# Samba Installation and Configuration Guide

Samba is an Open Source/Free Software suite that provides seamless file and print services to SMB/CIFS clients. In laymen speak, it allows windows machines to connect to a Linux samba share :)

### 1) Install

Download the source from the following URL: [http://www.samba.org/samba/download](http://www.samba.org/samba/download/) You should get a file that looks similar to this: **samba-3.4.0.tar.gz**.

|  |  |
| --- | --- |
|  | mv samba-3.4.0.tar.gz /tmp  cd /tmp  tar xzf samba-3.4.0.tar.gz  cd samba-3.4.0/source3 |

Setup up appropriate environment variables  
\*For solaris use /usr/sfw/bin/gcc

|  |  |
| --- | --- |
|  | set path = ( /usr/sfw/bin $path )  setenv CC "gcc -Wl,-rpath,/usr/local/samba/lib" |

\*For Linux

|  |  |
| --- | --- |
|  | setenv LD\_RUN\_PATH /usr/local/samba/lib |

Now for the configure

|  |  |
| --- | --- |
|  | ./configure --localstatedir=/var/adm/log/samba --prefix=/usr/local/samba --with-configdir=/usr/local/adm/config/samba --with privatedir=/usr/local/adm/config/samba/private --with-sendfile-support=no  make  mkdir /var/adm/log/samba  mkdir /usr/local/adm/config/samba  make install |

### 2) Post Install

Strip all the binaries to save space

|  |  |
| --- | --- |
|  | find /usr/local/samba/bin/ -type f -perm -100 -exec strip {} ;  find /usr/local/samba/sbin/  -type f -perm -100 -exec strip {} ; |

Create the man pages  
Linux

|  |  |
| --- | --- |
|  | sudo makewhatis -u -w /usr/local/samba/share/man/  cp /var/cache/man/whatis /usr/local/samba/share/man |

Clean up and Create the appropriate links

|  |  |
| --- | --- |
|  | rm -rf /usr/local/samba/swat  rm -rf /usr/local/samba/private  rm /usr/local/samba/lib/\*.msg  rm /usr/local/samba/sbin/swat  ln -s /usr/local/adm/config/samba/smb.conf /usr/local/samba/lib/smb.conf  ln -s /usr/local/adm/config/samba/smbusers /usr/local/samba/lib/smbusers  ln -s /var/adm/log/samba /usr/local/samba/var  ln -s /usr/local/adm/config/samba/private /usr/local/samba/private  chmod a-x /usr/local/samba/include/\*  chown -R -h 0:0 /usr/local/samba  chmod -R go-w /usr/local/samba |

### 3) Start-up Scripts

Now that we have samba compiled and installed we need to make up a startup script. Under **/usr/local/samba/sbin**, we will now have some daemons. Among them, we will have nmbd and smbd: **nmbd** is a server that understands and can reply to NetBIOS over IP name service requests, like those produced by Windows clients. When windows start up, they may wish to locate an SMB/CIFS server. That is, they wish to know what IP number a specified host is using. Nmbd will listen for such requests, and if its own NetBIOS name is specified it will respond with the IP number of the host it is running on. **smbd** is the server daemon that provides filesharing and printing services to Windows clients. (Most of this information can be found in the man pages for nmdb and smbd). Now then, when you have a samba server it is usually a good idea to start both services: nmdb and smbd. Here are examples of start-up scripts from a fedora install with my custom changes encorporated :).

**SMBD**

|  |  |
| --- | --- |
|  | #!/bin/sh  #  # chkconfig: - 91 35  # description: Starts and stops the Samba smbd daemon  # used to provide SMB network services.  #  # pidfile: /var/run/samba/smbd.pid  # config: /usr/local/samba/lib/smb.conf    SMBCONFIG="/usr/local/samba/lib/smb.conf"  SMBDOPTIONS="-D -s/usr/local/samba/lib/smb.conf"    # Source function library.  if [ -f /etc/init.d/functions ] ; then  . /etc/init.d/functions  elif [ -f /etc/rc.d/init.d/functions ] ; then  . /etc/rc.d/init.d/functions  else  exit 1  fi    # Avoid using root's TMPDIR  unset TMPDIR    # Source networking configuration.  . /etc/sysconfig/network    # Check that networking is up.  [ ${NETWORKING} = "no" ] && exit 1    # Check that smb.conf exists.  [ -f $SMBCONFIG ] || exit 6    RETVAL=0    start() {  KIND="SMB"  echo -n $"Starting $KIND services: "  daemon smbd $SMBDOPTIONS  RETVAL=$?  echo  [ $RETVAL -eq 0 ] && touch /var/lock/subsys/smb ||  RETVAL=1  return $RETVAL  }    stop() {  KIND="SMB"  echo -n $"Shutting down $KIND services: "  killproc smbd  RETVAL=$?  echo  [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/smb  return $RETVAL  }    restart() {  stop  start  }    # Check that we can write to it... so non-root users stop here  [ -w $SMBCONFIG ] || exit 4    case "$1" in  start)  start  ;;  stop)  stop  ;;  restart)  restart  ;;  \*)  echo $"Usage: $0 {start|stop|restart}"  exit 2  esac    exit $? |

