Netkiller Version 手札

DVCS(Distributed Version Control System)

netkiller Neo Chan

2009-12-12

文档出处: http://netkiller.sourceforge.net/ | http://netkiller.github.com

文档最近一次更新于 Tue Dec 6 08:45:21 UTC 2011

下面是我多年积累下来的经验总结，整理成文档供大家参考:

Netkiller Architect 手札	Netkiller Linux 手札	Netkiller Developer 手札	Netkiller Database 手札
Netkiller Debian 手札	Netkiller CentOS 手札	Netkiller FreeBSD 手札	Netkiller Shell 手札
Netkiller Web 手札	Netkiller Monitoring 手札	Netkiller Storage 手札	Netkiller Mail System 手札
Netkiller MySQL 手札	Netkiller LDAP 手札	Netkiller Security 手札	Netkiller Version 手札
Netkiller Intranet 手札	Netkiller Cisco IOS 手札	Netkiller Writer 手札	Netkiller Studio Linux 手札

目录

下一页

自述

自述

目录

1. 内容简介

当前文档档容比较杂，涉及内容广泛。

慢慢我会将其中章节拆成新文档.

文档内容简介:

1. Network
2. Security
3. Web Application
4. Database
5. Storage And Backup/Restore
6. Cluster
7. Developer

1.1. Audience(读者对象)

This book is intended primarily for Linux system administrators who are familiar with the following activities:

Audience

1. Linux system administration procedures, including kernel configuration
2. Installation and configuration of cluster, such as load balancing, High Availability,
3. Installation and configuration of shared storage networks, such as Fibre Channel SANs
4. Installation and configuration of web server, such as apache, nginx, lighttpd, tomcat/resin ...

本文档的读者对象:

文档面向有所有读者。您可以选读您所需要的章节,无需全篇阅读,因为有些章节不一定对你有用,用得着就翻来看看,暂时用不到的可以不看.

大体分来读者可以分为几类:

1. 架构工程师
2. 系统管理员
3. 系统支持,部署工程师

不管是谁,做什么的,我希望通过阅读这篇文档都能对你有所帮助。

1.2. 写给读者

欢迎提出宝贵的建议,如有问题请到 邮件列表 讨论

为什么写这篇文章

有很多想法,工作中也用不到所以未能实现，所以想写出来,和大家分享.有一点写一点,写得也不好,只要能看懂就行,就当学习笔记了.

开始零零碎碎写过一些文档，也向维基百科供过稿，但维基经常被ZF封锁，后来发现sf.net可以提供主机存放文档，便做了迁移。并开始了我的写作生涯。

这篇文档是作者8年来对工作的总结,是作者一点一滴的积累起来的，有些笔记已经丢失，所以并不完整。

因为工作太忙整理比较缓慢。目前的工作涉及面比较窄所以新文档比较少。

我现在花在技术上的时间越来越少，兴趣转向摄影，无线电。也想写写摄影方面的心得体会。

写作动力:

曾经在网上看到外国开源界对中国的评价，中国人对开源索取无度，但贡献却微乎其微.这句话一直记在我心中，发誓要为中国开源事业做我仅有的一点微薄贡献

另外写文档也是知识积累，还可以增加在圈内的影响力.

人跟动物的不同,就是人类可以把自己学习的经验教给下一代人.下一代在上一代的基础上再创新,不断积累才有今天.

所以我把自己的经验写出来,可以让经验传承

没有内容的章节:

目前我自己一人维护所有文档，写作时间有限，当我发现一个好主题就会加入到文档中，待我有时间再完善章节，所以你会发现很多章节是空无内容的.

文档目前几乎是流水帐试的写作，维护量很大，先将就着看吧.

我想到哪写到哪,你会发现文章没一个中心,今天这里写点,明天跳过本章写其它的.
文中例子绝对多,对喜欢复制然后粘贴朋友很有用,不用动手写,也省时间.
理论的东西,网上大把,我这里就不写了,需要可以去网上查.
我爱写错别字,还有一些是打错的,如果发现请指正.
文中大部分试验是在Debian/Ubuntu/Redhat AS上完成.

1.3. 获得文档

1.3.1. PDF

Download PDF Document 下载PDF文档1

Download PDF Document 下载PDF文档2

1.3.2. EPUB

http://netkiller.sourceforge.net/technology.html

1.3.3. 获得光盘介质

如有特别需要，请联系我

上一页		下一页
Netkiller Version 手札	起始页	2. 作者简介

2. 作者简介

主页地址：http://netkiller.sourceforge.net, http://netkiller.github.com/

陈景峰 (ㄔㄣ ㄐㄧㄥ ㄈㄥ)

Nickname： netkiller | English name: Neo chen | Nippon name: ちんけいほう (音訳) | Korean name: | Thailand name:

IT民工， UNIX like Evangelist, 业余无线电爱好者（呼号：BG7NYT）,户外运动以及摄影爱好者。

《PostgreSQL实用实例参考》，《Postfix 完整解决方案》，《Netkiller Linux 手札》的作者

2001年来深圳进城打工,成为一名外来务工者.

2002年我发现不能埋头苦干,埋头搞技术是不对的,还要学会"做人".

2003年这年最惨,公司拖欠工资16000元,打过两次官司2005才付清.

2004年开始加入分布式计算团队,目前成绩

2004-10月开始玩户外和摄影

2005-6月成为中国无线电运动协会会员

2006年单身生活了这么多年,终于找到归宿.

2007物价上涨,金融危机，休息了4个月（其实是找不到工作）

2008终于找到英文学习方法，，《Netkiller Developer 手札》，《Netkiller Document 手札》

2008-8-8 08:08:08 结婚,后全家迁居湖南省常德市

2009《Netkiller Database 手札》,年底拿到C1驾照

2010对电子打击乐产生兴趣，计划学习爵士鼓

2011 职业生涯路上继续打怪升级

2.1. 联系作者

Mobile: +86 13113668890

Tel: +86 755 2981-2080

Callsign: BG7NYT QTH: Shenzhen, China

注：请不要问我安装问题！

E-Mail: openunix@163.com
IRC irc.freenode.net #ubuntu / #ubuntu-cn

Yahoo: bg7nyt
ICQ: 101888222
AIM: bg7nyt

TM/QQ: 13721218
MSN: netkiller@msn.com
G Talk: 很少开
网易泡泡：很少开

写给火腿:

欢迎无线电爱好者和我QSO,我的QTH在深圳宝安区龙华镇溪山美地12B7CD,设备YAESU FT-50R,FT-60R,FT-7800 144-430双段机,拉杆天线/GP天线 Nagoya MAG-79EL-3W/Yagi

如果这篇文章对你有所帮助,请寄给我一张QSL卡片,qrz.cn or qrz.com or hamcall.net

Personal Amateur Radiostations of P.R.China

ZONE CQ24 ITU44 ShenZhen, China

Best Regards, VY 73! OP. BG7NYT

Home | Mirror | Search

Google™ Custom Search Search

3. 支持这个项目(Support this project)

