

# Subversion

---

1. 도구 개요

---

2. 설치 및 실행

---

3. 주요 기능

---

4. 활용 예제

---

# 1. 도구 개요



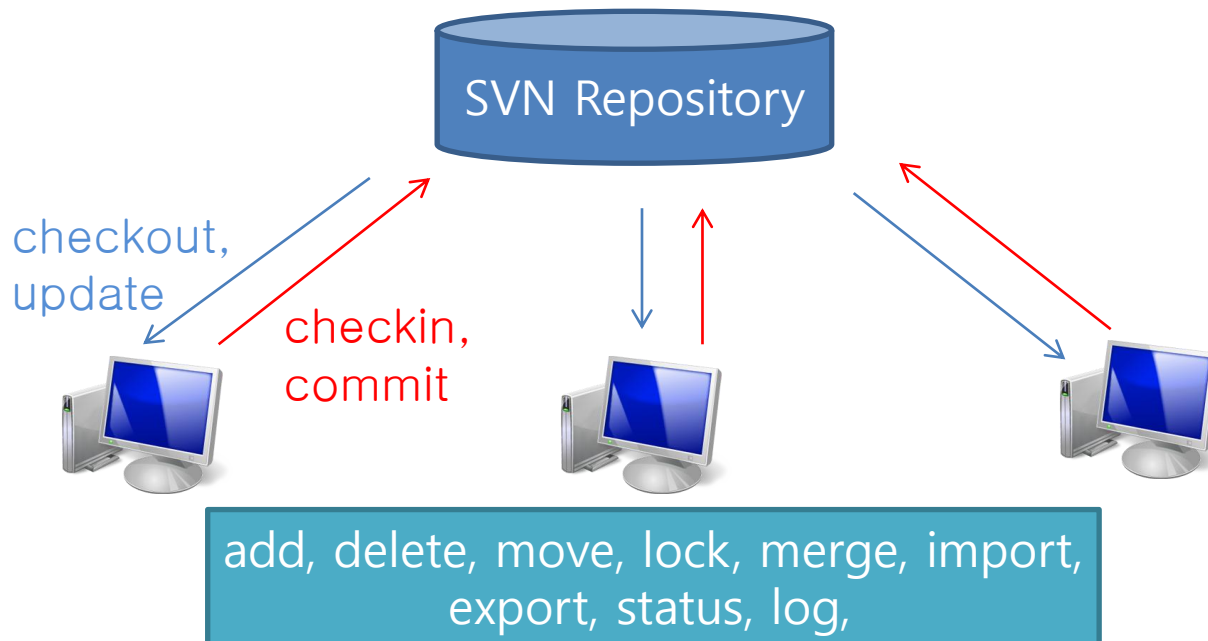
## 1.1 도구 정보 요약

도구명	Subversion ( <a href="http://subversion.apache.org/">http://subversion.apache.org/</a> )	라이선스	Apache License v2.0
소개	• CVS의 단점을 보완하여 작성된 중앙 집중형 방식의 형상 관리 도구		
특징	• 파일이나 디렉토리를 이동해도 이력 보존 • gzip 압축을 통한 저장 공간 절약 • CVS에 비해 빠른 속도 • 다양한 써드파티 GUI 도구 존재		
주요기능	• checkout/checkin, update/commit 등 서버 저장소와 클라이언트의 변경 사항 전송 • diff를 통한 파일 내용 비교 • 바이너리 문서 형상 관리 • 작업 단위의 변경 사항 관리 • atomic commit • svn:ignore를 통한 형상 관리 배제 자원 지정 기능		
실행환경	• Windows, Linux, MacOS	사전설치도구	•없음
카테고리	• 형상관리	최신버전	• v1.8.10 (2014.12)
관련도구	• CVS, Git, Mercurial, Subclipse		

# 1. 도구 개요

## 1.2 스크린 캡처 및 주요 기능

- checkout/checkin, update/commit 등 서버 저장소와 클라이언트의 변경 사항 전송
- 파일 단위가 아닌 여러 파일을 포함하는 작업 단위의 변경 사항 관리
- 텍스트 뿐 아니라 바이너리 파일도 형상 관리
- atomic commit으로 커밋 도중 오류 발생 시 롤백
- 컴파일된 오브젝트 파일이나 클래스 파일, 로그 파일 등 프로젝트 디렉터리 내에 존재하지만 형상 관리할 필요가 없는 자원은 svn:ignore로 형상 관리에서 배제 가능
- SVN은 CLI(Command Line Interface) 기반 도구로 스크린 캡처할 도구 화면은 없음



