

Using Windows Shares

Share With Me!

Linux computers are flexible – apart from running Samba to serve files up to Windows computers, they can also provide convenient access to genuine Windows shares. We explore using the Samba client to make the most of any Windows shares rather than needing a full Samba server.

BY NICO LUMMA

When Windows users set up shares to provide other users access to their data, Linux users do not need to resort to rebooting a dual-boot system to gain access.

All they need to know is that Samba (<http://www.samba.org/>) is not only a software package that provides a file server for Windows after installation, but also comprises programs that allow Linux to access Windows (or Samba) shares.

Now it is slightly over the top to install a Samba server on your desktop just to access the data on a Windows computer, and for this reason most distributions divide the Samba goodies up into a number of packets.

The task we have in hand merely necessitates installing the Samba client package (and paying attention to any dependencies); you do not need the Samba server.

Windows Files in the Command Line

Admittedly, the Samba project initially provides command line only access via the *smbclient* program; if you prefer a GUI approach and rely on your mouse, you will need to call on something different.

The *smbclient* is not as bad as it sounds; you can think of the program as a kind of FTP client for Windows servers. The following command:

```
smbclient -L computername
```

sensitive) turns out to be a Samba server with a number of shares, such as *nico*, which contains the home directory for the user *nico*. *nico* can use the following syntax to access the share:

```
smbclient //TURNIP/nico -U nico
```

This syntax reminds one of Windows, although it uses normal slashes instead of the Windows backslash (“\”) to specify the server and the share.

The *smbclient* uses the *smb: \>* prompt to show that it has logged on to the Windows server and can now upload or download files, in other words, it can access the share. There are some similarities to FTP here, reflected in commands such as *dir* (directory listing), *put* and *get* (upload or download files), *mkdir* (make directory) and *cd* (change directory).

The commands in Listing 2 create a new directory, *Test*, on the Windows share, change to that directory, and copy a file called *hello.pl* from a local working directory on the Linux machine to the directory.

The final command downloads the same file from the Windows share across the network to the local machine – of course this would not make much sense in a production environment, unless the file had been modified in the meantime (by the user of the machine providing



allows you to locate Windows shares on a specific computer. Just ignore the password prompts for the time being and keep hitting the [Enter] key. If this does not work, you will need to authenticate yourself to the target machine by providing a user name and a password:

```
smbclient -L computername  
-U username
```

In the example in Listing 1 the target machine *turnip* (the name is not case

the share, for example). To log off, simply type the `smbclient quit` command.

Backup Solution

The `smbclient` can do a lot more. Listing 3 shows the user `nico` using `smbclient` to log on to a file server, where he creates a tarball with the `Test` directory (`smbclient` option `-T` – the `-c` option comes from “create”). The interesting thing is that the archive file, `test.tar`, is created on the local machine and not on the Windows share.

Returning to the Linux shell let’s just ensure that the tarball really has been created: `ls` shows that `test.tar` really was stored locally; and typing `tar -tfv` to list the contents shows all the data available in the `Test` directory on the Windows share. So, if you have been looking for a quick and convenient way to back up Windows files on your Linux machine, this is it. Of course, you can just as easily expand the data in a local tarball on a remote Windows share. To do so, you need `smbclient`-Option `-x` for “extract” in addition to `-T`:

```
smbclient //turnip/nico -U nico
-Tx test.tar Test
```

Message in a Bottle

Incidentally, you can misuse `smbclient` as a primitive communications system. If you send a message to a Windows computer, as shown in Listing 4, the Windows machine pops up a window displaying the message contents. The `-M` is followed by the Windows target; if the target allows you to deliver a message, you simply type the text and then press [Ctrl-D] to send it.

You should not expect to reach a Samba server admin using this technique. If you require a similar functionality in the Linux world, you might like to take a look at LinPopup (http://www.littleigloo.org/downloads_002.php3).

Nico Lumma is the Head of IT at Orangemedia.de GmbH and looks back on years of experience with the practical application of Linux in enterprise environments.



