

- [openSUSE Home](#)
- [Swerdna](#)
- [Ubuntu](#)
- [Open Source](#)
- [Open Sauce](#)

Samba HowTo: Mount a CIFS Network Share [Mapped Drive] in openSUSE

This tutorial shows how to mount shares as network drives on a Linux client and also how to set up shares on a Linux server that are suitable for mounting on Windows and Linux clients.

Versions: openSUSE 11.x [should also work in openSUSE 10.2 & 10.3]

Index & in-page links

- [Preliminaries: software & glossary](#)
- [Temporary Mounts \(from the Command Line\)](#)
 - [Unsecured/guest shares \(no credentials required\)](#)
 - [Secure shares \(username & password required\)](#)
 - [Prescribing the owner of the mount on the client](#)
- [Permanent Mounts set in fstab](#)
 - [Editing the file fstab](#)
 - [Unsecured/guest shares \(no credentials required\)](#)
 - [Secure shares \(username & password required\)](#)
 - [Prescribing the owner of the mount on the client](#)
 - [Setting the cifs/smbfs daemon \(the CIFS mount helper daemon\)](#)
- [Frequently Asked Questions / Issues with Mounts](#)
 - [Escape code for spaces in fstab](#)
 - [Hiding the username and password](#)
 - [Using OpenOffice.Org and Microsoft Office files](#)
 - [Mapping shares on legacy servers \(OS/2, Windows 98 and Windows ME\)](#)
 - [Mount fails when using the server's network/netBIOS name](#)
 - [Permanent mount fails at boot time](#)
- [Setting up shares on a Linux Server](#)

- [Unsecured shares with guest access on a server](#)
- [Shares secured by credentials on a server](#)

↑↑↑↑ Preliminaries: software & glossary

Software: Install the RPMs cifs-mount and samba-client on a client machine (the machine where you mount the external resource shared from the server). Install the full Samba suite if your machine is the server.

Glossary of parameters & names that I use in the Tutorial: Here are the parameters, names, passwords and so on that I made up to illustrate this tutorial. Refer back here when you get confused about them in the examples that follow.

Names that relate to the server:

- 192.168.44.100: The server's IP address
- server: The network name of the server (this is the name you see in your network browser for the server computer).
- share: The network name of the share on the server.
- server_user: The username of the owner of the shared directory on the server.
- secret: The password to access the share on the server. In Windows it is the login password of the share's owner on the server. In Linux it is the Samba user's password for any authentic user on the server.

Names that relate to the client:

- mount: The directory on the client where the mapped drive is mounted. The filesystem and files for the share on the server appear in this directory. NB: you must create the mount directory before you can use it.
- /path_to/mount: The full path to the mount in the Linux client.
- client_user: Once the remote share is mounted, the ownership changes as designated in the mount command to a cifs owner; e.g. in this tutorial he/she is "client_user". Note that the usernames on the server and the client machines can be different, but if you wish it, they may be the same; i.e. "client_user" and "server_user" can be the same name/person.

↑↑↑↑ Temporary Mounts (from the Command Line)

Your mapped drives (AKA network drives) can be mounted either temporarily or permanently. We start with temporary mounts. Permanent

mounts are [detailed below](#).

Temporary Mounts: Unsecured/guest shares (no credentials required)

Shares that can be accessed without credentials (i.e. no username/password), e.g. share-level shares in Windows or world-readable Samba Linux shares, these shares can be mounted from the command line in a root console/terminal as follows:

```
mount -t cifs -o guest //192.168.44.100/share /path_to/mount
```

Notes:

(1) the only option here is `-o guest`

(2) you may use the network name of the server instead of the IP address, as follows:

```
mount -t cifs -o guest //server/share /path_to/mount
```

Temporary Mounts: Secure shares (username & password required)

Some Vista and Windows 7 shares and many Samba Linux shares require credentials in the form of username/password pairs before access is granted. These credentials are set on the server. In Windows Vista and Windows 7 they are the login username and password of the owner on the server and in Linux they are any member of the Samba User Database on the server who is "allowed" by the file that defines the share.

You use the option string `username=server_user,password=secret` instead of the `guest` option used above to generate the mount as follows:

```
mount -t cifs -o username=server_user,password=secret //192.168.44.100/share /path_to/mount
```

Once again, you could use the alternate form `//server/share` instead of `//IPaddress/share` for the address.

Temporary Mounts: Prescribing the owner of the mount on the client

So far, these commands haven't specified an owner for the mount. If the client maps a Windows share-level share, the mount on the client will automatically assume permissions `drwxrwxrwx` and the ownership will stay with the Linux owner of the directory `mount_point`. If the client maps a Linux share the mount will automatically assume the permissions on the share on the server and will automatically switch ownership to match the owner on the server (which can be tricky if the owner on the Linux server doesn't exist on the client).

