

[autofs](#), [cifs](#), [filesystem](#), [howto](#), [linux](#), [mount](#), [mounting](#), [samba](#), [smb](#), [suse](#), [tumbleweed](#)



2022-11-30: Please note that I have discovered, that this howto is more complicated than it needs to be and is not taking advantage of the flexibility offered by autofs. But... It will work for ONE user who needs shares from ONE server mounted into his/her/... home.

Anyway, this howto needs refactoring... stay tuned.

automount CIFS share with autofs

This HowTo will prepare a Linux client to automatically mount CIFS shares from a remote Samba server on access/demand. Since I am mounting different filesystems, I have structured my mountpoints as follows:

```

/
├── home
│   ├── USER
│   │   └── mnt
│   │       ├── cifs
│   │       │   ├── smb-server-a.fqdn
│   │       │   │   ├── share-a
│   │       │   │   ├── share-b
│   │       │   │   └── share-c
│   │       │   ├── smb-server-b.fqdn
│   │       │   │   ├── share-b
│   │       │   │   ├── share-b
│   │       │   │   └── share-c
│   │       └── sshfs
│   │           └── ssh-server-a.fqdn

```

From here on, I will use “mysambaserver.local” as the Samba servers FQDN, “mysambaserver” as its hostname, “myusername” as my username, “mygroup” as group and “myPassWord” as the password.

At time of writing, the server is running Ubuntu 18.04.4 LTS and the client is running Ubuntu 20.04.1 LTS.

This HowTo got compiled by trial and error and from these sources:

- <https://wiki.ubuntuusers.de/Autofs/#Samba-Freigabe>
- <https://www.elektronik-kompodium.de/sites/raspberry-pi/2102201.htm>
- https://wiki.ubuntuusers.de/Samba_Client_cifs/
- <https://wiki.ubuntuusers.de/Gigolo/>
- <https://blog.marvin-menzerath.de/artikel/samba-freigaben-mit-autofs-automatisch-ein-aushaengen/>
- https://www.howtoforge.com/accessing_windows_or_samba_shares_using_autofs
- https://wiki.samba.org/index.php/Mounting_samba_shares_from_a_unix_client
- <https://kalitut.com/samba-shares-fstab/>

Install Required packages, check supported filesystems

Install the required packages on the client (gigolo is just “nice to have”) and check if its kernel supports CIFS.

```
zypper install autofs cifs-utils gigolo gvfs-backends gvfs-fuse fuse
ls -l /lib/modules/$(uname -r)/kernel/fs | grep "cifs"
```

```
cifs
```

Check remote SAMBA connection

Check remote connection to the Samba server:

```
smbclient -N -L //<mysambaserver.local>/
```

```
Sharename      Type      Comment
-----      -
share-a        Disk      Share A
share-b        Disk      Share B
share-c        Disk      Share C
IPC$           IPC       IPC Service (mysambaserver server (Samba, Ubuntu))
SMB1 disabled -- no workgroup available
```

Check authenticated login

Check an authenticated remote login. If the command line asks for a password, enter the SMB password which is configured for the user at the Samba server (via smbpasswd).

```
smbclient -U <myusername> -L //<mysambaserver.local>/
```

```
Enter WORKGROUP\myusername's password: [myPassWord] <--- enter the password
Sharename      Type      Comment
-----      -
share-a        Disk      Share A
share-b        Disk      Share B
share-c        Disk      Share C
IPC$           IPC       IPC Service (mysambaserver server (Samba, Ubuntu))
SMB1 disabled -- no workgroup available
```

Create mount point

Create the mount point in the users home directory:

```
mkdir -pv /home/<myusername>/mnt/cifs
chown -R <myusername>:<mygroup> /home/<myusername>/mnt/
```

autofs configuration

The configuration consists of the master map file (/etc/auto.master), the corresponding map file (/etc/auto.mysambaserver-cifs) and the key file which contains the credentials for authentication. In the following we will configure autofs to mount shares to /home/<myusername>/mnt/cifs/<mysambaserver.local>/<share-name>.

