

Accessing Shares on a Windows Computer from a Linux Workstation

A logical first step for new Linux users is, after building a Linux workstation, to want to connect it to their existing Windows network. This document describes how to connect to shares on a Windows computer from a Linux workstation.

There are zillions of network configurations out there, and it'd be impossible to document and cover each of them separately. Instead, I'm going to describe how to connect to a popular type of Windows network and access shares on a Windows computer on that network. From there, you'll probably know enough to make changes if your network is different.

Overview of the Process

There are four basic steps to accessing shares on a Windows computer from your Linux workstation. First, you'll configure an accessible share on the Windows computer. Next, you'll make sure that the Samba client software is installed and running on your Linux workstation. Third, you'll set up a mount point on your Linux workstation that will map to the share on your Windows computer. And, finally, you'll make changes to three configuration files on your Linux workstation so that your Linux workstation can communicate with the share on the Windows computer.

Definitions

I'm going to assume that you've got a Linux workstation that you want to attach to an existing Windows network. For the purposes of this discussion, it doesn't matter what type of Windows network you're attaching to - simply that there is a computer on the network that has data that you want to be able to get to from your Linux workstation. This Windows computer might be a server, but it could also just be another workstation on the network. In either case, however, it has directories and/or files that you want to share.

One scenario is to have just a single specific directory, say, D:\common\data, on the Windows computer, that needs to be accessed from the Linux workstation. Another scenario is to have an entire drive, say, E:\, on the Windows computer that needs to be accessed from the Linux workstation.

In both cases, the entity on the Windows computer that you want to access from your Linux workstation is called a 'share'.

1. Creating an Accessible Share on the Windows Computer

There are two steps in creating an accessible share on the Windows computer. The first is to create a share, and the second is to allow it to be accessed from another computer.

1a. Creating a share

On the Windows computer, open Windows Explorer and select the drive or directory that is to be shared. Right click on the item and select the "Sharing" menu option from the resulting context menu. The Properties dialog, with the Sharing tab selected, will display, as shown in Figure 1.

Note: The screen shots for this section were all taken on a Windows 2000 workstation. Other versions of Windows look similar, although not necessarily exactly the same.

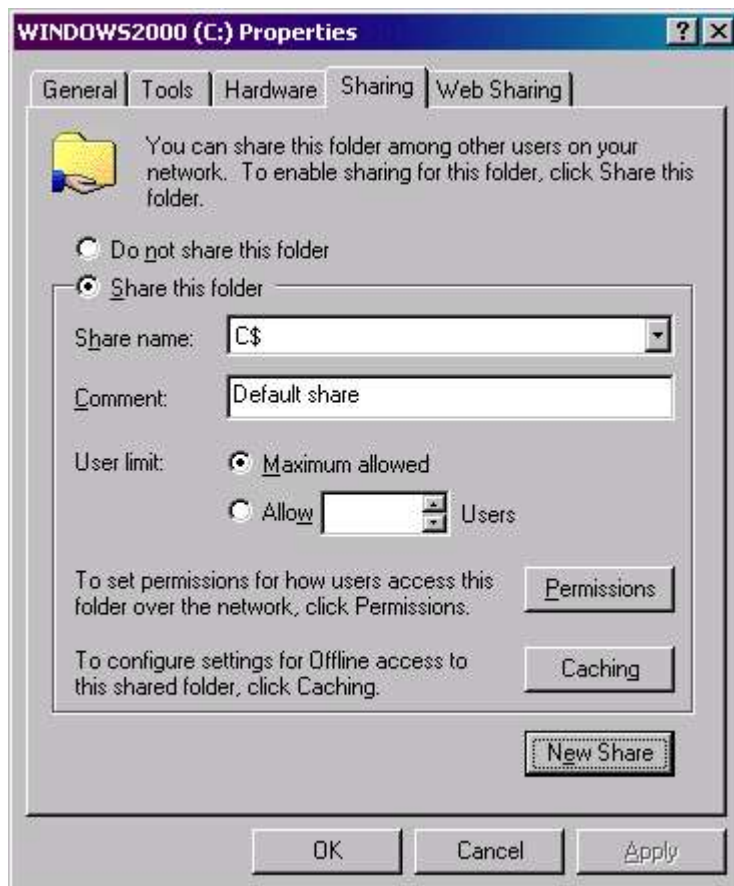


Figure 1. The Sharing tab of the Properties dialog.

Depending on which drive or directory you've selected, there might already be an administrative share created by the operating system, as there is in Figure 1. You can ignore it. However, if this is the case, the Share this folder option button will be selected and the New Share button will be enabled. If not, you'll need to select the Share this folder option button in order to enable the New Share button. Then click on the New Share button to bring forward the New Share dialog, as shown in Figure 2.

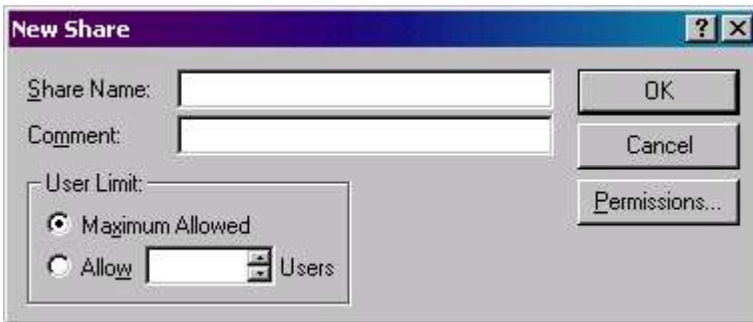


Figure 2. The New Share dialog.

Enter a text string that will be used to identify this share in the Share Name text box. This Share Name will be visible on other workstations and will be used in some configuration files as well, so make it descriptive but not too complicated. Don't give it a name that is too much like other resources on the computer, in order to avoid as much confusion as possible later.

In Figure 3, I've used the Share Name of "commondata" for the entire C drive of the Windows workstation simply to make the example easy to follow. In real life, I wouldn't be sharing the C drive, of course, but would have named the share something more specific. For example, if you've got a directory of customer blueprints, you might use a Share Name of "blueprints", while if you're sharing an entire drive of conference related data, you might use a Share Name of "conference_data". I recommend that you do not use spaces in your Share Name even though it's technically permitted, as it can introduce complications later on.



Figure 3. The Share Name for this drive is "commondata".

