**SVN手册**

**Concept：**The Repository / The Working Copy / Branch

基本命令：

显示信息 svn list [svn\_url]

查看日志 svn log –l n 其中n为代码更新层数

查看状态 svn status**( stat, st)**

下载代码 svn checkou**t(co)** [svn\_url]

上传代码 svn commi**t(ci)** –m “info” 注意在代码未完全确认下不要针对trunk执行

更新版本 svn update**(up)**

回退版本 svn revert [file\_path] //单个文件回滚

svn revert [dir] --depth=infinity //整个目录进行递归回滚

创建分支 svn copy**(cp)** –m “info” [trunk svn\_url] [branch\_url]

获得分支 svn co [branch\_url]

合并分支 svn merge –r[branch \_createversion]:[branch\_curversion] [branch\_url]

注意在trunk目录下执行

合并冲突产生解决之后执行：svn resolve –accept = working [file \_path]

生成patch： svn diff **(di)**> patchfile //整个工程项目变动生成patch

svn diff [file\_path] >patchfile //某个文件单独变动的patch

安装patch patch -p0 < patchfile //–p0选项要从当前目录查找的文件目录

**工作逻辑**



svn代码回滚命令

取消对代码的修改分为两种情况：

第一种情况：改动没有被提交（commit）。

这种情况下，使用svn revert就能取消之前的修改。

svn revert用法如下：

# svn revert [-R] something

其中something可以是（目录或文件的）相对路径也可以是绝对路径。

当something为单个文件时，直接svn revert something就行了；当something为目录时，需要加上参数-R(Recursive,递归)，否则只会将something这个目录的改动。

在这种情况下也可以使用svn update命令来取消对之前的修改，但不建议使用。因为svn update会去连接仓库服务器，耗费时间。

注意：svn revert本身有固有的危险，因为它的目的是放弃未提交的修改。一旦你选择了恢复，Subversion没有方法找回未提交的修改。

第二种情况：改动已经被提交（commit）。

这种情况下，用svn merge命令来进行回滚。

回滚的操作过程如下：

1、保证我们拿到的是最新代码：

svn update

假设最新版本号是28。

2、然后找出要回滚的确切版本号：

svn log [something]

假设根据svn log日志查出要回滚的版本号是25，此处的something可以是文件、目录或整个项目

如果想要更详细的了解情况，可以使用svn diff -r 28:25 [something]

3、回滚到版本号25：

svn merge -r 28:25 something

为了保险起见，再次确认回滚的结果：

svn diff [something]

发现正确无误，提交。

4、提交回滚：

svn commit -m "Revert revision from r28 to r25,because of ..."

提交后版本变成了29。

将以上操作总结为三条如下：

1. svn update，svn log，找到最新版本（latest revision）

2. 找到自己想要回滚的版本号（rollbak revision）

3. 用svn merge来回滚： svn merge -r : something