Map file

Create the mapfile

```
vim /etc/auto.<mysambaserver>-cifs
```

```
#!/bin/bash
# $Id$
# This file must be executable to work! chmod 755!
set -x
KEY="${1}"
# Note: create a cred file for each windows/Samba-Server in your network
#       which requires password authentication. The file should contain
#       exactly two lines:
#       username=user
#       password=*****
#       Please don't use blank spaces to separate the equal sign from the
#       user account name or password.
CREDFILE="/etc/autofs/keys/${KEY}"
# # !!!!!!!!!!!!!!!!!!!!! PAY ATTENTION TO the CIFS VERSION in MOUNTOPTS !!!!!!!!!!!!!!!!!!!!!
```

```
# https://www.raspberrypi.org/forums/viewtopic.php?t=201727 #
https://www.raspberrypi.org/forums/viewtopic.php?t=211987
# http://krisko210.blogspot.com/2016/06/autofs-automount-nfs-share.html
# Note: Use cifs instead of smbfs:
MOUNTOPTS="-fstype=cifs,file_mode=0644,dir_mode=0755,nounix,uid=1000,gid=100"
SMBCLIENTOPTS=""
for EACH in /bin /sbin /usr/bin /usr/sbin
do
    if [ -x $EACH/smbclient ]
    then
        SMBCLIENT=$EACH/smbclient
        break
    fi
done
[ -x $SMBCLIENT ] || exit 1
if [ -e "${CREDFILE}" ]
then
    MOUNTOPTS=$MOUNTOPTS",credentials=${CREDFILE}"
    SMBCLIENTOPTS="-A "${CREDFILE}
else
    SMBCLIENTOPTS="-N"
fi
${SMBCLIENT} ${SMBCLIENTOPTS} -gL "${KEY}" 2>/dev/null \
| awk -v key="${KEY}" -v opts="${MOUNTOPTS}" -F'|' -- '
BEGIN { ORS=""; first=1 }
/Disk/ { if (first) { print opts; first=0 };
        gsub(/ /, "\\ ", $2);
        sub(/\$/ , "\\$", $2);
        print " \\n\t/" $2, "://" key "/" $2 }
END { if (!first) print "\n"; else exit 1 }
'
```

```
chmod 755 /etc/auto.<mysambaserver>-cifs
```

This file is a slightly modified version of the file auto.smb which usually comes as part of the autofs package. You need to modify the line defining the mountopts above and change userid and groupid to the uid/gid of your personal account.

key file

Now you have to give autofs the credentials needed to access shares on your network. To do this create a key file

```
mkdir -pv /etc/autofs/keys/
vim /etc/autofs/keys/<mysambaserver.local>
```

```
username=<myusername>
password=<myPassWord>
```

```
chown root:root /etc/autofs/keys/<mysambaserver.local>
chmod 600 /etc/autofs/keys/<mysambaserver.local>
```

Master-Map

The maps to be monitored are specified in this file. Execute the following command to append the line `"/home/myusername/mnt/cifs /etc/auto.mysambaserver-cifs -timeout=60"` at the end of the `/etc/auto.master` file:

```
echo "/home/<myusername>/mnt/cifs /etc/auto.<mysambaserver>-cifs --timeout=60 --ghost"
>>/etc/auto.master
```

The syntax here is: `<Directory> <Map-File> [Parameter]` The shares should be unmounted after an inactivity of 60 seconds (`-timeout=60`) and empty directories should be created for the individual shares before mounting (`-ghost`).

Debugging

for debugging output stop the daemon and interactively start autofs with verbose output enabled

```
service autofs stop
automount -f -v
```

You can restart autofs with

```
systemctl enable autofs
service autofs start
```

Test

Use the following command to test if your setup is working

```
ls -als /home/<myusername>/mnt/cifs/<mysambaserver.fqdn>/<share-name>
```

