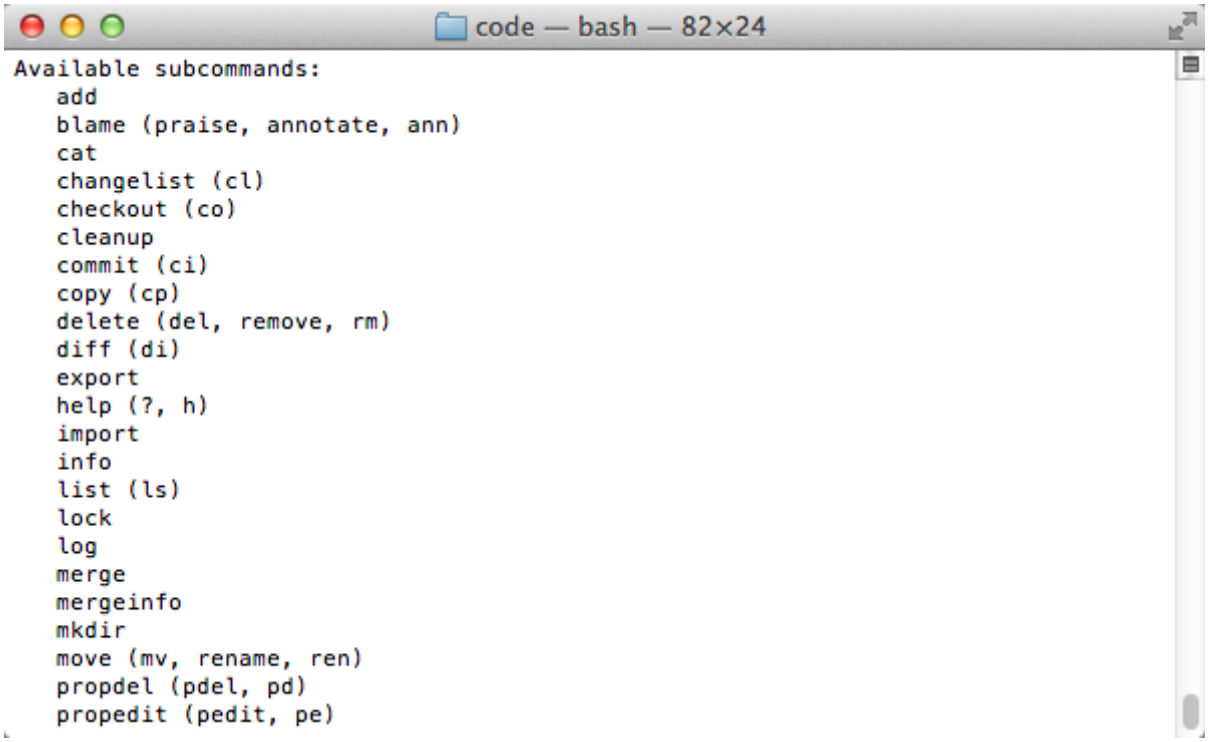
没有任何提示就说明启动成功了

5.关闭svn服务器

如果你想要关闭svn服务器，最有效的办法是打开实用工具里面的

“活动监视器”





img

这里列出一大堆svn指令，后面括号中的内容的一般代表着指令的简称，比如我们可以用svn ci代替svn commit，用svn co代替svn checkout

注：首次配置完先关闭svn服务器再进行数据的上传和下载操作。

综合上述，我们就可以轻松搭建svn服务器环境了

三、使用svn客户端功能

1.从本地导入代码到服务器(第一次初始化导入)

在终端中输入

svn import /Users/apple/Documents/eclipse_workspace/weibo svn://localhost/mycode/weibo --username=mj --password=123 -m "初始化导入"

我解释下指令的意思：将/Users/

apple

/Documents/eclipse_workspace/weibo中的所有内容，上传到服务器mycode仓库的weibo目录下，后面双引号中的"初始化导入"是注释

注：apple是用户名

2.从服务器端下载代码到客户端本地

在终端中输入svn checkout svn://localhost/mycode --username=mj --password=123 /Users/apple/Documents/code

我解释下指令的意思：将服务器中mycode仓库的内容下载到/Users/apple/Documents/code目录中

注：localhost（本地服务器IP地址）可以替换成你本地服务器的IP地址。当你和别人同用一个svn时，你可以输入你要进行数据请求的服务器的IP地址。

3.提交更改过的代码到服务器

在步骤2中已经将服务器端的代码都下载到/Users/apple/Documents/code目录中，现在修改下里面的一些代码，然后提交这些修改到服务器

1> 打开终端，先定位到/Users/apple/Documents/code目录，输入：cd /Users/apple/Documents/code

2> 输入提交指令：svn commit -m "修改了main.m文件"

这个指令会将/Users/apple/Documents/code下的所有修改都同步到服务器端，假如这次我只修改了main.文件

可以看到终端的打印信息：

Sending weibo/weibo/main.m

Transmitting file data .

