

File Sharing in Novell Nterprise Linux Services 1.0

Novell Cool Solutions AppNote

www.novell.com/coolSolutions

FEBRUARY 2004

Sander van Vugt

Sander van Vugt is an MCNI and CLE working as a technical trainer for Azlan Training in the Netherlands. Besides being a Novell expert for many years, he also is a Linux expert and has written several books on this subject; unfortunately, at present, these are only available in Dutch.

Acknowledgement

Many thanks to Richard Millet, Kees Bres, and Alexander Danoyan for reviewing this article.

TABLE OF CONTENTS

Samba and NNLS 1.0	2
Installation	3
Create Samba Users	5
Modify smb.conf for User Access to the Samba Server	9
Access Samba shares	12

Introduction

In a Microsoft environment, traditionally the Server Message Blocks (SMB) protocol was used to share resources between networked computers. This protocol can be implemented over a number of protocols including NetBIOS and TCP/IP. The working of SMB nowadays is upgraded to Common Internet File System (CIFS). This is an SMB implementation that uses TCP/IP connections to share files.

CIFS is in use on all Windows servers and clients where files are shared. Because of the advantages of CIFS, the Samba server implements this protocol on many platforms such as NetWare, Apple OS X and different brands of UNIX such as Linux. Since it has many advantages such as an extreme flexibility and it is an open source server, the Samba server has become the preferred way on many platforms to share files and printers. The major advantage of this server is that it is supported by all Windows clients. Therefore all applications that can be used in a Windows networked environment can use it. With Samba installed on a Linux server, the Linux server can provide the same functionality as a Windows server: with Samba, Linux can provide SMB based (Windows) file and print services to end-users, and there are even claims that the Samba server does this faster than Windows.



SAMBA AND NNLS 1.0

Novell compiled Samba to enable LDAP authentication by default. These options have the purpose to enable user authentication against an LDAP server. The Novell version of the Samba server has the following characteristics:

- eDirectory users can be turned into Linux/Samba users very easily.
- Home directories are created automatically when the Linux/Samba user first logs in to a Linux box.
- Home directories of Samba users are automatically shared so they can also be accessed from a Windows client.
- In order to access files shared by the Samba server, users must authenticate against eDirectory. Secure LDAP is used for this authentication.

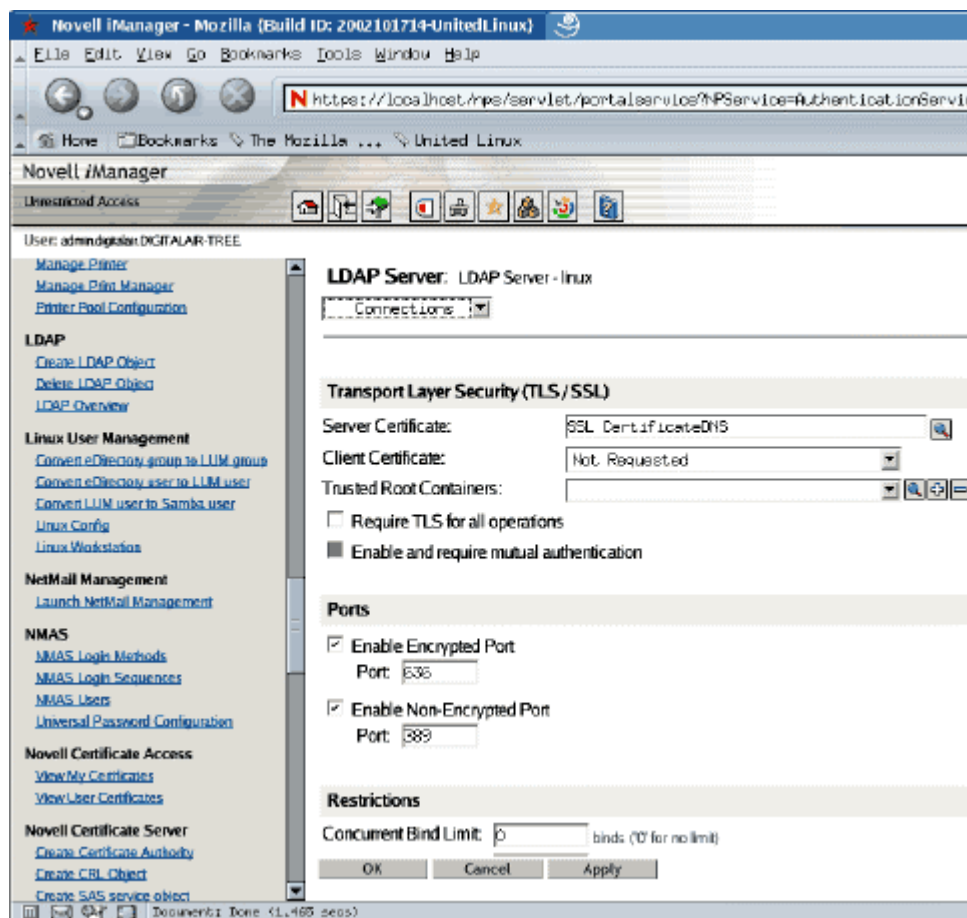
The purpose of the Samba server, is to share files, folders and printers on the network. Files and folders shared by the NNLS Samba server can be accessed in many ways:

