„Petru Maior" University, Târgu-Mureș
Science Department
Information Technolgy – Master Course

Distributed File System

Students:

Bardosi Florin

Cifor Crina

Danciu Ioana

Hintea Dan Alexandru

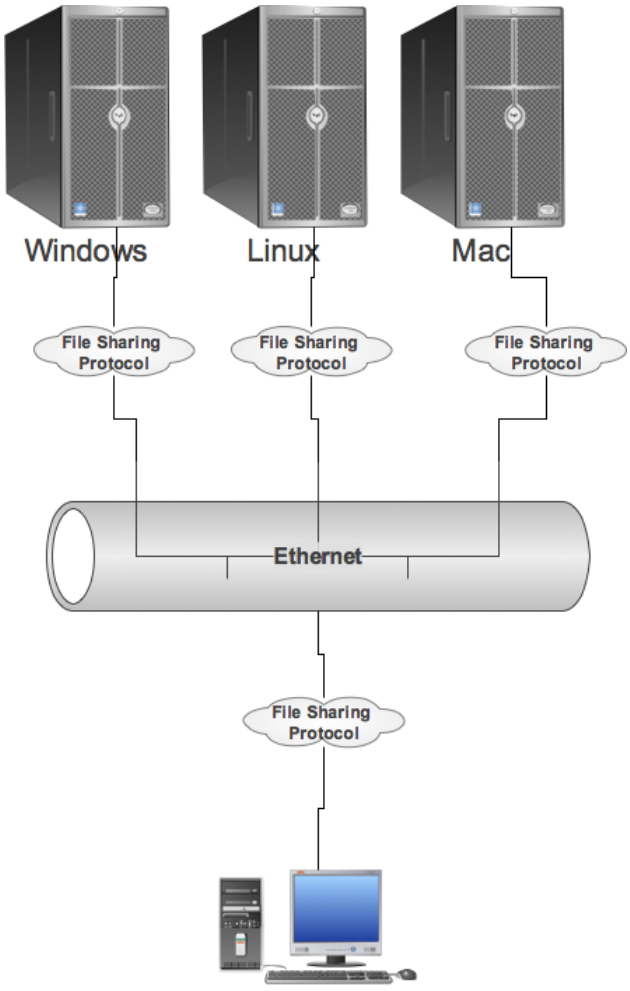
Table of Contents

Introduction

By "*Distributed File System*" (a.k.a. DFS) we understand a transparent mechanism, which allows multiple users on multiple machines to access files located on a remote host, as though these files would actually be located on the host computer. A DFS mechanism implies a series of computers connected to each other, which use a common file sharing protocol.

The current document presents several DFS solutions, assuming the machines are connected via a standard, 100Mbps Local Area Network (a.k.a. LAN), and running one of the following Operating Systems (a.k.a. OS): Microsoft® Windows® (versions XP, Vista, 7), Linux (distribution Ubuntu) and Mac OS X (version 10.6.2 Snow Leopard).

Below is a high-level view of DFS architecture:



User Machine (Windows, Linux or Mac)

The main focus will be on DFS solutions that are provided out-of-the-box by the upper-mentioned OSs', and maybe on 3rd party solutions if some specific OS lacks the implementation of a specific DFS. What we research in the following paragraphs are the DFS protocols that allow these systems to share files with each other, the ease-of-use of

each of these protocols taking into account the configuration process and the way of using it; the speed of the protocol and last but not least the reliability of the protocol.

This document does not intend in any way to re-invent the wheel, so to say, but simply wants to provide the reader with a clear picture on how to successfully configure and use a file-sharing environment between any or all of the three mentioned operating systems.

Before going into details, a few things should be said about file sharing between any of these systems. Whenever a DFS environment is established, one machine acts as a client and another one or more act as a server. The main idea in this scenario is that the client sees all the shared files on the servers, as they were located on the local computer. Depending on the file-sharing configuration the user should be able to view the share list, read/write existing files, create/remove files and create/remove folders. The configuration is usually made on per user basis, and can enable/disable specific rights for specific users.

What is Samba and when do you need it?

Samba is a set of tools to share files and printers. It implements the SMB network protocol, which is the heart of Windows networking. OS X also recognizes Samba shares.

Samba can be used to:

- Act as a server for Windows (or Samba) clients: share folders and printers, including PDF pseudo-printers so all the computers in your network may write PDF files
- Act as a domain controller in a Windows network (authenticating users, etc.)
- Do some more complex things, such as using a Windows domain controller to authenticate the users of a Linux/UNIX machine

Samba is freely available under the GNU General Public License. More information about Samba can be found at http://www.samba.org.

Configure Samba Client on Windows Vista

The system of file and printer sharing on Windows Vista is based on Samba tools.

Configuring file and printer sharing behavior in Windows Vista consists of the following:

- Setting the workgroup name to be the same as the other computers
- Setting the network location type
- Enabling file and printer sharing options

Setting the Workgroup Name

For easier and faster discovery of computers on your home network, it is better that all computers on a home network would be configured for the same workgroup name. If computers are in multiple workgroups, it can take additional time and effort to discover all of the computers on the network.

In the **System** window, the workgroup name is listed in the **Computer name, domain, and workgroup settings** section.

