

# Subversion

- [SVN Subversion Mini Cheat Sheet](#)

# SVN Subversion Mini Cheat Sheet

auf dem SVN-Host ein neues Repository anlegen:

```
cd /var/www/svn.meinserver.de/svn
svnadmin create meinProjekt
```

erster Import in das neue Repository:

```
svn import localDirectory https://svn.meinserver.de/svn/meinProjekt -m "First Import"
```

Jetzt muss das neue Projekt gleich ausgecheckt werden, damit auch das lokale Verzeichnis in die Versionsverwaltung kommt:

```
# erstmal das vorhandene Verzeichnis umbenennen:
mv meinProjekt meinProjekt.old
svn co https://svn.meinserver.de/svn/meinProjekt meinProjekt
```

ausgechecktes Repostory updaten:

```
svn up
```

Änderungen einchecken:

```
svn ci
```

lokale Änderungen rückgängig machen:

```
svn revert <filename>
```

Executable Property setzen:

```
svn ps svn:executable on *.sh
```

## Keyword Substitution

aktivieren:

```
svn ps svn:keywords 'Id' *.sh
```

bei jedem Checkin wird nun \$Id\$ durch einen String „Id: subversion-config 122 2013-01-08 15:01:58Z magenbrot \$“ ersetzt.

Um das Property bei jedem neuen File automatisch zu setzen muss das Subversion-Configfile angepasst werden. Dies liegt üblicherweise in ~/.subversion/config

Hier mal mein aktuelles als Beispiel:

```
#
# grundlegende Subversion Konfiguration
```

```
# wird in ~/.subversion/config abgelegt
#
# $Id: subversion-config 122 2013-01-08 15:01:58Z magenbrot $
#

[auth]
# store-passwords = no

[miscellany]
enable-auto-props = yes

[auto-props]
*.sh = svn:keywords=Id;svn:eol-style=native;svn:executable
*.py = svn:keywords=Id;svn:eol-style=native;svn:executable
*.exe = svn:executable
*.pl = svn:keywords=Id;svn:eol-style=native;svn:executable
*.pm = svn:keywords=Id;svn:eol-style=native
*.php = svn:keywords=Id;svn:eol-style=native
*.pas = svn:keywords=Id;svn:eol-style=native
*.c = svn:keywords=Id;svn:eol-style=native
*.cpp = svn:keywords=Id;svn:eol-style=native
*.h = svn:keywords=Id;svn:eol-style=native
Makefile = svn:eol-style=native
*.txt = svn:keywords=Id;svn:eol-style=native
*.png = svn:mime-type=image/png
*.jpg = svn:mime-type=image/jpeg
*.gif = svn:mime-type=image/gif
```

## Umzug eines Repositories

Der folgende Befehl stellt ein ausgechecktes Repository von einer ungesicherten Verbindung auf einen SSL geschützten Vhost um.

```
svn switch --relocate http://meinserver.de/svn/tools https://meinserver.de/svn/tools
```

Vorher sowie nachher lassen sich die Einstellungen über `svn info` abfragen:

```
$ svn info

Path: .
URL: https://meinserver.de/svn/tools
Repository Root: https://meinserver.de/svn/tools
Repository UUID: 3b6bfc9a-cf72-4c89-941c-58777a1bedca
Revision: 178
Node Kind: directory
Schedule: normal
Last Changed Author: magenbrot
Last Changed Rev: 178
Last Changed Date: 2013-06-05 14:57:14 +0200 (Mi, 05 Jun 2013)
```

## libapache2-svn und WebSVN über die gleiche Domain

Hier eine kurze Anleitung, wie man das Apache-Modul für Subversion und WebSVN über die gleiche Domain handlen kann. Ich setze bei mir ISPconfig ein. Damit dann alle User entsprechend zugreifen können, müssen ein paar Dinge konfiguriert werden.

Damit sowohl der Zugriff über WebDAV (läuft als User www-data) und WebSVN (wird mit dem PHP-User ausgeführt (web256)) funktioniert, müssen wir auf Filesystem ACLs zurückgreifen. Ist damit leider etwas umständlich und das Dateiesystem muss mit einer zusätzlich Mountoption gemounted werden.

SVN-Modul aktivieren:

```
a2enmod dav_svn
```

Wer auch ISPconfig verwendet kommentiert den Inhalt von /etc/apache2/sites-enabled/dav\_svn.conf am besten aus. Das machen wir dann direkt über den VHost.

Apache neu starten und danach den Vhost konfigurieren. Meine Beispieldomain ist svn.meinserver.de.

TODO

Ergänzung um „acl“ in /etc/fstab, z.B.

```
/dev/xvda1      /              ext4    errors=remount-ro,acl,noatime,nodiratime,usrjquota=aquot
```

Filesystem remounten:

```
mount -o remount /
```

Und User www-data und web256 gleichermaßen auf die Verzeichnisse berechtigen:

```
setfacl --set user::rwx,group::r-x,other:r-x,group:www-data:rwx,default:user:web256:rwx,default:
```