Donations

招商银行(China Merchants Bank) 陈景峰 9555500000007459

部分 I. Subversion

目录

上一页 下一页
3. 支持这个项目(Support this project) 起始页 第 1 章 版本控制及如何运作

Home | Mirror | Search

Custom Search Search

第 1 章 版本控制及如何运作

目录

1. 版本库布局

版本库布局

1. trunk
2. branches
3. tags
4. releases

多项目版本库布局

```
                                           /-> branches
                       .---> project 2 ---> trunk
                      /                    \-> tags
                     /
                    /                                   /-> branches
repositories -------> project 1 ---> trunk
                        \                                       \-> tags
                         \
                          \                        /-> branches
                           `---> project 3 ---> trunk
                                               \-> tags
```

2. 策略

trunk 主干，作为主干你要保证他的代码是可运行的。

branches 分支，代码来自主干，可以有很多分支，分支用于多个团队同步开发，最终要将代码合并到主干。例如：branches/member, branches/blog, branches/shop 分别来自不同的团队

tags 一般用于做快照，一旦建立永不更改

releases 发行本版，代码来自主干， 与tags功能一样，命名更直观。例如每个月为一个周期，发布一次代码 releases/v1.0, releases/1.5

```
                          .-----------------------------------------------------
------------------> cart r100 ---> ...
                         /
^
                        .---------------------> blog r3 ---> r4 ... ... r(10) -
--> r(n) ---> r(...) --->/---> ...
                       /                                    ^
\                          /
               .-> branches r1  ---> member r2 ---/---------> r4 ... r(10) --\---
-> r(14) ---> r(...)--/---> ...
              /                        ^          /                \           \
\             /
repositories ----> trunk    r1  -----/----------/--------------> r(11) ---> r(12)
---> r(15) ---> r(...) ---> ...
              \
\         \          \
               `-> tags     r1  ----------------------------------------->
v1.0.0 r(13)   \          \
                       \
\
                        `-----------------------------------------------------
-------> v1.0.1 (r16)   \
                         `----------------------------------------------------
--------------------> v1.0.2 r(n)
```

```
                                                                            .>
unstable 2.1.1

/ \
                                              unstable 1.1          unstable 2.1
unstable 3.1
                                                          ^ \
^ \         / \
               .-> branches ---> unstable 1.0 ---/   \   unstable 2.0 ---/   \
unstable 3.0    unstable---> ...
              /                      ^                \            ^          \
^    \          ^
repositories ----> trunk    --------/----------------> stable ---/-----> stable
---/---> stable ---/-> stable ---> ...
              \                                        \                     \
\
               `-> tags     ----------------------------> stable 1.0          \
\
                       \                                                       \
\
                        `-----------------------------------------------------
> stable 2.0   \
                         `----------------------------------------------------
-------------> stable 3.0
```

怎样访问版本库

美工、页面人员采用WebDav访问Subversion。Photoshop,Dreamwaver 软件对WebDav有很好地支持，他们不需要学习如何使用Subversion。

开发者通过Svn客户端访问代码库，既可以使用开发IDE集成工具也可以使用单独工具。像TortoiseSVN等等。

Home | Mirror | Search

Google™ Custom Search Search

3. nightly version

Home | Mirror | Search

Google™ Custom Search Search

4. rc1,rc2,rc4

第 2 章 Invoking the Server

目录

配置开发环境版本控制Subversion

Squid + Subversion 请参考Squid一节

过程 2.1. subversion

1. installation

$ sudo apt-get install subversion

```
$ sudo apt-get install subversion
```

2. create svn group and svnroot user

```
$ sudo groupadd svn
$ sudo adduser svnroot --ingroup svn
```

3. create repository

```
$ svnadmin create /home/svnroot/test
```

4. testing

```
svnroot@netkiller:~$ svnserve -d --foreground -r /home/svnroot/
```

check out

neo@netkiller:/tmp$ svn list svn://localhost/test

you may see some file and directory

```
neo@netkiller:/tmp$ ls test/.svn/
entries  format  prop-base  props  text-base  tmp
```

5. configure

$ vim repositories/conf/svnserve.conf

```
[general]
anon-access = read
auth-access = write
password-db = passwd
# authz-db = authz
# realm = My First Repository
```

$ vim repositories/conf/passwd

```
[users]
# harry = harryssecret
# sally = sallyssecret
neo = chen
```

如果不允许匿名用户checkout代码，配置文件这样写anon-access = none

```
[general]
anon-access = none
auth-access = write
```

6. firewall

```
$ sudo ufw allow svn
```

1. standalone “daemon” process

svn daemon

```
$ svnserve --daemon --root /home/svnroot
```

1.1. starting subversion for debian/ubuntu

/etc/init.d/subversion for debian/ubuntu

```

debian:/etc/init.d# cat subversion
#!/bin/sh
### BEGIN INIT INFO
# Provides:          subversion
# Required-Start:    $remote_fs $network
# Required-Stop:     $remote_fs $network
# Should-Start:      fam
# Should-Stop:       fam
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: Start the subversion subversion server.
### END INIT INFO

#########################
# Author: Neo <openunix@163.com>
#########################

PATH=/sbin:/bin:/usr/sbin:/usr/bin
DAEMON=/usr/bin/svnserve
NAME=subversion
DESC="subversion server"
PIDFILE=/var/run/$NAME.pid
SCRIPTNAME=/etc/init.d/$NAME
SVNROOT=/srv/svnroot
DAEMON_OPTS="-d -T -r $SVNROOT --pid-file $PIDFILE"

test -x $DAEMON || exit 0

set -e

. /lib/lsb/init-functions
```

```
case "$1" in
    start)
        log_daemon_msg "Starting $DESC" $NAME
        echo
        $DAEMON $DAEMON_OPTS
        echo `pgrep -o $NAME` > $PIDFILE > /dev/null 2> /dev/null
        ;;
    stop)
        log_daemon_msg "Stopping $DESC" $NAME
        echo
        killall `basename $DAEMON` > /dev/null 2> /dev/null
        rm -rf $PIDFILE
        ;;
    restart)
        $0 stop
        $0 start
        ;;
    status)
        ps ax | grep $NAME
        ;;
    *)
        echo "Usage: $SCRIPTNAME {start|stop|restart|status}" >&2
        exit 1
        ;;
esac

exit 0
```

1.2. starting subversion daemon script for CentOS/Radhat

```
#!/bin/bash
#
# /etc/rc.d/init.d/subversion
#
# Starts the Subversion Daemon
#
# chkconfig: 345 90 10
#
# description: Subversion Daemon

# processname: svnserve

source /etc/rc.d/init.d/functions

[ -x /usr/bin/svnserve ] || exit 1

### Default variables
SYSCONFIG="/etc/sysconfig/subversion"

### Read configuration
[ -r "$SYSCONFIG" ] && source "$SYSCONFIG"

RETVAL=0
USER="svnroot"
prog="svnserve"
desc="Subversion Daemon"

start() {
   echo -n $"Starting $desc ($prog): "
   daemon --user $USER $prog -d $OPTIONS
   RETVAL=$?
   [ $RETVAL -eq 0 ] && touch /var/lock/subsys/$prog
   echo
}

stop() {
   echo -n $"Shutting down $desc ($prog): "
   killproc $prog
   RETVAL=$?
   [ $RETVAL -eq 0 ] && success || failure
   echo
   [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/$prog
   return $RETVAL
}

case "$1" in
  start)
   start
   ;;
  stop)
   stop
   ;;
  restart)
   stop
   start
   RETVAL=$?
   ;;
  condrestart)
        [ -e /var/lock/subsys/$prog ] && restart
   RETVAL=$?
   ;;
  *)
   echo $"Usage: $0 {start|stop|restart|condrestart}"
   RETVAL=1
esac

exit $RETVAL
```