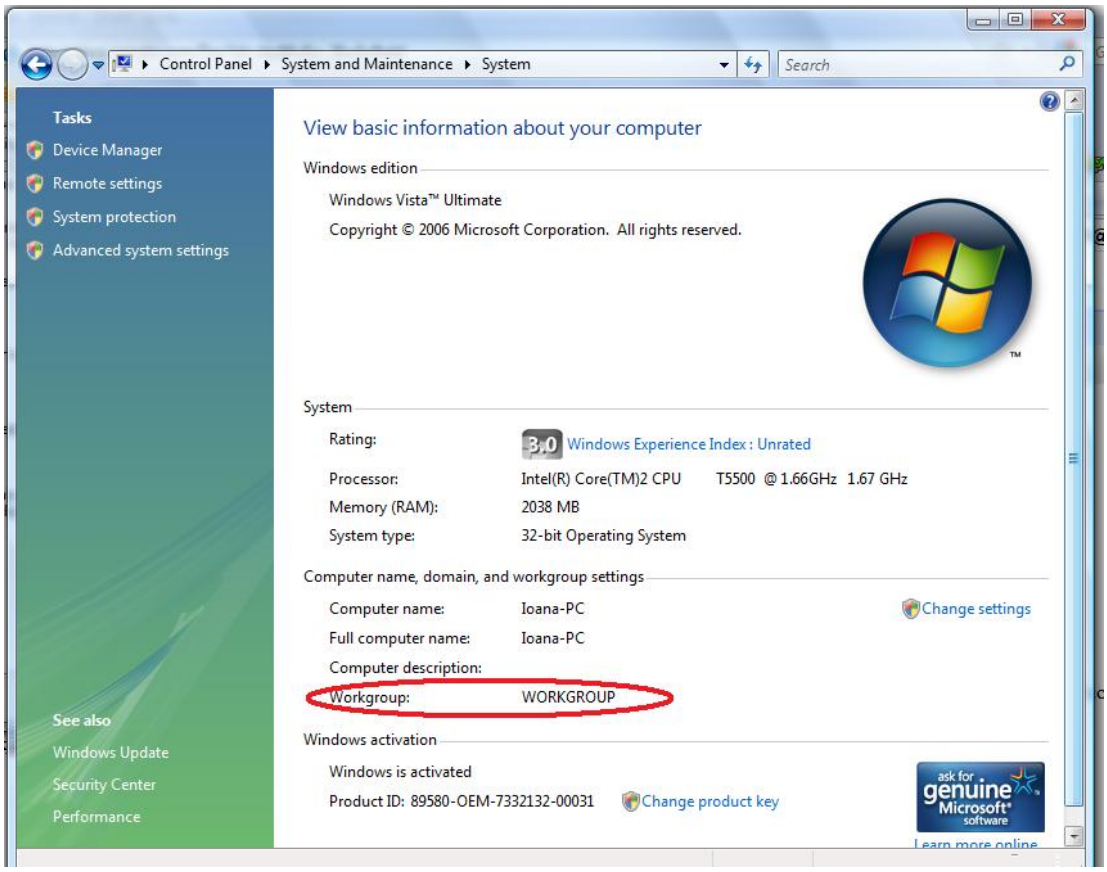


Figure 1: The System file on Windows Vista

If you want to change the Workgroup, you need to click on Change settings:

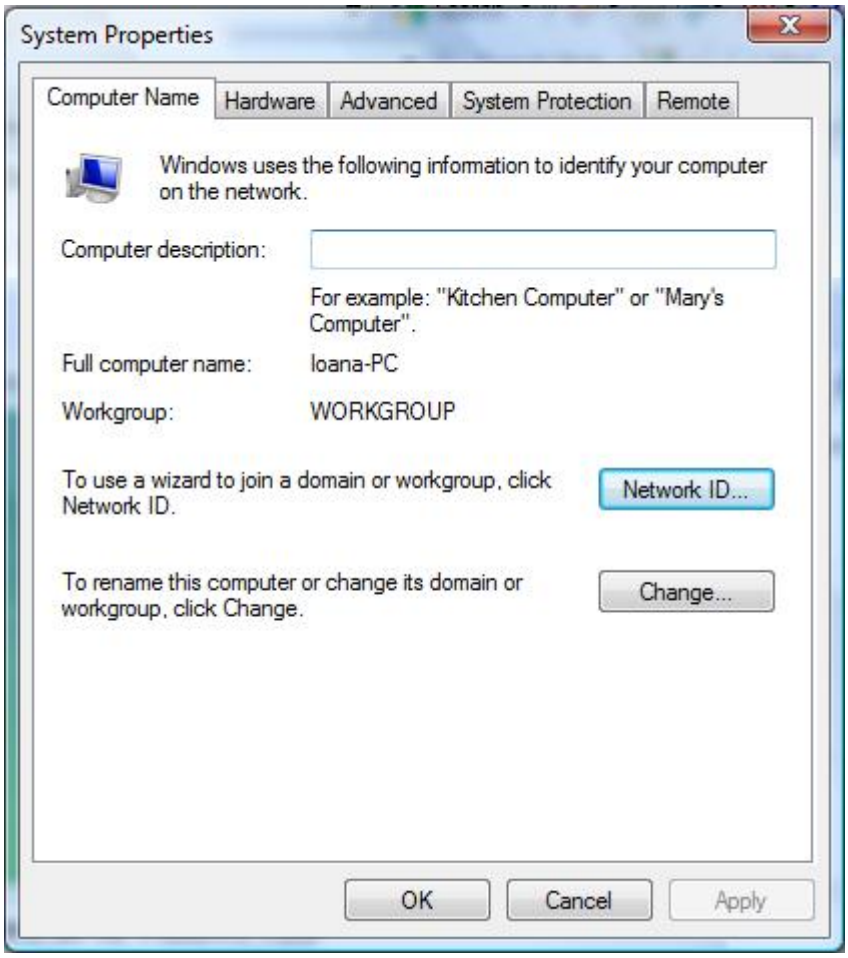


Figure 2: Change Workgroup on Windows Vista

Setting the Network Location Type

The network location type in Windows Vista is a setting that allows Windows Vista to automatically configure security and other settings based on the type of network to which the computer is connected. The Windows Vista network location types are the following:

Domain The computer is connected to a network that contains an Active Directory domain controller for the domain to which the computer is joined. An example a domain network type is an organization intranet.

Public The computer is connected to a network that has a direct connection to the Internet. Examples of public network types are public Internet access networks such as those found in airports, libraries, and coffee shops.

Private The computer is connected to a network that has some level of protection from the Internet and contains known or trusted computers. Examples of private network

types are home networks or small office networks that are located behind an Internet gateway device that provides firewall against incoming traffic from the Internet.