You can control this by specifying a new owner in the mount command. This owner is a Linux user on the client (who might not be a user on the server, that doesn't matter), e.g. `client_user`. You can specify the group too, e.g. `users`. The command is now like this for a guest-accessible share:

```
mount -t cifs -o guest,uid=client_user,gid=users //192.168.44.100/share /path_to/mount
```

and if the server requires authentication, it's like this:

```
mount -t cifs -o username=server_user,password=secret,uid=client_user,gid=users //192.168.44.100/share /path_to/mount
```

Remember, the user you specify on the client doesn't have to be the username of the owner of the shared directory on the server. Also, you can use the server's network name instead of the IP address. Finally you can use the numerical forms like `uid=1002`, `gid=100` instead of the alphabetical forms like `uid=client_user`, `gid=users`.

↑↑↑↑ Permanent Mounts set in fstab

If you want the mapped drive to be always present then you don't use the on-the-fly command line approach [detailed above](#). Instead you mount the drive at boot by adding a line to the text file that contains the file system table. It's called `fstab` and it's located at `/etc/fstab`.

Permanent Mounts: Editing the file `fstab`

To open file `fstab` in the text editor "gedit", a Gnome user issues the following command in a terminal:

```
gnomesu gedit /etc/fstab
```

and a KDE user opens `fstab` in the editor "kwrite" with this command:

```
kdesu kwrite /etc/fstab
```

Type the line for the mount, using the templates below. The line can be placed anywhere in the list of lines. I recommend at the bottom and make sure the very last line is a blank line. The description that follows is almost identical to the description given above for on-the-fly command line temporary shares. The only difference is that the syntax for lines in `fstab` is marginally different from the command line syntax.

Permanent Mounts: Unsecured/guest shares (no credentials required)

Shares that can be accessed without credentials (i.e. no username/password), e.g. share-level shares in Windows or world-readable Samba Linux shares, these shares can be mounted in `fstab` as follows:

```
//192.168.44.100/share /path_to/mount cifs guest,_netdev 0 0
```

Notes:

- (1) the credentials issue is bypassed by the option `guest`
- (2) The option `_netdev` is always recommended for cifs mounts in `fstab`. Option `_netdev` delays mounting until the network has been enabled. `_netdev` is known to the command "mount" but not to the command "mount.cifs". Even though `mount.cifs` doesn't recognise `_netdev`, you should include it in your mount command anyway.
- (3) you may use the network name of the server instead of the IP address, as follows:

```
//server/share /path_to/mount cifs guest,_netdev 0 0
```

Permanent Mounts: Secure shares (username & password required)

Some Vista and Windows 7 shares and many Samba Linux shares require credentials in the form of username/password pairs before access is granted. These credentials are set on the server. In Windows Vista and Windows 7 they are the login username and password of the owner on the server and in Linux they are any member of the Samba User Database on the server who is "allowed" by the file that defines the share.

You use the option string `username=server_user,password=secret` instead of the `guest` option used above to generate the mount as follows:

```
//192.168.44.100/share /path_to/mount cifs username=server_user,password=secret,_netdev 0 0
```

Once again, you could use the alternate form `//server/share` instead of `//IPaddress/share` for the address.

Permanent Mounts: Prescribing the owner of the mount on the client

So far, these commands haven't specified an owner for the mount. If the client maps a Windows share-level share, the mount on the client will automatically assume permissions `drwxrwxrwx` and the ownership will stay with the Linux owner of the directory `mount_point`. If the client maps a Linux share the mount will automatically assume the permissions on the share on the server and will automatically switch ownership to match the owner on the server (which can be tricky if the owner on the Linux server doesn't exist on the client).

You can control this by specifying a new owner in the line/mount that you put in `fstab`. This owner is a Linux user on the client (who might not be a user on the server, that doesn't matter), e.g. `client_user`. You can specify the group too, e.g. `users`. The line in `fstab` is now like this for a guest-accessible share:

```
//192.168.44.100/share /path_to/mount cifs guest,_netdev,uid=client_user,gid=users 0 0
```

and if the server requires authentication, it's like this (all one line):

```
//192.168.44.100/share /path_to/mount cifs username=server_user,password=secret,_netdev,uid=client_user,gid=users 0 0
```

Remember, the user you specify on the client doesn't have to be the username of the owner of the shared directory on the server. Also, you can use the server's network name instead of the IP address. Finally you can use the numerical forms like `uid=1002, gid=100` instead of the alphabetical forms like `uid=client_user, gid=users`.

Permanent Mounts: Setting the `smbfs/cifs` daemon (the CIFS mount helper daemon)

This segment is for permanent mounts. Ignore it for temporary mounts.