- Windows users can navigate to the Samba share using the Windows Explorer.
- Linux users can navigate to Samba shares using an smb-compatible browser such as the Gnome Nautilus.
- Linux users can mount a Samba share locally on their computers.
- Linux users can access files in a Samba share with the client utility smbclient.
- Windows users can access files on the Samba server when they create a web folder.
- All users can access files on the Samba server from a browser if Novell NetStorage is installed and a storage location for the Samba server is created. In this case files can be uploaded and downloaded but can't be modified directly.

INSTALLATION

In most cases, the Samba server is installed automatically when NNLS was installed. If, however, this is not the case in your situation, you can install it afterwards using an NNLS custom install. Make sure eDirectory 8.7.1 or later is installed before you start installing the Samba server, otherwise it will not work. This is because you need eDirectory to store the Samba user accounts.

1. Run the NNLS install script `install.sh` with the command `./install.sh`.
2. Hit Enter to start the installation.
3. Make sure Samba is selected in the overview you see now. If it isn't, you can toggle it by hitting 8. Press f to finish making your selection.
4. Read the license agreement and accept it to continue.
5. Enter the name or IP address for an eDirectory server. On this server, the eDirectory schema must be extended for Samba installation. Make sure this server has a master or read/write replica of eDirectory installed.
6. Enter the Admin name with context. This should be a fully qualified admin name like `cn=admin.o=digitalair`. This information is needed for extension of the eDirectory schema.
7. Enter the admin password.
8. Enter the LDAP server IP-address or DNS hostname. This information is needed because users use LDAP to authenticate against eDirectory before they can access folders shared by the Samba server.
9. Specify the name of a Samba Proxy user with context. This is the name of a user account that has enough rights to search the LDAP tree for Samba users. The default value for this username is the same as the name of your admin account. After specifying the name of the Samba proxy user account, you have to specify the password of this user. For a more secure environment, you could choose to create a specific user account with limited rights for this task.
10. The LDAP ports must be identified. After the secure LDAP port is specified, the exact path and name for the CA certificate file must be provided. If you fail to do this you will still be able to use encrypted connections over port 636, but you will lack client server authentication. If, at this stage, there is no certificate file available on your system, you can extract it from the SSL certificate object in base64 format. Only if you are installing Samba into a new eDirectory tree that you are also creating on this server, this information is optional.



If you fail to refer to a certificate file during installation, you can always extract the certificate from the LDAP server object to specify the certificate file to be used when the installation is finished.

