

[English](#) • [Deutsch](#) • [Español](#) • [Français](#) • [Italiano](#) • [???](#) • [Polski](#) • [Português](#) • [??????](#) • [Svenska](#) • [???\(????\)?](#) • [???\(??\)?](#) •

Samba is one of a number of [network file systems](#) available to the DD-WRT platform. A network file system allows for sharing the contents of directories or entire drives across a network. In DD-WRT this can work both ways: You can run a Samba server on your main computer and run a client on your router (thus gaining writable storage for the router) or you can use Samba to share a drive connected (typically by [USB](#)) to the router among all the computers connected to your network.

*Standard and larger versions of DD-WRT contain the Samba/CIFS client. Mini and Micro versions of DD-WRT do not contain Samba/CIFS. See the Wiki page [Version Features](#).*

## Contents

- [1 Running a Samba client on DD-WRT](#)
  - ◆ [1.1 Mount Manually](#)
  - ◆ [1.2 Allowing General Write Access for Windows Users](#)
  - ◆ [1.3 Filesystem to use that works](#)
  - ◆ [1.4 Allowing write access for any OS i.e. Mac OSX, Windows](#)
  - ◆ [1.5 Note for Windows XP](#)
  - ◆ [1.6 Note for Windows Server 2003/ Vista/ 7](#)
- [2 Running a Samba Server on DD-WRT](#)
  - ◆ [2.1 Samba Server - option 1](#)
  - ◆ [2.2 Samba Server - option 2](#)
  - ◆ [2.3 Samba Server - option 3: the easy way](#)
  - ◆ [2.4 Samba Server - option 4: Samba3](#)
- [3 Troubleshooting](#)

## Running a Samba client on DD-WRT

1. Create a SMB network share on the host computer.
  - ◆ On Linux, use [Samba](#) to create a SMB network share. [HOWTO](#). Also you can find a tutorial specific for JFFS in [Configure Samba in Linux](#)
  - ◆ On Windows, this is the default file sharing system. Simply right click and select "Sharing and Security" to set up sharing.
  - ◆ On MacOS X, go to "Sharing" in the System Preferences and turn on "Windows Sharing" for [10.4](#) or in [10.5](#) "Share files and folders using SMB." from the File Sharing options.
2. Use the [Web Interface](#) and go to the **Administration** tab.
3. Enable the **Samba FS Automount** option (or **CIFS Automount**). If new options do not appear the "Save Changes" and go back to the page.
4. Enter the share location in the 'share' field. eg: `//192.168.1.160/myshare`
5. In 'user' and 'password' enter the user credentials of a user allowed to access the share.
6. You may optionally enter a Startscript. For example if you have a script on your share `\\nanserver\myshare\startup.sh` you enter `startup.sh` in the 'Startscript' box. This script (`/tmp/smbshare/startup.sh` on the router) will be executed when the router mounts your share on boot.

You can see sample of startup script on [Jffs](#) page.

## Mount Manually

You can temporarily mount a share manually via a Telnet/SSH session:

First, insert a necessary kernel module:

```
insmod /lib/modules/`uname -r`/smbfs.o
```

Then enter these commands to mount:

```
mkdir /tmp/Path  
smbmount //hostname/Share /tmp/Path/ -o username=<user>,password=<password_user>
```

Note: If using CIFS, the procedure is slightly different:

```
insmod cifs  
nvram set samba_user="username"  
nvram set samba_password="password"  
mkdir /tmp/smbshare  
mount.cifs //hostname/Share /tmp/smbshare -o username=$(nvram get samba_user),password=$(nvram ge
```

- `/tmp/Path/` is the path you want mounted
- Only `/tmp/`, `/dev/` and potentially `/jffs/` are writeable!

Receiving errors like the following seem normal:

```
load_client_codepage: filename /usr/lib/codepages/codepage.850 does not exist.  
load_unicode_map: filename /usr/lib/codepages/unicode_map.850 does not exist.  
session request to 192.168.1.99 failed  
Can't get /etc/mstab~ lock filesmbmnt failed: 1
```

To unmount a share, enter this command in a shell prompt:

```
umount /tmp/Path/  
rmdir /tmp/Path/  
rmmmod smbfs
```

## Allowing General Write Access for Windows Users

*On 4/23/09, DD-WRT User locketine advised:* You have to mount the drives as rw for the user "nobody". By default drives are mounted read only for everyone but root. The following link told me what I needed to know. I think using "-o user" is the important part when mounting.