Listing 1: What shares are available on *turnip*?

```
linux:~ # smbclient -L TURNIP -U nico
added interface ip=192.168.1.247 bcast=192.168.1.255 nmask=255.255.255.0
Got a positive name query response from 192.168.1.2 ( 192.168.1.2 )
Password:
Domain=[ORANGEMEDIA] OS=[Unix] Server=[Samba 2.2.7-SuSE]
```

Sharename	Type	Comment
print\$	Disk	Printer Drivers
docs	Disk	Documentation
IPC\$	IPC	IPC Service (Samba 2.2.7-SuSE)
ADMIN\$	Disk	IPC Service (Samba 2.2.7-SuSE)
nico	Disk	Home Directories

Server	Comment
TURNIP	Samba 2.2.7-SuSE

Workgroup	Master
ORANGEMEDIA	TURNIP

Listing 2: Data manipulation on the share

```
smb: \> mkdir Test
smb: \> cd Test
smb: \Test\> put hello.pl
putting file hello.pl as \Test\hello.pl (3.3 kb/s) (average 3.3 kb/s)
smb: \Test\> dir
.                D            0   Tue Dec 17 08:21:55
2002
..               D            0   Tue Dec 17 08:19:55
2002
  hello.pl       262   Tue Dec 17 08:21:55
2002

          39265 blocks of size 1048576. 15507 blocks available
smb: \Test\> get hello.pl
getting file \Test\hello.pl of size 262 as hello.pl (8.3 kb/s) (average
8.3 kb/s)
smb: \Test\> quit
```

Listing 3: Compressing files on the Windows host locally

```
nico@linux:~> smbclient //turnip/nico -U nico -Tc test.tar Test
added interface ip=192.168.1.247 bcast=192.168.1.255 nmask=255.255.255.0
Got a positive name query response from 192.168.1.2 ( 192.168.1.2 )
Password:
Domain=[ORANGEMEDIA] OS=[Unix] Server=[Samba 2.2.7-SuSE]
          directory \Test\
          262 (262000.0 kb/s) \Test\hello.pl
tar: dumped 2 files and directories
Total bytes written: 512
nico@linux:~> ls -al test.tar
-rw-r--r-- 1 nico users      2560 2002-12-17 08:27 test.tar
nico@linux:~> tar -tfv test.tar
tar: Record size = 5 blocks
drwxr-xr-x 0/0                0 2002-12-17 08:21:55 ./Test/
-rw-r--r-- 0/0                262 2002-12-17 08:21:55 ./Test/hello.pl
```

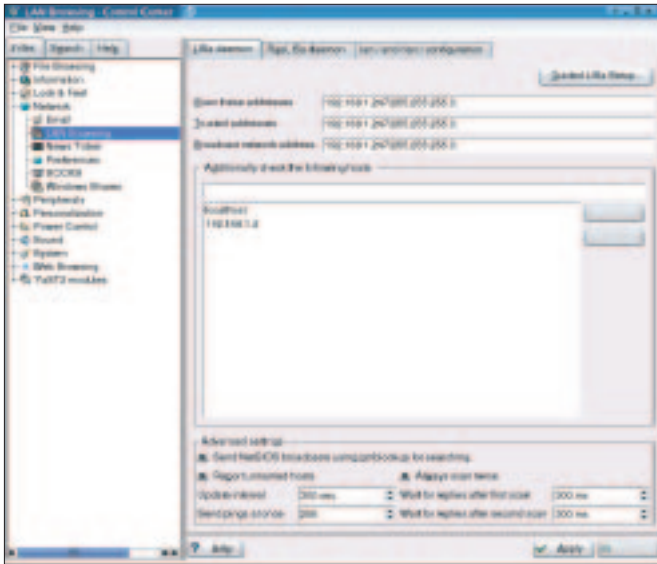


Figure 1: LISA needs to be installed to allow Konqueror to be able to access the SMB shares. Simple setup is in the KDE control center