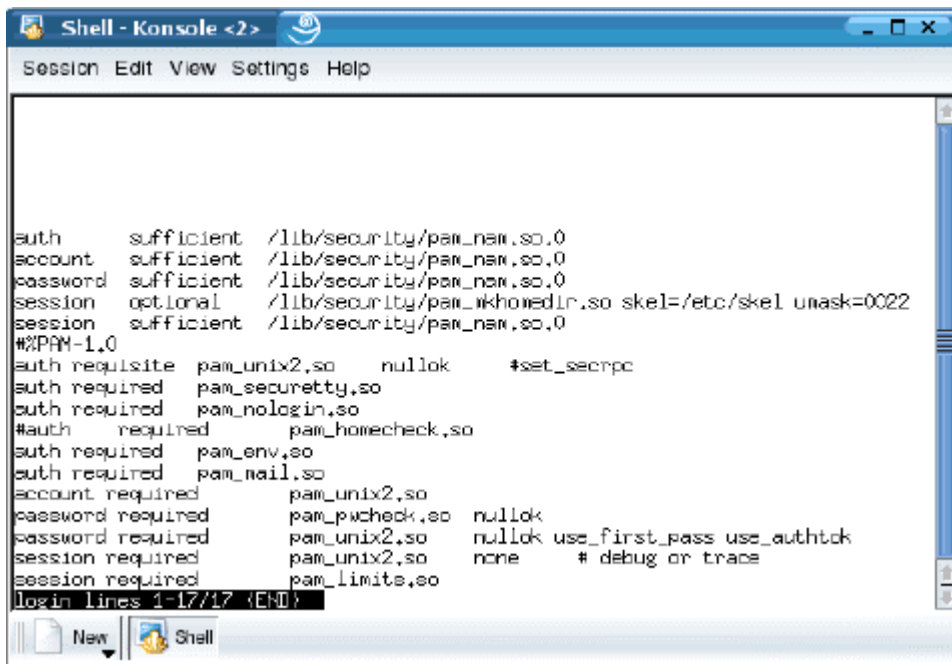
11. Finally, the installer asks if you want to remove existing Samba packages. Always say Yes here, you need the new Novell Samba server to be able to take advantage of specific Novell features. Conflicts might arise if you don't remove the old Samba installation.

CREATE SAMBA USERS

After the Samba server is installed, you have to enable your eDirectory users to allow them to use it. It is possible to create entirely new eDirectory users that are also enabled as Samba users, alternatively it is also possible to convert existing eDirectory users so they become Samba users. In both cases, LUM is the keyword.

LUM stands for Linux User Management. It is the system that creates Linux users with all necessary properties in eDirectory. Normally, Linux users are created in the local files `/etc/passwd` and `/etc/shadow` on the Linux machine. The login program checks these files when a user tries to log in. You can imagine this situation is not ideal in a networked environment with many Linux workstations. eDirectory provides a solution: when you create users as LUM users, all traditional Linux properties of the users are stored in eDirectory. The advantage is clear, you don't have to create your Linux users on all Linux workstations where you want them to have access; you create them just once in eDirectory.

There is another change required on the Linux computer. Normally the login process on your Linux computer just checks the local files when users are providing their username and password. The mechanism behind this is PAM (Pluggable Authentication Modules) that uses a configuration file for each process that has to do something with logging in. When NNLS is installed, you have the choice to modify some of these PAM configuration files in the `/etc/pam.d` directory. The login module is always modified. This enables your users to log in against eDirectory instead of the local files. You can even instruct other programs such as the FTP-client and the SSH-client to log in on eDirectory instead of some local files on your Linux machines.



```

auth      sufficient  /lib/security/pam_nam.so.0
account  sufficient  /lib/security/pam_nam.so.0
password sufficient  /lib/security/pam_nam.so.0
session  optional    /lib/security/pam_mkhomeIn.so skel=/etc/skel unask=0022
session  sufficient  /lib/security/pam_nam.so.0
#%PAM-1.0
auth requisite pam_unix2.so nullok    #set_secure
auth required pam_securetty.so
auth required pam_nologin.so
#auth required pam_homecheck.so
auth required pam_env.so
auth required pam_nail.so
account required pam_unix2.so
password required pam_pwcheck.so nullok
password required pam_unix2.so nullok use_first_pass use_authok
session required pam_unix2.so none    # debug or trace
session required pam_limits.so
login lines 1-17/17 (END)

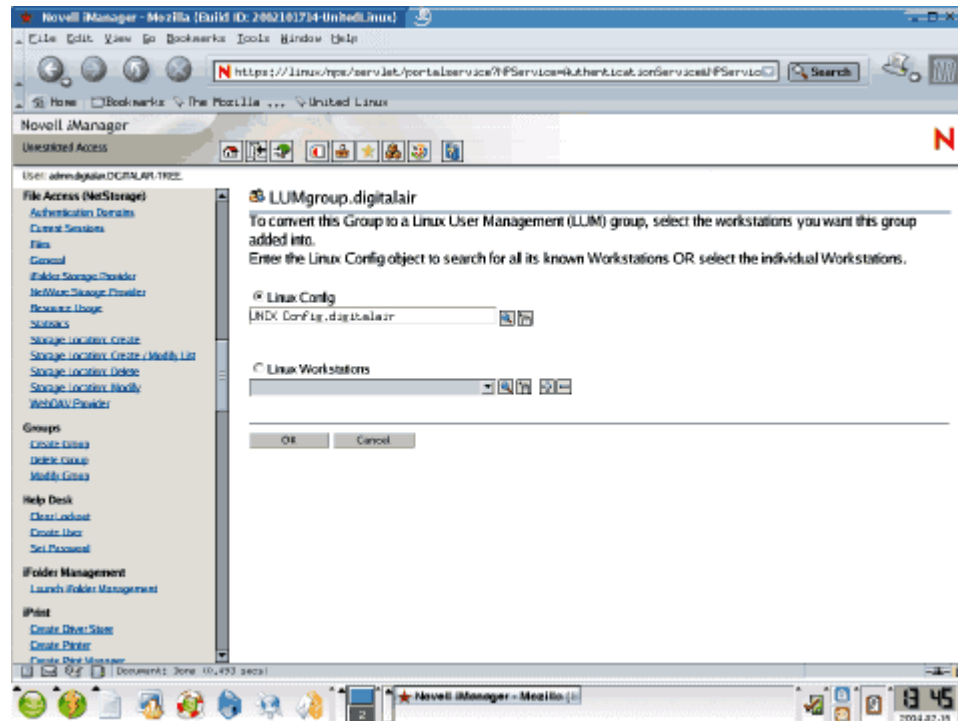
```