Once you've entered the Share Name, click OK. We'll address that interesting looking Permissions button shortly. The Properties dialog now shows the Share Name that you've created in the Share Name drop down combo box, as shown in Figure 4. If this is the first Share Name associated with this drive or directory, the Share Name drop down combo box can be opened but there will only be the share you created. If there are other shares associated with this drive or directory, you can open the drop down combo box and see what they are.



Figure 4. The Properties dialog after creating a share.

Windows will now allow this drive or directory to be accessed from other computers, provided that the permissions for that resource are set up appropriately. This is our next step.

1b. Making a share accessible

Now it's time to make the share available to others via permissions. Actually, it's more accurate to say that we're going to restrict access to the share to only those who should be able to access it. In the Properties dialog, make sure that the Share Name you are working with is displayed

in the Share Name drop down combo box, and then click the Permission button. The Permissions dialog for that share will display, as shown in Figure 5.

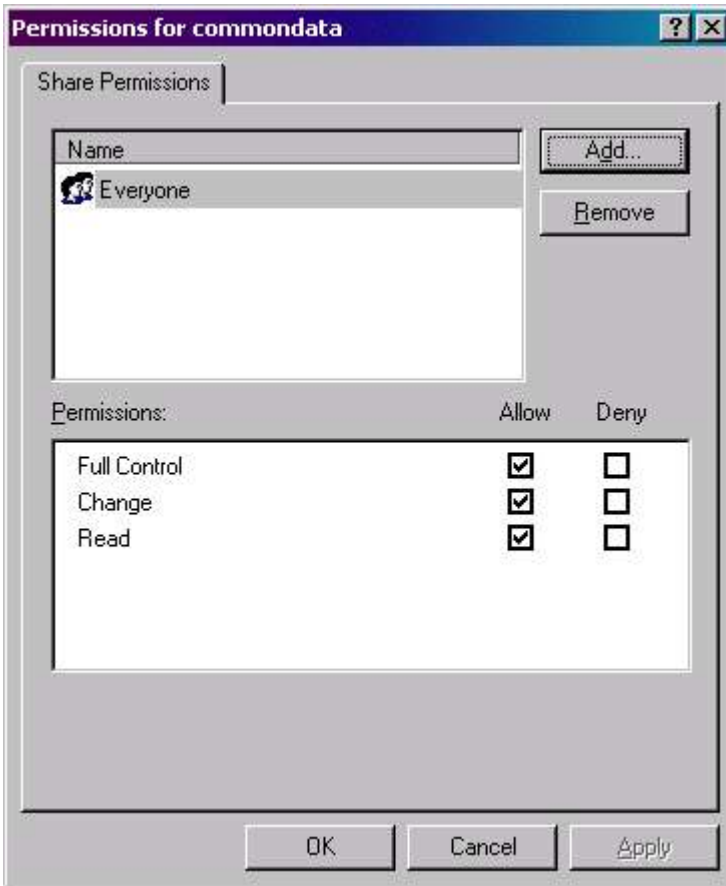


Figure 5. The Permissions dialog for the commondata share.

As you can see, this share is currently wide open - the Everyone group has Full Control, Change and Read permissions for this share. Since every user belongs to the Everyone group, there are no restrictions at all as to who can access this share. This is probably not what you're looking for. Instead, you probably want to restrict access to either a specific user, several individual users, or one or more groups. Click on the Add button in the Permissions dialog, and you'll get a list of all users and groups that are part of the domain (if this machine belongs to a domain) or that are part of the local machine (if this machine is on a peer-to-peer network.)

Individual users are identified by a single head icon to the left of the user's name while groups are identified by a icon with two heads. In Figure 6, you'll see some of the groups that are part of a standard Windows 2000 domain, with the Everyone group already selected and displayed in the edit box in the bottom half of the dialog.

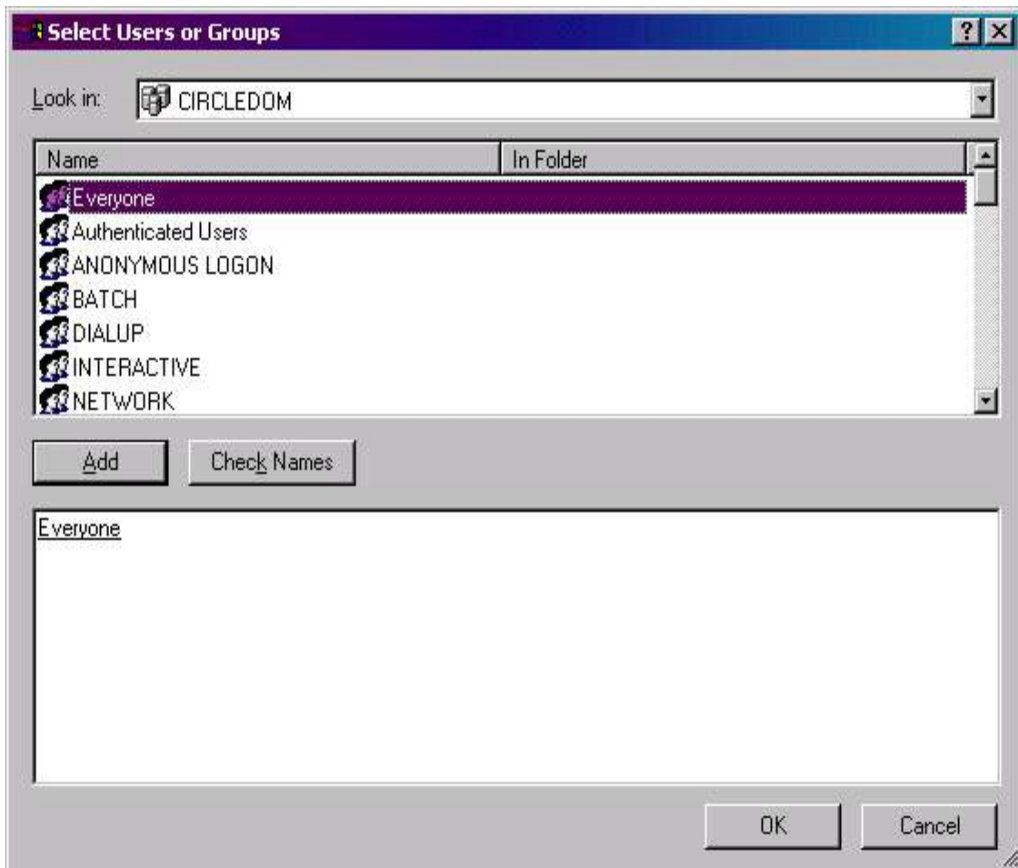


Figure 6. The Select Users or Groups dialog allows you to choose which entities have access to a share.