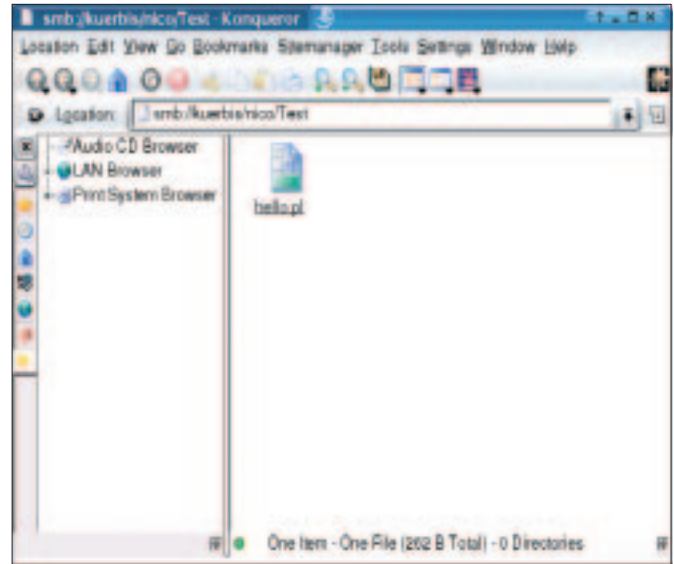


Figure 3: The Konqueror LAN browser is hidden in a tab that appears after clicking on the lower yellow asterisk ("KDE Services")

Konqueror for Share Access

If `smbclient` in the command line is a bit too cumbersome for your taste, you can use everybody's favorite browser and file manager, Konqueror, to access your Windows shares. To do so, you will need to install a few packages, the names of which differ from distribution to distribution.

The current SuSE 8.1 version requires the RPM packages `samba-client` (since `smbclient` will be working in the background in this case), `kdenetwork3-lisa`, `kdenetwork3-lan` and `kdebase3-samba`.

The ominous abbreviation, LISA, refers to the "LAN Information Server" for KDE, a daemon that scans the network for services in the background. `kdebase3-lisa` and `kdenetwork3-lan` provide plug-ins for Konqueror to allow convenient access to Windows shares.

Before you can tell LISA to retire to the background, you will need to perform a variety of configuration steps in the KDE control center (Figure 1). The easiest way to do this, is by clicking on the *Guided LISA setup ...* button of the *LISA Daemon* tab in *Network / LAN Browsing* (Figure 2).

No need to worry when you are prompted for all those numbers! You only need to know your own machine's IP and the netmask for your network, and the wizard will normally supply correct values.

If it comes to the worst, you can always ask your admin – or simply type the `/sbin/ifconfig` command to find out for yourself.

After completing the configuration, LISA needs to be relaunched by `root`; if you do not

have administrative privileges for your Linux machine, you should ask for help before clicking *Apply* in the control center (please note that some KDE installations require `root` privileges to configure LISA).

If everything works out okay, the network neighborhood will appear in your Konqueror browser and allow you to access Windows shares directly. You can use the *Address* line to enter a URL with the following format:

```
smb://computer/share
```

Figure 3 shows an example of this working in Konqueror. ■

Listing 4: Message for a Windows Box

```
01 nico@linux:~> smbclient -M WINBOX
02 added interface ip=192.168.1.247 bcast=192.168.1.255
   nmask=255.255.255.0
03 Got a positive name query response from 192.168.1.23
   ( 192.168.1.23 )
04 Connected. Type your message, ending it with a
   Control-D
05 Can you get me a coffee? :)
06 <I>[Ctrl-D]<I>
07 sent 26 bytes
08 nico@linux:~>
```

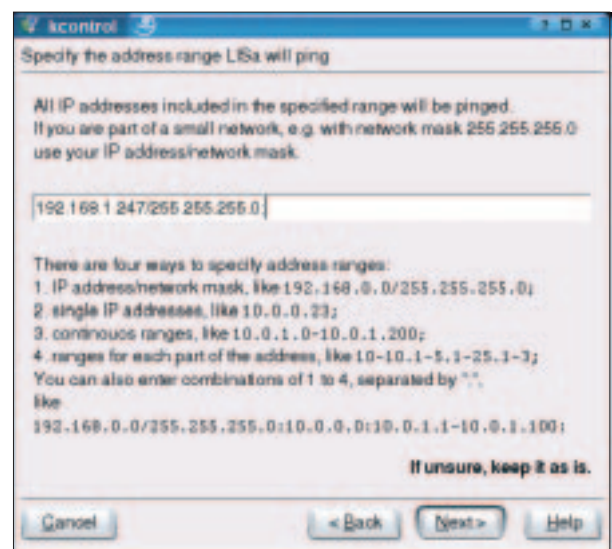


Figure 2: The LISA Configuration Wizard needs just a few simple facts to solve your connection needs