Committed revision 2.

4.更新服务器端的代码到客户端

这个应该是最简单的指令了，在终端中定位到客户端代码目录后，比如上面的/Users/apple/Documents/code目录，然后再输入指令：svn update

5.至于svn的其他用法，可以在终端输入：svn help

注：1、系统自带的svn无法支持本地删除文件的更新，即当本地删除文件后去更新服务器的文件时，删除的文件又从服务器上的文件夹中下载到你的本地文件夹中。但是它是支持文件修改后的更新操作的。

2、记得写注释，否则系统不会识别命令的。

mac svn 删除.svn隐藏文件的命令

打开终端,进到所在的目录,然后出入一下代码

find . -name ".svn" | xargs rm -Rf

1、将文件checkout到本地目录

svn checkout path（path是服务器上的目录）
例如：svn checkout svn://192.168.1.1/pro/domain
简写：svn co

2、往版本库中添加新的文件

svn add file
例如：svn add test.PHP(添加test.php)
svn add *.php(添加当前目录下所有的php文件)

svn add xxx@2x.png 文件时， 正常命令 svn add xxx@2x.png 会报 xxx not found

需用 svn add xxx@2x.png@ 来添加，也就是图片名字后面再添加一个@ 符号，

这是因为 svn 命令最后需要用@符号来指定一个版本导致的

遇到 xxx@2x.png文件时，如果用svn命令行添加到 版本库的话，只能手动一个一个添加，不能批量添加

3、将改动的文件提交到版本库

svn commit -m “LogMessage“ [-N][--no-unlock] PATH(如果选择了保持锁，就使用–no-unlock开关)
例如：svn commit -m “add test file for my test“ test.php
简写：svn ci

4、加锁/解锁

svn lock -m “LockMessage“ [--force] PATH
例如：svn lock -m “lock test file“ test.php
svn unlock PATH

5、更新到某个版本

svn update -r m path
例如：
svn update如果后面没有目录，默认将当前目录以及子目录下的所有文件都更新到最新版本。
svn update -r 200 test.php(将版本库中的文件test.php还原到版本200)
svn update test.php(更新，于版本库同步。如果在提交的时候提示过期的话，是因为冲突，需要先update，修改文件，然后清除svn resolved，最后再提交commit)
简写：svn up

6、查看文件或者目录状态

1）svn status path（目录下的文件和子目录的状态，正常状态不显示）
【?：不在svn的控制中；M：内容被修改；C：发生冲突；A：预定加入到版本库；K：被锁定】
2）svn status -v path(显示文件和子目录状态)
第一列保持相同，第二列显示工作版本号，第三和第四列显示最后一次修改的版本号和修改人。
注：svn status、svn diff和 svn revert这三条命令在没有网络的情况下也可以执行的，原因是svn在本地的.svn中保留了本地版本的原始拷贝。
简写：svn st

7、删除文件

svn delete path -m “delete test fle“
例如：svn delete svn://192.168.1.1/pro/domain/test.php -m “delete test file”
或者直接svn delete test.php 然后再svn ci -m ‘delete test file‘，推荐使用这种
简写：svn (del, remove, rm)

8、查看日志

svn log path
例如：svn log test.php 显示这个文件的所有修改记录，及其版本号的变化

9、查看文件详细信息

svn info path
例如：svn info test.php

10、比较差异

svn diff path(将修改的文件与基础版本比较)
例如：svn diff test.php
svn diff -r m:n path(对版本m和版本n比较差异)
例如：svn diff -r 200:201 test.php
简写：svn di

11、将两个版本之间的差异合并到当前文件

svn merge -r m:n path
例如：svn merge -r 200:205 test.php（将版本200与205之间的差异合并到当前文件，但是一般都会产生冲突，需要处理一下）