**NMDB**

#!/bin/sh

#

# chkconfig: - 91 35

# description: Starts and stops the Samba nmbd daemons

# used to provide SMB network services.

#

# pidfile: /var/run/samba/nmbd.pid

# config: SMBCONFIG

SMBCONFIG="/usr/local/samba/lib/smb.conf"

NMBDOPTIONS="-D -l/var/log/samba -s/usr/local/samba/lib/smb.conf"

# Source function library.

if [ -f /etc/init.d/functions ] ; then

. /etc/init.d/functions

elif [ -f /etc/rc.d/init.d/functions ] ; then

. /etc/rc.d/init.d/functions

else

exit 1

fi

# Avoid using root's TMPDIR

unset TMPDIR

# Source networking configuration.

. /etc/sysconfig/network

# Check that networking is up.

[ ${NETWORKING} = "no" ] && exit 1

# Check that smb.conf exists.

[ -f SMBCONFIG ] || exit 6

RETVAL=0

start() {

KIND="NMB"

echo -n $"Starting $KIND services: "

daemon nmbd $NMBDOPTIONS

RETVAL=$?

echo

[ $RETVAL -eq 0 ] && touch /var/lock/subsys/nmb ||

RETVAL=1

return $RETVAL

}

stop() {

KIND="NMB"

echo -n $"Shutting down $KIND services: "

killproc nmbd

RETVAL=$?

echo

[ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/nmb

return $RETVAL

}

restart() {

stop

start

}

# Check that we can write to it... so non-root users stop here

[ -w SMBCONFIG ] || exit 4

case "$1" in

start)

start

;;

stop)

stop

;;

restart)

restart

;;

\*)

echo $"Usage: $0 {start|stop|restart}"

exit 2

esac

exit $?

Setup the startup scripts.

|  |  |
| --- | --- |
|  | cp smb /etc/init.d/smb  cp nmb /etc/init.d/nmb  chkconfig --add smb  chkconfig --add nmb |

### 4) Creating the smb.conf file

Samba is a very power service, it can be used as a regular sharing service, it can be setup as a primary domain controller, and much more. It can also use a variety of authentication methods such as tdbsam (samba’s own database), smbpasswd (just a text file with password hashes) , or even against an active directory. I will provide 3 examples: 1) smb.conf as a standard sharing service sharing everyone’s home directory. 2) smb.conf as a primary domain controller, and 3) smb.conf authenticating against an AD server.

#### Example as a standard sharing service

[global]

workgroup = MYSMBWG

server string = Samba Server Version %v

# logs split per machine

log file = /var/log/samba/log.%m

# max 50KB per log file, then rotate

max log size = 50

# Verbosity level of logging goes from 0-10

# the value below is usually used for debugging

log level = 10

# authentication method

security = user

passdb backend = tdbsam

#============ Share Definitions ===============

[homes]

comment = Home Directories

browseable = no

writable = yes

#### Example as a Primary Domain Controller

[global]

workgroup = MYSMBWG

netbios name = MYMACHINE

server string = %h server (Samba)

# Primary Controller Settings

domain logons = yes

preferred master = yes

wins support = yes

#Authentication method

security = user

passdb backend = smbpasswd

smb passwd file = /usr/local/samba/private/smbpasswd

# Default logon: for roaming profiles, this must be here

logon drive = H:

logon path = \%Nprofile%U

# Useradd scripts: this is custom, so when a new machine

# joins this domain, a machine account is created.

add machine script = /usr/sbin/useradd -d /dev/null -g 902 -s /bin/false -M -r %u