/etc/sysconfig/subversion

```
# Configuration file for the Subversion service

#
# To pass additional options (for instace, -r root of directory to server) to
# the svnserve binary at startup, set OPTIONS here.
#
#OPTIONS=
OPTIONS="--threads --root /srv/svnroot"
```

上一页 上一级 下一页
4. rc1,rc2,rc4 起始页 2. classic Unix-like inetd daemon

2. classic Unix-like inetd daemon

/etc/inetd.conf

```
svn stream tcp nowait svn /usr/bin/svnserve svnserve -i -r
/home/svnroot/repositories
```

xinetd.d

/etc/xinetd.d/subversion

```
$ sudo apt-get install xinetd
$ sudo vim /etc/xinetd.d/subversion

service subversion
{
    disable = no
    port = 3690
    socket_type = stream
    protocol = tcp
    wait = no
    user = svnroot
    server = /usr/bin/svnserve
    server_args = -i -r /home/svnroot
}
```

restart xinetd

```
$ sudo /etc/init.d/xinetd restart
```

3. hooks

```
$ sudo apt-get install subversion-tools
```

3.1. post-commit

install SVN::Notify

```
perl -MCPAN -e 'install SVN::Notify'
```

```

$ sudo cp post-commit.tmpl post-commit
$ sudo chown svnroot:svn post-commit
$ sudo vim post-commit

REPOS="$1"
REV="$2"

#/usr/share/subversion/hook-scripts/commit-email.pl "$REPOS" "$REV"
openunix@163.com
/usr/share/subversion/hook-scripts/commit-email.pl "$1" "$2" --from
neo@netkiller.8800.org -h localhost  -s "[SVN]" --diff y openunix@163.com
openx@163.com
```

另一种方法

```

#!/bin/sh

REPOS="$1"
REV="$2"

/usr/local/bin/svnotify                        \
    --repos-path    "$REPOS"                   \
    --revision      "$REV"                     \
    --subject-cx                               \
    --with-diff                                \
    --handler       HTML::ColorDiff            \
    --to            <your e-mail address>      \
    --from          <from e-mail address>
```

```
/usr/bin/svnotify --repos-path "$REPOS" --revision "$REV" \
--from neo@netkiller.8800.org --to openunix@163.com --smtp localhost \
--handler "HTML::ColorDiff"  --with-diff --charset zh_CN:GB2312  -g zh_CN --
svnlook /usr/bin/svnlook --subject-prefix '[SVN]'
```

如果你没有安装邮件服务器，你可以使用服务商的SMTP如163.com

```
/usr/bin/svnotify --repos-path "$REPOS" --revision "$REV" \
--from openx@163.com --to openunix@163.com --smtp smtp.163.com  --smtp-user
openunix --smtp-pass ****** \
--handler "HTML::ColorDiff"  --with-diff --charset UTF-8 --language zh_CN --
svnlook /usr/bin/svnlook --subject-prefix '[SVN]'
```

Charset

```
REPOS="$1"
REV="$2"

svnnotify --repos-path "$REPOS" --revision "$REV" \
        --subject-cx \
        --from neo.chen@example.com \
        --to group@example.com,manager@example.com \
        --with-diff \
        --svnlook /usr/bin/svnlook \
        --subject-prefix '[SVN]' \
        --charset UTF-8  --language zh_CN
```

上一页 上一级 下一页

2. classic Unix-like inetd daemon 起始页 4. WebDav

4. WebDav

Apache SVN

$ sudo apt-get install libapache2-svn

```
netkiller@neo:/etc/apache2$ sudo apt-get install libapache2-svn
```

vhost

```

<VirtualHost *>
        ServerName svn.netkiller.8800.org
        DocumentRoot /var/svn

      <Location />
                DAV svn
                SVNPath /var/svn
                AuthType Basic
                AuthName "Subversion Repository"
                AuthUserFile /etc/apache2/svn.passwd
                <LimitExcept GET PROPFIND OPTIONS REPORT>
                        Require valid-user
                </LimitExcept>
        </Location>
</VirtualHost>
```

建立密码文件

建立第一个用户需要加-c参数

```
netkiller@neo:/etc/apache2$ sudo htpasswd2 -c /etc/apache2/svn.passwd svn
New password:
Re-type new password:
Adding password for user svn
```

输入两次密码

建立其他用户

```
sudo htpasswd2 /etc/apache2/svn.passwd otheruser
```

4.1. davfs2 - mount a WebDAV resource as a regular file system

install

```
$ sudo apt-get install davfs2
```

mount a webdav to directory

```
$ sudo mount.davfs https://opensvn.csie.org/netkiller /mnt/davfs/
Please enter the username to authenticate with server
https://opensvn.csie.org/netkiller or hit enter for none.
Username: svn
Please enter the password to authenticate user svn with server
https://opensvn.csie.org/netkiller or hit enter for none.
```

```
Password:
mount.davfs: the server certificate is not trusted
  issuer:      CSIE.org, CSIE.org, Taipei, Taiwan, TW
  subject:     CSIE.org, CSIE.org, Taipei, TW
  identity:    *.csie.org
  fingerprint: e6:05:eb:fb:69:5d:25:4e:11:3c:83:e8:7c:44:ee:bf:a9:85:a3:64
You only should accept this certificate, if you can
verify the fingerprint! The server might be faked
or there might be a man-in-the-middle-attack.
Accept certificate for this session? [y,N] y
```

test

```
$ ls davfs/
branches  lost+found  tags  trunk
```

第 3 章 repository 管理

目录

1. user admin

```
#!/bin/bash
################################
# Author: Neo<openunix@163.com
# Home: http://netkiller.sf.net
################################
SVNROOT=/srv/svnroot/project
adduser(){
    echo $1  $2
    if [ -z $1 ]; then
        usage
    else
        local user=$1
    fi
    if [ -z $2 ]; then
        usage
    else
        local passwd=$2
    fi
    echo "$1 = $2" >> $SVNROOT/conf/passwd
}
deluser(){
    local user=$1
    if [ -z $user ]; then
        usage
    else
        ed -s $SVNROOT/conf/passwd <<EOF
/$user/
d
wq
EOF
    fi
}
list(){
    cat $SVNROOT/conf/passwd
}
usage(){
    echo $"Usage: $0 {list|add|del} username"
}
case "$1" in
    list)
        list
        ;;
    add)
        adduser $2 $3
        ;;
    del)
        deluser $2
        ;;
    restart)
        stop
        start
        ;;
    condrestart)
        condrestart
        ;;
    *)
        usage
        exit 1
esac
```