12、SVN 帮助

svn help
svn help ci

13、版本库下的文件和目录列表

svn list path
显示path目录下的所有属于版本库的文件和目录
简写：svn ls

14、创建纳入版本控制下的新目录

svn mkdir: 创建纳入版本控制下的新目录。
用法: 1、mkdir PATH…
2、mkdir URL…
创建版本控制的目录。
1、每一个以工作副本 PATH 指定的目录，都会创建在本地端，并且加入新增
调度，以待下一次的提交。
2、每个以URL指定的目录，都会透过立即提交于仓库中创建。
在这两个情况下，所有的中间目录都必须事先存在。

15、恢复本地修改

svn revert: 恢复原始未改变的工作副本文件 (恢复大部份的本地修改)。revert:
用法: revert PATH…
注意: 本子命令不会存取网络，并且会解除冲突的状况。但是它不会恢复
被删除的目录

16、代码库URL变更

svn switch (sw): 更新工作副本至不同的URL。
用法: 1、switch URL [PATH]
2、switch –relocate FROM TO [PATH...]
1、更新你的工作副本，映射到一个新的URL，其行为跟“svn update”很像，也会将
服务器上文件与本地文件合并。这是将工作副本对应到同一仓库中某个分支或者标记的
方法。
2、改写工作副本的URL元数据，以反映单纯的URL上的改变。当仓库的根URL变动
(比如方案名或是主机名称变动)，但是工作副本仍旧对映到同一仓库的同一目录时使用
这个命令更新工作副本与仓库的对应关系。

17、解决冲突

svn resolved: 移除工作副本的目录或文件的“冲突”状态。
用法: resolved PATH…
注意: 本子命令不会依语法来解决冲突或是移除冲突标记；它只是移除冲突的
相关文件，然后让 PATH 可以再次提交。

18、输出指定文件或URL的内容。

svn cat 目标[@版本]…如果指定了版本，将从指定的版本开始查找。
svn cat -r PREV filename > filename (PREV 是上一版本,也可以写具体版本号,这样输出结果是可以提交的)

19、配置忽略文件 vi ~/.subversion/config

找到 global-ignores 一行，去掉注释，编辑成

global-ignores = build *~.nib *.so *.pbxuser *.mode *.perspective
找到 enable-auto-props = yes 把注释去掉，在[auto-props] Section声明以下文本文件
.mode = svn:mime-type=text/X-xcode
*.pbxuser = svn:mime-type=text/X-xcode
.perspective = svn:mime-type=text/X-xcode
*.pbxproj = svn:mime-type=text/X-xcode

==========

svn 命令共同的选项

--targets list 读取list并将其解释为一个将要操作的参数列表

--non-recurisive, –N 只操作单个目录， 不处理子目录

--verbose, –v 打印额外的信息

--quiet, –q 打印的信息尽可能少

--username, name 指定在连接授权时使用的用户名

--password, pawd 指定要使用的密码

--no-auth-cache 不要缓存身份令牌

--non-interactive 不要提示输入额外的信息

--config-dir dir 从dir读取用户配置

--editor-cm cmd 使用cmd作为日志消息的编辑器

svn add

把文件及目录的名称添加给版本控制系统。他们会在下次提交时被添加到项目仓库

svn add path

--auto-props 在添加他们的时候自动设置文件的属性

--no-auto-props 禁用自动属性设置

svn blame

显示文件每行的版本及作者信息

--revision, –r rev 如果指定的rev是单个版本，显示该版本作者信息。如果是范围rev1:rev2, 显示rev2版本作者的信息，但只检查版本到rev1.

svn cat

输出指定文件或者URL的内容

svn cat target…

--revision, –r rev

svn checkout

从项目仓库牵出一个工作拷贝

svn checkout url…path

如果没有指定path,签出的本地目录名使用URL的base name.

svn cleanup

清理工作拷贝，移除锁，完成未完成的操作，等等。

svn cleanup path…

svn commit path

把改动从你的工作拷贝发送到项目仓库

--message, –m msg 使用msg作为提交日志消息。

--file, –F file 使用file的内容作为提交日志消息。

--no-unlock 不要在提交的时候释放锁。

svn copy

在工作拷贝或者项目仓库中制造包括历史在内的复本

svn copy src dest

src和dest可以是工作拷贝(WC)的路径或者URL.