<http://www.linuxforums.org/forum/debian-linux-help/79749-mount-fat32-shared-partition-write-permission.html>

## Filesystem to use that works

I initially format my usb hard drive as Fat32 because I wanted to be able to use the drive on any computer. I couldn't get [Slim\\_samba2](#) to run. Here is a link to the exact same issue I was having.

## Samba\_filesystem

/opt/etc/init.d/S80samba: line 18: /opt/sbin/smbd: not found

I messed with this for a while and just decided to go with what was recommended in the [Slim\\_samba2](#) tutorial i.e. etx3/swap/ext3 as defined here [How to - Format and Partition External Storage Device](#). It works great.

[Ericmagaha](#) 15:12, 31 March 2010 (CEST)

## Allowing write access for any OS i.e. Mac OSX, Windows

I used [Slim\\_samba2](#) and formatted my drive as ext3/swap/ext3 i.e. Optware/Swapfile/Data as outlined [How to - Format and Partition External Storage Device](#). Initially I had readonly access to the drive from Windows and Mac. Had to change the permissions of the Data partition i.e. /mnt/data. It defaulted to 700 i.e. rwx-----. I used chmod and changed it to 777 i.e. rwxrwxrwx. I could then write to the Data partition from Windows (Windows explorer) and Mac (Finder).

[Ericmagaha](#) 15:12, 31 March 2010 (CEST)

## Note for Windows XP

Windows XP has what is called "simple file sharing" enabled by default. This forces guest access to the share and does not allow permissions (read, write, etc.) based on specific user accounts. If you run into problems getting CIFS to work, consider disabling simple file sharing.

To disable it, go to: My Computer -> Tools -> Folder Options -> View Under Advanced Settings: Uncheck "Use Simple File Sharing" Click Apply, OK

Optionally, run the Network Setup Wizard in Windows Control Panel in order to set up your share(s).

Password Note: Make sure the user account that you wish to access the share has a password. Windows XP will reject any logins over the network which don't use a password.

## Note for Windows Server 2003/ Vista/ 7

Especially on Windows Server 2003, assert that your server accepts NTLM v1.

1. Goto Start > Run
2. Type "gpedit.msc" and push enter.
3. Goto Computer Configuration / Windows Settings / Security Settings / Local Policies / Security Options.
4. Verify that "Network security: LAN Manager authentication level" is **NOT** "Send NTLMv2 response only/refuse LM & NTLM" (on Vista Home Premium set HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa\LmCompatibilityLevel to 1, because there is no gpedit.msc as in Ultimate )

**NOTE** For Windows Vista and Windows 7 (mainly), you must use a version of Samba that is 3.4 or higher!

## Samba\_filesystem

If you have a USB capable router with Optware, the Right Way installed, it will be a breeze for you. For those with only JFFS2 space available, your options are very limited!

To effortlessly get the latest version of Samba working (Samba 3.5.4 as of 08/13/2010) with Optware, the Right Way, you must do the following:

First make a backup of your current Samba, just in case something happens:

For vi users:

```
cp /opt/etc/samba/smb.conf /opt/etc/samba/smb.conf.copy
cp /opt/etc/init.d/S80samba /opt/etc/init.d/S80samba.copy
```

For users who want to edit those files on their computers:

```
cp /opt/etc/samba/smb.conf /mnt/smb.conf.copy
cp /opt/etc/init.d/S80samba /mnt/S80samba.copy
```

Now do the following:

```
service samba stop
service samba off
```

Then

```
ipkg-opt remove samba2
ipkg-opt install samba35
```

Let the packages install, and don't worry about Dbus, you won't be needing it. After install has completed, it will prompt you that a new smb.conf file needs to be created. Easy enough...