Script

This script automates the whole process above. Please keep in mind to change the variables to fit your environment.

```
#!/bin/bash

#
# Variables
#

_SMB_FQDN="your.smbserver.fqdn"           # set to FQDN of your SMB-Server
_SMB_USER="your SMB username"           # set to your username
_SMB_PASS="your SMB password"           # set to your password
_LOCAL_USER="your username on local machine" # set to your local username
_LOCAL_GROUP="your local group"         # set to your local group
_SMB_MOUNTPOINT="/home/${_LOCAL_USER}/mnt/cifs" # set to path of your desired SMB base-
mountpoint
_BASE_MOUNTPOINT="/home/${_LOCAL_USER}/mnt"

# don't touch
_UNAME="$(uname -r)"
_CIFSKM="/lib/modules/${_UNAME}/kernel/fs/cifs/cifs.ko.xz"
_UID="$(id -u ${_LOCAL_USER})"
_GID="$(id -g ${_LOCAL_USER})"

zypper install -y autofs cifs-utils gigolo gvfs-backends gvfs-fuse fuse

# exit if cifs kernel module is not present
[ -f "${_CIFSKM}" ] || { echo "${_CIFSKM} does not exist. Exiting."; exit 1; }

# test unauthenticated connection to SMB server
smbclient -N -L //"${_SMB_FQDN}"/ > /dev/null 2>&1 || { echo "Unauthenticated connection to ${_SMB_FQDN}
failed. Exiting."; exit 1; }

# test authenticated connection to SMB server
smbclient --user="${_SMB_USER}%${_SMB_PASS}" -L //"${_SMB_FQDN}"/ > /dev/null 2>&1 || { echo
"Authenticated connection to ${_SMB_FQDN} failed. Exiting."; exit 1; }

# create mountpoint, set permissions
mkdir -pv "${_SMB_MOUNTPOINT}"
chown -R "${_LOCAL_USER}":"${_LOCAL_GROUP}" "${_BASE_MOUNTPOINT}"
```

```

# create the mapfile
cat > /etc/auto."${_SMB_FQDN}"-cifs << "ENDOFFILE"
#!/bin/bash
# $Id$
# This file must be executable to work! chmod 755!
#set -x
KEY="${1}"
# Note: create a cred file for each windows/Samba-Server in your network
#       which requires password authentication. The file should contain
#       exactly two lines:
#           username=user
#           password=*****
#       Please don't use blank spaces to separate the equal sign from the
#       user account name or password.
CREDFILE="/etc/autofs/keys/${KEY}"
# # !!!!!!!!!!!!!!!!!!!!!!! PAY ATTENTION TO the CIFS VERSION in MOUNTOPTS !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
# https://www.raspberrypi.org/forums/viewtopic.php?t=201727 #
# https://www.raspberrypi.org/forums/viewtopic.php?t=211987
# http://krisko210.blogspot.com/2016/06/autofs-automount-nfs-share.html
# Note: Use cifs instead of smbfs:
MOUNTOPTS="-fstype=cifs,file_mode=0644,dir_mode=0755,nounix,uid=___UID___,gid=___GID___"
SMBCLIENTOPTS=""
for EACH in /bin /sbin /usr/bin /usr/sbin
do
    if [ -x $EACH/smbclient ]
    then
        SMBCLIENT=$EACH/smbclient
        break
    fi
done
[ -x $SMBCLIENT ] || exit 1
if [ -e "${CREDFILE}" ]
then
    MOUNTOPTS=$MOUNTOPTS",credentials=${CREDFILE}"
    SMBCLIENTOPTS="-A "${CREDFILE}
else
    SMBCLIENTOPTS="-N"
fi
$SMBCLIENT $SMBCLIENTOPTS -gL "${KEY}" 2>/dev/null \
| awk -v key="${KEY}" -v opts="${MOUNTOPTS}" -F'|' -- '
BEGIN { ORS=""; first=1 }
/Disk/ { if (first) { print opts; first=0 };
        gsub(/ /, "\\ ", $2);
        sub(/\$/, "\\$", $2);
        print " \\n\\t /" $2, "://" key "/" $2 }
END { if (!first) print "\\n"; else exit 1 }
'
ENDOFFILE

# insert UID and GID
sed -i "s/___UID___/${_UID}/" /etc/auto."${_SMB_FQDN}"-cifs
sed -i "s/___GID___/${_GID}/" /etc/auto."${_SMB_FQDN}"-cifs

# set permissions for mapfile
chmod 755 /etc/auto."${_SMB_FQDN}"-cifs

# create keyfile
mkdir -pv /etc/autofs/keys/
cat > /etc/autofs/keys/"${_SMB_FQDN}" << "ENDOFFILE"
username=___USERNAME___
password=___PASSWORD___
ENDOFFILE

# insert username and password
sed -i "s/___USERNAME___/${_SMB_USER}/" /etc/autofs/keys/"${_SMB_FQDN}"
sed -i "s/___PASSWORD___/${_SMB_PASS}/" /etc/autofs/keys/"${_SMB_FQDN}"

# set permissions for keyfile
chown root:root /etc/autofs/keys/"${_SMB_FQDN}"
chmod 600 /etc/autofs/keys/"${_SMB_FQDN}"

```

Last update: 2022/11/30 14:53 pages:howtos:suse:automount-cifs-share-with-autofs <https://wiki.nanoscopic.de/doku.php/pages/howtos/suse/automount-cifs-share-with-autofs?rev=1669820026>

```
# Master-Map
echo "${_SMB_MOUNTPOINT} /etc/auto.${_SMB_FQDN}-cifs --timeout=60 --ghost" >>/etc/auto.master

systemctl enable autofs
service autofs start
```

~~DISCUSSION~~

From: <https://wiki.nanoscopic.de/> - nanoscopic wiki

Permanent link: <https://wiki.nanoscopic.de/doku.php/pages/howtos/suse/automount-cifs-share-with-autofs?rev=1669820026>

Last update: **2022/11/30 14:53**