## 2. 설치 및 실행



### 세부 목차

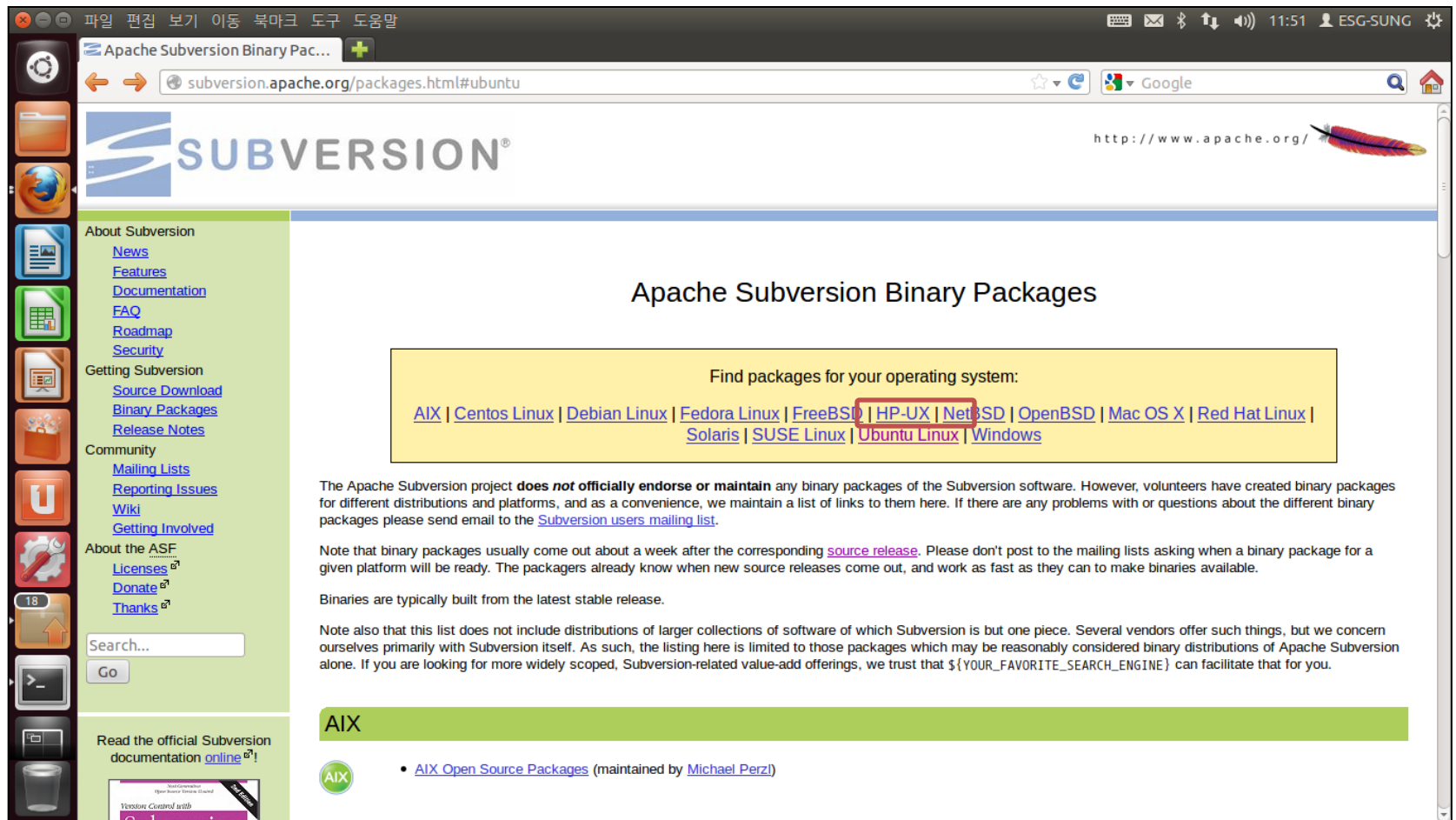
---

- 2.1 Subversion 다운받기
- 2.2 Subversion 설치 하기
- 2.3 Subversion 설치 및 확인하기

# 2. 설치 및 실행

## 2.1 Subversion 다운 받기

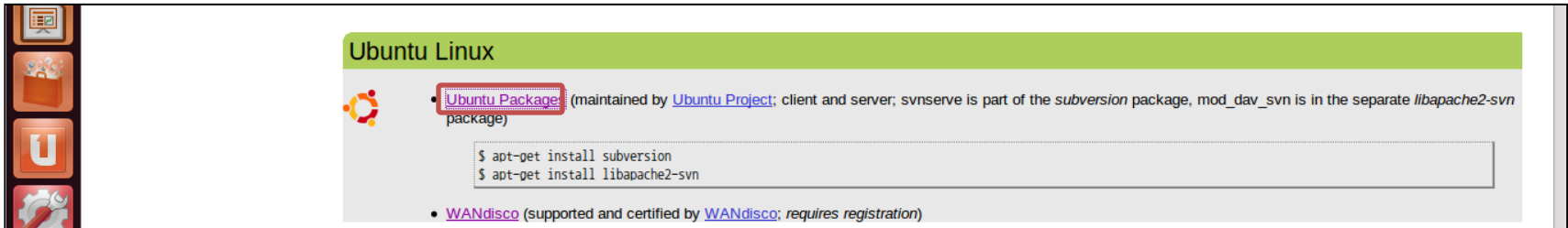
- <http://subversion.apache.org/packages.html>에서 Ubuntu Linux 클릭
  - 대부분의 개발 환경에서 Subversion을 Linux에 사용하므로, 본 매뉴얼은 Ubuntu 환경에서 기술