用法

```
./svnuser list
./svnuser add user passwd
./svnuser del user
```

上一页	上一级	下一页
4. WebDav	起始页	2. authz

2. authz

$ svnadmin create /home/svnroot/project

$ svnserve --daemon --root /home/svnroot/project

```
[groups]
member = neo
blog = neo,netkiller
wiki = bg7nyt,chen,jingfeng

[/]
* =

[/member]
@member = rw
* = r

[/app/blog]
@blog = rw
* =

[/app/wiki]
@blog = rw
* =

# [repository:/baz/fuz]
# @harry_and_sally = rw
# * = r
```

$ svnadmin create /home/svnroot/project1

$ svnadmin create /home/svnroot/project2

$ svnserve --daemon --root /home/svnroot

```
[groups]
member = neo
blog = neo,netkiller
wiki = bg7nyt,chen,jingfeng

[project1:/]
* =
[project2:/]
* = r

[project1:/member]
@member = rw
* = r

[project2:/app/blog]
@blog = rw
* =

[project2:/app/wiki]
@blog = rw
* = r
```

Home | Mirror | Search

Google™ Custom Search Search

3. create repository

```
$ su - svnroot
$ svnadmin create /home/svnroot/neo
```

Home | Mirror | Search

Google™ Custom Search Search

4. dump

```
svnadmin dump /svnroot/project | gzip > svn.gz
```

第 4 章 使用Subversion

目录

1. Initialized empty subversion repository for project

```
svn co svn://127.0.0.1/project
cd project
mkdir trunk
mkdir tags
mkdir branches
svn ci -m "Initialized empty subversion repository in your_project"
```

Home | Mirror | Search

Custom Search Search

2. ignore

svn propset svn:ignore [filename] [folder]

```
$ svn propset svn:ignore 'images' .
$ svn ci -m 'Ignoring a directory called "images".'
```

```
$ svn propset svn:ignore '*' images
$ svn ci -m 'Ignoring a directory called "images".'
```

```
$ svn export spool spool-tmp
$ svn rm spool
$ svn ci -m 'Removing inadvertently added directory "spool".'
$ mv spool-tmp spool
$ svn propset svn:ignore 'spool' .
$ svn ci -m 'Ignoring a directory called "spool".'
```

.ignore

```
svn propset svn:ignore -F .cvsignore .
```

3. 关键字替换

```
Date
         这个关键字保存了文件最后一次在版本库修改的日期，看起来类似于$Date: 2006-07-22
21:42:37 -0700 (Sat, 22 Jul 2006) $，它也可以用LastChangedDate来指定。
Revision
         这个关键字描述了这个文件最后一次修改的修订版本，看起来像$Revision: 144 $，也可以通
过LastChangedRevision或者Rev引用。
Author
         这个关键字描述了最后一个修改这个文件的用户，看起来类似$Author: harry $，也可以
用LastChangedBy来指定。
HeadURL
         这个关键字描述了这个文件在版本库最新版本的完全URL，看起来类似$HeadURL:
http://svn.collab.net/repos/trunk/README $，可以缩写为URL。
Id
         这个关键字是其他关键字一个压缩组合，它看起来就像$Id: calc.c 148 2006-07-28
21:30:43Z sally $，可以解释为文件calc.c上一次修改的修订版本号是148，时间是2006年7月28日，作
者是sally。
```

```
$ cat weather.txt
$Id$

$ svn propset svn:keywords "Id" weather.txt
property 'svn:keywords' set on 'weather.txt'

$ cat weather.txt
$Id: weather.txt 25 2009-03-17 09:37:23Z netkiller $
```

Home | Mirror | Search

Google™ Custom Search Search

4. lock 加锁/ unlock 解锁

$ svn lock -m "LockMessage" [– force] PATH

```
$ svn lock -m “lock test file“ test.php
$ svn unlock PATH
```

Home | Mirror | Search

Google™ Custom Search Search

5. import/export

```
svn import [PATH] URL
svn export URL [PATH]
```

Home | Mirror | Search

Google™ Custom Search Search

6. export 指定版本

```
svn log file
svn export -r rxxxxx file
or
svn export -r rxxxxx file newfile
svn ci -m "restore rxxxxxx"
```

上一页 上一级 下一页
5. import/export 起始页 7. 修订版本关键字

Home | Mirror | Search

Google™ Custom Search Search

7. 修订版本关键字

HEAD

版本库中最新的（或者是“最年轻的”）版本。

BASE

工作拷贝中一个条目的修订版本号，如果这个版本在本地修改了，则“BASE版本”就是这个条目在本地未修改的版本。

COMMITTED

项目最近修改的修订版本，与BASE相同或更早。

PREV

一个项目最后修改版本之前的那个版本，技术上可以认为是COMMITTED -1。

```
$ svn cat -r PREV filename > filename
$ svn diff -r PREV filename
```

Home | Mirror | Search

Google™ Custom Search Search

8. 恢复旧版本

svn没有恢复旧版本的直接功能，不过可以使用svn merge命令恢复。比如说当前HEAD为2，而我要恢复成1版本，怎么做？

用svn merge：

```
svn update
svn merge --revision 2:1 svn://localhost/lynn
svn commit -m "restore to revision 1"
```

svn merge --r HEAD:1 svn://localhost/lynn

第 5 章 branch

目录

1. create

create a new branch using copy

```
svn cp http://www.domain.com/truck/project
http://www.domain.com/branches/project_branch_1
```

2. remove

remove

```
svn rm http://www.domain.com/branches/project_branch_1
```

Home | Mirror | Search

Google™ Custom Search Search

3. switch

```
svn switch http://www.domain.com/branches/project_branch_2 .
```

Home | Mirror | Search

Google™ Custom Search Search

4. merge

```
svn -r 148:149 merge svn://server/trunk branches/module
```

Home | Mirror | Search

Google™ Custom Search Search

5. relocate

switch --relocate FROM TO [PATH...]

```
svn switch --relocate svn://192.168.3.9/neo svn://192.168.3.5/neo .
```

Home | Mirror | Search

Google™ Custom Search Search

第 6 章 FAQ

目录

1. 清除项目里的所有.svn目录

```
find . -type d -iname ".svn" -exec rm -rf {} \;
```

2. color diff

http://colordiff.sourceforge.net/

```
$ sudo apt-get install colordiff
```

add the following to your ~/.bashrc

```
alias svndiff='svn diff --diff-cmd=colordiff'
```

3. cvs2svn

http://cvs2svn.tigris.org/

```
[root@development ~]# cvs2svn --encoding=gb2312 --fallback-encoding=utf_8 --
existing-svnrepos --svnrepos /home/svnroot /home/cvsroot
[root@development ~]# cvs2svn --encoding=gb2312 --fallback-encoding=utf_8 --
svnrepos /home/svnroot /home/cvsroot
```