In Figure 7, I've granted a single user named 'john' permission to access the share by scrolling down through the list box in the top half of the Add Users or Groups dialog in Figure 6, and clicking the Add button. By default, new users and groups are only granted Read permission.

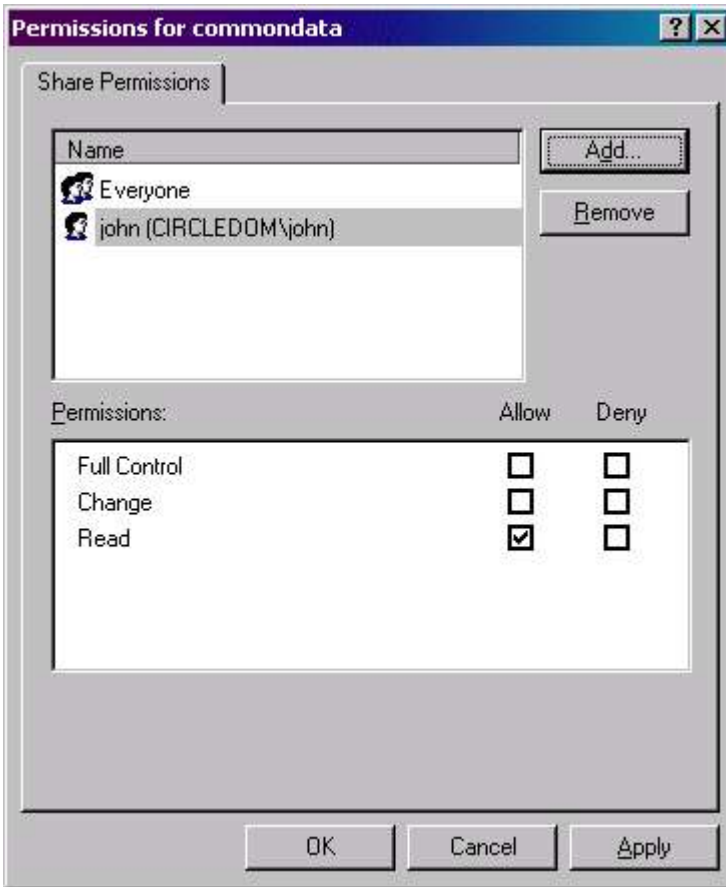


Figure 7. Adding user 'john' to the commondata share.

In Figure 8, I've removed the user 'john' but given the group Alex & Friends permission to access the 'commondata' share in a similar fashion.

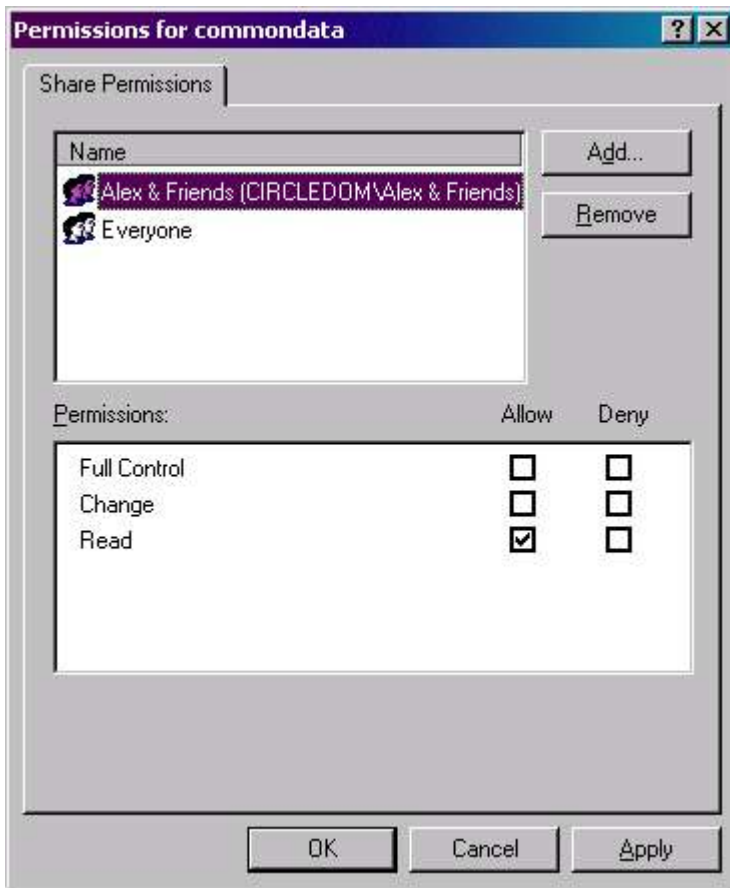


Figure 8. Adding group "Alex & Friends" to the commondata share.

Then, in Figure 9, I updated the permissions that the Alex & Friends group have, so that they now include Full Control and Change permissions as well.

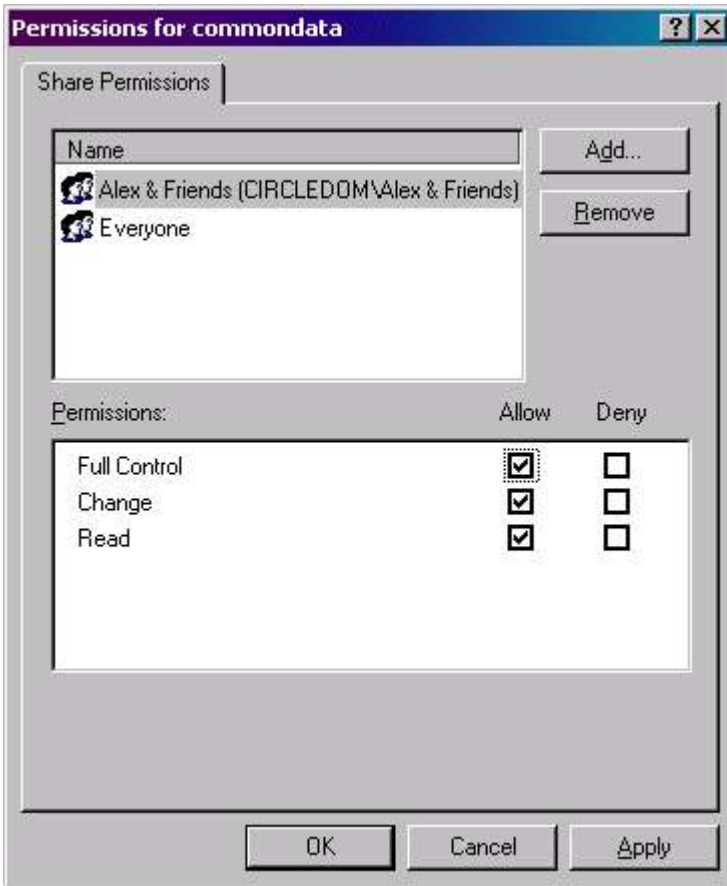


Figure 9. Giving group "Alex & Friends" additional permissions on the commondata share.