# set the loglevel

log level = 3

#============ Share Definitions ===============

[homes]

comment = Home

valid users = %S

read only = no

browsable = no

[netlogon]

comment = Network Logon Service

path = /home/samba/netlogon

admin users = Administrator

valid users = %U

read only = no

[profile]

comment = User profiles

path = /home/samba/profiles

valid users = %U

create mode = 0600

directory mode = 0700

writable = yes

browsable = no

#### Example with AD authentication

\* To use this setup there are many prerequisites :) One of the them is that another samba daemon (winbindd) must be running. I will write another guide on how to set this up in the future :)

[global]

workgroup = AD

realm = YOUR.AD.SERVER (DOMAIN.INTERNAL)

security = ads

password server = your.ads.server1 your.ads.server2 (domainserver.domain.internal)

encrypt passwords = yes

domain logons = no

domain master = no

winbind separator = +

# Disable idmapping of Windows SIDs to Unix UIDs

idmap config AD:readonly = yes

log level = 2

max log size = 20

log file = /var/log/samba/log.%m

#============ Share Definitions ===============

[homes]

comment = Home directorys from bechtel

guest ok = no

read only = no

force user = %S

[example-share-with-ad-users]

comment = Temp Share

path = /tmp

valid users = AD+<ad\_user1>,AD+<ad\_user2>

write list = AD+<ad\_user1>,AD+<ad\_user2>

read list = AD+<ad\_user1>,AD+<ad\_user2>

#ie write list = AD+elatov

[example-share-with-ad-groups]

comment = Temp Share

path = /data/tmp

writeable=yes

browseable=yes

valid users = @AD+"<ad\_group1>" @AD+"<ad\_group2>"

#ie valid users = @AD+"Finance Department Users"

### 5) Starting the samba service

Once you have the start-up scripts in place and your smb.conf all setup it is now time to start the samba services :)

Check to make sure not errors exist in the smb.conf

|  |  |
| --- | --- |
|  | /usr/local/samba/bin/testparm |

\* If you see any errors, fix them :)

Start the services

|  |  |
| --- | --- |
|  | service smb start  service nmb start |

Check to make sure they are running

|  |  |
| --- | --- |
|  | ps -eaf | grep smbd  ps -eaf | grep nmbd |

If the services are not running or the service fails to start, check under /var/log/samba for the reasons why it’s not running. Also setting the “log level” variable to “10″ will help in debugging startup error. Well enjoy!! :)

## 6. General Configuration (/etc/smb.conf)

Samba configuration on a Linux (or other UNIX machine) is controlled by a single file, /etc/smb.conf. This file determines which system resources you want to share with the outside world and what restrictions you wish to place on them.

Since the following sections will address sharing Linux drives and printers with Windows machines, the smb.conf file shown in this section is as simple as you can get, just for introductory purposes.

Don't worry about the details, yet. Later sections will introduce the major concepts.

Each section of the file starts with a section header such as [global], [homes], [printers], etc.

The [global] section defines a few variables that Samba will use to define sharing for all resources.

The [homes] section allows a remote users to access their (and only their) home directory on the local (Linux) machine). That is, users trying to connect to this share from Windows machines, will be connected to their personal home directories. Note that to do this, they must have an account on the Linux box.

The sample smb.conf file below allows remote users to get to their home directories on the local machine and to write to a temporary directory. For a Windows user to see these shares, the Linux box has to be on the local network. Then the user simply connects a network drive from the Windows File Manager or Windows Explorer.

Note that in the following sections, additional entries for this file will be given to allow more resources to be shared.

; /etc/smb.conf

;

; Make sure and restart the server after making changes to this file, ex:

; /etc/rc.d/init.d/smb stop

; /etc/rc.d/init.d/smb start

[global]

; Uncomment this if you want a guest account

; guest account = nobody

log file = /var/log/samba-log.%m

lock directory = /var/lock/samba

share modes = yes

[homes]

comment = Home Directories

browseable = no

read only = no

create mode = 0750

[tmp]

comment = Temporary file space

path = /tmp

read only = no

public = yes

Having written a new smb.conf, it is useful to test it to verify its correctness. You can test the correctness of a smb.conf file , using the 'testparm' utility (man page: testparm); if testparm reports no problems, smbd will correctly load the configuration file.

Here's a good trick: If your Samba server has more than one ethernet interface, the smbd may bind to the wrong one. If so, you can force it to bind to the intended one by adding a line that looks like this to the [global] section of /etc/smb.conf:

interfaces = 192.168.1.1/24

where you replace the IP address above with the one that is assigned to the correct ethernet interface. The "24" is correct for a Class C network, but may have to be recalculated if you have subnetted the network. The number relates to the netmask. Numbers for other classes of networks are given in the IP-Masquerade mini-HOWTO.