4. Macromedia Dreamweaver MX 2004 + WebDAV +Subversion

首先进入站点管理

单击新建(New…)按钮选择站点(Site)

显示站点设置面版 Local Info 中设置

Remote Info 中设置

单击设置按钮（settings)

单击确定

单击Done完成

连接 WebDAV 服务器

单击

连接

部分 II. Git － Fast Version Control System

目录

homepage: http://git.or.cz/index.html

过程 2. Git

1. install

```
sudo apt-get install git-core
```

2. config

```
$ git-config --global user.name neo
$ git-config --global user.email openunix@163.com
```

3. Initializ

```
$ mkdir repository
$ cd repository/

/repository$ git-init-db
Initialized empty Git repository in .git/
```

to check .gitconfig file

```
$ cat ~/.gitconfig
[user]
        name = chen
        email = openunix@163.com
```

上一页 下一页

4. Macromedia Dreamweaver MX 2004 + WebDAV +Subversion 起始页 第 7 章 Creating and Commiting

第 7 章 Creating and Commiting

```
$ cd (project-directory)
$ git init
$ (add some files)
$ git add .
$ git commit -m 'Initial commit'
```

第 8 章 command

目录

1. git-add - Add file contents to the index

```
$ echo 'hello world!!!'> newfile
$ git-add newfile
```

2. git-status - Show the working tree status

```
$ git-status newfile
# On branch master
#
# Initial commit
#
# Changes to be committed:
#   (use "git rm --cached <file>..." to unstage)
#
#       new file: newfile
#
```

3. git-commit - Record changes to the repository

```
$ git-commit -m 'add a new file' newfile
Created initial commit f6fda79: add a new file
 1 files changed, 1 insertions(+), 0 deletions(-)
 create mode 100644 newfile
```

Home | Mirror | Search

Google™ Custom Search Search

4. git-show - Show various types of objects

```
$ git-show
commit f6fda79f2f550ea3b2c1b483371ed5d12499ac35
Author: chen <openunix@163.com>
Date:   Sat Nov 1 08:50:45 2008 -0400

    add a new file

diff --git a/newfile b/newfile
new file mode 100644
index 0000000..b659464
--- /dev/null
+++ b/newfile
@@ -0,0 +1 @@
+hello world!!!
```

上一页 上一级 下一页
3. git-commit - Record changes to the repository 起始页 5. git-checkout - Checkout and switch to a branch

5. git-checkout - Checkout and switch to a branch

5.1. checkout master

```
$ git-checkout master
Switched to branch "master"
```

5.2. checkout branch

```
$ git-branch
* master
  mybranch

$ git-checkout mybranch
Switched to branch "mybranch"

$ git-branch
  master
* mybranch
```

第 9 章 Manipulating branches

目录

git-branch - List, create, or delete branches

1. list branches

```
$ git-branch
* master
```

2. create branches

```
$ git-branch mybranch
$ git-branch
* master
  mybranch
```

3. delete branches

```
$ git-branch -d mybranch
Deleted branch mybranch.

$ git-branch
* master
```

Home | Mirror | Search

Google™ Custom Search Search

4. switch branch

```
$ git-branch
* master
  mybranch

$ git-checkout mybranch
Switched to branch "mybranch"

$ git-branch
  master
* mybranch
```

5. git-show-branch - Show branches and their commits

```
$ git-show-branch
! [master] add a new file
 * [mybranch] add a new file
--
+* [master] add a new file
```

Home | Mirror | Search

Google™ Custom Search Search

第 10 章 Cloning and Creating a Patch

```
$ git clone git://github.com/git/hello-world.git
$ cd hello-world
$ (edit files)
$ git add (files)
$ git commit -m 'Explain what I changed'
$ git format-patch origin/master
```

第 11 章 Setting up a git server

First we need to setup a user with a home folder. We will store all the repositories in this users home folder.

```
sudo adduser git
```

Rather than giving out the password to the git user account use ssh keys to login so that you can have multiple developers connect securely and easily.

Next we will make a repository. For this example we will work with a repository called example. Login as the user git and add the repository.

login to remote server

```
ssh git@REMOTE_SERVER
```

once logged in

```
sudo mkdir example.git
cd example.git
sudo git --bare init
Initialized empty Git repository in /home/git/example.git/
```

That’s all there is to creating a repository. Notice we named our folder with a .git extension.

Also notice the ‘bare’ option. By default the git repository assumes that you’ll be using it as your working directory, so git stores the actual bare repository files in a .git directory alongside all the project files. Since we are setting up a remote server we don’t need copies of the files on the filesystem. Instead, all we need are the deltas and binary objects of the repository. By setting ‘bare’ we tell git not to store the current files of the repository only the diffs. This is optional as you may have need to be able to browse the files on your remote server.

Finally all you need to do is add your files to the remote repository. We will assume you don’t have any files yet.

```
mkdir example
cd example
git init
touch README
git add README
git commit -m 'first commit'
git remote add origin git@REMOTE_SERVER:example.git
git push origin master
```

replace REMOTE_SERVER with your server name or IP

第 10 章 Cloning and Creating a Patch 起始页 第 12 章 Manager remote

Home | Mirror | Search

Google™ Custom Search Search

第 12 章 Manager remote

remote add

```
git remote add origin git@localhost:example.git
```

remote show

```
git remote show
origin
```

remote rm

```
git remote rm origin
```

上一页 上一级 下一页

第 11 章 Setting up a git server 起始页 第 13 章 git-daemon - A really simple server for git repositories

第 13 章 git-daemon - A really simple server for git repositories

在 /home/neo/ 上运行 git 守护进程

过程 13.1. git-daemon

1. /etc/shells

   /etc/shells 最后一行添加 ’/usr/bin/git-shell’

```
$ grep git /etc/shells
/usr/bin/git-shell
```

2. add new user ’git’ and ’gitroot’ for git

   you need to assign shell with /usr/bin/git-shell

```
$ sudo adduser git --shell /usr/bin/git-shell
$ sudo adduser gitroot --ingroup git --shell /bin/bash
```

   /etc/passwd

```
$ grep git /etc/passwd
git:x:1001:1002:,,,:/home/git:/usr/bin/git-shell
gitroot:x:1002:1002:,,,:/home/gitroot:/bin/bash
```

3. /etc/services

```
$ grep 9418 /etc/services
git             9418/tcp                        # Git Version Control System
```

4. /etc/inet.conf

```
$ grep git /etc/inet.conf
git     stream  tcp     nowait  nobody \
  /usr/bin/git-daemon git-daemon --inetd --syslog --export-all /home/gitroot
```

   reload inetd

```
$ sudo pkill -HUP inetd
```

   or

   /etc/xinetd.d/

```
$ cat /etc/xinetd.d/git
# default: off
# description: The git server offers access to git repositories
service git
{
        disable = no
        type            = UNLISTED
        port            = 9418
        socket_type     = stream
        protocol                = tcp
        wait            = no
        user            = git
        server          = /usr/bin/git-daemon
        server_args     = --inetd --export-all --base-path=/home/gitroot/
        log_on_failure  += USERID
}
```