Finally, in Figure 10, I've removed the Everyone group so that only members of the Alex and Friends group can access the 'commondata' share.

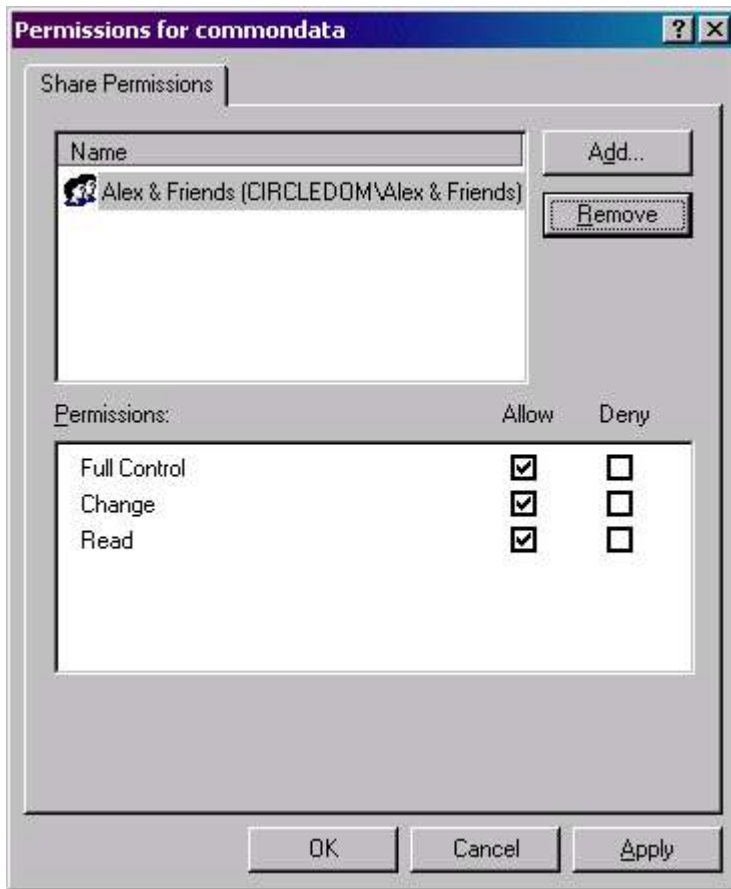


Figure 10

Click the Apply button and the OK button, and you are done with configuring the Windows workstation's share.

2. Make Sure the Samba Client is Installed and Running

Linux relies on a piece of software called Samba to communicate with non-Linux networks. There are two parts to Samba; the Samba client and the Samba server. The client is installed on Linux workstations and is used so the workstation can communicate with machines on, say, Windows networks. The server is installed on Linux servers so that non-Linux machines (such as Windows workstations) can access shares on the Linux server, and even authenticate against a Linux server's database of users.

Since we have a Linux workstation accessing a Windows computer, we need to have the Samba client installed on the Linux workstation, and then we have to make sure the Samba service is running.

2a. Make Sure Samba is Installed

Samba is included with all popular Linux distributions, including Red Hat, SuSE and Mandrake, so it's highly likely that it's already installed on your Linux workstation. In order to check, you can do one of several things. The most foolproof way is to look in the package management tool for your distribution and see if it's been selected. In RedHat, select System Settings | Packages to display the Package Management dialog, and scroll down the list of package groups to System as shown in Figure 11.

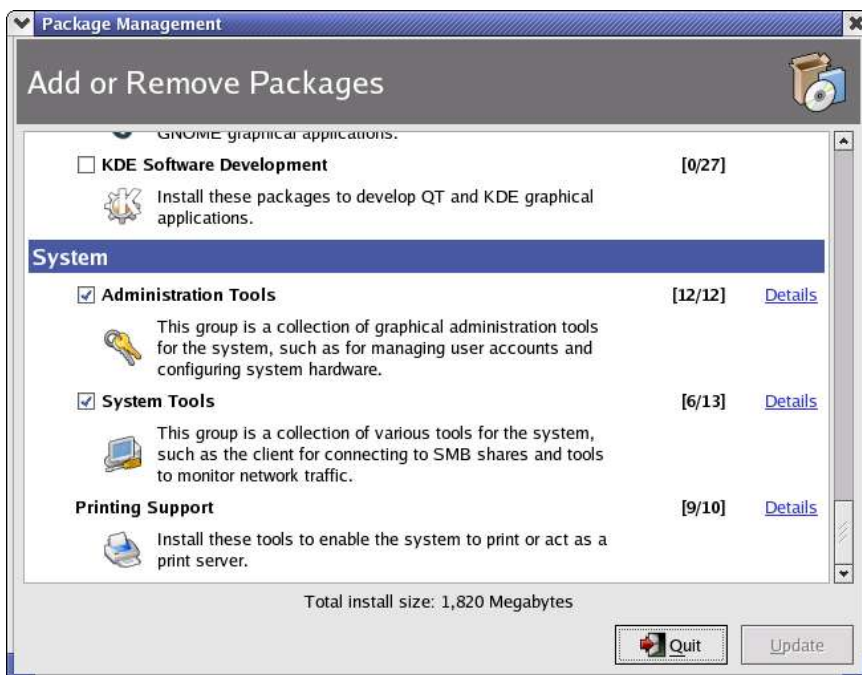


Figure 11. The System package group includes System Tools, which includes the Samba software.

If the checkbox next to System Tools is selected, click on the Details link to display which packages have been selected. If the System Tools checkbox is not selected, the Details link will not be enabled. Thus, select the checkbox, and then click on the Details link. In both cases, the System Tools Package Details dialog will display next, as shown in Figure 12.