In order to log in against eDirectory, your PAM login file is modified.

The LUM account is necessary for the users to be able to access the files and folders shared by the Samba server. This is because in order to access these files, users need some local permissions on the Samba server. They can only have these local permissions if they have a user account that is locally valid. We will see in more detail how this works later.

Before you can create an eDirectory user which is also a Samba and a LUM user, you must create a LUM group object. This is necessary, because Linux users must have a primary group, a Linux user that doesn't have such a group can't log in to the system. If you are using NNLS, the LUM group can be used as primary group for your Linux users.

1. Open a browser and enter the URL <https://yourservername/nps/iManager>. This will start the iManager login screen. Note that some browsers are not supported by iManager. We recommend you use Netscape 7, Mozilla 1.4 or higher or Internet Explorer 5.5 or later to access iManager.
2. Select **Groups, Create Group** and enter a name for the group and the context where you want to create it. Click OK twice when finished.
3. Next you can specify the Linux workstations you want to add the LUM group to. On all workstations you select, the group will be available. If you have several workstations in your group, you can refer to the Linux Config object as well to refer to all Linux workstations in your tree. Use the magnifying glass to browse for this object, you will find the Linux Config object as **UNIX Config**.

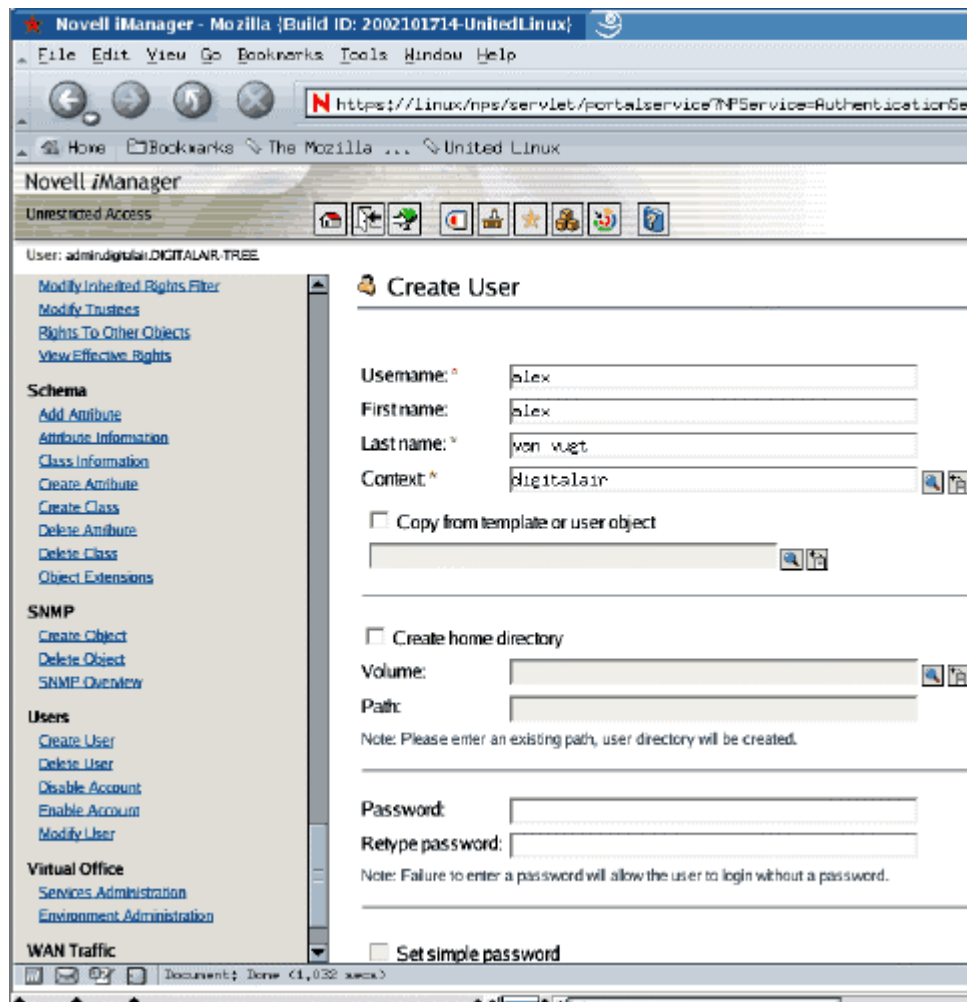


To add a group as a LUM group, you have to specify on which Linux workstations this group should be available.

4. Click OK twice when finished.

Now that the LUM group is created, you are ready to start creating LUM users and specify these users must also be created as Samba users. In order to create Samba users, you can use iManager:

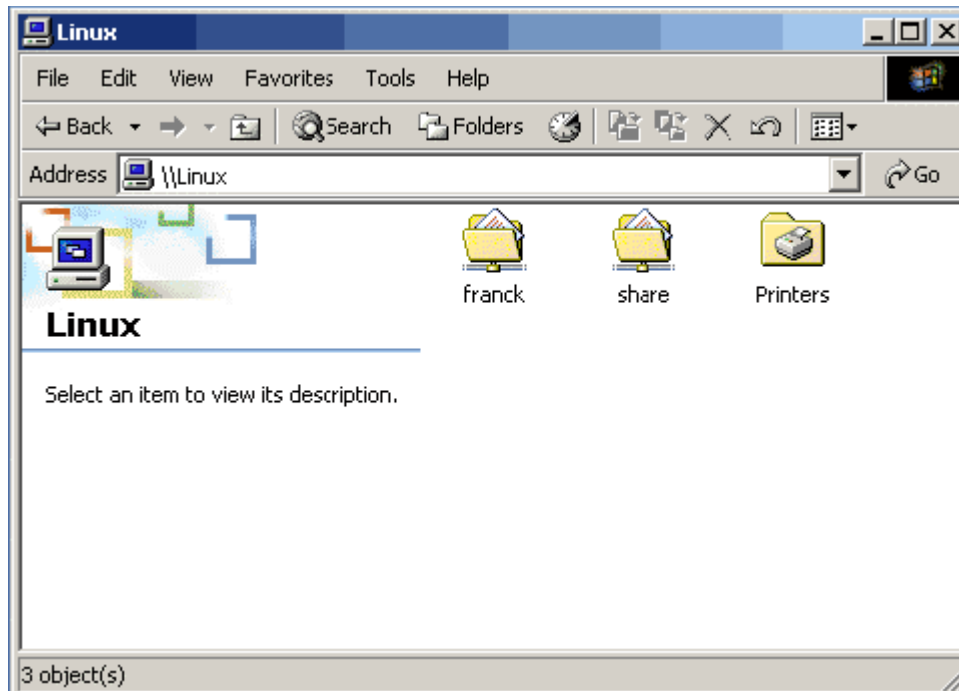
1. Select **Users, Create User**. Now enter all required properties for your eDirectory user. Note that you do not need to enter a home directory here, since from this interface only home directories on Netware volumes are supported. Click OK at the bottom of the page after specifying all required properties. Click OK once more.