<!-- service git { disable = no port = 9418 socket_type = stream wait = no user = git server = /usr/bin/git-daemon server_args = --inetd --export-all --base-path=/home/git/ log_on_failure += USERID } -->

reload xinitd

```
$ sudo /etc/init.d/xinetd reload
 * Reloading internet superserver configuration xinetd
[ OK ]
```

5. test

   git-clone git://localhost/example.git

```

$ git-clone git://localhost/example.git
Initialized empty Git repository in /tmp/example/.git/
remote: Counting objects: 3, done.
remote: Total 3 (delta 0), reused 0 (delta 0)
```

上一页 上一级 下一页

第 12 章 Manager remote 起始页 第 14 章 git-svn - Bidirectional operation between a single Subversion branch and git

第 14 章 git-svn - Bidirectional operation between a single Subversion branch and git

```
sudo apt-get install git-svn
```

clone

```
git-svn clone -s svn://netkiller.8800.org/neo
cd neo
git gc

git commit -a
git-svn dcommit
```

从 svn 仓库更新

```
git-svn rebase
```

git-svn init svn://netkiller.8800.org/neo/public_html

第 15 章 .gitignore

```
find ./ -type d -empty | grep -v \.git | xargs -i touch {}/.gitignore
```

Home | Mirror | Search

Google™ Custom Search Search

第 16 章 Web Tools

viewgit

http://viewgit.sourceforge.net/

上一页 上一级 下一页

第 15 章 .gitignore 起始页 部分 III. cvs - Concurrent Versions System

部分 III. cvs - Concurrent Versions System

目录

过程 4. install cvs

1. install

```
$ sudo apt-get install xinetd
$ sudo apt-get install cvs
```

   show the cvs version

```
$ cvs -v

Concurrent Versions System (CVS) 1.12.13 (client/server)
```

2. create cvs group and cvsroot user

```
$ sudo groupadd cvs
$ sudo adduser cvsroot --ingroup cvs
```

   change user become cvsroot

```
$ su - cvsroot
```

3. initialization 'CVSROOT'

```
$ cvs -d /home/cvsroot init
```

if you have successed, you can see CVSROOT directory in the '/home/cvsroot'

```
$ ls /home/cvsroot/
CVSROOT
```

4. authentication

default SystemAuth=yes, you can use system user to login cvs.

but usually, we don't used system user because it isn't security.

SystemAuth = no

edit '/home/cvsroot/CVSROOT/config' make sure SystemAuth = no

```
$ vim /home/cvsroot/CVSROOT/config
SystemAuth = no
```

create passwd file

the format is user:password:cvsroot

you need to using htpasswd command, if you don't have, please install it as the following

```
$ sudo apt-get install apache2-utils
```

or

```
$ perl -e 'print("userPassword: ".crypt("secret","salt")."\n");'
```

or

```
$ cat passwd
#!/usr/bin/perl
srand (time());
my $randletter = "(int (rand (26)) + (int (rand (1) + .5) % 2 ? 65 : 97))";
my $salt = sprintf ("%c%c", eval $randletter, eval $randletter);
my $plaintext = shift; my $crypttext = crypt ($plaintext, $salt);
print "${crypttext}\n";

$ ./passwd "mypasswd"
atfodI2Y/dcdc
```

let's using htpasswd to create a passwd

```
$ htpasswd -n neo
New password:
Re-type new password:
neo:yA50LI1BkXysY
```

copy 'neo:yA50LI1BkXysY' and add ':cvsroot' to the end

```
$ vim /home/cvsroot/CVSROOT/passwd
neo:yA50LI1BkXysY:cvsroot
nchen:GXaAkSKaQ/Hpk:cvsroot
```

5. Go into directory '/etc/xinetd.d/', and then create a cvspserver file as the following.

```
$ sudo vim /etc/xinetd.d/cvspserver

service cvspserver
{
   disable = no
   flags = REUSE
```

```
    socket_type = stream
    wait = no
    user = cvsroot
    server = /usr/bin/cvs
    server_args = -f --allow-root=/home/cvsroot pserver
    log_on_failure += USERID
}
```

6. check cvspserver in the ’/etc/services’

```
$ grep cvspserver /etc/services
cvspserver      2401/tcp                        # CVS client/server
operations
cvspserver      2401/udp
```

7. restart xinetd

```
$ /etc/init.d/xinetd
Usage: /etc/init.d/xinetd {start|stop|reload|force-reload|restart}
```

8. port

```
$ nmap localhost -p cvspserver

Starting Nmap 4.53 ( http://insecure.org ) at 2008-11-14 16:21 HKT
Interesting ports on localhost (127.0.0.1):
PORT     STATE SERVICE
2401/tcp open  cvspserver

Nmap done: 1 IP address (1 host up) scanned in 0.080 seconds
```

9. firewall

```
$ sudo ufw allow cvspserver
```

environment variable

CVSROOT=:pserver:username@ip:/home/cvsroot

```
vim .bashrc

export CVS_RSH=ssh
export CVSROOT=:pserver:neo@localhost:/home/cvsroot
```

test

```
$ cvs login
Logging in to :pserver:neo@localhost:2401/home/cvsroot
CVS password:
neo@netkiller:/tmp/test$ cvs co test
cvs checkout: Updating test
U test/.project
U test/NewFile.xml
U test/newfile.php
neo@netkiller:/tmp/test$
```

1. chroot

```
$ sudo apt-get install cvsd
```

environment variable

```
neo@netkiller:~/workspace/cvs$ export CVSROOT=:pserver:neo@localhost:/home/cvsroot
```

ssh

```
export CVS_RSH=ssh
export CVSROOT=:ext:$USER@localhost:/home/cvsroot
```

上一页 下一页
第 16 章 Web Tools 起始页 第 17 章 cvs login | logout

Home | Mirror | Search

Google™ Custom Search Search

第 17 章 cvs login | logout

```
neo@netkiller:~/workspace/cvs$ cvs login
Logging in to :pserver:neo@localhost:2401/home/cvsroot
CVS password:
```

logout

```
$ cvs logout
Logging out of :pserver:neo@localhost:2401/home/cvsroot
```

第 18 章 cvs import

cvs import -m "write some comments here" project_name vendor_tag release_tag

```
$ cvs import -m "write some comments here" project_name vendor_tag release_tag
```

上一页 上一级 下一页

第 17 章 cvs login | logout 起始页 第 19 章 cvs checkout

第 19 章 cvs checkout

```
$ cvs checkout project_name
cvs checkout: Updating project_name
```

checkout before

cvs checkout -r release_1_0 project_name

```
$ cvs checkout -r release_1_0 project_name
cvs checkout: Updating project_name
U project_name/file
cvs checkout: Updating project_name/dir1
U project_name/dir1/file1
cvs checkout: Updating project_name/dir2
U project_name/dir2/file1
U project_name/dir2/file2
```

第 20 章 cvs update

about update

```
$ cvs update
$ cvs update -r HEAD
$ cvs update -r 1.5
$ cvs update -D now
$ cvs update -D now file
```

