

Inhaltsverzeichnis

- Subversion-Repository sichern und kopieren mit SVNSync** 3
- Sichern*** 3
- Kopieren*** 4
- Schritt 1: Kopieren des Repositories 4
- Schritt 2: Umstellen der lokalen Arbeitskopie auf den neuen Server 4
- Migrieren von SVN zu Git(Hub) 5

Subversion-Repository sichern und kopieren mit SVNSync

Sichern

Mit `svnsync` lässt sich ein remote Repository zur Datensicherung (u.a.) auf die eigene Platte übertragen. Die dafür notwendigen Schritte in der richtigen Reihenfolge mit der richtigen Syntax der URLs, Pfade usw. können einem aber schon zu schaffen machen, darum hier wieder mal ein „so hat's bei mir geklappt“:

```
mkdir skds-script-repository
```

```
svnadmin create skds_repository_svnsync/
```

Achtung: Voller Pfad mit 3! / hinter file: !

```
svn info file:///mnt/smbdata/Server_Backups/skds-script-repository/skds_repository_svnsync
```

```
gedit skds_repository_svnsync/hooks/pre-revprop-change
```

containing

```
#!/bin/sh
USER="$3"
if [ "$USER" = "stko" ]; then exit 0;
fi
echo "Only the svnsync user can change revprops" >&2
exit 1
```

Ausführbar machen

```
chmod +x skds_repository_svnsync/hooks/pre-revprop-change
```

Lokales Repository initialisieren:

```
svnsync init --username stko file:///mnt/smbdata/Server_Backups/skds-script-repository/skds_repository_svnsync
https://svn2.hosted-projects.com/stko/skdsscripts
```

mit der Rückmeldung

```
Eigenschaften für Revision 0 kopiert.
```

Dann kommt der eigentliche Kopiervorgang:

```
svnsync sync --username stko file:///mnt/smbdata/Server_Backups/skds-script-repository/skds_repository_svnsync
```

Und am Ende transferiert man die Repository- Struktur noch in ein Dump- File:

```
svnadmin dump skds_repository_svnsync/ > skds_repository_svnsync.dvndump
```

Kopieren

Wechselt man den Provider, dann muß das SVN- Repository ja irgendwie vom alten zum neuen Provider rübergeschaufelt werden. Hier mal am Beispiel von SourceForge zu GoogleCode: ¹⁾

Schritt 1: Kopieren des Repositories

```
svnadmin create /home/steffen/SF_Svn_Backup

svnsync init --username stko https://oobd.svn.sourceforge.net/svnroot/oobd
file:///home/steffen/SF_Svn_Backup
svnsync sync --username stko https://oobd.svn.sourceforge.net/svnroot/oobd

joe /home/steffen/SF_Svn_Backup/hooks/pre-revprop-change # make it 'exit 0'
chmod +x /home/steffen/SF_Svn_Backup/hooks/pre-revprop-change

svnsync init --username steffen.koehlers.de@gmail.com
file:///home/steffen/SF_Svn_Backup https://oobd.googlecode.com/svn
svnsync sync --username steffen.koehlers.de@gmail.com
file:///home/steffen/SF_Svn_Backup
```

Falls die Übertragung mal abbrechen sollte, muß man erst wieder den Transfer-Lock löschen, bevor man weitermachen kann („Couldn't get lock on destination repos“):

```
svn proplist --revprop -r0 https://oobd.googlecode.com/svn
```

Schritt 2: Umstellen der lokalen Arbeitskopie auf den neuen Server

Das Wichtigste am Anfang: Auf den neuen Server umstellen mit dem bekannten SVN Befehl „relocate repository URL“ **geht nicht!**. Der funktioniert nämlich nur dann, wenn der alte Server sich verschoben hat. Ein neu aufgelegter Server dagegen hat eine andere ID als der Alte, und das mag SVN dann gar nicht...

Statt dessen geht das z.B. so:

- erstmal ein Backup des Hauptverzeichnis der lokalen Arbeitskopie machen (sicherheitshalber, man weiß ja nie...)
- dieses Hauptverzeichnis umbenennen (also MeineArbeitskopie → MeineArbeitskopieAlt)

- vom neuen Repository aus auschecken, dabei auf die Pfadangaben achten, dass der neue Verzeichnisbaum auch wieder so heißt und genauso aussieht wie der Alte.
- im alten Hauptverzeichnis über alle Unterordner alle Verzeichnisse suchen und löschen, die `.svn` heißen. **Wichtig: Wirklich alle löschen!**
- dann den gesamten Inhalt des alten Hauptverzeichnisses ins neue Hauptverzeichnis rüberkopieren.