# 2. 설치 및 실행

## 2.1 Subversion 다운 받기

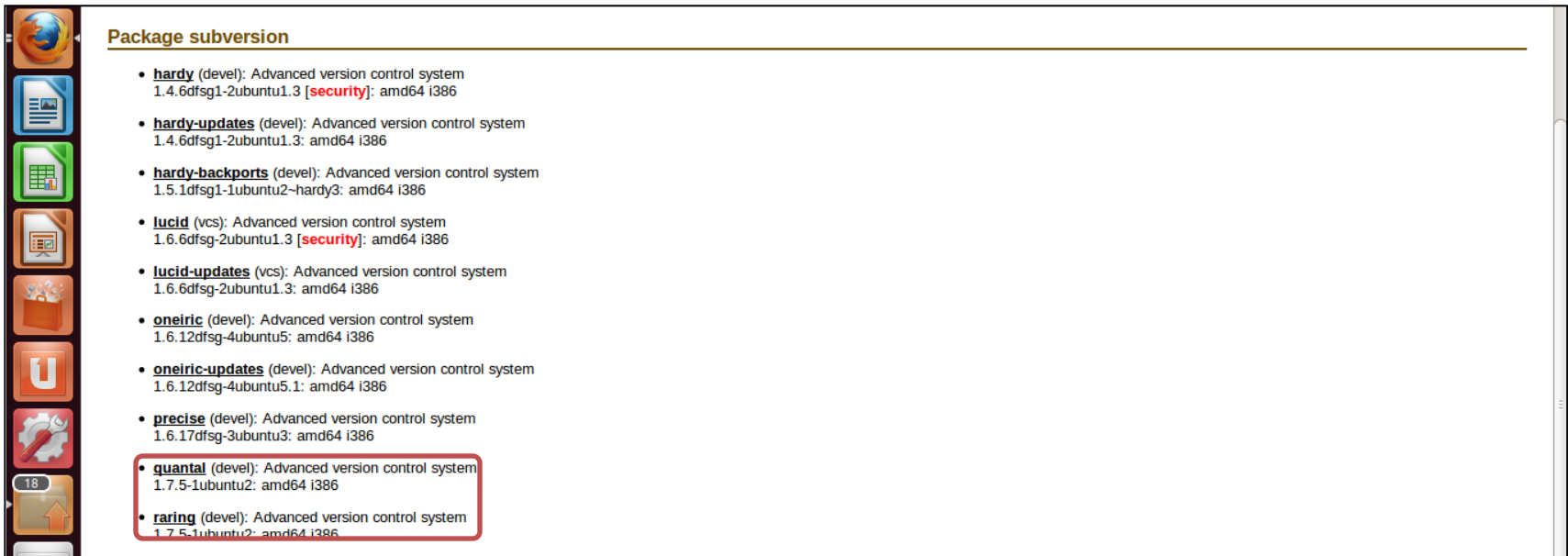
- Ubuntu Package → 미러사이트 선택



Ubuntu Linux

- **Ubuntu Package** (maintained by [Ubuntu Project](#); client and server; svnserve is part of the *subversion* package, mod\_dav\_svn is in the separate *libapache2-svn* package)

```
$ apt-get install subversion
$ apt-get install libapache2-svn
```
- [WANdisco](#) (supported and certified by [WANdisco](#); requires registration)



Package subversion

- **hardy** (devel): Advanced version control system  
1.4.6dfsg1-2ubuntu1.3 [security]: amd64 i386
- **hardy-updates** (devel): Advanced version control system  
1.4.6dfsg1-2ubuntu1.3: amd64 i386
- **hardy-backports** (devel): Advanced version control system  
1.5.1dfsg1-1ubuntu2-hardy3: amd64 i386
- **lucid** (vcs): Advanced version control system  
1.6.6dfsg-2ubuntu1.3 [security]: amd64 i386
- **lucid-updates** (vcs): Advanced version control system  
1.6.6dfsg-2ubuntu1.3: amd64 i386
- **oneiric** (devel): Advanced version control system  
1.6.12dfsg-4ubuntu5: amd64 i386
- **oneiric-updates** (devel): Advanced version control system  
1.6.12dfsg-4ubuntu5.1: amd64 i386
- **precise** (devel): Advanced version control system  
1.6.17dfsg-3ubuntu3: amd64 i386
- **quantal** (devel): Advanced version control system  
1.7.5-1ubuntu2: amd64 i386
- **raring** (devel): Advanced version control system  
1.7.5-1ubuntu2: amd64 i386

