

Inhaltsverzeichnis

SVNC Skript

Skript zum Anlegen, Löschen, ... von Subversion Repositories

svnc

```
#!/bin/bash
#####
#####
#
# Wrapper Script for svn command.
#
#####
#####
#
# JHE Development / IT-Service Huber,
# Chiemgastr. 20,
# 83552 Evenhausen,
# Germany
#
# Version: 1.2
# Date:    03.09.2014
#####
#####

SVNBASE="/var/local/svn"
SVNBASEURL="file:///SVNBASE"
SVNEXEC="/usr/bin/svn"
CONFIGDIR="/etc/subversion"
WEBCONFIGDIR="/var/www/vhosts/it-service-
huber.de/subdomains/svn/include"
BACKUPDIR="/private-backup/auto_backup/subversion"
BACKUPSCRIPT="/private-backup/auto_backup/subversion/runSvnBackup.sh"
REPOFILE="$WEBCONFIGDIR/repo_list.php"
SVNADMINUSER="admin"
SVNADMINPASSWD="*****"
AUTHZFILENAME="dav_svn.authz"
USERFILENAME="dav_svn.passwd"
GROUPDEVELOPER="developer"

quit()
{
    clear
    exit
}
```

```
}

interactiveSelectCommands()
{
    clear
    echo ""
    echo "  Subversion (SVN) Administration  "
    echo "  ===== "
    echo ""

    printf " 1 - empty Erstellt leeres Repository in SVN\n"
    printf " 2 - project Erstellt neues Projekt in SVN mit\n"
    Grundstruktur\n"
    printf " 3 - delete Löscht Repository (ACHTUNG! Daten für immer\n"
    verloren)\n"
    printf " 4 - dump Repository dump\n"
    printf " 5 - load Repository wiederherstellen\n"
    printf " 6 - info Zeige Repository Infomrationen\n"
    printf " 7 - log Zeige Repository Logs\n"
    printf " 8 - list Zeige Repository Inhalt\n"
    printf " u - adduser Erstelle neuen SVN-Benutzer oder aendere\n"
    Passwort\n"
    printf " t - tag/branch Erstelle einen Tag oder Branch\n"

    echo ""
    echo "  editCfg  - Konfigurations-Dateien bearbeiten"
    echo ""
    echo "  q - Abbruch"
    echo ""

    printf " Auswahl: "
    read sel

    case $sel in
        1|empty) COMMAND="empty"
            ;;
        2|project) COMMAND="project"
            ;;
        3|delete) COMMAND="delete"
            ;;
        4|dump) COMMAND="dump"
            ;;
        5|load) COMMAND="load"
            ;;
        6|info) COMMAND="info"
            ;;
    esac
}
```

```
7|log) COMMAND="log"
;;
8|list) COMMAND="list"
;;
u|adduser) COMMAND="adduser"
;;
t|tag|branch) COMMAND="copy"
;;
reconfig) COMMAND="reconfig"
;;
editCfg) COMMAND="editCfg"
;;
q)
    quit
    ;;
esac

return
}

getRepo()
{
    clear
    if [ "$1" = "" ]
    then
        echo ""
        echo " Repository-Auswahl"
        echo " ===== "
        echo ""

        repos=`find $SVNBASE -maxdepth 1 -type d -printf "%f\n"`
        cnt=0

        for srepo in $repos;
        do
            if [ "$cnt" = "0" ]
            then
                repolist[cnt]=$srepo
            else
                printf " $cnt:\t${srepo} \n"
                repolist[cnt]=$srepo
            fi
            ((cnt=$cnt+1));
        done

        echo ""
        echo " q - Abbruch"
```

```
    echo ""

    printf "\n Auswahl: "
    read sel

    if [ "$sel" = "q" ]
    then
        quit
    fi

    REPO=${repolist[$sel]}
    echo " -> $REPO"
    # read hallo
fi

return
}

createRepo()
{
    if [ "$1" = "" ]
    then
        printf " Repository Name: "
        read REPO
    else
        REPO=$1
    fi

    svnadmin create $SVNBASE/$REPO

    printf " Repository Titel: "
    read title
    if [ "$title" = "" ]
    then
        title=$REPO
    fi

    chown -R www-data:www-data $SVNBASE/$REPO

    return
}

createStructure()
{
    REPO=$1
    svn co file://$SVNBASE/$REPO /tmp/$REPO --username $SVNADMINUSER
--password $SVNADMINPASSWD
}
```