OpenSUSE uses a helper daemon for mounting the network share at boot time. The daemon cannot be activated until a `cifs` mount is configured in the filesystem table, `fstab`, located at `/etc/fstab` (you may also use `smbfstab` in ≤ 11.2 or `cifstab` in 11.3). Check the daemon's status in Yast --> System --> System Services (Runlevel). In openSUSE 11.3 the daemon is called `cifs`. In earlier versions (11.0, 11.1, 11.2) it was called `smbfs`. Scroll to service `cifs` (or `smbfs`). It must show status `Enabled=Yes` to facilitate the mount process at boot time. If the status is `No`, then toggle it on with the `Enable` button.

If you get a status of `Yes*` with an asterisk, that's fine and indicates that the daemon is working but no `cifs` mounts are currently active. If you attempt to `Enable` `cifs` (or `smbfs`) but get an error message like `"cifs start returned 6 -- program is not configured"` and the status remains at `No`, then you don't have a valid `cifs` mount configured in `fstab` (or in `smbfstab`); first configure a mount before returning to enable the daemon again.

[↑↑↑↑](#)Frequently Asked Questions / Issues with Mounts

Escape code for spaces in `fstab`

If you have a space in a network name in `fstab`, you can "escape" the space by replacing it with the escape code `\040`. For example, if a share's name is "music library", then the share's name in a mount in `fstab` becomes like this:

```
//192.168.44.100/music\040library
```

Hiding the username and password

Any Linux user on the client machine who chooses to view the file `fstab` can of course read the `{username,password}` credentials required to be sent to the server to effect the mount. If this is undesirable, you can put the credentials in a hidden file, e.g. `".creds"`. For additional security you can make the file `".creds"` readable only by the owner (`rw----`). The entry in `fstab` then becomes (all on one line):

```
//192.168.44.100/share /path_to/mount cifs
```

```
credentials=/path_to/.creds,_netdev,uid=client_user,gid=users 0 0
```

And the contents of ".creds" are two lines, like this:

```
username=server_user  
password=secret
```

Using OpenOffice.Org and Microsoft Office files

A drive mapped from a Windows Server will create problems with shared OpenOffice or Microsoft Office files being manipulated on the Linux client in OpenOffice software. The OpenOffice Writer program will freeze when an edited .doc/.odt file is saved using the "Save As" feature. This occurs because OpenOffice doesn't use the cifs-style mandatory byte range locks. The problem is fixed when you add the option `nobrl` into the options for the cifs mount.

Mapping shares on legacy servers (OS/2, Windows 98 and Windows ME)

I have not tried these. I am just reporting them for you. Older servers require a few extra options in the mount strings. You appear to need the security option `sec=lanman` and the server-name option (`servern=SERVER_NAME`) in mounts for OS/2 and Windows 98/ME servers. The network name (`SERVER_NAME`) should be in upper case. Thanks to jimoe666, who also recommends adding the `nocase` option for OS/2 servers (which are case-insensitive).

Mount fails when using the server's network/netBIOS name

You can designate the address of the server two ways, either by the IP address or with the network (netBIOS) name of the server; e.g. `//192.168.44.100/share` or `//server/share`. Sometimes the use of the network name doesn't work, only the IP address does. I find if you can ping the server by network name, you can also mount it by network name.

- Check that you can browse to the network share on the server using the Network Browser on the client (e.g. Nautilus or Dolphin)
- Check that the `smbfs` daemon is running (only for a permanent mount [see above](#))
- Check that `wins` is switched on in the hosts line in the file `/etc/nsswitch.conf`. Locate the line beginning with `hosts` and see that `wins` comes before `dns` as follows:
hosts: files mdns4_minimal [NOTFOUND=return] wins dns
Reference for [/etc/nsswitch.conf](#)

A permanent mount fails at boot time

Is your `smbfs` helper daemon switched on? Check in Yast --> System --> System Services (runlevels) that the `smbfs` daemon is on. [See](#)

[section above](#).

Are you using the network name of the server? If you're using the network name of the server in the address (//server/share), make sure you can ping the server by network name in a console or terminal window. If not, fix your samba name resolution mechanism and the file nsswitch.conf [see section above](#). **[Tip:** I prefer to use the IP address for the server name in fstab instead of the network/netBIOS name; I find IP addressing more reliable in openSUSE.]

Is the syntax for the mount correct in fstab? You can test the syntax several ways. First open a terminal and run the command `su` to get rootly powers. Then run the command `mount -a`. Check the return for error messages. If the mount executes, then perhaps the network is delayed at boot time (see next). Second, translate the syntax in fstab to a command line form (see section above: [Temporary Mounts](#)). Run the command for the temporary mount and check for error messages.