src dest 效果……

WC WC 拷贝并添加

WC URL 立即提交WC的拷贝到URL

URL WC 签出URL到WC, 添加

URL URL 完全服务器端拷贝；用于制作分支和打标签

--revision, –r rev要拷贝的src的版本。只在src是项目仓库的URL时才有意义。

svn delete target

从项目仓库删除文件或者目录。如果target是工作拷贝中的文件或者目录，它被从工作拷贝中移除并且预计在下次提交时删除掉。如果target是项目仓库URL,通过一次立即的提交从项目仓库中删除。

--message, –m msg

--file, –F file

svn diff

显示两个路径之间的差异

svn diff –r rev1:rev2 target…

svn diff oldurl newurl

svn export

创建一个无版本记录的拷贝.

svn export –r rev URL path

从项目仓库的指定URL导出一个干净的目录树到path中，如果指定了rev参数，导出rev版本的，否则到处最新版本。

svn import

提交一个无版本的文件或者树到项目仓库

svn import path URL

svn info

显示文件或者目录的信息。

svn list

列出项目仓库中的目录条数。

svn lock

锁住文件让其它用户不能提交改动。

svn lock target

--message, –m msg 使用msg作为锁信息消息

--force 强制加锁成功，通过从其他用户或者工作拷贝把锁给偷过来。

svn log

显示一些版本或者文件的日志消息.

--stop-on-copy 在遍历历史的时候不要穿越拷贝（对于查找分支的起点很有用）

svn merge

把两个来源的差异应用给工作拷贝路径。

svn merge –r rev1:rev2 source wcpath

svn mkdir

创建版本控制下的新目录

svn mkdir target

svn move src dest

移动或者重命名工作拷贝或者项目仓库中文件或者目录。

--revision, –r rev使用版本rev作为源来执行这次移动。

svn propdel

删除文件或者目录的属性

svn propdel propname path…

svn propedit

编辑文件或者目录的属性

svn propedit propname path…

svn propget

打印文件或者目录的属性值

svn propget propname path…

--strict 禁用额外的换行和其它的美化措施（在把二进制属性重定向到文件时会有用处)

svn proplist

列出文件或者目录的所有属性

--verbose

--recursive

--revision, –r rev 列出path在版本rev定义的属性

svn propset(pset, ps)

svn propset propname propval path…

--file, –F file 读取file的内容，使用它作为属性值.

--recursive

--encoding enc 把值作为用enc编码的字符集

svn resolved

移除工作拷贝文件或者目录的冲突状态

--recursive

svn revert

恢复工作拷贝的文件（撤销最新的本地修改）

svn revert path 这个命令不需要网络连接

--recursive

svn status

打印工作拷贝中文件或者目录的状态

svn status path…

--show-updates, –u 联系服务器显示更新信息

--no-ignore 忽视默认设置和svn:ignore属性设置的忽略项

--non-recursive, –N

--verbose, –v

svn switch

把工作拷贝转向到其他的URL

svn switch URL path

更新工作拷贝让其使用项目仓库的新URL.这个行为类似svn update 而且是一种把工作拷贝转向到同一项目仓库中的分支或者标签的办法。

--revision, –r rev 转向到版本rev

--non-recursive, –N

--diff3-cm 使用cmd作为合并命令

svn unlock

解开工作拷贝文件或者项目仓库URL的锁。

svn unlock target…

--force 砸坏现有对target的锁，甚至它不是被当前工作拷贝所拥有的。

svn update

把改动从项目仓库带到工作拷贝来。

svn update path…

--revision, –r rev 更新到版本rev

--non-recrusive, –N

--diff3-cmd

==========

svn 出错信息总汇

-svn 出错信息总汇 . Subversion 错误信息一览表 注意： 不同的客户端（命令行，TortoiseSVN, AnkhSVN, Subclipse 等）的出错信息可能稍有不同。 下面表格中的出错信息以 http://svn.moon.ossxp.com/svn/test 版本库做示例，仅供参考。 编号 出错信息 问题剖析 解决方案 1. svn: Server sent unexpected return value (500 I-

svn 出错信息总汇 . Subversion 错误信息一览表

注意：

不同的客户端（命令行，TortoiseSVN, AnkhSVN, Subclipse等）的出错信息可能稍有不同。
下面表格中的出错信息以 http://svn.moon.ossxp.com/svn/test 版本库做示例，仅供参考。

编号
出错信息
问题剖析
解决方案

1.

svn: Server sent unexpected return value (500 Internal Server Error) in response to OPTIONS request for 'http://svn.moon.ossxp.com/svn/test'