# 2. 설치 및 실행



## 2.2 Subversion 설치 하기

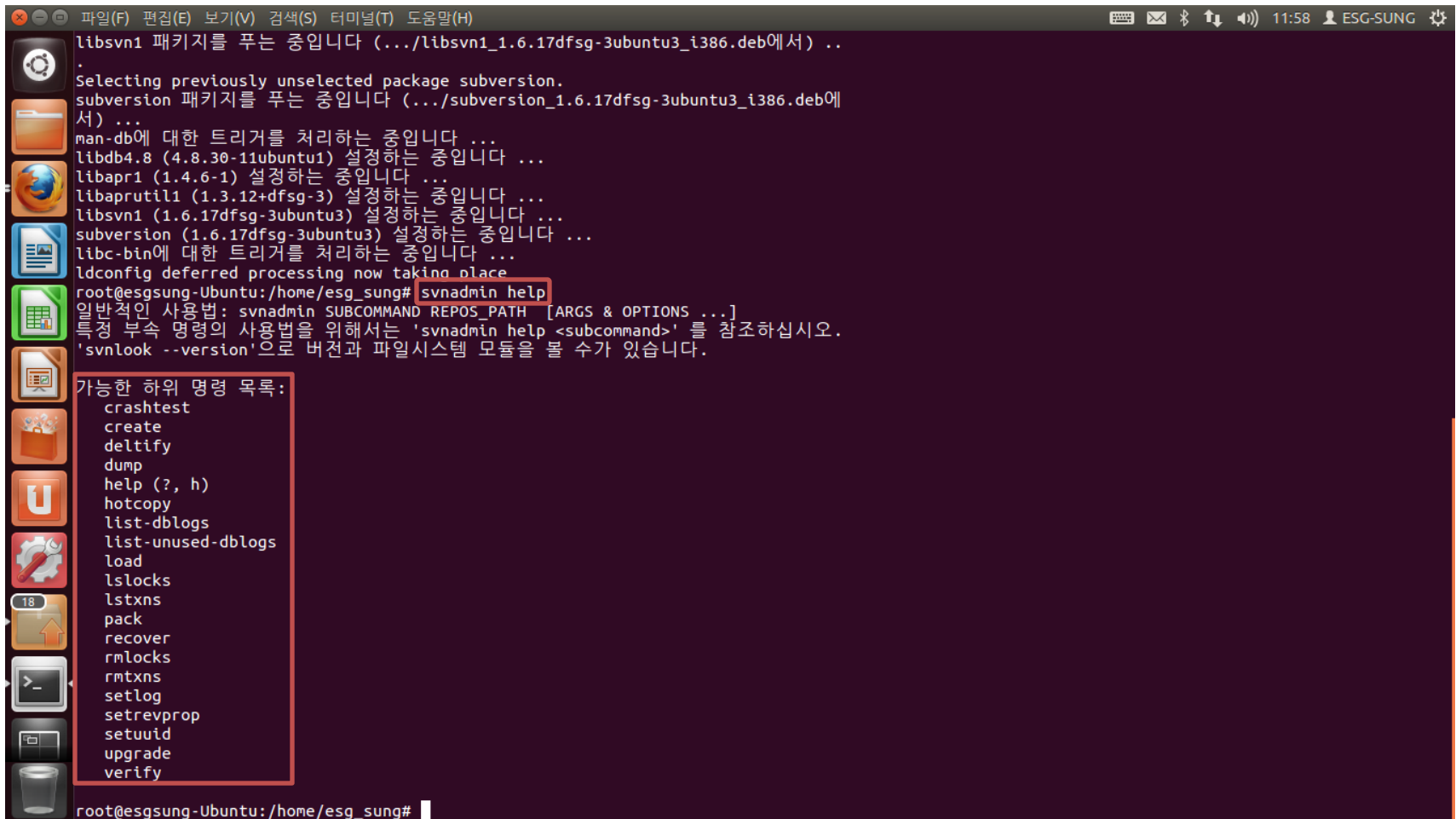
- Root권한으로 이동 → apt-get install subversion
  - Root권한에서 설치가 가능하며, apt명령을 실행하여 설치를 진행

```
esg_sung@esgsung-Ubuntu:~$ su root
암호:
root@esgsung-Ubuntu:/home/esg_sung# apt-get install subversion
패키지 목록을 읽는 중입니다... 완료
의존성 트리를 만드는 중입니다
상태 정보를 읽는 중입니다... 완료
다음 패키지가 자동으로 설치되었지만 더 이상 필요하지 않습니다:
linux-headers-3.2.0-29 linux-headers-3.2.0-29-generic-pae
이들을 지우려면 'apt-get autoremove'를 사용하십시오.
다음 패키지를 더 설치할 것입니다:
libapr1 libaprutil1 libdb4.8 libsvn1
제안하는 패키지:
subversion-tools db4.8-util
다음 새 패키지를 설치할 것입니다:
libapr1 libaprutil1 libdb4.8 libsvn1 subversion
0개 업그레이드, 5개 새로 설치, 0개 제거 및 18개 업그레이드 안 함.
2,009 k바이트 아카이브를 받아야 합니다.
이 작업 후 5,472 k바이트의 디스크 공간을 더 사용하게 됩니다.
계속 하시겠습니까 [Y/n]? y
받기:1 http://kr.archive.ubuntu.com/ubuntu/ precise/main libdb4.8 i386 4.8.30-11ubuntu1 [711 kB]
받기:2 http://kr.archive.ubuntu.com/ubuntu/ precise/main libapr1 i386 1.4.6-1 [91.3 kB]
받기:3 http://kr.archive.ubuntu.com/ubuntu/ precise/main libaprutil1 i386 1.3.12+dfsg-3 [75.4 kB]
받기:4 http://kr.archive.ubuntu.com/ubuntu/ precise/main libsvn1 i386 1.6.17dfsg-3ubuntu3 [839 kB]
받기:5 http://kr.archive.ubuntu.com/ubuntu/ precise/main subversion i386 1.6.17dfsg-3ubuntu3 [293 kB]
내려받기 2,009 k바이트, 소요시간 4초 (502 k바이트/초)
Selecting previously unselected package libdb4.8.
(데이터베이스 읽는중 ...현재 169774개의 파일과 디렉터리가 설치되어 있습니다.)
libdb4.8 패키지를 푸는 중입니다 (.../libdb4.8_4.8.30-11ubuntu1_i386.deb에서) ...
Selecting previously unselected package libapr1.
libapr1 패키지를 푸는 중입니다 (.../libapr1_1.4.6-1_i386.deb에서) ...
Selecting previously unselected package libaprutil1.
libaprutil1 패키지를 푸는 중입니다 (.../libaprutil1_1.3.12+dfsg-3_i386.deb에서)
...
Selecting previously unselected package libsvn1.
libsvn1 패키지를 푸는 중입니다 (.../libsvn1_1.6.17dfsg-3ubuntu3_i386.deb에서) ..
.
```