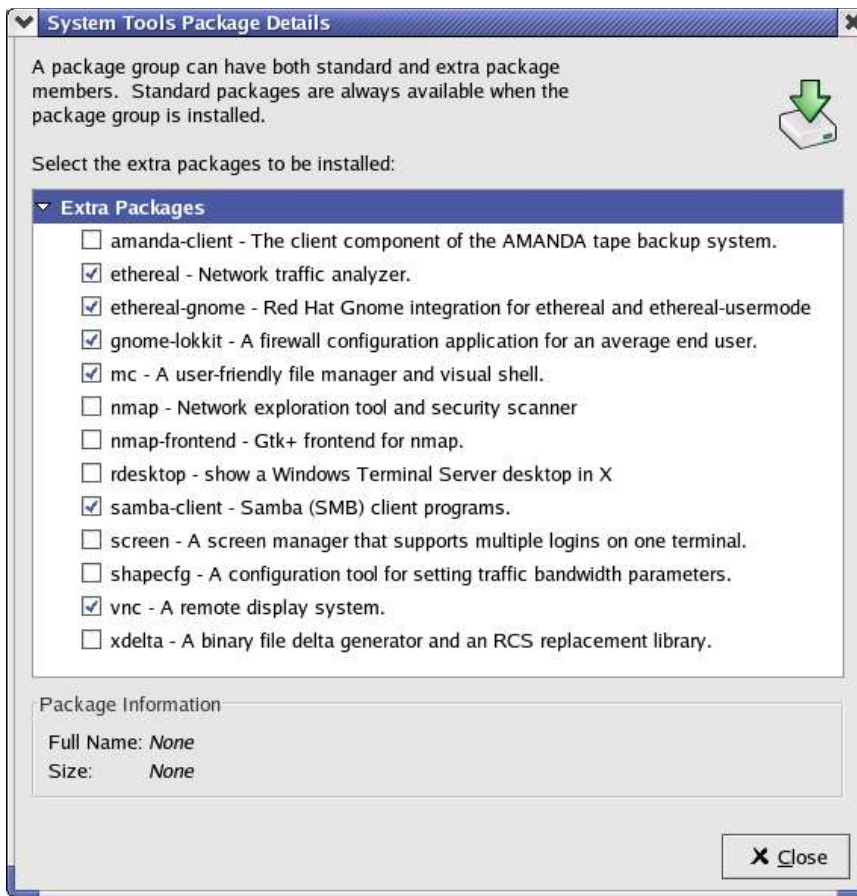


Figure 12. The System Tools Package Details dialog shows which packages have been installed.

The samba-client checkbox should be selected. If it is, click Close and move on to the next section, 2b, Make Sure Samba is Running. If it isn't Samba isn't installed on this Linux workstation yet. Select the checkbox, click Close, and then click the Update button on the Package Management dialog from Figure 11.

In SuSE, select System | YaST to bring up the YaST Control Center, as shown in Figure 13.



Figure 13. Use the YaST Control Center to get to the Install and Remove Software applet.

Then select the Install and Remove Software applet to bring forward the software management dialog. Select "Package Groups" in the Filter drop down combo box in the upper left corner to display a tree view of the available and installed software package groups, as shown in Figure 14.

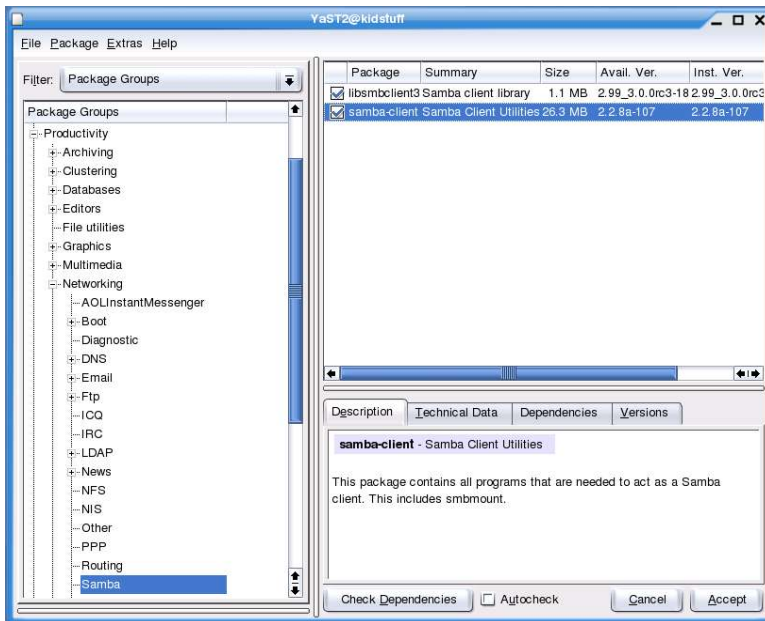


Figure 14. The software management dialog displays a hierarchy of all possible software packages on the system.

The tree view shown in Figure 14 might be expanded, so you might have to scroll down to the Productivity node, or you could just minimize all of the nodes before Productivity. In either case, open the Networking node and scroll down until you see the Samba item. In the list in the upper right, you'll see all of the individual packages that make up the Samba client on this machine; make sure that they are selected. If they aren't, check the check boxes and then click the Accept button in the lower right to install them.

It can be difficult to find which node a particular package belongs to; if you know the name of the packages themselves, you can close all of the nodes and then click on the "zzz All" node at the bottom of the tree view in order to display every package in one big master list, as shown in Figure 15.

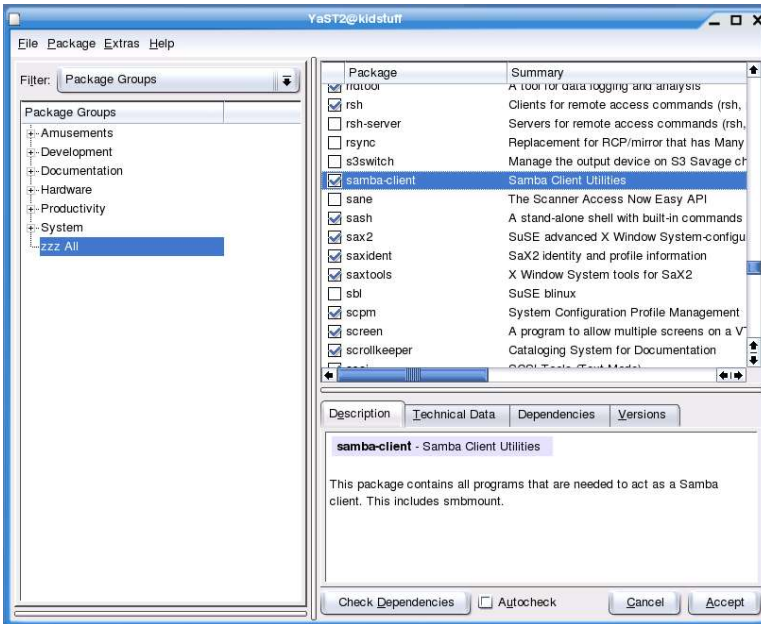


Figure 15. An alternative to drilling through the various Package Group nodes is to select the *zzz All* node.