The easiest way to create a Samba user, is from the Create User interface.

2. On Linux every user needs a Primary Group. Without a primary group, the user will not be able to log in to their Linux workstation. You can specify the group you just created as the primary group for your user now.
3. Select the option **Also convert this User object to a Samba User object** and enter the desired Samba password next. Click OK when finished.

You now created a LUM user who is also a Samba user. This means everything necessary to authenticate on the Linux workstations in your network and on your Samba server as well, is now present in eDirectory. Before you continue, you should log in once as the user you just created. Do this on the server where your Samba server is running. This will create a Linux home directory for the user. This home directory will be automatically shared by your Samba server.



If existent, Samba automatically shares all home directories it finds for Samba users.

1. From your NNLS server, use Ctrl-Alt-F2 to open a virtual console.
2. On the login prompt, enter the name and password of the LUM user you just created. You will see that after successful login, a home directory will automatically be created for this user. This home directory is shared automatically by the Samba server, so your user will be ready to access their home directory from a Samba share right away!

```

Welcome to SuSE SLES 8 (powered by UnitedLinux 1.0) (i586)
Kernel 2.4.19-64GB-SMP (tty2).

linux login: franck
Password:
1 failure since last login. Last was 13:53:46 on tty2.
Creating directory '/home/franck'.
Creating directory '/home/franck/Documents'.
Creating directory '/home/franck/public_html'.
Creating directory '/home/franck/.xemacs'.
franck@linux:~> _

```

After logging in on your NNLS server, a home directory will be created automatically for your LUM user.

MODIFY SMB.CONF FOR USER ACCESS TO THE SAMBA SERVER

As mentioned before, all access to your Samba server is defined in the configuration file `smb.conf`. You will find this file in `/etc/opt/novell/samba/smb.conf`. You will find this file is well documented, there are working examples in the file, all you have to do to activate them, is remove the remark sign (`#`) at the start of the line. In most cases that will work right after you have saved your changes and restarted the Samba server.

The first lines in `smb.conf` that are of particular interest, are the entries you will find on the NNLS version of the Samba server. Here you will find most options you entered during the installation of the NNLS Samba server:

```
ldap admin dn = cn=admin,o=digitalair
ldap ssl = on
ldap port = 636
ldap server = Linux.local
```

You see that in these lines the access to your LDAP server is defined. If ever you need to change any of these values (which you will probably never have to do), you can find them in this file in the section `[global]`.

Later in the configuration file, you will find some examples of how to create a shared directory. This, however, is not as easy as just removing the pound signs at the beginning of the lines. For a user to get access to a shared directory, he has to have some rights to this directory. You can compare this to a scenario in a Windows environment where a user has sufficient access to a share, but no NTFS permissions to the shared files: that simply doesn't work.

Before we dive into this problem, there are some things you should know about permissions on a Linux system.

- There are just three basic permissions: Read Write and Execute (`rwX`). The execute permission makes a files executable, so that leaves just two "real" permissions as compared to the eight rights you can assign when working on a Netware volume.
- To determine the permissions of a certain user, the file system checks if this user is the owner of the specified file and next if the user is a member of the group that is owner of the file. If neither is true, the user automatically gets the permissions assigned to "others".
- The user `root` can make a user owner of a file or directory when he uses the command `chown`, `chgrp` is used to make a group owner of a file or directory.
- A user on a Linux system is member of one primary group. He can be member of more groups, but this system is that unsophisticated that normally users are member of just one group.
- You can see the permissions assigned to files and directories, as well as the user and group that is owner of these files and directories with the command `ls -l`.
- There are some additional permissions available: SGID, SUID and Sticky bit, they are not relevant if you only want to give a user basic access to some files and folders.

```

Shell - Konsole <2>
Session Edit View Settings Help

linux:/ # ls -l
total 119
drwxr-xr-x 21 root root 512 Feb 15 14:15 .
drwxr-xr-x 21 root root 512 Feb 15 14:15 ..
drwx----- 2 root root 48 Feb 13 07:46 .x11-unix
drwxr-xr-x 2 root root 2416 Feb 13 03:05 bin
drwxr-xr-x 3 root root 472 Feb 13 03:15 boot
drwxr-xr-x 29 root root 94752 Feb 15 03:50 dev
drwxr-xr-x 73 root root 7632 Feb 15 13:15 etc
drwxr-xr-x 4 root root 96 Feb 15 13:56 home
drwxr-xr-x 7 root root 2784 Feb 13 07:34 lib
drwxr-xr-x 5 root root 128 Feb 13 02:50 media
drwxr-xr-x 3 root root 72 Feb 13 03:33 mnt
drwxr-xr-x 10 root root 240 Feb 13 07:01 opt
dr-xr-xr-x 523 root root 0 Feb 15 03:49 proc
drwx----- 17 root root 848 Feb 15 14:15 root
drwxr-xr-x 3 root root 7512 Feb 15 13:15/sbin
drwxr-xr-x 2 franck LUMgroup 48 Feb 15 14:15 share
drwxr-xr-x 4 root root 96 Feb 13 02:50 srv
drwxr-xr-x 15 root root 704 Feb 15 14:21 tmp
drwxr-xr-x 15 root root 424 Oct 22 05:12 usr
drwxr-xr-x 25 root root 696 Feb 13 07:31 var
-rw-r--r-- 1 root root 132 Feb 15 03:51 versions.lns
linux:/ #