# 2. 설치 및 실행

## 2.2 Subversion 설치 하기

- 설치 완료 후 설치 확인 : "svnadmin help"
  - 아래와 같은 리스트가 나온다면 설치가 완료 된 것



```
libsvn1 패키지를 푸는 중입니다 (.../libsvn1_1.6.17dfsg-3ubuntu3_i386.deb에서) ..
.
Selecting previously unselected package subversion.
subversion 패키지를 푸는 중입니다 (.../subversion_1.6.17dfsg-3ubuntu3_i386.deb에
서) ...
man-db에 대한 트리거를 처리하는 중입니다 ...
libdb4.8 (4.8.30-11ubuntu1) 설정하는 중입니다 ...
libapr1 (1.4.6-1) 설정하는 중입니다 ...
libaprutil1 (1.3.12+dfsg-3) 설정하는 중입니다 ...
libsvn1 (1.6.17dfsg-3ubuntu3) 설정하는 중입니다 ...
subversion (1.6.17dfsg-3ubuntu3) 설정하는 중입니다 ...
libc-bin에 대한 트리거를 처리하는 중입니다 ...
ldconfig deferred processing now taking place
root@esgsung-Ubuntu:/home/esg_sung# svnadmin help
일반적인 사용법: svnadmin SUBCOMMAND REPOS_PATH [ARGS & OPTIONS ...]
특정 부속 명령의 사용법을 위해서는 'svnadmin help <subcommand>' 를 참조하십시오.
'svnlook --version'으로 버전과 파일시스템 모듈을 볼 수가 있습니다.

가능한 하위 명령 목록:
crashtest
create
deltify
dump
help (?, h)
hotcopy
list-dblogs
list-unused-dblogs
load
lslocks
lstxns
pack
recover
rmlocks
rmtxns
setlog
setrevprop
setuuid
upgrade
verify

root@esgsung-Ubuntu:/home/esg_sung#
```



## 2. 설치 및 실행

### 2.3 저장소 생성

---

- 초기 설정 : 저장소 생성
  - Svn을 구동하는 계정을 포함한 Main folder를 생성

```
root@hacs-laptop:/usr/bin# mkdir /home/hacs/svn
root@hacs-laptop:/usr/bin#
root@hacs-laptop:/usr/bin#
root@hacs-laptop:/usr/bin# cd /home/hacs/svn
root@hacs-laptop:/home/hacs/svn#
```

- Folder 생성 후 svnadmin도구를 생성

```
root@hacs-laptop:/home/hacs/svn# svnadmin create --fs-type fsfs sample
root@hacs-laptop:/home/hacs/svn# ls
sample
root@hacs-laptop:/home/hacs/svn# ls -al
합계 40
drwxr-xr-x  3 root root  4096 2010-11-29 17:36 .
drwx----- 40 hacs hacs 28672 2010-11-29 17:36 ..
drwxr-xr-x  6 root root  4096 2010-11-29 17:36 sample
root@hacs-laptop:/home/hacs/svn#
```

## 2. 설치 및 실행

### 2.4 접속 권한 및 계정 설정

---

- chmod 명령어를 사용하여 쓰기 권한을 추가

```
hacs@hacs-laptop:~/svn$ ls -al
합계 40
drwxr-xr-x  3 hacs vde2-net  4096 2010-11-29 17:57 .
drwx----- 40 hacs hacs     28672 2010-11-29 17:57 ..
drwxr-xr-x  6 hacs vde2-net  4096 2010-11-29 17:57 sample
hacs@hacs-laptop:~/svn$ chmod -R g+w sample
hacs@hacs-laptop:~/svn$ ls -al
합계 40
drwxr-xr-x  3 hacs vde2-net  4096 2010-11-29 17:57 .
drwx----- 40 hacs hacs     28672 2010-11-29 17:57 ..
drwxrwxr-x  6 hacs vde2-net  4096 2010-11-29 17:57 sample
hacs@hacs-laptop:~/svn$
```

- 생성된 파일을 확인 후 svnserve.conf 파일을 수정

```
hacs@hacs-laptop:~/svn$ cd sample
hacs@hacs-laptop:~/svn/sample$ ls
README.txt  conf  db  format  hooks  locks
hacs@hacs-laptop:~/svn/sample$ cd conf
hacs@hacs-laptop:~/svn/sample/conf$ ls
authz  passwd  svnserve.conf
hacs@hacs-laptop:~/svn/sample/conf$ vim svnserve.conf
```

## 2. 설치 및 실행



### 2.4 접속 권한 및 계정 설정

---

- 익명 및 인증 사용자에 대한 권한 설정
  - 익명 설정(anon-access)값에는 읽기, 인증된 사용자는 쓰기까지 설정, 계정정보는 passwd 설정 예시

```
[general]
### These options control access to the repository for unauthenticated
### and authenticated users.  Valid values are "write", "read",
### and "none".  The sample settings below are the defaults.
anon-access = read
auth-access = write
### The password-db option controls the location of the password
### database file.  Unless you specify a path starting with a /,
### the file's location is relative to the directory containing
### this configuration file.
### If SASL is enabled (see below), this file will NOT be used.
### Uncomment the line below to use the default password file.
password-db = passwd
### The authz-db option controls the location of the authorization
### rules for path-based access control.  Unless you specify a path
### starting with a /, the file's location is relative to the the
### directory containing this file.  If you don't specify an
### authz-db, no path-based access control is done.
### Uncomment the line below to use the default authorization file.
# authz-db = authz
### This option specifies the authentication realm of the repository.
### If two repositories have the same authentication realm, they should
### have the same password database, and vice versa.  The default realm
### is repository's uuid.
realm = SampleRepository
```

## 2. 설치 및 실행

### 2.4 접속 권한 및 계정 설정

---

- passwd 파일 설정
  - "passwd" file을 editor로 열고 접속을 위한 ID, Password를 입력

```
hacs@hacs-laptop:~/svn/sample/conf$ vim passwd
```

```
[users]
# harry = harryssecret
# sally = sallysecret
embedded = 1111
```