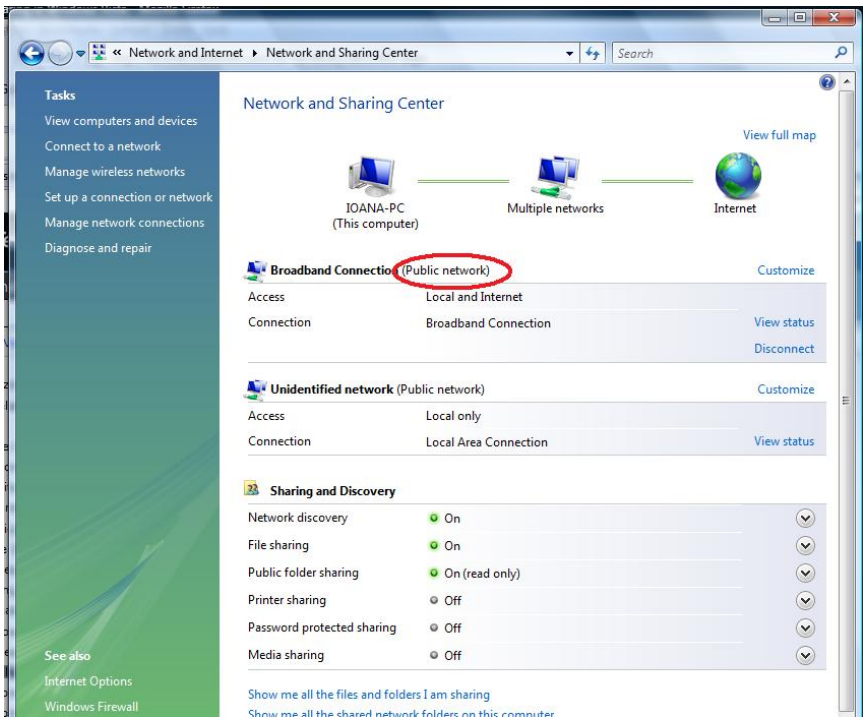


Figure 3: Network and Sharing CenterWindow

You are able to customize the location:

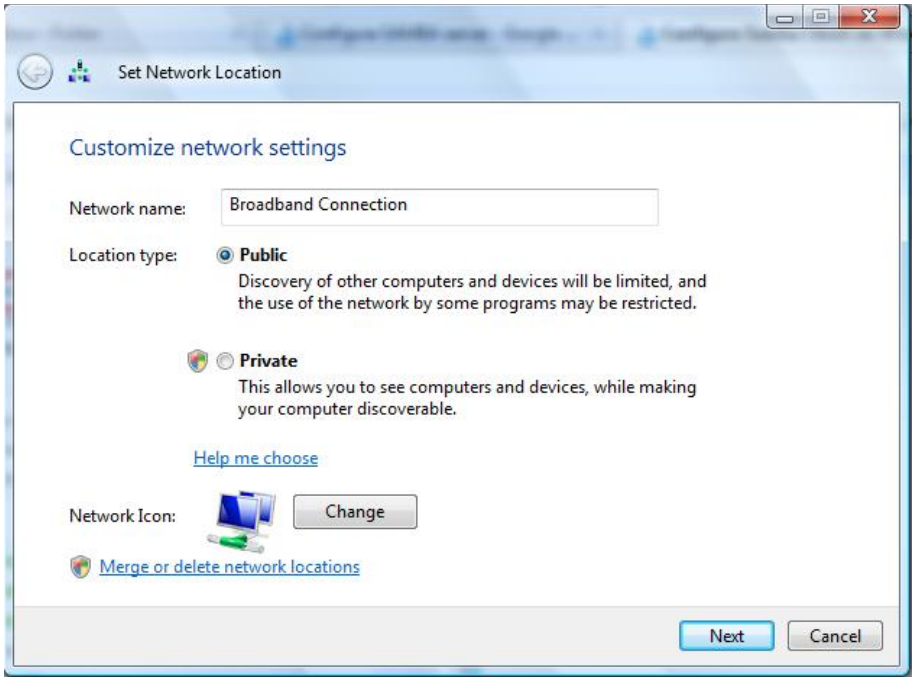


Figure 4: Setting the Network Location

Enabling File and Printer Sharing Options

By changing your network location type to private, network discovery is automatically enabled in the **Sharing and Discovery** section of the **Network and Sharing Center** window. The following additional file and printer sharing options must be manually enabled:

File sharing

Public folder sharing

Printer sharing

Password protected sharing

When all of these sharing and discovery options are enabled, the computer can:

Locate other computers and devices on your home network and have other computers locate your computer

Share its folders

Share its Public folder

Share its printers

Require user names and passwords for other computers that connect to the shared folders and printers of this computer

Figure 5: Sharing and Discovery Tab

With password protected sharing enabled, other computers on your network will not be able to access your shared folders, including the Public folder, without a user name or password that corresponds to a user account on the computer with the shared folder. When a user on another computer tries to connect to the shared folder, they will send the user name and password of the account that they used to log on to their own computer. For example, if they logged on to their computer with the “Bob” account and a password, then the “Bob” name with its password is sent when connecting to a shared folder on another computer.