There is now a GUI configuration tool for Samba: GtkSamba. See <http://www.open-systems.com/gtksamba.html>.

## 7. Sharing A Linux Drive With Windows Machines

As shown in the simple smb.conf above, sharing Linux drives with Windows users is easy. However, like everything else with Samba, you can control things to a large degree. Here are some examples:

To share a directory with the public, create a clone of the [tmp] section above by adding something like this to smb.conf:

[public]

comment = Public Stuff

path = /home/public

public = yes

writable = yes

printable = no

To make the above directory readable by the public, but only writable by people in group staff, modify the entry like this:

[public]

comment = Public Stuff

path = /home/public

public = yes

writable = yes

printable = no

write list = @staff

It used to be that easy; you would now be able to start Samba and browse the shares from a Windows PC. However, Microsoft has recently made life slightly more difficult for those using Samba. Windows 98, Windows NT (service pack 3 or higher) and later builds of Windows 95 now use encrypted passwords by default. Samba uses unencrypted passwords by default. You can't browse servers when either the client or server is using encrypted passwords, because a connection cannot be made anonymously.

You can tell if you have a password type mismatch between client and server if when you try to connect to a share you see a dialog box which reads something like "You are not authorized to access that account from this machine".

You can either configure your Samba server to use encrypted passwords, or configure the Windows machines to use unencrypted passwords.

To get Windows to work with encrypted SMB passwords:

Windows 95/98 =============

Using the registry editor (regedit), create the registry setting HKEY\_LOCAL\_MACHINE\System\CurrentControlSet\Services\VxD\VNETSUP Add a new DWORD value: Value Name: EnablePlainTextPassword Data: 0x01.

Windows NT ==========

Using the registry editor (regedit), create the registry setting HKEY\_LOCAL\_MACHINE\System\CurrentControlSet\Services\Rdr\Parameters Add a new DWORD value: Value Name: EnablePlainTextPassword Data: 0x01

Windows 2000 ============

Using the registry editor (regedit), create the registry setting HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\LanmanWorkStation\Parameters Add a new DWORD value: Value Name: EnablePlainTextPassword Data: 0x01

Once these registry changes have been made, reboot the Windows machine and try to map a network drive on the Samba server again. It should work as long as the Samba server is using plain text passwords.

To configure Samba to use encrypted passwords:

In the [global] section of /etc/smb.conf, add the following lines:

encrypt passwords = yes

smb passwd file = /etc/smbpasswd

You are highly encouraged to read the files ENCRYPTION.txt, Win95.txt and WinNT.txt in the Samba documentation before doing this!

If your clients and server are using encrypted passwords, you will not be able to browse the available shares on the server until an initial connection has been made with the appropriate authentication. To get the initial connection, enter the share name manually in the Windows File Manager or Explorer dialog box, in the form '\\<hostname>\<sharename>'. Log onto the server with a username and password that is valid on the server!

If you suspect that your NetBIOS name service is not correctly configured (perhaps because you get 'host not found' errors when trying to connect), try using just the IP address of the server: '\\<host ip address>\<sharename>'.

In order to get filenames to appear correctly, you may also need to set some options in the appropriate share section. These work for Windows 95/98/NT clients, but may need to be modified if you have Windows 3.X clients:

; Mangle case = yes seems to give the correct filenames

; for Win95/98/NT.

mangle case = yes

; If samba is case sensitive when looking for files

case sensitive = no

; Default case of files that are created

default case = lower

; Preserve case for all filenames

preserve case = yes

; Preserve case for dos (8.3) filenames

short preserve case = no

For other tricks to play with drive shares, see the Samba documentation or man pages.

interfaces = 192.168.1.1/24

Note: The bit after the / is a reference to the subnet mask. "24" is the value to use for an unsegmented Class C network. For more information on subnet calculations, you might want to see <http://www.ralphb.net/IPSubnet/>.

There is a lot more to Samba configuration than this, but this will get you started. If you want to do something more advanced, I refer you to the Samba Web site mentioned above.

## 8. Accessing an SMB Share With Linux Machines

Linux (UNIX) machines can also browse and mount SMB shares. Note that this can be done whether the server is a Windows machine or a Samba server!

An SMB client program for UNIX machines is included with the Samba distribution. It provides an ftp-like interface on the command line. You can use this utility to transfer files between a Windows 'server' and a Linux client.