## 2. 설치 및 실행

### 2.5 접속 테스트

- Svn Server를 실행 후 테스트
  - "svnserve -d -r /home/hacs/svn"입력 → Editor 화면 → "Ctrl-X"로 나오기 → "C"

```
GNU nano 2.2.2          파일: svn-commit.tmp
--이 줄 이하는 자동으로 제거됩니다--
A   http://localhost/sample/trunk

^G 도움말 보기 ^O WriteOut   ^R 파일 읽기   ^Y 이전 페이지^K 문자열 자르 ^C 커서 위치
^X 끝내기     ^J Justify     ^W Where Is   ^V 다음 페이지^U UnCut Text ^T To Spell
```

```
hacs@hacs-laptop:~/svn$ svnserve -d -r /home/hacs/svn
No command 'svnserve' found, did you mean:
 Command 'svnserve' from package 'subversion' (main)
svnserve: command not found
로그 메시지가 변경되지 않았거나 지정되지 않았습니다
취소(A), 계속(C), 수정(E):
c
svn: OPTIONS of 'http://localhost/sample': 200 OK (http://localhost)
hacs@hacs-laptop:~/svn/sample/conf$ cd ..
```

### 2.5 접속 테스트

---

- Folder 생성 테스트
  - 계정 및 비밀번호 입력 (passwd 파일 작성에 사용한 계정 및 비밀번호)

```
hacs@hacs-laptop:~/svn/sample/conf$ svn mkdir svn://localhost/sample/trunk
로그 메시지가 변경되지 않았거나 지정되지 않았습니다
취소(A), 계속(C), 수정(E):
C
인증 영역(realm): <svn://localhost:3690> Samplesv Repository
'hacs'의 암호:
인증 영역(realm): <svn://localhost:3690> Samplesv Repository
사용자명: embedded
'embedded'의 암호:

커밋된 리비전 1.
hacs@hacs-laptop:~/svn/sample/conf$ svn list svn://localhost/sample
svn: 알 수 없는 URL 스키마입니다. 'svn://localhost/sample'
hacs@hacs-laptop:~/svn/sample/conf$ svn list svn://localhost/sample
trunk/
```

# 3. 주요 기능



## 개요 및 세부목차

---

3.1 svnserve 명령어

3.2 svnadmin 명령어

3.3 svn 명령어

# 3. 주요 기능



## 3.1 svnserve 명령어

---

- Svnserve : svn의 서버 Daemon
  - 본 서버가 실행되고 있어야 svn client 도구들이 접속 가능
    - > --daemon (또는-d) : svnserve 도구를 daemon 형태로 구동
    - > --help (-h) : 사용량 및 도움말을 표시한다.
    - > --listen-port=PORT : TCP/IP 접속시의 port 번호를 설정
    - > --root=root (-r=ROOT) : repository가 있을 Main folder경로
- 사용 예
  - /home/hacs/svn 폴더에 svn 서버를 구동하고자 하는 경우,
    - > Folder 지정을 위한 -r option과 daemon 구동을 위한-d option을 모두 설정하고 실행

```
hacs@hacs-laptop:~/svn$ svnserve -d -r /home/hacs/svn
No command 'svnserve' found, did you mean:
  Command 'svnserve' from package 'subversion' (main)
svnserve: command not found
로그 메시지가 변경되지 않았거나 지정되지 않았습니다
취소(A), 계속(C), 수정(E):
c
svn: OPTIONS of 'http://localhost/sample': 200 OK (http://localhost)
hacs@hacs-laptop:~/svn/sample/conf$ cd ..
```



# 3. 주요 기능



## 3.2 svnadmin 명령어

---

- Svn의 전반적인 관리를 위한 도구 : svnadmin
  - Svn서비스 구동 전, repository 생성, lock 제거, 전체정보 출력 등
  - svnadmin create : repository를 생성

```
#svnadmin create /var/svn/repos --fs-type fsfs
```

- svnadmin recover : repository를 제일 안정적인 version의 상태로 복구

```
#svnadmin recover /var/svn/repos/  
repository lock acquired.  
Please wait; recovering the repository may take some time...  
Recovery completed.  
The latest repos revision is 34.
```

# 3. 주요 기능



## 3.3 svn 명령어

- Svn의 명령어
  - svn add : 지정된 folder 아래에 file 또는 folder를 추가

```
#svn add foo.c  
A foo.c
```

- svnadmin verify : repository를 점검하고 최종 version을 알림

```
hanmomhanda@apexsoft-svr1 /home/svn $ svnadmin verify sample  
* Verifying repository metadata ...  
* 검토된 리비전 0.  
* 검토된 리비전 1.  
* 검토된 리비전 2.  
* 검토된 리비전 3.
```

- svn update : 지정된 folder아래의 모든 file들을 최신 version으로 update

```
#svn update  
A newdir/toggle.c  
A newdir/disclose.c  
A newdir/launch.c  
D newdir/README  
updated to revision 32.
```

# 3. 주요 기능



## 3.3 svn 명령어

- svn import : 현재 Prompt가 있는 folder 아래에 file또는 folder들을 모두 서버로 upload/Commit

```
#svn import -m "New import" myproj http://localhost/sample/trunk Adding
  myproj/sample.txt
...
Transmitting file data .....
Committed revision 16.
```

- svn blame : 해당 URL의 file에 대한 작성자 정보등의 내용들을 출력

```
apexsoft@apexsoft-svr2 ~/svnWks/sample $ svn blame newFolderBySvr2/text1.txt
4 hanmomhand 1.2 스크린 캡처 및 주요 기능
5 apexsoft 라인 하나 더 추가 이걸 다른 계정으로 commit
```