错误的用户名
检查登录的用户名是否输入错误

svn: 服务器发送了意外的返回值(500 Internal Server Error)，在响应 “OPTIONS” 的请求 “http://svn.moon.ossxp.com/svn/test” 中

2.

svn: OPTIONS of 'http://svn.moon.ossxp.com/svn/test': authorization failed: Could not authenticate to server: rejected Basic challenge (http://svn.moon.ossxp.com)

错误的口令
用正确的用户名/口令登录

svn: 方法 OPTIONS 失败于 “http://svn.moon.ossxp.com/svn/test”: 认证失败: Could not authenticate to server: rejected Basic challenge (http://svn.moon.ossxp.com)

3.

svn: Server sent unexpected return value (403 Forbidden) in response to OPTIONS request for 'http://svn.moon.ossxp.com/svn/test'

用户无权限
联系管理员，为用户分配权限

svn: 服务器发送了意外的返回值(403 Forbidden)，在响应 “OPTIONS” 的请求 “http://svn.moon.ossxp.com/svn/test” 中

4.

svn: OPTIONS of 'http://www.moon.ossxp.com/svn/test': 200 OK (http://www.moon.ossxp.com)

服务器地址错误，是普通Web页面，不支持SVN的 WebDAV 协议

确认输入正确的 SVN 服务地址。可以在浏览器中输入该地址进行确认

svn: 方法 OPTIONS 失败于 “http://www.moon.ossxp.com/svn/test”: 200 OK (http://www.moon.ossxp.com)

5.

The version of your subversion (client) is below 1.5.0, upgrade to 1.5.0 or above. SVN below 1.5.0 can not handle mergeinfo properly. It can mess up our automated merge tracking!

是由于客户端的软件版本低于1.5.0造成的。服务器端对客户端软件版本进行了限制，以免对合并跟踪破坏。

升级本地的Subversion客户端软件到1.5.0或以上版本。

6.

svn: This client is too old to work with working copy '.'. You need to get a newer Subversion client, or to downgrade this working copy. See http://subversion.tigris.org/faq.html#working-copy-format-change for details.

安装了多个版本的SVN客户端(TSVN,Subclipse,...)，且各个客户端的版本不一致。高版本的SVN客户端会自动更新本地工作目录中的 .svn 目录下的文件格式，导致旧版本的SVN客户端不能继续访问该本地工作目录

将本机安装的所有的SVN客户端都更新到同一个大版本，以避免本地工作目录的格式不一致

-svn: 此客户端对于工作副本 . 太旧。你需要取得更新的 Subversion 客户端，或者降级工作副本。 参见 http://subversion.tigris.org/faq.html#working-copy-format-change 以获得更详细的信息。 7. svn: Working cop-

svn: 此客户端对于工作副本 “.” 太旧。你需要取得更新的 Subversion 客户端，或者降级工作副本。 参见 http://subversion.tigris.org/faq.html#working-copy-format-change 以获得更详细的信息。

7.

svn: Working copy 'trunk/src' locked svn: run 'svn cleanup' to remove locks (type 'svn help cleanup' for details)

异常操作导致目录没有解锁。

一个简单的重现方法：在 .svn 目录下创建空的名为 lock 的文件

使用命令行 "svn cleanup" 或者类似的“清理”动作删除锁定

svn: 工作副本“trunk/src”已经锁定 svn: 运行“svn cleanup”删除锁定 (输入“svn help cleanup”得到用法)

8.

日志中没有作者信息： ------------------------------------ r9 | (没有作者信息) | … ossxp.com anonymous commit test
匿名提交导致没有作者信息
检查版本库权限控制,禁止匿名提交

9.

正在发送 ... 传输文件数据.svn: 提交失败(细节如下): svn: Commit blocked by pre-commit hook (exit code 1) with output: 提交说明至少应包含 4 个字符, 或者太简单了。

这是由于用户提交的提交说明(commit log)，太过简单了。在提交时需要输入有意义的 commit log。

写有意义的提交说明，或者请求管理员更改版本库插件
10.