Now that you know that the Samba client is installed, it's time to make sure it's running.

2b. Make Sure Samba is Running

Samba is a service that runs in the background, but as with every Linux service, it can be stopped. In Red Hat, click on Server Settings | Services, enter the root password, and the Service Configuration dialog will display, as shown in Figure 16.

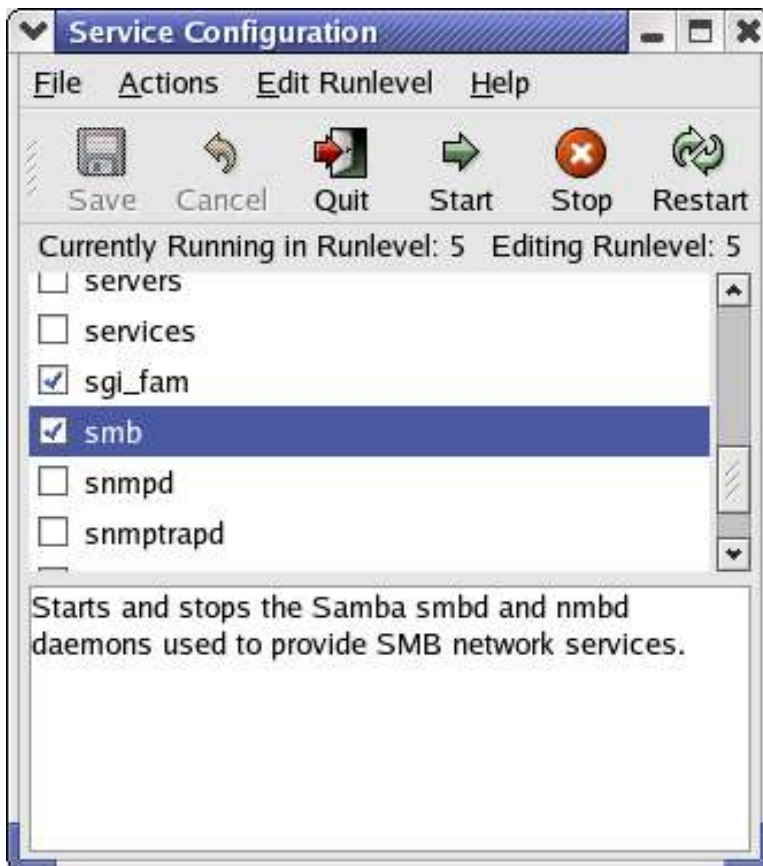


Figure 16. The Services dialog displays which services are running in Red Hat.

In SuSE, select System | YaST to open the YaST control Center, and then click on the System item in the left hand list, as shown in Figure 17.



Figure 17. The YaST Control Center is also used for determining which services are running.

Select the Runlevel Editor applet, enter Administrator Mode, enter the root password, and the Runlevel Editor will display, as shown in Figure 18. Similarly to the Red Hat tool, you can start and stop services in this tool.

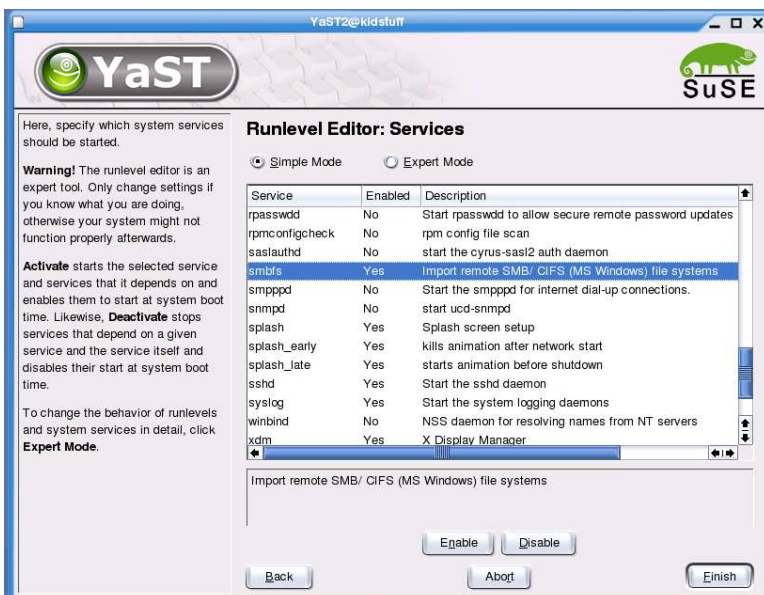


Figure 18. The Runlevel Editor allows you to start and stop services in SuSE.

Now that we know that Samba is ready for us, we need to set up a mount point on the Linux workstation.

3. Set up a Mount Point on the Linux Workstation

There are several ways to map a share on another computer to a drive letter on a workstation in Windows. One easy way is to open Windows Explorer, drill down to the share on the other computer (found under the My Network Places node) and right click to open the context menu or select the Tools menu item in the menu bar. Then you select the Map Network Drive menu item to bring forward the Map Network Drive dialog as shown in Figure 19.

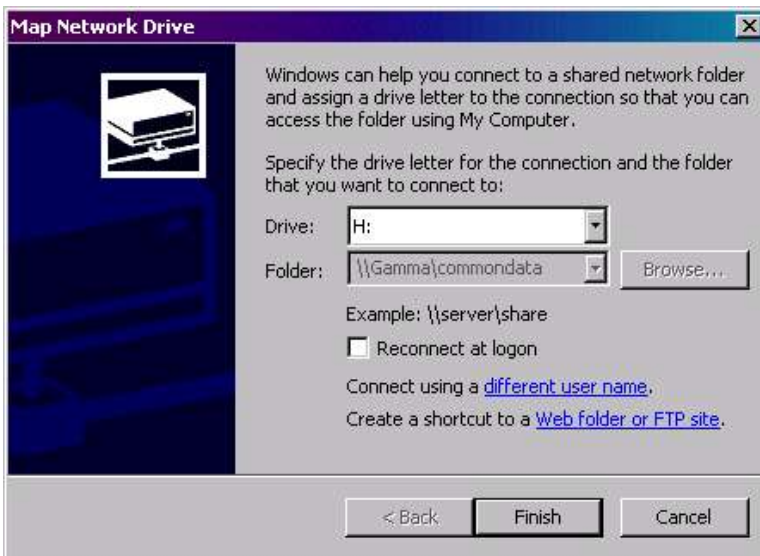


Figure 19. Mapping a drive letter to a share on another Windows computer via the Map Network Drive called from a share's context menu in Windows Explorer.