- svn checkout : 해당 svn에 있는 repository에서 file들을 모두 최신 version으로 update

```
#svn checkout file:///var/svn/repos/test mine
A mine/a
A mine/b
A mine/c
A mine/d
Checked out revision 20.
```

# 3. 주요 기능



## 3.3 svn 명령어

---

- svn cleanup : local에 있는 모든 file및 folder들을 새로 정리(별도의 응답 없음)

```
#svn cleanup
```

- svn commit : 수정 또는 신규로 올린 file이나 folder가 확정 version임을 Server에 통보

```
# svn commit --force-log --file file_under_vc.txt foo.c
Sending foo.c
Transmitting file data .
Committed revision 6.
```

- svn copy : local에 있는 file및 folder들을 서버의 repository에 복사(upload)

```
# svn cp bat.c baz.c qux.c src
A src/bat.c
A src/baz.c
A src/qux.c
```

- svn move : 파일이나 디렉토리를 다른 위치로 이동하거나 이름 변경

```
apexsoft@apexsoft-svr2 ~/svnWks/sample $ svn move newFolderBySvr2/text1.txt ./
A      text1.txt
D      newFolderBySvr2/text1.txt
```

# 3. 주요 기능



## 3.3 svn 명령어

- svn lock : repository에 있는 folder나 file에 대하여 다른 사용자가 update하지 못하도록 막음

```
#svn lock tree.jpg house.jpg  
'tree.jpg' locked by user 'harry'.  
'house.jpg' locked by user 'harry'.
```

- svn diff : repository에 있는 두 file에 대하여 비교하여 상호 틀린 부분을 출력

```
#svn diff -r 3000:3500 http://svn.collab.net/repos/svn/trunk/COMMITTERS  
Index: COMMITTERS  
-----  
---COMMITTERS (revision 3000)  
+++ COMMITTERS (revision 3500)
```

- svn mkdir : repository에 새로운 folder를 생성

```
# svn mkdir newdir  
A newdir
```

# 3. 주요 기능

## 3.3 svn 명령어

---

- svn status : 작업 사본의 파일이나 디렉터리 상태 표시

```
apexsoft@apexsoft-svr2 ~/svnWks/sample $ svn status
D      newFolderBySvr2/text1.txt
      > moved to text1.txt
A +    text1.txt
      > moved from newFolderBySvr2/text1.txt
```

- svn info : 작업 사본이나 저장소의 파일, 디렉토리의 정보 출력

```
apexsoft@apexsoft-svr2 ~/svnWks/sample $ svn info
경로: .
작업 사본의 루트 경로: /home/apexsoft/svnWks/sample
URL: svn://apexsoft-svr1/sample
Relative URL: ^/
저장소 루트: svn://apexsoft-svr1/sample
저장소 UUID: bea94cfd-7b6b-469b-a629-c4e46309401a
리비전: 5
노드 종류: 디렉토리
스케줄: 일반
마지막 수정 작업자: apexsoft
마지막 수정 리비전: 5
마지막 수정 일자: 2014-12-13 18:50:15 +0900 (2014-12-13, 토)
```

# 3. 주요 기능



## 3.4 version 1.8 기준 페지 내용

---

- neon 기반의 HTTP 클라이언트 지원 폐지
  - 대신 serf 기반의 HTTP 클라이언트 지원
  
- Berkeley DB 저장소 지원 중지
  - 1.8 이후로 FSFS 저장소에 대해서만 기능 추가 및 업데이트 진행
  - 기능 추가 및 업데이트가 없을 뿐 Berkeley DB로 운영 중인 저장소를 FSFS로 반드시 전환할 필요는 없음

# 4. 활용 예제



## 세부 목차

---

- 4.1 Subversion 폴더 생성
- 4.2 Subversion 업로드
- 4.3 Subversion 확인
- 4.4 Check Out
- 4.5 Subversion Reversion
- 4.6 Subversion lock



## 4.1 Subversion 폴더 생성

---

- 여러 명이 동시에 하나의 과제 내에서 개발하는 예제
  - Sourcefile들을 upload하고 lock을 걸어 file이 다른 사람으로부터 쓰여지는 것을 막는 처리
  - 설치과정에서 생성된 "sample" repository를 이용, trunk라는 root folder를 만들어 접속
    - > Folder 및 file들을 관리
  
- > svn mkdir을 이용하여 "cpp\_trunk"라는 folder를 생성

```
$ svn mkdir --username apache2 --password 1111 svn://localhost/sample/cpp_trunk
```

로그 메시지가 변경되지 않았거나 지정되지 않았습니다  
취소(A), 계속(C), 수정(E):

로그 메시지가 변경되지 않았거나 지정되지 않았습니다  
취소(A), 계속(C), 수정(E):  
C

커밋된 리비전 3.

# 4. 활용 예제

## 4.2 Subversion 업로드

---

- Sample folder 아래 "cpp\_trunk"라는 folder를 생성

```
$ mkdir cpp_trunk
```

- 해당 folder로 들어가서 editor를 이용, C sourcefile을 작성

```
$ pwd  
/home/apache2/sample/cpp_trunk  
$  
$  
$  
$ vi printf_func.c
```

- 작성된 file을 import 명령을 이용, svn server에 upload

```
$ svn import --username apache2 --password 1111 cpp_trunk svn://localhost/sample  
/cpp_trunk  
로그 메시지가 변경되지 않았거나 지정되지 않았습니다  
취소(A), 계속(C), 수정(E):  
C  
추가      cpp_trunk/printf_func.c  
커밋된 리비전 4.  
$
```