Is something delaying the network at boot time? This has been a problem with openSUSE and cifs since cifs was introduced years ago. A quick workaround is to tell the system to run the mount commands again after the network has settled down. You create a file named `after.local` in the directory `/etc/init.d` and give it these contents:

```
#!/bin/sh
mount -a
```

You must also make the file executable. You can do that by altering permissions in your file browser or with this command in a console:

```
sudo chmod a+x /etc/init.d/after.local
```

Sometimes, very rarely, you may need to delay the procedure even more. The extra delay is effected by setting up a cron job in the crontab(1e) for the root user, whereby the command `mount -a` is execute approximately 10 or 20 seconds after the boot process. That command mounts all devices listed for mounting in fstab that are not already mounted. I've included a [discussion of cron and crontab](#) on this site and here's the entry you can put into root's crontab if the `after.local` workaround fails to delay the mounting procedure sufficiently:

```
@reboot sleep 10;mount -a
```

↑↑↑↑ Setting up shares on a Linux Server

The variety of network shares that you might seek to mount is immense. Most can be mapped across to a Linux client using CIFS. Some of them will cause problems. I'll describe two that shouldn't cause problems. One allows guest access and the other requires credentials.

Unsecured shares with guest access on a server

A typical read-only share for guests looks like this in the samba configuration file at `/etc/samba/smb.conf`:

```
[share]
path = /path_to/share
guest ok = yes
```

The typical read/write share should look like this:

```
[share]
path = /path_to/share
force user = server_user
read only = no
guest ok = yes
```

The directory "share" and all its contents are owned by user "server_user". Permissions on the directory are `drwxr-xr-x`. The user on the client will see the files as belonging to users on the client, and the "force user" parameter is a simple device to keep things tidy on the server.

Shares secured by credentials on a server

These shares always require authentication built into the mount process.

A typical read-only share for guests looks like this in the samba configuration file at `/etc/samba/smb.conf`:

```
[share]
path = /path_to/share
```

The typical read/write share should look like this:

```
[share]
path = /path_to/share
force user = server_user
read only = no
```

The directory "share" and all its contents are owned by user "server_user". Permissions on the directory are `drwxr-xr-x`. The user on the client will see the files as belonging to users on the client, and the "force user" parameter is a simple device to keep things tidy on the server. The credentials that must be passed to the server from the client during the mount are `username=server_user,password=secret`.

You must add the user "server_user" to the samba user database on the server to allow these shares to work, using the command smbpasswd. Here's the dialogue:

```
sudo smbpasswd -a server_user
root's password: xxxxxxxxxxxx
New SMB password: secret
Retype new SMB password: secret
Added user server_user.
```

End of story -- Phew!

Be as well as you can

Swerdna: 27 April 2007; last updated 21 August 2010

[How you can help](#)

Samba: Browsing Shares

[SOHO LAN Primer](#)

[Advanced Samba LAN](#)

[Troubleshoot Samba](#)

[Configure Browsing](#)

[On-the-fly File Shares](#)

[About Usershares](#)

[CIFS: Mapped Drives
& Mounted Shares](#)

[Samba and Vista](#)

[Samba & Windows 7](#)

Lan & Net Connectivity

[SuSE Firewall](#)

[Configure Network Card](#)

[Configure Wireless Card](#)

[Configure \(K\)torrent](#)

[Nokia Mobile Phone](#)

HSDPA Wireless

[BigPond NextG HSDPA](#)

Servers & Clients

[Apache + SSI & CGI](#)

[Windows Print Server](#)

[Suse Samba Print Server](#)

[Suse IPP Print Server](#)

[Samba File Server](#)

[ICS IP Masquerading](#)

[Xvnc TightVNC](#)

[Windows RDP in Linux](#)

Partition Filesystem Boot Backup Restore

[Mount FAT32 Read/Write](#)

[Mount NTFS Read/Write](#)

[Multiboot Windows & openSUSE with Grub](#)

[Five ways to boot when Grub is broken](#)

[Multiboot Ubuntu from Grub in openSUSE](#)

[Multiboot openSUSE with Windows bootloader](#)

[Mount /home on a Separate Partition](#)

[Linux Ghost Clone](#)

Installing Printers

[Brother Laser HL2040](#)

[HP Laserjet 1020](#)

Miscellaneous

[Cron Tables & CronTab](#)

[Install FreeBASIC](#)

[Install VMware Server](#)

- [openSUSE Home](#)
- [Swerdna](#)
- [Ubuntu](#)
- [Open Source](#)
- [Open Sauce](#)

[Valid XHTML](#)

[Valid CSS](#)

[Copyright © Asserted]

[Site & Software by Swerdna](#)

359436 2436784