```
OldPWD=`pwd`
cd /tmp
cd $REPO
mkdir tags
mkdir work
mkdir public
mkdir work/docs
mkdir work/source
touch work/docs/Anforderungen.xls
svn add *
svn commit -m "Projekt $REPO angelegt" --username $SVNADMINUSER -
-p password $SVNADMINPASSWD
cd ..
rm -rf $REPO
cd $OldPWD

return
}

registerRepo()
{
    REPO=$1

    # register in WebSVN
    printf "\$config->addRepository('\$title\',"
    '\file://$SVNBASE/$REPO\');\n" >> $REPOFILE

    case $COMMAND in
        empty)
            ;;
        project)
            printf "\$config->setMinDownloadLevel(1, '\$title\');\n" >>
$REPOFILE
            ;;
    esac

    printf "\n" >> $REPOFILE

    # register in AUTHZ file
    printf "\n[$REPO:] \n@$GROUPDEVELOPER = rw\n" >>
$CONFIGDIR/$AUTHZFILENAME
    cp -f $CONFIGDIR/$AUTHZFILENAME $WEBCONFIGDIR/$AUTHZFILENAME

    # register for backup
    printf "backupSVN '$REPO'\n" >> $BACKUPSCRIPT

    return
}
```

```
}

editSvnFiles()
{
    mcedit $REPOFILE
    mcedit $CONFIGDIR/$AUTHZFILENAME
    cp -f $CONFIGDIR/$AUTHZFILENAME $WEBCONFIGDIR/$AUTHZFILENAME

    return
}

unregisterRepo()
{
    printf "Bitte manuell austragen...\n";
}

deleteRepo()
{
    REPO=$1

    printf " LÖSCHEN bitte mit JA bestätigen: "
    read ja
    if [ "$ja" = "JA" ]
    then
        rm -rf $SVNBASE/$REPO
        unregisterRepo $REPO
        echo " $REPO unwiederruflich gelöscht!"
    fi

    return
}

dumpRepo()
{
    REPO=$1
    FILE=$2

    if [ "$FILE" = "" ]
    then
        FILE=$BACKUPDIR/$REPO/dump_$REPO

        if [ ! -d "$BACKUPDIR/$REPO" ]; then
            mkdir $BACKUPDIR/$REPO
        fi
    fi

    svnadmin dump "$SVNBASE/$REPO" > "$FILE"
}
```

```
    gzip -f9 $FILE

    return
}

loadRepo()
{
    REPO=$1
    FILE=$2

    if [ "$FILE" = "" ]
    then
        if [ -e "$BACKUPDIR/dump_$REPO" ]
        then
            svnadmin load $SVNBASE/$REPO <
$BACKUPDIR/$REPO/dump_$REPO
        else
            ls -l $BACKUPDIR/$REPO
            printf " Dateinamen bitte angeben (ohne Endung .gz): "
            read file
            gunzip -kc $BACKUPDIR/$REPO/$file >
$BACKUPDIR/$REPO/dump.tmp
            svnadmin load $SVNBASE/$REPO < $BACKUPDIR/$REPO/dump.tmp
            rm $BACKUPDIR/$REPO/dump.tmp
            printf "\n"
        fi
    else
        gunzip -kc $FILE > $BACKUPDIR/$REPO/dump.tmp
        svnadmin load $SVNBASE/$REPO < $BACKUPDIR/$REPO/dump.tmp
        rm $BACKUPDIR/$REPO/dump.tmp
        printf "\n"
    fi

    return
}

addUser()
{
    USER=$1
    PASSWD=$2

    if [ "$USER" = "" ]
    then
        echo
        cat $CONFIGDIR/$USERFILENAME
        echo
        printf " Benutzername (Login): "
```

```

        read USER
    fi

    if [ "$PASSWD" = "" ]
    then
        printf " Passwort für \"$USER\": "
        read PASSWD
    fi

    htpasswd -b $CONFIGDIR/$USERFILENAME $USER $PASSWD

    mcedit $CONFIGDIR/$AUTHZFILENAME

    cp -f $CONFIGDIR/$USERFILENAME $WEBCONFIGDIR/$USERFILENAME
    cp -f $CONFIGDIR/$AUTHZFILENAME $WEBCONFIGDIR/$AUTHZFILENAME

    return
}

cpRepo()
{
    clear
    REPO=$1

    cpDirList=`svnlook tree $SVNBASE/$REPO/ --full-paths | egrep
"/$"`
    cnt=0

    printf "Wähle Quell- und Ziel-Verzeichnis von $REPO:
\n=====\\n"
    for sdir in $cpDirList;
    do
        if [ "$cnt" = "0" ]
        then
            dirlist[$cnt]=$sdir
        else
            printf " $cnt:\t${sdir} \\n"
            dirlist[$cnt]=$sdir
        fi
        ((cnt=$cnt+1));
    done

    echo ""
    echo " q - Abbruch"
    echo ""

    printf " Quelle: "