```

In this picture you see almost all directories are owned by the user root and the group root, with the exception of the directory /share.

So the bottom line is: if you want a user to have access to some directory by means of a Samba share, you have to grant him permissions on the underlying Linux file system as well. In the next procedure you can read how a new directory with the name "documents" is created and how user franck and group LUMgroup are made owner of this directory. We assume this user and group are already created as LUM user and LUM group on your NNLS server. After the objects are assigned as owners of the directory, both the user and the group get all available rights to this directory.

1. In order to make the directory, from a console window, use the commando `mkdir /documents`.
2. Use `chown franck.LUMgroup /documents` to make user franck and the group LUMgroup owners of this directory.
3. Use `chmod 770` to give the read, write and execute permissions to user franck as well as the group LUMgroup. Now locally on the Linux filesystem, user franck as well as any user who is a member of LUMgroup, has access to the directory /share.

Now that you have arranged the local access for your users, you can define the Samba share. We will keep it easy: if you add the following lines to `/etc/opt/novell/samba/smb.conf`, you will make the share /documents accessible and writable for everyone who is member of LUMgroup:

```

[documents]

comment = User documents

path = /documents

public = yes

writable = yes

printable = no

write list = @LUMgroup

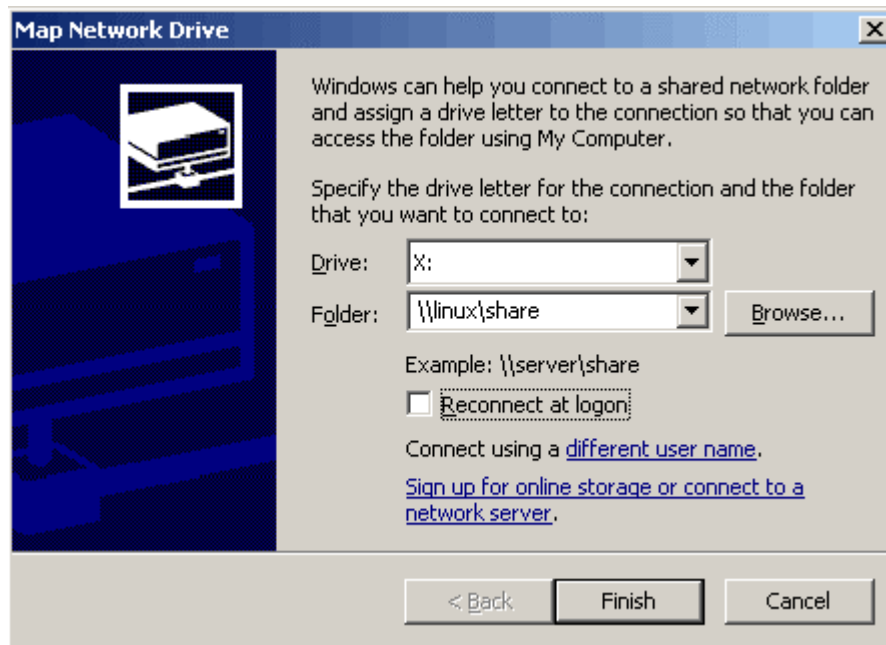
```

The last thing you need to do, is to restart the Samba server. Once this is done, your Samba share will be accessible for anyone who is member of LUMgroup. You can restart your Samba server with the command `/etc/init.d/novell-smb restart`. This command however starts the Samba server just once. If you want it to start automatically each time your server boots, use the `chkconfig` command to add it to your default runlevel. On SUSE Linux, you can use the command `chkconfig novell-smb 235` to add the services to be started automatically. On other distributions, check the man-pages help file for exact syntax.