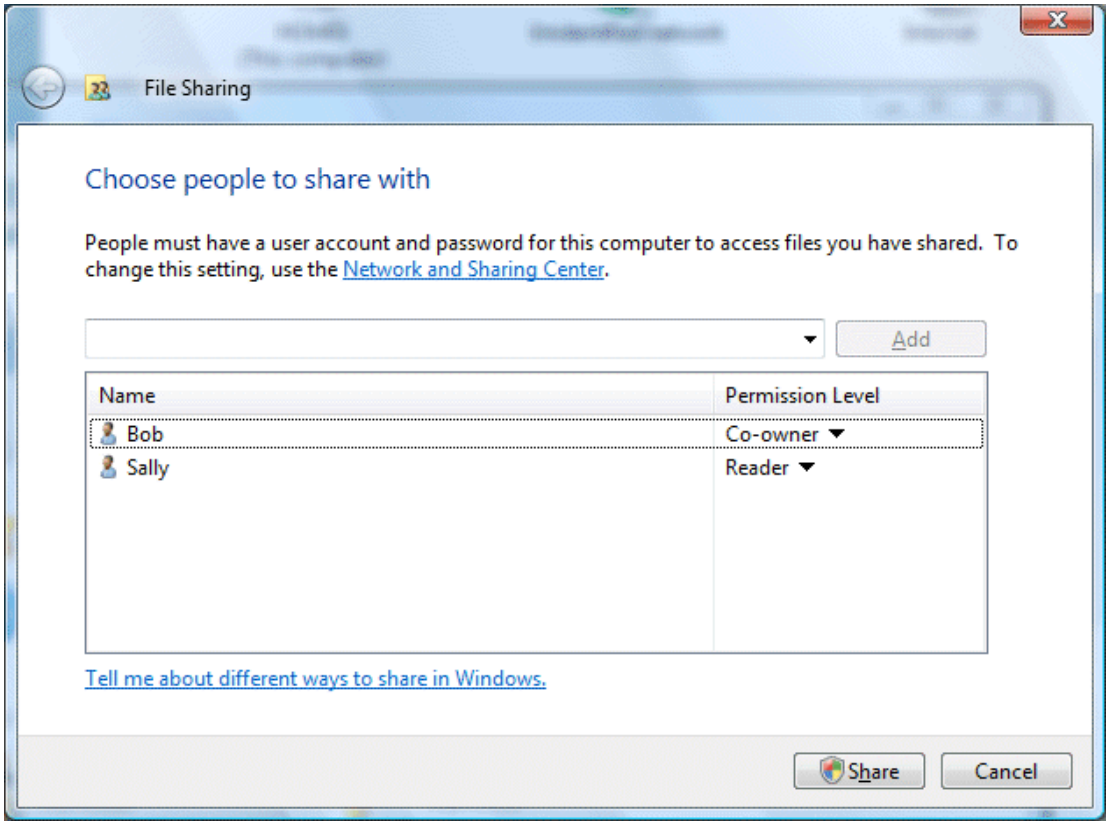


Figure 6: File Sharing Window

You are able to add computers, you want to share your files with; and you are able to set the levels of permissions for each people. System administrators and users with administrator accounts on computers can assign permissions to individual users or groups. The following list shows the typical permissions levels for files and folders:

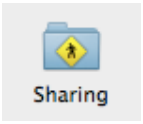
Full Control - Users can see the contents of a file or folder, change existing files and folders, create new files and folders, and run programs in a folder.
Modify - Users can change existing files and folders but cannot create new ones.
Read and Execute - Users can see the contents of existing files and folders and can run programs in a folder.

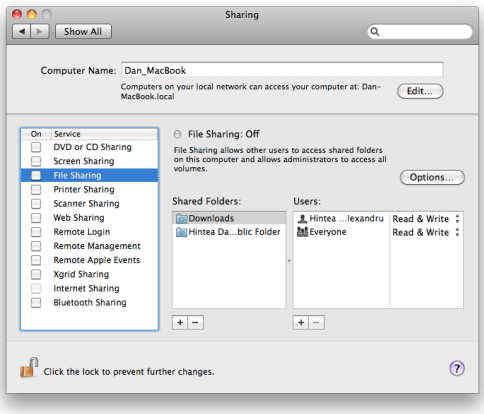
- Read - Users can see the contents of a folder and open files and folders.
- Write - Users can create new files and folders and make changes to existing files and folders.

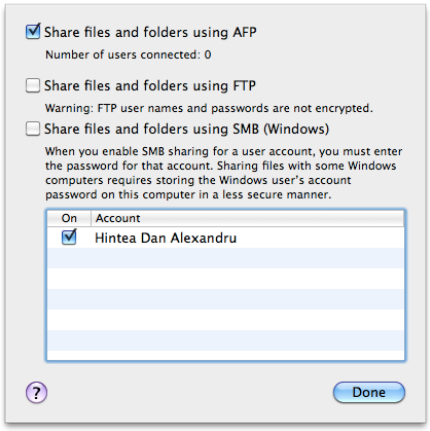
File sharing on Mac OS X

Setting Mac OS X to act as a DFS Server

This paragraph describes how to configure a Mac system to act as a file-sharing server. The Mac environment provides three protocols for this: Apple Filing Protocol (a.k.a. AFP), File Transfer Protocol (a.k.a. FTP) and Samba protocol (a.k.a. SMB). All of these are included in the latest version of the Mac OS X operating system. As the AFP is primarily used to share files between Mac systems only, we'll focus only on the last two sharing protocols as they are more widely spread among operating systems. To select via which of these protocols the user wants to share the files, he must navigate to *System Preferences -> Sharing* (under Internet and Wireless group). Check the *File Sharing* section from the left column and click *Options* to bring the protocols screen.





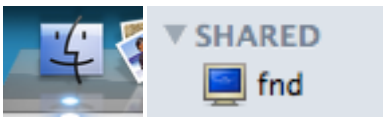


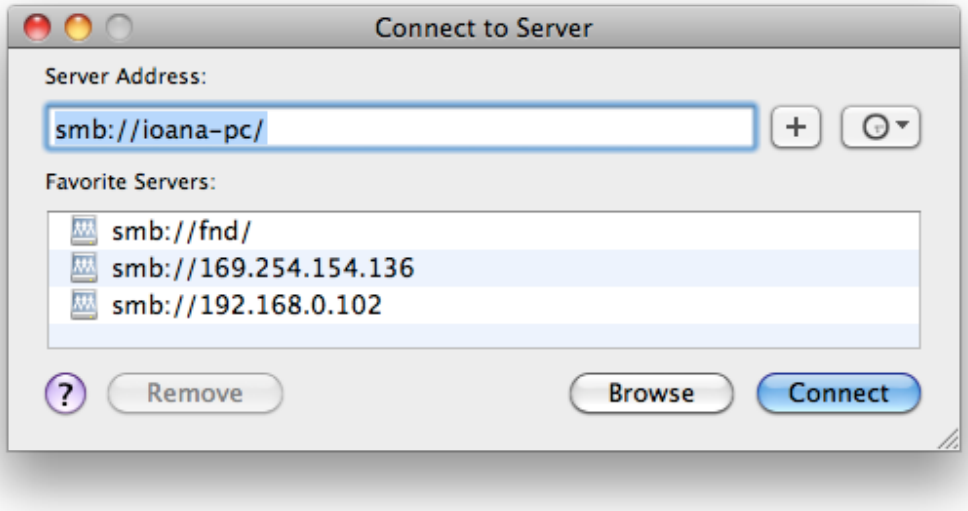
You can see on this screen the list with the three mentioned protocols. As we said before we should check the FTP and SMB options only, as we'll not be using the AFP sharing. From this screen we also need to select which users will have access to the file shares. The account list below the protocol list displays the local accounts, as these will be the credentials via which the shares will be made available. There is also an *Everyone* account which is used for anonymous acces. The rights set on this account will specify what type of access will anonymous clients have. On this screen the user can also select the type of access each client will have on the specified share; possible options are: Read&Write, Read Only and Write Only.