Most Linux distributions also now include the useful smbfs package, which allows one to mount and umount SMB shares. More on smbfs below.

To see which shares are available on a given host, run:

/usr/bin/smbclient -L host

where 'host' is the name of the machine that you wish to view. this will return a list of 'service' names - that is, names of drives or printers that it can share with you. Unless the SMB server has no security configured, it will ask you for a password. Get it the password for the 'guest' account or for your personal account on that machine.

For example:

smbclient -L zimmerman

The output of this command should look something like this:

Server time is Sat Aug 10 15:58:27 1996

Timezone is UTC+10.0

Password:

Domain=[WORKGROUP] OS=[Windows NT 3.51] Server=[NT LAN Manager 3.51]

Server=[ZIMMERMAN] User=[] Workgroup=[WORKGROUP] Domain=[]

Sharename Type Comment

--------- ---- -------

ADMIN$ Disk Remote Admin

public Disk Public

C$ Disk Default share

IPC$ IPC Remote IPC

OReilly Printer OReilly

print$ Disk Printer Drivers

This machine has a browse list:

Server Comment

--------- -------

HOPPER Samba 1.9.15p8

KERNIGAN Samba 1.9.15p8

LOVELACE Samba 1.9.15p8

RITCHIE Samba 1.9.15p8

ZIMMERMAN

The browse list shows other SMB servers with resources to share on the network.

To use the client, run:

/usr/bin/smbclient service <password>

where 'service' is a machine and share name. For example, if you are trying to reach a directory that has been shared as 'public' on a machine called zimmerman, the service would be called \\zimmerman\public. However, due to shell restrictions, you will need to escape the backslashes, so you end up with something like this:

/usr/bin/smbclient \\\\zimmerman\\public mypasswd

where 'mypasswd' is the literal string of your password.

You will get the smbclient prompt:

Server time is Sat Aug 10 15:58:44 1996

Timezone is UTC+10.0

Domain=[WORKGROUP] OS=[Windows NT 3.51] Server=[NT LAN Manager 3.51]

smb: \>

Type 'h' to get help using smbclient:

smb: \> h

ls dir lcd cd pwd

get mget put mput rename

more mask del rm mkdir

md rmdir rd prompt recurse

translate lowercase print printmode queue

cancel stat quit q exit

newer archive tar blocksize tarmode

setmode help ? !

smb: \>

If you can use ftp, you shouldn't need the man pages for smbclient.

Although you can use smbclient for testing, you will soon tire of it for real work. For that you will probably want to use the smbfs package. Smbfs comes with two simple utilties, smbmount and smbumount. They work just like mount and umount for SMB shares.

One important thing to note: You must have smbfs support compiled into your kernel to use these utilities!

The following shows a typical use of smbmount to mount an SMB share called "customers" from a machine called "samba1":

[root@postel]# smbmount "\\\\samba1\\customers" -U rtg2t -c 'mount /customers -u 500 -g 100'

Added interface ip=192.168.35.84 bcast=192.168.255.255 nmask=255.255.0.0

Got a positive name query response from 192.168.168.158 ( 192.168.168.158 )

Server time is Tue Oct 5 10:27:36 1999

Timezone is UTC-4.0

Password:

Domain=[IPM] OS=[Unix] Server=[Samba 2.0.3]

security=user

Issuing a mount command will now show the share mounted, just as if it were an NFS export:

[root@postel]# mount

/dev/hda2 on / type ext2 (rw)

none on /proc type proc (rw)

none on /dev/pts type devpts (rw,mode=622)

//SAMBA1/CUSTOMERS on /customers type smbfs (0)

Please see the manual pages for smbmount and smbumount for details on the above operation.

## 9. Sharing A Linux Printer With Windows Machines

To share a Linux printer with Windows machines, you need to make certain that your printer is set up to work under Linux. If you can print from Linux, setting up an SMB share of the printer is stright forward.

Note that Windows users must have an account on the Linux/Samba server in order to print. Windows 95/98 will attempt to authenticate to the print server using the username and password used on login to the Windows box.This means that if you clicked 'Cancel' when logging onto Windows, you can't print (or connect to other SMB services)! Windows NT allows one to explicitely provide a username and password when connecting to a printer.

See the Printing HOWTO to set up local printing.