You'll see that the Folder drop down combo box is filled in with the name of the share you selected, and that the Folder drop down combo box and Browse button are both disabled because a share was already selected. Select a drive letter to map to the share, select Finish, and you're done.

Another way to bring forward the Map Network Drive is to select the Tools menu item in the menu bar. Then select the Map Network Drive menu item, and a similar version of the Map Network Drive will be displayed, as shown in Figure 20.

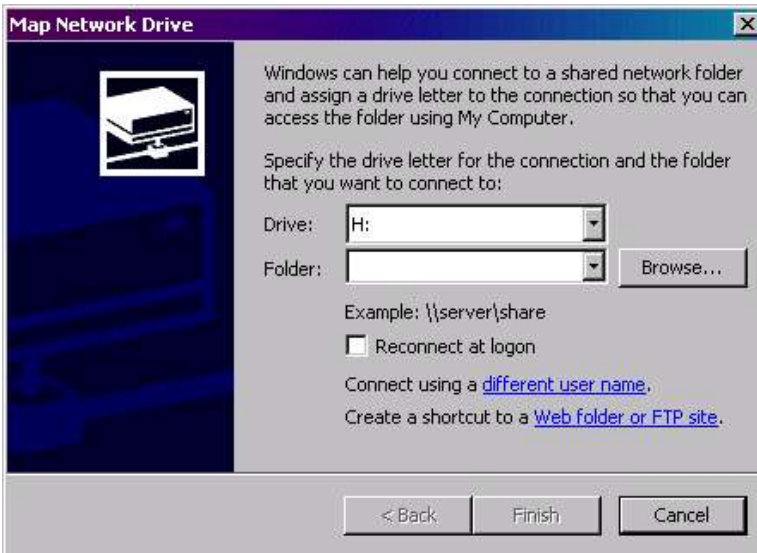


Figure 20. The Map Network Drive dialog called from the Tools menu.

The difference is since you haven't selected a specific share on the other computer, the Folder drop down combo box will be enabled and empty, and the Browse button will be enabled as well. Clicking on the Folder drop down combo box will display a list of available shares (their Windows Share Names) from which you can select your target, or you can navigate through the Folder hierarchy via the Browse button. Then select a Drive letter as before and you're all set. You can access the folders and files on the share on the other computer simply by referring to them via the mapped drive letter, just as you reference files on your local computer via drive letters like "C" and "D".

Since Linux doesn't have drive letters as such, there needs to be a different mechanism to point to the share.

The "mnt" directory is used to provide points of access to devices such as CD-ROMs and floppy drives, and some people create directories under mnt for shares as well. The folder structure in the share will be displayed and available under the directory created under "mnt".

I prefer to create directories under the user's home directory that map to each share they're going to have access to so that the user can stay in their home directory for everything. Since you've gone to the trouble of creating a Windows Share Name that is user-friendly, I think it makes sense to use the same label for the directory. In Figure 21, I've created a directory named "commondata" under /home/whil and mounted the Windows share to it (How I did so is covered in the next two steps.) You can see the directories that are accessible via the /home/while/commondata directory - they're all from the shared C drive of the Windows computer.

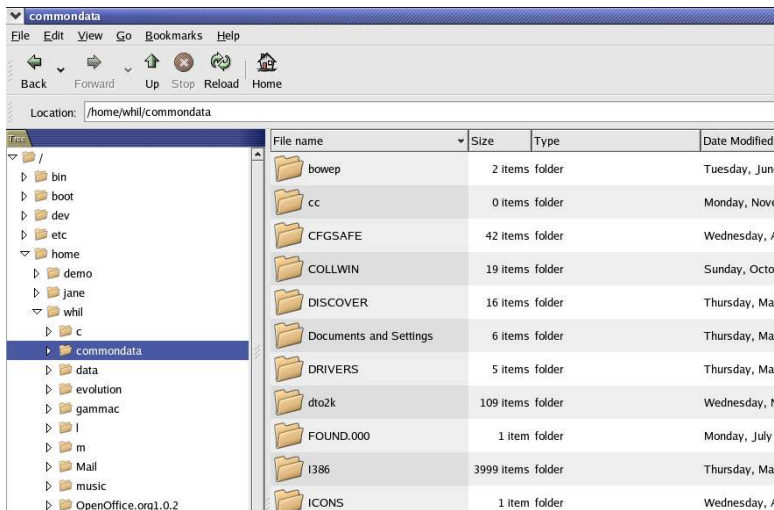


Figure 21. A directory under a user's home directory and named the same as the Windows Share Name serves as a useful mounting point for shares.

In order to create a directory, you can use the File | New Folder menu option in Nautilus, or simply open a terminal window and issue the command

```
mkdir /home/<username>/<directoryname>
```

where <username> and <directoryname> are replaced with the appropriate values. In my example, I would have typed

```
mkdir /home/whil/commondata
```

Your preferences on where to create mount points for shares may vary, of course. Now that the Linux workstation has a mounting point for the Windows computer share, it's time to set up the configuration files on the Linux workstation so that it can find the share.

4. Make Changes to Linux Workstation Config Files

There are three configuration files on a Linux computer that need to be modified. Unlike Windows, many files on Linux computers do not have extensions, and these three are no different. However, they're all simple text files, and thus can be modified with any program that can create and save a plain text file. There are a lot of ways to modify a config file; here's one that I prefer.

First, open up a terminal window and become root by typing the "su" command and then entering the root password when prompted. Next, open a text editor by typing the "gedit" command. Then use the File | Open command or the Open button to open the file to be edited. You can use a different editor if you like, of course.

4a. Change /etc/hosts

The first job is to deal with name resolution (mapping IP addresses to words). You can do this via DNS or WINS if either is installed, but if you're using static IP addresses and you've got a small network, it's pretty easy just to edit the hosts config file. The first config file to modify is the /etc/hosts file. This file contains entries that correlate IP addresses and machine names on the network. (There is a similar file in Windows - in Windows 2000, it's found in c:\winnt\system32\drivers\etc\, for example.) Before it's modified, hosts looks like this:

```
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1    localhost    localhost.localdomain    localhost
```

If you have created a machine name for your Linux workstation, the first "localhost" entry will be replaced by the machine name.