Once these settings are made the specified shares will be available on the current network and are accesible via any Samba and FTP Client.

Accesing network shares

The network shares (FTP and Samba) can easily be accesed from Mac OS via it's main file explorer, *Finder*. To access a network share, open a new Finder Window via the Finder icon in the launcher bar. Notice the left column of the finder window and the *Shared* section. Normally Mac should auto-detect all the shares in the current network and display a list of network devices here. A click on the desired device will display the shares on that specific device. However, if the share you are looking for is not in the list, you can manually specify the network path to that share via the keyboard shortcut *Cmd+K* or via Finder's menu command: *Go->Connect To Server*....





File sharing on Linux

What to install

The samba package is a meta-package intended to be installed on servers. Clients do not need this meta-package (you are acting as a client if you need to access files on another computer). For example, installing samba is not necessary if you only need your Ubuntu system to do any of the following:

- Access shared folders, drives and printers on a Windows computer (that is, act as a client with Windows servers). To do this, you only need the **smbfs** plug-in.
- Have your Windows computer use (via a network) a printer that is attached to a Linux computer. CUPS can be configured to make the printer accessible to the network.
- Share directories between two Linux computers. You can use NFS or setup an SSH server on one computer and access it from other computers using an scp or sftp client, or Places -> Connect to Server... and choose "SSH" as the service type.

To install Samba you must access Ubuntu Software Download, and choose Samba application, click on Install button, and is there, you can use it.

The other way to install samba: open a terminal window and enter the following command:

```
sudo apt-get install samba smbfs
```

Configuring a Samba Server

The default configuration file (/etc/samba/smb.conf) allows users to view their home directories as a Samba share. It also shares all printers configured for the system as Samba shared printers. In other words, you can attach a printer to the system and print to it from the Windows machines on your network.

Using Graphical Configuration

To configure Samba using a graphical interface, use the **Samba Server Configuration Tool**.

The **Samba Server Configuration Tool** is a graphical interface for managing Samba shares, users, and basic server settings. It modifies the configuration files in the /etc/samba/ directory. Any changes to these files not made using the application are preserved.

To use this application, you must have root privileges. To start the **Samba Server Configuration Tool** from the desktop, go to the **Main Menu Button** => **System Settings** => **Server Settings** => **Samba**.

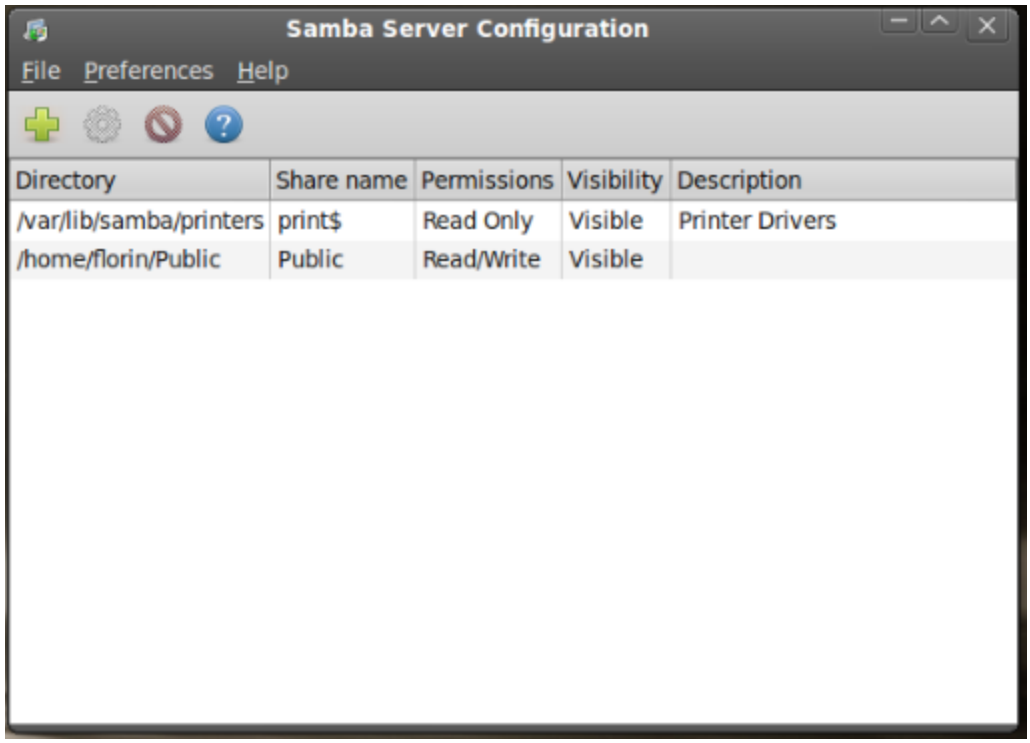


Figure 1. Samba Server Configuration Tool

The **Samba Server Configuration Tool** does not display shared printers or the default stanza that allows users to view their own home directories on the Samba server.

Configuring Server Settings

The first step in configuring a Samba server is to configure the basic settings for the server and a few security options. After starting the application, select **Preferences** => **Server Settings** from the pull down menu. The **Basic** tab is displayed as shown in Figure 2.

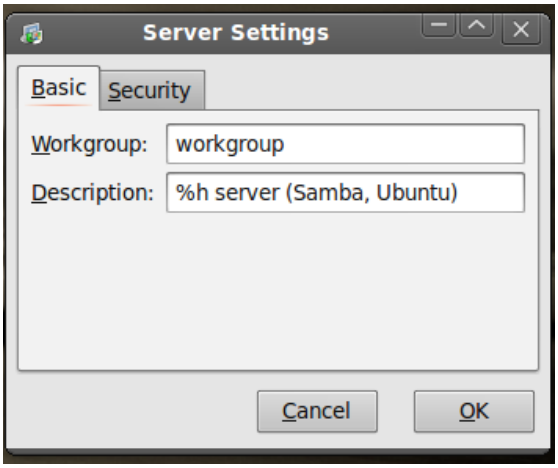


Figure 2. Configuring Basic Server Settings

On the **Basic** tab, specify which workgroup the computer should be in as well as a brief description of the computer. They correspond to the workgroup and server string options in smb.conf.