ACCESS SAMBA SHARES

Now that you have successfully created a Samba share, you can access it both from Windows and Linux computers. The easiest way is to connect from a Windows computer, we'll discuss just one of the ways this works:

1. Open Windows Explorer and select **Tools, Map Network Drive**.
2. Enter a valid username and password if asked for. You will see the mapped network drive opened in a new Window.

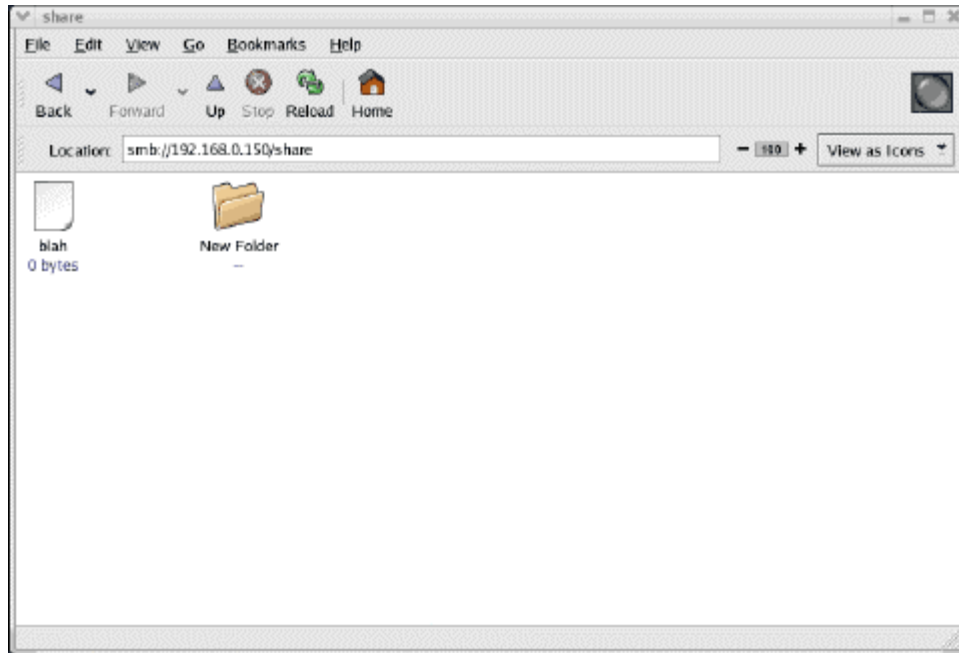


You can map network drives from Windows to the Samba server like you map network drives to Windows servers.

Besides mapping drives from Windows, you can also access shared Samba drives from a Linux computer. One way to access a shared Samba drive, is with the use of the mount command. Since only root can make a mount, you should be root to do this. This means the usage is rather limited, but it is an easy way to test if a shared Samba drive is accessible at all. In order to make the mount, you must have some available mount point on your local Linux computer. If nothing else is already mounted on it, you can use the directory /mnt for this purpose. If, for example, you want to access the share **documents** on the server named **nns1**, you can use the mount command like:

```
mount -t smbfs -o username=franck //nns1/documents /mnt.
```

After you provided a valid password for this share, you can access all files in it from the directory /mnt on which it is mounted.



Some browsers such as Red Hat's Nautilus support URL based access to Samba shares.

Summary

In this AppNote you read about the NNLS Samba server. We provided a short overview of how you can configure your own Samba server to shares files and directories on an NNLS Samba server. The basic point is that on one side, you have to create a LUM user and LUM groups. These are needed to provide permissions on the Linux file system. To access a home directory from the Samba server, nothing else has to be done. If however you want to give access to other shared directories on the Samba server, you need to provide the necessary permissions on the Linux file system before the share can be used.