So gehen nämlich z.B. lokale Config- Dateien und bereits gemachte Änderungen nicht verloren, die ja natürlich nicht im frisch downgeloadeten Repository enthalten sind.

Das war die händische Anleitung für alle Windows-Abhängigen. Unter Unix geht natürlich auch das etwas einfacher, auch hier wieder exemplarisch für ein GoogleCode- Repository. Das Script wird mit dem GoogleAccount als Parameter aufgerufen.

Aber vor dem Script erst doch noch die alten Daten sichern

```
cp -r oobd oobd_orig
```

Dann kommt das Script:

[migrateGoogleCode.sh](#)

```
mv oobd oobd_SF_old
svn checkout https://oobd.googlecode.com/svn/ oobd --username $1
find oobd_SF_old -name ".svn" -exec rm -rf {} \;
cp -r oobd_SF_old/* oobd
```

und Fertig - Jetzt kann man in der neuen Arbeitskopie wieder genauso arbeiten wie vorher im Alten



Migrieren von SVN zu Git(Hub)

Wenn einen das Schicksal trifft und man muß vom aktbekanntesten SVN auf Git wechseln, muß man zwei Aspekte beachten: Zum einen natürlich den Übergang vom alten Repository auf das Neue, zum anderen aber auch sollte man nicht vergessen, daß man ja normalerweise in der lokalen svn- Arbeitskopie noch einen Haufen unversionierter Dateien hat, die man zwar einerseits braucht, aber die nichts im Repository zu suchen haben.

Das folgende Script nimmt ein bestehendes SVN- Repository, legt davon ein Backup an, checkt stattdessen das neue bereits bestehende GIT- Repository unter dem gleichen Namen aus und kopiert die ganzen unversionierten Dateien aus dem alten SVN- ins Git- Repository.

[migrate_svnDir2Github.sh](#)

```
#!/bin/bash

function runcommand {
    "$@"
}
```

```
local status=$?
if [ $status -ne 0 ]; then
    echo "error with $1" >&2
fi
return $status
}

PROGRAMNAME=$(basename $0)
if [ $# -ne 2 ] ;
then echo "Usage $PROGRAMNAME <gitusername> <svndirname>"
echo "$PROGRAMNAME creates a <svndirname>_svnbackup Dir"
echo "and a <svndirname>_unversioned Dir"
echo "then it clones your Github repository by 'clone
https://github.com/<gitusername>/<svndirname>.git'"
echo "and finally copies all unversioned files from the old svn
data into the new git directory"
exit
fi
if [ -d $2_svnbackup ] ;
then echo "$2_svnbackup does already exist !"
echo "Did the program maybe run before? Please check before try
$PROGRAMNAME again"
exit
fi
if [ -d $2_unversioned ] ;
then echo "$2_unversioned does already exist."
echo "For security, this need to be removed manually before try
$PROGRAMNAME again"
exit
fi
echo "Create directory copies:"
runcommand cp -r $2 $2_unversioned
runcommand mv $2 $2_svnbackup
echo "Clone repostitory:"
runcommand git clone https://github.com/$1/$2.git
echo "remove svn artifacts:"
runcommand cd $2_unversioned
runcommand svn list -R | xargs rm
find . -name ".svn" -exec rm -rf {} \; 2>/dev/null
runcommand cd ..
echo "copy remaining unversioned files into new git repository:"
runcommand cp -r $2_unversioned/* $2
echo "Check results between old svn directory and new git repository:"
diff -r -q $2_svnbackup $2
```

1)

see

http://code.google.com/p/support/wiki/SubversionFAQ#How_do_I_import_an_existing_Subversion_repository?

From:

<http://koehlers.de/wiki/> - **Steffen Köhlers Online- Bastelbuch**

Permanent link:

<http://koehlers.de/wiki/doku.php?id=pc:svnsync>

Last update: **2015/04/05 08:54**