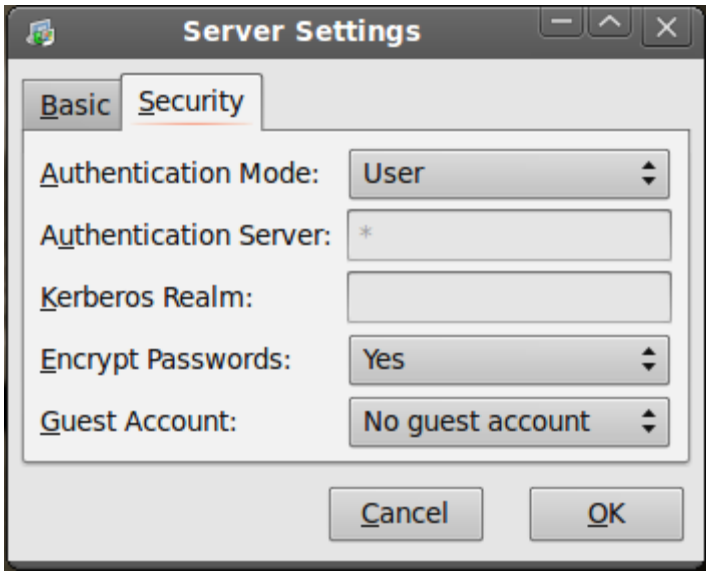


Figure 3. Configuring Security Server Settings

The **Security** tab contains the following options:

- **Authentication Mode** — This corresponds to the security option. Select one of the following types of authentication.
  - **ADS** — The Samba server acts as a domain member in an Active Directory Domain (ADS) realm. For this option, Kerberos must be installed and configured on the server, and Samba must become a member of the ADS realm using the net utility, which is part of the samba-client package. Refer to the net man page for details. This option does not configure Samba to be an ADS Controller.
  - **Domain** — The Samba server relies on a Windows NT Primary or Backup Domain Controller to verify the user. The server passes the username and password to the Controller and waits for it to return. Specify the NetBIOS name of the Primary or Backup Domain Controller in the **Authentication Server** field.

    The **Encrypted Passwords** option must be set to **Yes** if this is selected.
  - **Server** — The Samba server tries to verify the username and password combination by passing them to another Samba server. If it can not, the server tries to verify using the user authentication mode. Specify the NetBIOS name of the other Samba server in the **Authentication Server** field.
  - **Share** — Samba users do not have to enter a username and password combination on a per Samba server basis. They are not prompted for a username and password until they try to connect to a specific shared directory from a Samba server.

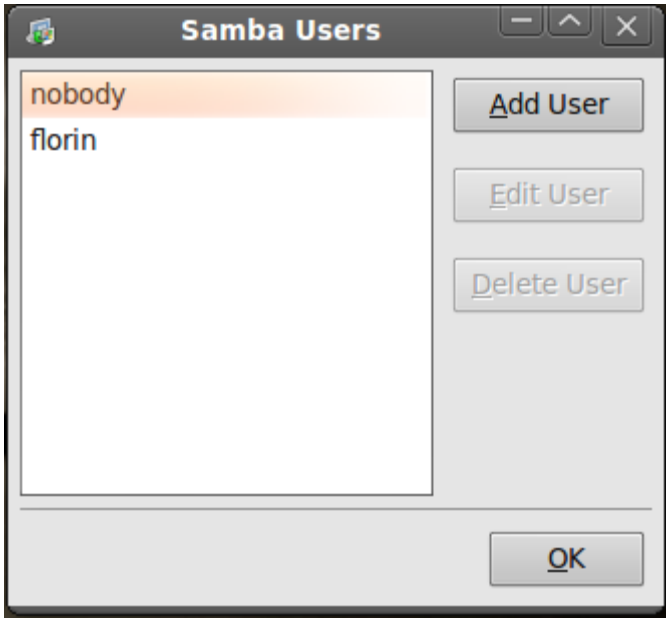
- **User** — (Default) Samba users must provide a valid username and password on a per Samba server basis. Select this option if you want the **Windows Username** option to work.

- **Authentication Server**
- **Kerberos Realm**
- **Encrypt Passwords** — This option must be enabled if the clients are connecting from a Windows 98, Windows NT 4.0 with Service Pack 3, or other more recent versions of Microsoft Windows. The passwords are transferred between the server and the client in an encrypted format instead of in as a plain-text word that can be intercepted. This corresponds to the encrypted passwords option.
- **Guest Account** — When users or guest users log into a Samba server, they must be mapped to a valid user on the server. Select one of the existing usernames on the system to be the guest Samba account. When guests logs in to the Samba server, they have the same privileges as this user. This corresponds to the guest account option.

After clicking **OK**, the changes are written to the configuration file and the daemon is restart; thus, the changes take effect immediately.

Managing Samba Users

The **Samba Server Configuration Tool** requires that an existing user account be active on the system acting as the Samba server before a Samba user can be added. The Samba user is associated with the existing user account.



Adding a Share

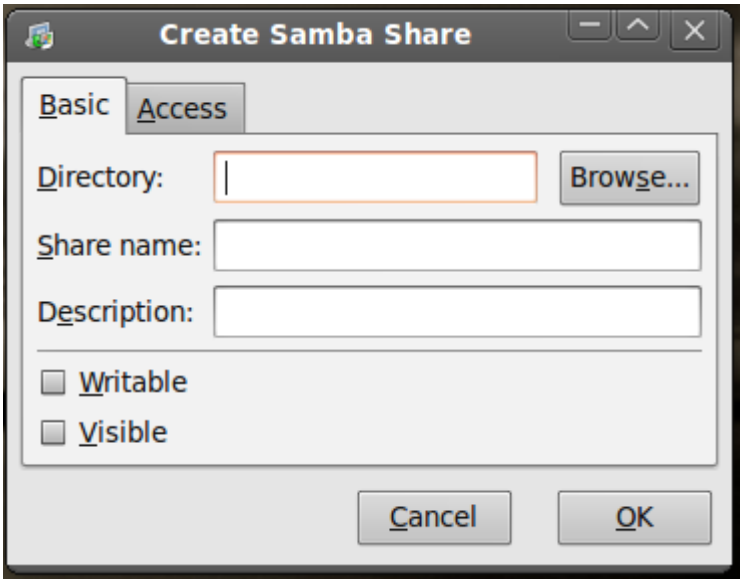


Figure 6. Adding a Share

To add a share, click the **Add** button. The **Basic** tab configures the following options:

- **Directory** — The directory to share via Samba. The directory must exist.
- **Share name**
- **Descriptions** — A brief description of the share.
- **Basic Permissions** — Whether users should only be able to read the files in the shared directory or whether they should be able to read and write to the shared directory.

