

## A backup volume switcher for Apple's TimeMachine

So here's another piece of code I hacked up tonight.

Since I'm roaming around with my MacBook every now and then, the need arised, that I would need to switch my TimeMachine destination volumes based on location.

So while in the office, I'd like to backup to my external USB drive there.

Being at my home office, i'd like to backup to my NAS, while on the road, I'd love to habe my external mobile drive to kick in (and yes, I know about the "mobile backup feature" of OS X Lion, but that's not the point ...)

So I hacked up a script, which will check if a TimeMachine volume is actually connected or not. If none is connected, it will check which targets are available by either looking for connected USB drives (by the use of "diskutil") or by checking connectivity to specified NAS servers. If it sees necessity to switch volumes it will also force creating an immediate backup in that case.

So, below is the code. I hope the comment section is clear enough. You'd need to change the USB\_Volumes and NET\_Volumes sections to fit your needs.

And yes, it works with the TimeMachine mobile feature enabled or disabled, that makes no difference.

```
#!/bin/bash
#
# tm_switcher -- a backup volume switchr for Apple's TimeMachine
#
# #####
# This script will help to automatically switch backup destination
# on Apple's TimeMachine according to available destination volumes.
# You can have one or more of both USB and network devices, of which
# the first available destination is used
# #####
# released to the public "as-is" under the terms of the GPL Version 2
# #####
# r0.1 2012/01/25
# - initial release

# specify zero or more USB volumes as backup destinations
# you must give only the volume label
# specify multiple volumes like this:
# USB_Volumes=("Volume1" "Volume2")
#
USB_Volumes=("My Passport")

# specify zero or more NETWORK volumes as backup destinations
# you must give the full AFP volume path
# specify multiple volumes identically as described
# in the "USB Volumes" section above
#
NET_Volumes=("afp://TMUsername:TMPasswOrd@HOSTorIPAddress/TMShare")

# get current TimeMachine directory
#
current_tm_dir=`/usr/bin/tmutil machinedirectory`
```

```
# check if we are disconnected
#
[ "$current_tm_dir" == "" ] && tm_status=CONNECT_NONE

# check if we are connected to an USB disk
#
for volume in "${USB_Volumes[@]}"; do
  echo $current_tm_dir | grep -e "$volume" > /dev/null 2>&1
  [ "$?" == "0" ] && tm_status=CONNECT_USB
done

# check if we are connected to an Network Volume (TimeCapsule)
#
for volume in "${NET_Volumes[@]}"; do
  _volume_basename=`basename ${volume}`
  /sbin/mount | grep " on /Volumes/${_volume_basename}" > /dev/null 2>&1
  [ "$?" == "0" ] && tm_status=CONNECT_NET
done

# check if we need to switch TimeMachine locations
#
if [ "${tm_status}" == "CONNECT_NONE" ]; then
  echo "NOTICE: TimeMachine is disconnected, current status is: ${tm_status}"

  # check if we can reach any of the NET_Volumes hosts given
  #
  for volume in "${NET_Volumes[@]}"; do
    _dest_server=`echo $volume | grep -P -o '@([[:digit:]]{1,3}\.[:digit:]]{1,3}\.[:digit:]]{1,3}\.[:digit:]]{1,3}(b.*b.?)+)' | sed -E
    's:(@/):g`

    # always recheck that TimeMachine is not yet connected
    #
    if [ "${tm_status}" == "CONNECT_NONE" ]; then
      # perform a simple connectivity test
      ping -c 1 $_dest_server > /dev/null 2>&1
      if [ "$?" == "0" ]; then
        echo "NOTICE: We found this volume to be available for backups: $volume"
        echo "    We will be switching TimeMachine volumes now ..."
        sudo tmutil setdestination "$volume"
        tm_status=CONNECT_NET
      fi
    fi
  done

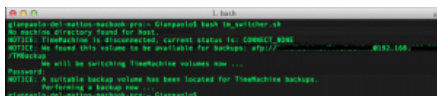
  # check if we can reach any of the NET_Volumes hosts given
  #
  for volume in "${USB_Volumes[@]}"; do
```

```
# always recheck that TimeMachine is not yet connected
#
if [ "${tm_status}" == "CONNECT_NONE" ]; then
# check if the volume is available
diskutil info "$volume" > /dev/null 2>&1
if [ "$?" == "0" ]; then
# try to mount the volume
diskutil mount "/Volumes/$volume" > /dev/null 2>&1
if [ "$?" == "0" ]; then
sleep 5
echo "NOTICE: We found this volume to be available for backups: $volume"
echo "    We will be switching TimeMachine volumes now ..."
sudo tmutil setdestination "/Volumes/$volume"
tm_status=CONNECT_USB
fi
fi
fi
done

# bail out if we failed on finding a proper backup volume
#
if [ "${tm_status}" == "CONNECT_NONE" ]; then
echo "ERROR: Failed in finding a TimeMachine Voloume on USB or on the LAN available for backups."
echo "    Exiting now."
exit
else
echo "NOTICE: A suitable backup volume has been located for TimeMachine backups."
echo "    Starting backup now ..."
tmutil startbackup
fi
else
if [ "${tm_status}" == "CONNECT_USB" -o "${tm_status}" == "CONNECT_NET" ]; then
echo "NOTICE: TimeMachine is already connected, current status is: ${tm_status}"
echo "    TimeMachine destinations will not be switched!"
else
echo "ERROR: Unable to figure current TimeMachine status."
fi
fi

exit
```

And here's the script in action:



To complement this it's a great idea to add this to Launch Daemon to have this script run on a recurring basis, so you wont need to run it manually.