```

```
read SOURCE
if [ "$SOURCE" = "q" ]
then
    quit
fi
SOURCE=${dirlist[$SOURCE]}

printf " -> $SOURCE \n Ziel: "
read TARGET
if [ "$TARGET" = "q" ]
then
    quit
fi
TARGET=${dirlist[$TARGET]}

printf " -> $TARGET \n Tag/Branch-Name: "
read NAME
if [ "$NAME" = "q" ]
then
    quit
fi

svn copy $SVNBASEURL/$REPO/$SOURCE
$SVNBASEURL/$REPO/$TARGET/$NAME -m "Creating tag/branch $TARGET/$NAME
from $SOURCE"

return
}

reconfig()
{
    clear
    echo -e "\nInstall svnc script ... "
    cp $BACKUPDIR/svnc /bin/svnc
    chmod a+x /bin/svnc
    chown root:root /bin/svnc

    echo -e "Copy authz files from $CONFIGDIR to $WEBCONFIGDIR ..."
    \n"
    cp -f $CONFIGDIR/$USERFILENAME $WEBCONFIGDIR/$USERFILENAME
    cp -f $CONFIGDIR/$AUTHZFILENAME $WEBCONFIGDIR/$AUTHZFILENAME

    exit
}

if [ ! -e "/bin/svnc" ]
```

```
then
    reconfig
fi

COMMAND=$1

if [ "$COMMAND" = "" ]
then
    interactiveSelectCommands
else
    shift
fi

REPO=$1
FILE=$2

echo "$COMMAND $REPO $FILE"

case $COMMAND in
    empty)
        createRepo $REPO
        registerRepo $REPO
        editSvnFiles
        ;;
    project)
        createRepo $REPO
        registerRepo $REPO
        createStructure $REPO
        editSvnFiles
        ;;
    delete)
        getRepo $REPO
        deleteRepo $REPO
        editSvnFiles
        ;;
    dump)
        getRepo $REPO
        shift
        dumpRepo $REPO $FILE
        ;;
    load)
        getRepo $REPO
        shift
        loadRepo $REPO $FILE
        ;;
    info)
```

```
    getRepo $REPO
    svn info $SVNBASEURL/$REPO --username $SVNADMINUSER --
password $SVNADMINPASSWD
    ;;
log)
    getRepo $REPO
    svn log $SVNBASEURL/$REPO --username $SVNADMINUSER --password
$SVNADMINPASSWD
    ;;
list)
    getRepo $REPO
    svn list -v --depth files --recursive $SVNBASEURL/$REPO --
username $SVNADMINUSER --password $SVNADMINPASSWD
    ;;
adduser)
    # REPO = Benutzername, FILE = Passwort
    addUser $REPO $FILE
    ;;
copy)
    getRepo $REPO
    cpRepo $REPO
    ;;
reconfig)
    reconfig
    ;;
editCfg)
    editSvnFiles
    ;;
quit)
    quit
    ;;
esac
```