On the **Access** tab, select whether to allow only specified users to access the share or whether to allow all Samba users to access the share. If you select to allow access to specific users, select the users from the list of available Samba users.

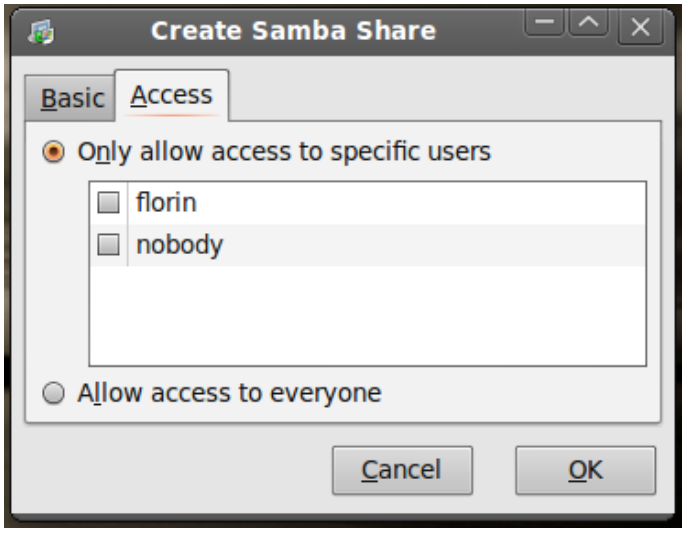


Figure 7. Give access to users

The share is added immediately after clicking **OK**.

Samba uses /etc/samba/smb.conf as its configuration file. If you change this configuration file, the changes do not take effect until you restart the Samba daemon with the command service smb restart.

To specify the Windows workgroup and a brief description of the Samba server, edit the following lines in your smb.conf file:

workgroup = *WORKGROUPNAME*
server string = *BRIEF COMMENT ABOUT SERVER*

Replace *WORKGROUPNAME* with the name of the Windows workgroup to which this machine should belong. The *BRIEF COMMENT ABOUT SERVER* is optional and is used as the Windows comment about the Samba system.

To create a Samba share directory on your Linux system, add the following section to your smb.conf file (after modifying it to reflect your needs and your system):

[*sharename*]
comment = *Insert a comment here*
path = */home/share/*
valid users = *tfox carole*
public = no
writable = yes
printable = no
create mask = 0765

The above example allows the users tfox and carole to read and write to the directory /home/share, on the Samba server, from a Samba client.

Encrypted Passwords

Encrypted passwords are enabled by default because it is more secure. If encrypted passwords are not used, plain text passwords are used, which can be intercepted by someone using a network packet sniffer. It is recommended that encrypted passwords be used.

The Microsoft SMB Protocol originally used plaintext passwords. However, Windows NT 4.0 with Service Pack 3 or higher, Windows 98, Windows 2000, Windows ME, and Windows XP require encrypted Samba passwords. To use Samba between a Linux system and a system running one of these Windows operating systems, you can either edit your Windows registry to use plaintext passwords or configure Samba on your Linux system to use encrypted passwords. If you choose to modify your registry, you must do so for all your Windows machines — this is risky and may cause further conflicts. It is recommended that you use encrypted passwords for better security.

To configure Samba to use encrypted passwords, follow these steps:

1. Create a separate password file for Samba. To create one based on your existing /etc/passwd file, at a shell prompt, type the following command:

```
cat /etc/passwd | mksmbpasswd.sh > /etc/samba/smbpasswd
```

2. If the system uses NIS, type the following command:

```
ypcat passwd | mksmbpasswd.sh > /etc/samba/smbpasswd
```

3. The mksmbpasswd.sh script is installed in your /usr/bin directory with the samba package.

4. Change the permissions of the Samba password file so that only root has read and write permissions:

```
chmod 600 /etc/samba/smbpasswd
```

5. The script does not copy user passwords to the new file, and a Samba user account is not active until a password is set for it. For higher security, it is recommended that the user's Samba password be different from the user's system password. To set each Samba user's password, use the following command (replace *username* with each user's username):

```
smbpasswd username
```

6. Encrypted passwords must be enabled. Since they are enabled by default, they do not have to be specifically enabled in the configuration file. However, they can not be disabled in the configuration file either. In the file /etc/samba/smb.conf, verify that the following line does not exist:

```
encrypt passwords = no
```

7. If it does exist but is commented out with a semi-colon (;) at the beginning of the line, then the line is ignored, and encrypted passwords are enabled. If this line exist but is not commented out, either remove it or comment it out.

8. To specifically enable encrypted passwords in the configuration file, add the following lines to etc/samba/smb.conf:

```
encrypt passwords = yes
smb passwd file = /etc/samba/smbpasswd
```

9. Make sure the smb service is started by typing the command service smb restart at a shell prompt.

10. If you want the smb service to start automatically, use **ntsysv**, chkconfig, or the **Services Configuration Tool** to enable it at runtime.

The pam_smbpass PAM module can be used to sync users' Samba passwords with their system passwords when the passwd command is used. If a user invokes the passwd command, the password he uses to log in to the Red Hat Enterprise Linux system as well as the password he must provide to connect to a Samba share are changed.

To enable this feature, add the following line to /etc/pam.d/system-auth below the pam_cracklib.so invocation:

password required /lib/security/pam_smbpass.so nullok use_authtok try_first_pass

Starting and Stopping the Server

On the server that is sharing directories via Samba, the smb service must be running.

View the status of the Samba daemon with the following command:

/sbin/service smb status

Start the daemon with the following command:

/sbin/service smb start

Stop the daemon with the following command:

/sbin/service smb stop

To start the smb service at boot time, use the command:

/sbin/chkconfig --level 345 smb on

You can also use chkconfig, **ntsysv** or the **Services Configuration Tool** to configure which services start at boot time.