Add printing configuration to your smb.conf:

[global]

printing = bsd

printcap name = /etc/printcap

load printers = yes

log file = /var/log/samba-log.%m

lock directory = /var/lock/samba

[printers]

comment = All Printers

security = server

path = /var/spool/lpd/lp

browseable = no

printable = yes

public = yes

writable = no

create mode = 0700

[ljet]

security = server

path = /var/spool/lpd/lp

printer name = lp

writable = yes

public = yes

printable = yes

print command = lpr -r -h -P %p %s

Make certain that the printer path (in this case under [ljet]) matches the spool directory in /etc/printcap!

The lines:

printcap name = /etc/printcap

load printers = yes

controls whether all the printers in /etc/printcap should be loaded by default. If you do this, there is no reason to set up printers individually. The section [printers] specifies options for the printers that you wish to explicitly difine. If the printing subsystem you are using doesn't work this way (BSD), you need to set up a fake printcap file (or to use the 'print command' technique, see below). For more information on the printcap system see the Printing HOWTO.

A useful technique to test the network connection is to change the print command to:

print command = cp %S /tmp/print.%P.%S

The resulting file can then be analyzed.

NOTE: There are some problems sharing printers on UNIX boxes with Windows NT machines using Samba. One problem is with NT seeing the shared printer properly. To fix this, see the notes in the Samba distribution in the file docs/WinNT.txt. The other deals with password problems. See the comments in the same file for an annoying gain of understanding and failure to fix the problem.

Oleg L. Machulskiy ( [machulsk@shade.msu.ru](mailto:machulsk@shade.msu.ru)) suggests that a better print command to use in the above example would be:

print command = smb2ps %s | lpr -r -h -P %p

where 'smb2ps' is a script which transforms the spool file received from Windows into usual a usable Postscript file. It must cut off first 3 lines and last 2 lines, because these lines contain some PJL or PCL codes.

That approach is only needed if your Windows machine is printing PCL and not real Postscript. I have found that Windows 95/98/NT don't have a generic Postscript driver per se, but the "Digital turbo Printserver 20" driver acts as a good general Postscript driver for most setups. I have also heard that the "Apple LaserWriter II NTX" driver works for this purpose.

If you are creating a printer spool directory instead of using one created by a Linux distribution's installation utility, be careful of permissions! Neil Fraser ( [neilf@necon.co.za](mailto:neilf@necon.co.za)) suggested setting the permissions of the spool directory (in his case, /var/spool/lpd/lpr) to 4755 (note the suid bit). This worked for him when the owner of the directory was 'root' and the group was 'lp'.

Jeff Stern ( [jstern@eclectic.ss.uci.edu](mailto:jstern@eclectic.ss.uci.edu)) reported that he had to set the permissions on his spool directory to 777 in order for non-priviledged users to print, although he notes that he could have also added users to the 'lp' group. This is a decision for local systems administrators; if printing security is an issue, then lock it down. In home environments, you will probably want everyone to be able to print.

Dr. Michael Langner ( [langner@fiz-chemie.de](mailto:langner@fiz-chemie.de)) points out that write permission problems on the /var/spool/lpd/ tree could be avoided by use something like "path = /tmp" and "print command = lpr -r -P%p %s" instead.

Sometimes, a Postscript parsing error will occur with Postscript printing from Windows machines that causes an extra page to be printed at the end of every print job. The last page will always have "%%[ Lastpage ]%%" at the top of it. This seems to happen with Windows 95 and 98 only and is because the Postscript is malformed.

One way to handle that is to use a script to remove that bit of bad Postscript from the spooled jobs. Another way is to try to find a better Windows Postscript driver. Probably the best way is to us LPRng instead of Postscript to print to a Samba server.

Erik Ratcliffe ( [erik@caldera.com](mailto:erik@caldera.com)) Caldera tells me that using LPRng means that any printer driver can be used from Windows machines. On the Samba server, they used an /etc/printcap entry that looked like this:

raw:\

:rw:sh:

:lp=/dev/lp1

:sd=/var/spool/lpd/raw

:fx=flp

LPRng doesn't require :\ at the end of every line. A printer entry will still need to be made in /etc/smb.conf for the physical printer. The print command line needs to use the "raw" entry in /etc/printcap and data must be sent to the printer in binary form. Try a print command line like this:

print command = lpr -b -Praw %s

You may also need to set the spooling on the Windows95 end to print directly to the printer instead of spooling.

If you constantly get a extra page printing at the end of print jobs from Windows clients, try adding an "sf" directive in /etc/printcap. This will suppress form feeds separating jobs, but will not effect form feeds within documents.