增加 Logger.c 传输文件数据.svn: 提交失败(细节如下): svn: Commit blocked by pre-commit hook (exit code 1) with output: Wide character in print at /opt/svn/svnroot/myrepos/hooks/scripts/check-case-insensitive.pl line 259. 发现文件名大小写冲突: trunk/src/Logger.c 已经存在于 logger.c

管理员设置了对新增文件是否重名（只有大小写不同）的文件进行检查。文件名只有大小写不同，在Windows上进行检出会造成麻烦

不要添加重名（仅大小写不同）文件

增加 src/文件aBc.txt 传输文件数据.svn: 提交失败(细节如下): svn: Commit blocked by pre-commit hook (exit code 1) with output: Clash: '/trunk/src/文件aBc.txt' '/trunk/src/文件abc.txt'

11.

svn: While preparing '/home/jiangxin/tmp/svn.test/trunk/src/README.txt' for commit svn: Inconsistent line ending style

提交的文件已经设置了 svn:eol-style 属性，但是该文本内的换行符有DOS的换行符CRLF，也有Unix换行符LF，不一致！

统一该文本文件内的换行符。Linux 下可以用dos2unix, unix2dos, sed等命令。Windows下可用 UltraEdit 进行转换。

svn: 当为提交操作准备“/home/jiangxin/tmp/svn.test/trunk/src/README.txt”时 svn: 不一致的行结束样式
12.
svn: Failed to add file 'Makefile': an unversioned file of the same name already exists

执行更新(svn up)时报错。因为其他人新增一个文件到服务器，而本地却存在一个同名文件（未版本控制）

先将本地重名文件改名，再执行 "svn up"，之后再比较、合并文件。或者执行 "svn up --force"

-svn: 增加文件 'Makefile' 失败: 同名未版本控制的文件已存在 13. Adding src/Makefile svn: Commit failed (details follow): svn: File '/svn/test/trunk/src/Makefile' already exists 添加新文件，提交时报错。-

svn: 增加文件 'Makefile' 失败: 同名未版本控制的文件已存在

13.

Adding src/Makefile svn: Commit failed (details follow): svn: File '/svn/test/trunk/src/Makefile' already exists

添加新文件，提交时报错。因为其他人已经先于我增加了该文件。

先执行更新操作（"svn up"），再根据提示进行操作：合并/提交...

增加 src/Makefile svn: 提交失败(细节如下): svn: 文件“/svn/test/trunk/src/Makefile”已存在

14.

$ svn up Conflict discovered in 'Makefile'. Select: (p) postpone, (df) diff-full, (e) edit, (mc) mine-conflict, (tc) theirs-conflict, (s) show all options: p C Makefile Updated to revision 5. Summary of conflicts: Text conflicts: 1

多人同时编辑同一个文件时，可能会遇到冲突。别人先于我提交，则当我提交时要先更新。更新可能遇到不能自动解决的冲突

使用工具进行冲突解决

$ svn up 在 “Makefile” 中发现冲突。 选择: (p) 推迟，(df) 显示全部差异，(e) 编辑, (mc) 我的版本, (tc) 他人的版本, (s) 显示全部选项: p C Makefile 更新到版本 5。 冲突概要： 正文冲突：1

15.

svn: Commit failed (details follow): svn: File 'Makefile' is out of date svn: File not found: transaction '6-d', path '/trunk/src/Makefile'

提交的文件已被他人删除

先执行更新操作（"svn up"），再根据提示解决该树冲突：删除文件或继续添加...

svn: 提交失败(细节如下): svn: 文件 “Makefile” 已经过时 svn: File not found: transaction '6-c', path '/trunk/src/Makefile'

16.

svn: Commit failed (details follow): svn: File or directory '/trunk/XXX' is out of date; try updating svn: resource out of date; try updating

基于旧版本修改是不允许的
先更新（"svn update"），再提交

svn: 提交失败(细节如下): svn: 文件或目录 “/trunk/XXX” 已经过时；请先更新 svn: resource out of date; try updating

17.

svn: DAV request failed; it's possible that the repository's pre-revprop-change hook either failed or is non-existent svn: At least one property change failed; repository is unchanged svn: Error setting property 'log': Repository has not been enabled to accept revision propchanges; ask the administrator to create a pre-revprop-change hook

修改提交说明等操作属于高风险操作，因为该操作没有被版本控制，属于不可恢复的操作。缺省禁止。

请联系管理员，启用该版本的相关钩子，允许修改“版本属性”。参见 管理员钩子设置

svn: DAV 请求失败；可能是版本库的 pre-revprop-change 钩子执行失败或者不存在 svn: 至少有一个属性变更失败；版本库未改变 svn: 设置属性 “log” 出错: Repository has not been enabled to accept revision propchanges; ask the administrator to create a pre-revprop-change hook

原文链接：http://blog.csdn.net/fys_0801/article/details/48469817