Browsing SMB shares

Ubuntu and Gnome make it easy to access files on a Windows network share.

Open the **Places** Menu, then click on **Network**. You will see a **Windows network** icon. Double-click to open it. The next window shows all the domains/workgroups found on your network. Inside each domain/workgroup you will see all the computers on the domain/workgroup with sharing enabled. Double-click on a computer icon to access its shares and files.

Before showing a computer's shares, your system may prompt you for a name and password. Fill in the form with the credentials of a valid user for the computer you are connecting to. You may additionally store that password in your keyring for convenience.

Note: The default installation of Samba does not synchronize passwords. You may have to run "smbpasswd" for each user that needs to have access to his Ubuntu home directory from Microsoft Windows.

Connect to a samba server

Ubuntu Client

On the Ubuntu client using the menu at the top, go to "Places" -> "Network". You will see an icon "Windows network" and should be able to browse to your shared folder. You will be asked for a password, leave it blank. Click the "Connect button.

Alternate: From the menu at the top select "Location" -> "Connect to a server". In the "Service type" pull down and select "Windows share". Enter the server ip address in the "Server:" box and the share name in the "Share:" box. Click "Connect" and then "Connect" again on the second dialog box (no need for a password).

If you would like to mount your SMB share using your (server) hostname rather than the IP Address, edit /etc/hosts and add your samba server (syntax IP Address hostname).

192.168.1.100 hostname

Where "hostname" = the name of your samba server.

Windows 7 Client

On the Windows 7 client you must use Network shortcut. There the Ubuntu smb server appears. Double click on server. If you will be asked for a username and password, complete with the username and password received from samba server administrator.

Sometimes linux users can't access Windows 7 shared folder. You can see them but you can't access them. There is a bug. The solution is simple. Uninstall Windows Live–ID Assistant and restart the operating system.

Advanced File Sharing

We started with the base of Samba file-sharing. The above-mentioned items should be enough to get you started. Next we will add details that you might or might not need.

If you have more than one network card

If you have more than one network card (or interface) then you have to define where you want Samba to run. In smb.conf under the [global] section, add:

interfaces = 127.0.0.1, 192.168.0.31/24
bind interfaces only = yes

The first address (127.0.0.1), is a loopback network connection (it's your own machine). The second address (eg. 192.168.0.31), is the address of the card you want Samba to run

on, the second number (24) is the subnet default for a CLASS-C network. It may vary depending on your network.

With "bind interfaces only" you limit which interfaces on a machine will serve SMB requests.

You can limit which IP address can connect to your Samba server adding these lines:

hosts allow = 127.0.0.1, 192.168.0.31, 192.168.0.32
hosts deny = 0.0.0.0/0

The loopback address must be present in the first line. The second line deny access from all IP address not in the first line.

Private and public shares in same config

First you'll want to set this up in the [global] section of your smb.conf

```
[global]
    security = user
    encrypt passwords = true
    map to guest = bad user
    guest account = nobody
```

security = user restricts logins to users on your server. encrypt passwords = true is necessary for most modern versions of Windows to login to your shares. map to guest = bad user will map login attempts with bad user names to the guest account you specify with guest account = nobody. That is, if you attempt to login to the share with a user name not set up with smbpasswd the you will be logged in as the user *nobody*.

Next the private share

```
[private]
    comment = Private Share
    path = /path/to/share/point
    browseable = no
    read only = no
```

If browsable is set to no the share will not show up on graphical browsers such a "My Network Places" on Windows or Places -> Network on Ubuntu.

path is the path to the directory that you want to share out. browseable = no will have the share not show up when users browse the network. read only = no will let you, as an authenticated user, write to the share.

Finally, the public share

```
[public]
    comment = Public Share
    path = /path/to/share/point
    read only = no
    guest only = yes
    guest ok = yes
```

Again, path is the path to the directory that you want to share out. read only = no will allow users to write to this share. guest only = yes and guest ok = yes will allow guest logins and also force users to login as guests. **The user you specified with guest account in the [global] section must have write permissions on /path/to/share/point in order to write files to the share.**

When Windows attempts to access a SMB share it will use the current Windows user name and password. The map to guest = bad user trick above allows access to the public share only if you give Samba an incorrect user name. If you give it a valid user name, but a bad password, the login will fail and Windows will give you a password prompt when you try to access the share. If you have the same user name for your Windows machine and your Ubuntu machine, you could be unwittingly giving the Samba server a valid user name, but invalid password. To resolve this you will either have to change the Windows user name, or to remove that user name from the Samba password file with sudo smbpasswd -x [username].

The above uses security = user. To access the private shares you will have to make sure the user exists in smbpasswd. These users must also already exist as normal users on your machine. You add users to smbpasswd simply by running sudo smbpasswd -a [username] and giving a password.

Setting permissions

To set permissions of newly created documents / files edit /etc/samba/smb.conf and in the [global] section add :

create mask = 0644
directory mask = 0755

Securing Samba

Here are some general advices on security considerations and is not an exhaustive review of samba security.

You can configure those in /etc/samba/smb.conf .

Network and shares

- Networking Section - use "hosts allow" and "hosts deny"

hosts allow = 127.0.0.1 192.168.1.0/24
hostal allow = 127.0.0.1 192.168.1.1 192.168.1.2
hosts deny = 0.0.0.0/0

- hosts deny 0.0.0.0/0 = all others.
- Shares
    - When defining a share, consider the following options :

1. browseable = no ~ Shares will not show up when browsing your network.
2. users = user1 user2 ~ List of users able to access the share

Users

When setting up a Samba share, you can limit the users who have access to your share

```
[private]
    comment = Private Share
    path = /path/to/share/point
    browseable = no
    read only = no
    valid users = user1 user2 user3
```

Now only samba users user1, user2, and user3 will have access to the share "private".

Firewall

Configure your firewall (iptables) to limit access to your server. Samba uses ports

- UDP ports 137 and 138
- TCP ports 139 and 445

Conclusions

The Distributed File System mechanism is a usefull and productive way to share resources between several devices using various operating systems. There are a lot of protocols defined for distributed file systems but two of them proved to be reliable over time and are implemented by most operating systems available on the market. The two protocols are Samba and FTP (File Transfer Protocol). These protocols are fairly easy to confgure and have quite good performances, the main limitation being the actual network speed.