In the terminal prompt (make sure to use Putty, not Telnet when editing files with Vi!), do the following:

```
rm -r /opt/etc/samba/smb.conf
```

Also you will need to delete the file in **/opt/etc/init.d** that Samba35 has created. It is called S08samba

```
rm -r /opt/etc/init.d/S08samba
```

Now create a new smb.conf file:

```
vi /opt/etc/samba/smb.conf
```

Once the editor has opened up, the new file should be named **smb.conf**, Hit the **Insert** key on your keyboard and paste the following into the new file (give a big thanks to okki, who discovered the interface problem!):

```
[global]
interfaces = 192.168.1.1/255.255.255.0
security = SHARE
# don't ask for a password
```

## Samba\_filesystem

```
[Optware]
path = /opt
force user = root
read only = No
guest ok = Yes
```

```
[Public]
path = /mnt
force user = root
read only = No
guest ok = Yes
```

```
#guest ok = yes:
#guest user 'nobody' is allowed and has access as 'valid users' is not specified (so all users are
```

Once properly pasted, hit the **ESCAPE** key and type **:wq** followed by **Enter**

All that's left to do now is to Reboot (although you really don't have to), then when logged back in, type:

```
service samba on
service samba start
```

Now you will be able to see your shares and Access them on your network in Windows 7!

**NOTE** I have not tested Samba35 S.W.A.T yet, but it should be as easy as issuing the following:

```
ipkg-opt install samba35-swat
```

Hope this helps!

Finally, for those who do not wish to install Samba 35, or are looking for an even newer version, issue the following:

```
ipkg-opt upgrade
ipkg-opt update
```

Then issue

```
ipkg-opt list | grep samba
```

## Running a Samba Server on DD-WRT

If you have a powerfull Router like the Asus WL-500gP(remium) you might want to use the USB 2.0 Ports for HD sharing with Windows Clients.

### Samba Server - option 1

The default way - using OpenWRT packages

## Samba\_filesystem

- a) install the USB stuff according to [USB storage](#)
- b) activate [jffs](#)
- c) update the [ipkg](#) system with the following command

```
ipkg update
```

- d) install samba with the following command

```
ipkg install samba2
```

samba2 is not a current package.

the samba install requires a libgcc version that is not trivial to find.

## Samba Server - option 2

The [optware](#) way - using Optware packages

A nice Howto for this is located here [Optware](#)

You should really try it the optware way because it works :)

## Samba Server - option 3: the easy way

See [Slim Samba2](#) - a stripped down version of Optware Samba2 for use on USB drives, just unpack a .tar.gz file, adjust a few settings in the web-GUI, and you are up and running.

See [HDSamba2](#) - a fully automated install and boot script for installing Optware Samba2 with SWAT, and more... on a USB HD. Format you HD, adjust a few settings in the web-GUI, run the install script, and you are up and running.

## Samba Server - option 4: Samba3

Since [samba2 has issues with files of size >2Gb and <4Gb](#) and has no unicode support, you may want to use [samba3](#).

## Troubleshooting

If you are having issues mounting your share, hopefully the following forum post will help:  
[Problems mounting with cifs](#)

Also note that the error "Error 22: invalid argument" is a very misleading catch-all error; very often, your arguments are correct and the error actually is on the server side - things such as incorrect permissions can cause it, for instance. There are quite a few variations to the SMB protocol and authentication. In theory, CIFS

## Samba\_filesystem

and the server should be able to figure it out automatically, but it may not always work reliably. These variations include:

Protocol: Netbios over TCP (port 139) or SMB over TCP (port 445)

Authentication: Plain text password, Lanman authentication (obsolete), NTLM authentication (most common) or NTLM2 authentication

These are just some of the most common parameters.

Start by trying to mount a share on a non-existent IP address. It should take a while before you get a timeout error. That tells you that the arguments you passed to mount.cifs are actually valid.

Also, you can try the -v option to mount.cifs to get a better idea of what it does behind the scenes.

```
mount.cifs //192.168.0.15/myshare /tmp/sambashares -v -o user=xxxx,pass=xxxx
```

On the server side, see if you can enable guest access (on a Samba server, set guest OK = yes in your smb.conf file). If that helps, the problem is a matter of either user name resolution or passwords.

If you are having windows 7 access issues try adding 'host msdf = no' into the global area of your samba config file. I am not sure exactly what is happening but windows 7 seems to sometimes make samba think its a ms distributed filesystem host and so refuses to show access to files unless you map the device to a drive. This cleared it up for me and is working great accessing with a windows 7 box.