## 4.3 Subversion 확인

---

- List명령어로 확인
  - Upload한 file이 해당 folder(cpp\_trunk)에 정상적으로 처리되었는지 확인이 가능

```
$ svn list svn://localhost/sample/cpp_trunk
printf_func.c
$
```

- Local에서 file이 삭제가 되었거나 다른 PC에서 svn에 있는 file및 folder들을 내려 받는 것으로 가정
  - > Local에 만들었던 cpp\_trunk folder전체를 아래 그림과 같이 삭제

```
$ ls
cpp_trunk printf_func.c trunk
$ rm -r cpp_trunk
$ ls
printf_func.c trunk
$ pwd
/home/apache2/sample
$
```

### 4.4 Check Out

- Checkout 명령어로 앞서 생성하고 file을 upload하였던 cpp\_trunk folder 전체를 받음
  - Local에 삭제하였던 folder가 자동으로 생성되어 있는 것을 확인

```
$ svn checkout --username apache2 --password 1111 svn://localhost/sample/cpp_trunk
A      cpp_trunk/printf_func.c
체크아웃된 리비전 4.
$ ls
cpp_trunk  printf_func.c  trunk
$ ls -al
합계 28
drwxr-xr-x 5 apache2 hacs      4096 2010-11-30 17:42 .
drwxr-xr-x 4 apache2 apache2 4096 2010-11-30 17:36 ..
drwxr-xr-x 6 apache2 hacs      4096 2010-11-30 17:42 .svn
drwxr-xr-x 3 apache2 hacs      4096 2010-11-30 17:42 cpp_trunk
-rw-r--r-- 1 apache2 hacs      6083 2010-11-30 17:36 printf_func.c
drwxr-xr-x 3 apache2 hacs      4096 2010-11-30 17:07 trunk
$
```

### 4.4 Check Out

---

- 해당 folder에 올려두었던 "printf\_func.c" sourcefile도 정상적으로 받아짐
  - update 명령어로 다른 사용자가 file을 수정하거나 또 다른 folder나 file업로드 여부사항 확인 가능

```
$ cd cpp_trunk
$ ls
printf_func.c
$ pwd
/home/apache2/sample/cpp_trunk
$ svn update
리비전 4.
$
```

## 4.5 Subversion Reversion

---

- File 내용을 수정하고, 수정된 file을 svn에 upload 및 commit 처리
  - Editor에서 주석 한 줄을 추가

```
/*
    putchar is the only external dependency for this file,
    if you have a working putchar, just remove the following
    define. If the function should be called something else,
    replace outbyte(c) by your own function call.
*/
/* I added the comment. */
#define putchar(c) outbyte(c)
static void printchar(char **str, int c)
{
```

## 4.5 Subversion Reversion

- Commit 명령어를 이용하면 수정된 file은 자동으로 upload되고 commit 처리 상태가 됨
  - "Ctrl-X"키로 나올 수 있음

```
GNU nano 2.2.2          파일: svn-commit.tmp
--이 줄 이하는 자동으로 제거됩니다--
M   printf_func.c
[ 4 행을 읽으십시오. ]
^G 도움말 보기 ^O WriteOut ^R 파일 읽기 ^Y 이전 페이지 ^K 문자열 자르 ^C 커서 위치
^X 끝내기      ^J Justify      ^W Where Is      ^V 다음 페이지 ^U UnCut Text ^T To Spell
```

### 4.5 Subversion Reversion

---

- Reversion 상태 확인
  - Commit 처리가 되면 아래 그림과 같이 수정된 file명이 보이고 Reversion

```
$ svn commit --username apache2 --password 1111
로그 메시지가 변경되지 않았거나 지정되지 않았습니다
취소(A), 계속(C), 수정(E):
C
전송중      printf_func.c
파일 데이터 전송중 .
커밋된 리비전 5.
$
```



## 4.5 Subversion Reversion

---

- Log 명령어 : 변경된 이력을 조회
  - Version4에서 5로 변경된 내역 및 어느 계정이 변경 처리했는지 알 수 있음

```
$ svn log printf_func.c
-----
r5 | apache2 | 2010-12-01 09:23:02 +0900 (2010-12-01, 수) | 1 개의 행
-----
r4 | apache2 | 2010-11-30 17:39:42 +0900 (2010-11-30, 화) | 1 개의 행
-----
$
```

### 4.5 Subversion Reversion

---

- 해당 version에서 내부 변경 사항 확인
  - Diff 명령어를 이용하여 source에서 어떤 부분이 변경되었는지 확인

```
$ svn diff -r 4:5
Index: printf_func.c
=====
--- printf_func.c      (리비전 4)
+++ printf_func.c      (리비전 5)
@@ -22,6 +22,9 @@
     define. If the function should be called something else,
     replace outbyte(c) by your own function call.
 */
+
+/* I added the comment. */
+
#define putchar(c) outbyte(c)

static void printchar(char **str, int c)
$
```

### 4.6 Subversion lock

---

- Lock : 다른 이용자의 접근 불허
  - 해당 file을 수정 중 이라면 사전에 미리 lock을 걸어서 file을 update할 수 없도록 막음
    - > Lock 명령어로 printf\_func.c file에 대하여 lock을 걸었을 경우

```
$ svn lock --username apache2 --password 1111 printf_func.c
'printf_func.c' (은)는 'apache2'(이)가 잠겼습니다.
$
$ svn unlock --usrname apache2 --password 1111 printf_func.c
svn: invalid option: --usrname
사용법은 'svn help'를 통해 볼 수 있습니다.
$
$ svn unlock --username apache2 --password 1111 printf_func.c
'printf_func.c'(은)는 잠금이 해제 되었습니다.
$ □
```