Your job in hosts is to add a line that identifies the IP address and the machine name of the Windows computer that has the share you want your Linux workstation to connect to. For example, if the Windows computer's IP address is 192.168.1.100 and the machine name of that computer is 'tesla' then you'd add a line so that the hosts file looks like this:

```
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1    localhost    localhost.localdomain    localhost
192.168.1.100 tesla
```

You can add as many entries to hosts as you like, if you want your Linux workstation to know who those other computers are. For example, if you were connecting to shares on two different computers, you'd add an entry for each computer.

Now that your Linux workstation knows how to identify other computers on the network, it's time to show it how to identify and access specific shares on those computers.

4b. Change /etc/fstab

The second config file to modify is /etc/fstab. This config file describes the filesystems used by the machine. Each line maps a filesystem with various information about how to access the filesystem. A typical fstab file without any modifications might look like this:

```
LABEL=/          /          ext3    defaults    1 1
LABEL=/boot      /boot      ext3    defaults    1 2
none             /dev/pts   devpts  gid=5,mode=620 0 0
none             /proc      proc    defaults    0 0
none             /dev/shm   tmpfs   defaults    0 0
/dev/hda3        swap       swap    defaults    0 0
/dev/cdrom        /mnt/cdrom iso9660 noauto,owner,kudzu,ro 0 0
/dev/fd0          /mnt/floppy auto     noauto,owner,kudzu 0 0
```

The second line identifies the filesystem used by the /boot directory (ext3) and the mount point (/boot) while the sixth line identifies the filesystem used by /dev/hda3 (swap.) Each share

you access from your Linux workstation will also have its own line that identifies the name of the share, the mount point, the type of filesystem to be used, and some additional information needed for the type of filesystem selected. For example, to mount the `comdata` share, the entry in `fstab` would look like this (this is all one line, but wraps to two lines here):

```
//gamma/comdata /home/whil/comdata smbfs
credentials=/etc/samba/crw,uid=whil
```

The first field is the fully qualified name of the share - the machine name of the Windows computer and the Windows Share Name. The second field is the mount point on your Linux workstation that you created in step 3. The third identifies the filesystem type - in this case, `smbfs` (samba filesystem).

The last field identifies the credentials file that will be used to authenticate against the share. In other words, access to the share is restricted by the permissions set up for that share on the Windows computer - as we did in Step 1b. In order for our Linux workstation to access that share, we have to pass a username and password to the Windows computer that matches the permissions for that share. Thus, we create a credentials file in the `/etc/samba` directory that contains that username and password, and point to the credentials file in `fstab`.

You can name the credentials file anything you want, but you should consider giving them clear names because if you're going to have more than one person using the Linux workstation, they'll each need their own credentials file. Thus, a name like "cred" is all fine and good for the first user, but the second and subsequent users will need different files, and you then end up with difficult to maintain naming conventions like "cred2", "cred3" and "crednew". Better to use something like "credbob" to begin with.

The `uid` part of the credentials field indicates which local user on the Linux workstation this line in `fstab` belongs to.

If you want to be able to access more than one share, you need to create separate lines in `fstab` for each share, just as you created Windows Share Names for each share on the Windows computer in Step 1a, and separate mount point directories for each share in Step 3.

Now let's look at the credentials file itself.

4c. Create `/etc/samba/<credentials file>`

The last config file to deal with is the credentials file. As mentioned in the last section, you'll need a separate credentials file for each user who is going to access shares on another computer. Since your Linux install doesn't know in advance what users you're going to add to the system, nor does it automatically create these credentials files when you modify `fstab`. Thus, you'll have to create them yourself. I suggest you name the credentials file for each user with a mnemonic that includes their username or other unique info. For the purposes of this chapter, I'll assume you name your credentials file along the lines of "credbob", where "bob" is the username of the person whose credential file it is.

A credentials file is a text file with two entries in it, like so:

```
username=bob
password=doo_97wah
```

That's it. Now it's time to access the share.

5. Access the Share

The first thing you'll need to do in order to access the share on the Windows computer from your Linux workstation is to notify your Linux machine of the changes that have been made to the config files. You can do this by rebooting your machine, as Windows users are used to doing, but it's not necessary to go that far. You can simply restart the `netfs` service, and your Linux workstation will now be aware of the shares.

In Red Hat, select the **Server Settings | Services** menu option, enter the root password, and then scroll down to the `netfs` checkbox. Click the **Restart** button at the top of the dialog, and in a few moments you will get a dialog that says the restart was successful.

Now open up Nautilus, a terminal window, or even just an application. Drill down through the folder hierarchy to the mount point you defined in Step 3, and you should see the mount point directory populated with the folders and files in the share, as shown earlier in Figure 21.

Checklist

A detailed step-by-step set of instructions is useful the first few times you go through a process. Once you're basically familiar with the concepts and nuances of each step, and need to perform the same operation a few months later, all the verbiage gets in the way. A checklist to remind you of each of the steps, listing the specific syntax and menu options, but without the explanations, is more useful now. So here's the checklist for Connecting a Linux workstation to a share on a Windows computer.

1. Create an accessible share on the Windows computer

1a. Create a share

Open Windows Explorer, select the drive or directory to share, right-click to open the context menu, select **Sharing**, select **Add New Share**, enter a user-friendly Windows Share Name.

1b. Set the permissions on the new share

Open Windows Explorer, select the drive or directory that was just shared, right-click to open the context menu, select the **Properties** menu, select the **Security** tab, select the **Permissions** button, add and remove users as appropriate according to who should have access to this share.

2. Make sure the Samba client is installed and running via the **Server Settings | Services** menu option in Red Hat or the **YaST | Software | Install** and **YaST | System | Runlevel Editor** applets.

3. Set up a mount point on the Linux workstation

Such as `/home/<username>/<sharename>`

4. Make changes to the Linux workstation config files

4a. `/etc/hosts`

```
192.168.1.100  tesla
```

4b. `/etc/fstab`

```
//gamma/commondata /home/bob/commondata smbfs
credentials=/etc/samba/crbobw,uid=bob
```

4c. /etc/samba/<credentials>

```
username=bob  
password=doo_97wah
```

5. Access the share by restarting the netfs service.

Thanks to many members of the Milwaukee Linux User Group as well as Paul McNett and Ted Roche for feedback on this whitepaper.

For updates to this whitepaper as well as other HOWTO whitepapers, please visit www.hentzenwerke.com.

Copyright 2004 Whil Hentzen. All rights reserved.