第 21 章 cvs add

```
$ cd project_name/
$ touch new_file
$ cvs add new_file
cvs add: scheduling file `new_file' for addition
cvs add: use `cvs commit' to add this file permanently
```

if the file is binary

cvs add -kb new_file.gif

add a directory

```
$ mkdir dir1
$ mkdir dir2
$ touch dir1/file1
$ touch dir2/file1
$ touch dir2/file2
$ cvs add dir1
? dir1/file1
Directory /home/cvsroot/project_name/dir1 added to the repository
$ cvs add dir2
? dir2/file1
? dir2/file2
Directory /home/cvsroot/project_name/dir2 added to the repository
```

add mulit files

```
$ cvs add dir1/file1
$ cvs add dir2/file?
```

Home | Mirror | Search

Google™ Custom Search Search

第 22 章 cvs status

```
$ cvs status dir1/file1
cvs status: use `cvs add' to create an entry for `dir1/file1'
===================================================================
File: file1             Status: Unknown

   Working revision:  No entry for file1
   Repository revision: No revision control file
```

第 23 章 cvs commit

```
$ cvs commit -m "add a new file"
cvs commit: Examining .
/home/cvsroot/project_name/new_file,v  <--  new_file
initial revision: 1.1
```

commit multi files

```
$ cvs commit -m "add a new file" dir1/* dir2/*
/home/cvsroot/project_name/dir1/file1,v  <--  dir1/file1
initial revision: 1.1
/home/cvsroot/project_name/dir2/file1,v  <--  dir2/file1
initial revision: 1.1
/home/cvsroot/project_name/dir2/file2,v  <--  dir2/file2
initial revision: 1.1
```

Home | Mirror | Search

Custom Search Search

第 24 章 cvs remove

```
$ rm -rf new_file
$ cvs remove new_file
cvs remove: scheduling `new_file' for removal
cvs remove: use `cvs commit' to remove this file permanently
$ cvs commit -m "delete file" new_file
/home/cvsroot/project_name/new_file,v  <--  new_file
new revision: delete; previous revision: 1.1
```

第 25 章 cvs log

let me create a file, and then modify the file to make several version

```
$ touch file
$ echo helloworld > file
$ cvs add file
cvs add: scheduling file `file' for addition
cvs add: use `cvs commit' to add this file permanently
$ cvs commit -m 'add file to cvs' file
/home/cvsroot/project_name/file,v  <--  file
initial revision: 1.1
$ echo I am Neo > file
$ cvs commit -m 'add file to cvs' file
/home/cvsroot/project_name/file,v  <--  file
new revision: 1.2; previous revision: 1.1
$ echo my nickname is netkiller > file
$ cvs commit -m 'modified file' file
/home/cvsroot/project_name/file,v  <--  file
new revision: 1.3; previous revision: 1.2
$ echo I am 28 years old > file
$ cvs commit -m 'modified file' file
/home/cvsroot/project_name/file,v  <--  file
new revision: 1.4; previous revision: 1.3
```

show log message

```
$ cvs log file

RCS file: /home/cvsroot/project_name/file,v
Working file: file
head: 1.4
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
total revisions: 4;   selected revisions: 4
description:
----------------------------
revision 1.4
date: 2008-11-24 15:42:49 +0800;  author: neo;  state: Exp;  lines: +1 -1;
commitid: V0iuptfP43iETPrt;
modified file
----------------------------
revision 1.3
date: 2008-11-24 15:42:20 +0800;  author: neo;  state: Exp;  lines: +1 -1;
commitid: YWfYHFSV10duTPrt;
modified file
----------------------------
revision 1.2
date: 2008-11-24 15:41:47 +0800;  author: neo;  state: Exp;  lines: +1 -1;
commitid: 4iRs5fm1g9diTPrt;
add file to cvs
----------------------------
revision 1.1
date: 2008-11-24 15:41:28 +0800;  author: neo;  state: Exp;  commitid:
zCWkxnWxLZHbTPrt;
add file to cvs
=============================================================================
```

cvs log -r1.2 file

```
$ cvs log -r1.2 file

RCS file: /home/cvsroot/project_name/file,v
Working file: file
head: 2.1
branch:
locks: strict
access list:
symbolic names:
        release_1_0_patch: 1.4.0.2
        release_1_0: 1.4
keyword substitution: kv
total revisions: 5;     selected revisions: 1
description:
----------------------------
revision 1.2
date: 2008-11-24 15:41:47 +0800;  author: neo;  state: Exp;  lines: +1 -1;
commitid: 4iRs5fm1g9diTPrt;
add file to cvs
=============================================================================
```

上一页	上一级	下一页
第 24 章 cvs remove	起始页	第 26 章 cvs annotate

第 26 章 cvs annotate

```
$ cvs annotate file

Annotations for file
***************
2.2          (nchen     26-Nov-08): I am Neo
2.2          (nchen     26-Nov-08): My nickname netkiller
2.3          (nchen     26-Nov-08): I'm from shenzhen
1.4          (neo       24-Nov-08): I am 28 years old
```

第 27 章 cvs diff

```
neo@netkiller:~/workspace/cvs/project_name$ cvs diff -r1.3 -r1.4 file
Index: file
===================================================================
RCS file: /home/cvsroot/project_name/file,v
retrieving revision 1.3
retrieving revision 1.4
diff -r1.3 -r1.4
1c1
< my nickname is netkiller
---
> I am 28 years old
neo@netkiller:~/workspace/cvs/project_name$ cvs diff -r1.2 -r1.4 file
Index: file
===================================================================
RCS file: /home/cvsroot/project_name/file,v
retrieving revision 1.2
retrieving revision 1.4
diff -r1.2 -r1.4
1c1
< I am Neo
---
> I am 28 years old
```

--side-by-side

```
neo@netkiller:/tmp/cvs/test/project_name$ cvs diff --side-by-side -r1.2 -r1.4
file
Index: file
===================================================================
RCS file: /home/cvsroot/project_name/file,v
retrieving revision 1.2
retrieving revision 1.4
diff --side-by-side -r1.2 -r1.4
I am Neo                                                       | I am 28 years
old
```

第 28 章 rename file

mv file_name new_file_name && cvs remove file_name
cvs add new_file_name

```
neo@netkiller:/tmp/cvs/project_name$ mv file file.txt
neo@netkiller:/tmp/cvs/project_name$ cvs remove file
cvs remove: scheduling `file' for removal
cvs remove: use `cvs commit' to remove this file permanently
neo@netkiller:/tmp/cvs/project_name$ cvs add file.txt
cvs add: scheduling file `file.txt' for addition
cvs add: use `cvs commit' to add this file permanently
neo@netkiller:/tmp/cvs/project_name$ cvs commit -m 'rename file to file.txt'
cvs commit: Examining .
cvs commit: Examining dir1
cvs commit: Examining dir2
/home/cvsroot/project_name/file,v  <--  file
new revision: delete; previous revision: 2.3
/home/cvsroot/project_name/file.txt,v  <--  file.txt
initial revision: 1.1
```

第 29 章 revision

```
neo@netkiller:~/workspace/cvs/project_name$ cvs update -r 1.2 file
U file
neo@netkiller:~/workspace/cvs/project_name$ cvs st file
===================================================================
File: file              Status: Up-to-date

   Working revision:    1.2
   Repository revision: 1.2     /home/cvsroot/project_name/file,v
   Commit Identifier:   4iRs5fm1g9diTPrt
   Sticky Tag:          1.2
   Sticky Date:         (none)
   Sticky Options:      (none)
```

last version

```
neo@netkiller:~/workspace/cvs/project_name$ cvs update -r HEAD file
U file
neo@netkiller:~/workspace/cvs/project_name$ cvs st file
===================================================================
File: file              Status: Up-to-date

   Working revision:    1.4
   Repository revision: 1.4     /home/cvsroot/project_name/file,v
   Commit Identifier:   V0iuptfP43iETPrt
   Sticky Tag:          HEAD (revision: 1.4)
   Sticky Date:         (none)
   Sticky Options:      (none)
```

第 30 章 cvs export

cvs export -r release_1_0 project_name

```
$ cvs export -r release_1_0 project_name
cvs export: Updating project_name
U project_name/file
cvs export: Updating project_name/dir1
U project_name/dir1/file1
cvs export: Updating project_name/dir2
U project_name/dir2/file1
U project_name/dir2/file2
```

cvs export -D 20081126 project_name

```
$ cvs export -D 20081126 project_name
cvs export: Updating project_name
U project_name/file
cvs export: Updating project_name/dir1
U project_name/dir1/file1
cvs export: Updating project_name/dir2
U project_name/dir2/file1
U project_name/dir2/file2
```

cvs export -D now -d nightly project_name

```
$ cvs export -D now -d nightly project_name
cvs export: Updating nightly
U nightly/file
cvs export: Updating nightly/dir1
U nightly/dir1/file1
cvs export: Updating nightly/dir2
U nightly/dir2/file1
U nightly/dir2/file2
neo@netkiller:/tmp/cvs$
```

第 31 章 cvs release

```
$ ls
project_name

$ cvs release -d project_name
You have [0] altered files in this repository.
Are you sure you want to release (and delete) directory `project_name': y

$ ls
```

Home | Mirror | Search

Google™ Custom Search Search

第 32 章 branch

目录

1. milestone

set up a release number

```
$ cvs tag release_1_0
cvs tag: Tagging .
T file
cvs tag: Tagging dir1
T dir1/file1
cvs tag: Tagging dir2
T dir2/file1
T dir2/file2
```

beginning next one milestone

```

$ cvs commit -r 2

Log message unchanged or not specified
a)bort, c)ontinue, e)dit, !)reuse this message unchanged for remaining dirs
Action: (continue) c

CVS: ----------------------------------------------------------------------
CVS: Enter Log.  Lines beginning with `CVS:' are removed automatically
CVS:
CVS: Committing in .
CVS:
CVS: Modified Files:
CVS:  Tag: 2
CVS:    file dir1/file1 dir2/file1 dir2/file2
CVS: ----------------------------------------------------------------------

/home/cvsroot/project_name/file,v  <--  file
new revision: 2.1; previous revision: 1.4
/home/cvsroot/project_name/dir1/file1,v  <--  dir1/file1
new revision: 2.1; previous revision: 1.1
/home/cvsroot/project_name/dir2/file1,v  <--  dir2/file1
new revision: 2.1; previous revision: 1.1
/home/cvsroot/project_name/dir2/file2,v  <--  dir2/file2
new revision: 2.1; previous revision: 1.1
```

other user

```
$ cvs up
cvs update: Updating .
P file
cvs update: Updating dir1
U dir1/file1
cvs update: Updating dir2
U dir2/file1
U dir2/file2

$ cvs st file
===================================================================
File: file              Status: Up-to-date
```

```
   Working revision:    2.1
   Repository revision: 2.1      /home/cvsroot/project_name/file,v
   Commit Identifier:   SuZpTC1gCRrH2Qrt
   Sticky Tag:          (none)
   Sticky Date:         (none)
   Sticky Options:      (none)
```

上一页 上一级 下一页
第 31 章 cvs release 起始页 2. patch branch

2. patch branch

create a branch release_1_0_patch from release_1_0 by cvs admin

```
$ cvs rtag -b -r release_1_0 release_1_0_patch project_name
cvs rtag: Tagging project_name
cvs rtag: Tagging project_name/dir1
cvs rtag: Tagging project_name/dir2
```

checkout release_1_0_patch by other user

```
$ cvs checkout -r release_1_0_patch project_name
cvs checkout: Updating project_name
U project_name/file
cvs checkout: Updating project_name/dir1
U project_name/dir1/file1
cvs checkout: Updating project_name/dir2
U project_name/dir2/file1
U project_name/dir2/file2
```

show the status, and you can see 'Sticky Tag' is 'release_1_0_patch'

```
$ cvs st file
===================================================================
File: file              Status: Up-to-date

   Working revision:    1.4
   Repository revision: 1.4     /home/cvsroot/project_name/file,v
   Commit Identifier:   V0iuptfP43iETPrt
   Sticky Tag:          release_1_0_patch (branch: 1.4.2)
   Sticky Date:         (none)
   Sticky Options:      (none)
```

第 33 章 keywords

$Author$
$Date$
$Name$
$Id$
$Header$
$Log$
$Revision$

add above keywords into a file, and then commit it.

```
$ cat file.txt
$Author: nchen $
$Date: 2008-11-27 01:33:29 $
$Name:  $
$Id: file.txt,v 1.2 2008-11-27 01:33:29 nchen Exp $
$Header: /home/cvsroot/project_name/file.txt,v 1.2 2008-11-27 01:33:29 nchen Exp $
$Log: file.txt,v $
Revision 1.2  2008-11-27 01:33:29  nchen
added some of keywords

$Revision: 1.2 $
```

部分 IV. Miscellaneous

目录

Home | Mirror | Search

Google™ Custom Search Search

第 34 章 怎样写注释信息

目录

1. Fixed Bug

```
svn ci -m "- Fixed bug #53412 (your comment)"
```

2. Implemented

```
svn ci -m "- Implemented FR #53271, FR #52410 (Building multiple XXXX binary)"
```

3. Add

```
svn ci -m "- Add (your message)"
```

Home | Mirror | Search

Google™ Custom Search [Search]

第 35 章 代码托管

目录

1. SourceForge

Home | Mirror | Search

Google™ Custom Search Search

2. Google Code

3. GitHub

http://www.github.com/

Global setup:

Download and install Git

```
git config --global user.name "Neo Chan"
git config --global user.email bg7nyt@gmail.com
```

Next steps:

```

mkdir neo
cd neo
git init
touch README
git add README
git commit -m 'first commit'
git remote add origin git@github.com:netkiller/neo.git
git push origin master
```

Existing Git Repo?

```
cd existing_git_repo
git remote add origin git@github.com:netkiller/neo.git
git push origin master
```

Home | Mirror | Search

Google™ Custom Search Search

第 36 章 GUI

目录

1. TortoiseSVN

2. TortoiseGit