Michael MacIsaac, IBM

Chapter 1.

Migrating Windows Servers to Samba

Samba is an open source software package often used with Linux distributions. Replacing Windows file and print servers with Samba results in a more stable server environment and reduces costs because no client licenses are needed. This paper describes considerations and issues when migrating Microsoft Windows servers to Samba running on Linux. Simple file serving function with Samba is relatively straightforward. However, duplicating some of the more advanced function available on Windows servers can be difficult to set up or is simply not supported with Samba. This paper describes file and print serving function from a Windows point of view.

When Samba and Linux are running on IBM zSeries (mainframe) hardware, there are some unique aspects of the solution which are described also. However, the majority of this paper applies to Linux running on any hardware platform.

This paper is divided into the following sections:

- Sections 1 and 2 mirror each other. The first describes much of the Windows Server function including browse lists, basic file and print serving, NT 4 Domains, Active Directory, etc. The second section, Section 2., “Equivalent Samba function” on page 21addresses how each of these file and print server functions is or can be duplicated using Samba and Linux.
- Section 3 “Samba scenarios” on page 43 describes in detail many of the scenarios described in the previous section. When you have made a decision on how to implement Samba, you should be able to sit down with the scenarios and get to work.
- Section 4 “Migration to zSeries considerations” on page 79 describes how to propose and design migrations from Windows to Samba servers. Additionally, unique zSeries characteristics and performance issues are addressed.
- Section 5 “Migration from Novell NetWare” on page 82 discusses a couple of basic approaches to migrating Novell NetWare data to Samba.
- The last section, “Advantages of Samba and IBM zSeries” on page 83 describes some advantages of file and print serving with Linux on zSeries.

Please send *any* feedback to mikemac@us.ibm.com

Credit

- Special thanks to Steve Heinbuch of IBM Canada for the basic requirements for and thorough reviews of this paper.
- Thanks goes to John Terpstra of the Samba team, lead author of the *Samba HOWTO Collection,* for input from that work and a review of this paper.
- Thanks to Malcolm Beattie, Steve French, Claudia Prawirakusumah, Bill Reeder, Andrew Tridgell and Michael Weisbach, all of IBM, for reviewing the paper.

Section 1 Microsoft Windows file and print serving function

The main functions that Windows file and print servers provides are *browse lists*, *file serving* and *print serving*. The protocol used is Server Message Block (SMB), sometimes called the Common Internet File System (CIFS). Necessary and important aspects of file and print serving are authentication and authorization. With these broad groups of function, the topics addressed in this section are:

- Browse lists and name resolution
- File Serving function
  - Basic server function
  - Basic client function
  - Distributed File System (Dfs)
  - Offline files/Client side caching
  - Encrypted File System (EFS)
  - Backup and restore
  - Anti-virus software
  - Quotas
- Print serving function
  - Basic server function
  - Basic client function
  - Uploading and automatic downloading of printer drivers
  - Printer pools
  - Accounting
- Time serving function
- Authentication, authorization and related function
  - NT Domains
  - NT Domain trusts
  - Active Directory
  - Permissions and Access Control Lists
  - Group policies
  - User profile and logon scripts
  - Folder redirection
  - Logon hours
  - Software distribution, RIS and Intellimirror
  - Desktop configuration control
- Client access issues
  - Windows 95/98/ME
  - Windows NT/2000/XP
  - Other clients

1.1 Browse lists and name resolution

A *browse list* is a list of SMB resources available for sharing. You can view browse lists via the DOS **`net view`** command or via the Windows NT *Network Neighborhood*, or the Windows 2000/XP *My Network Places* dialogs.

Browse lists grew out of peer-to-peer networking. They started as lists of computers and resources created via network broadcasts and were thus restricted to the local LAN. To address the issue of combining groups of browse lists, Microsoft introduced the Windows

Internet Naming System (WINS), which has a relatively flat namespace. The Internet's Domain Name System (DNS) is considered superior to browse lists because it has a hierarchical namespace and allows a single DNS server to supply every DNS name in the whole system (regardless of whether the data is on that server). With Active Directory, introduced in Windows 2000, WINS is deprecated in favor of DNS.

In order to create and maintain browse lists, one computer in the local area network becomes the *master browser* (sometimes called *local master browser* or *browse master*). This computer maintains the list of all computers, domains and workgroups.

When Windows NT/2000 domains are used, there is also a *domain master browser* which is the master browser for the domain. With just Windows servers in play, only an NT/2000/XP domain controller can function as a domain master browser.

The process by which computers become browsers is via elections. This function is beyond the scope of this paper, but to read more on it see the article *Name Resolution and Browsing in Samba*, Parts 1 and 2, on the Web at:

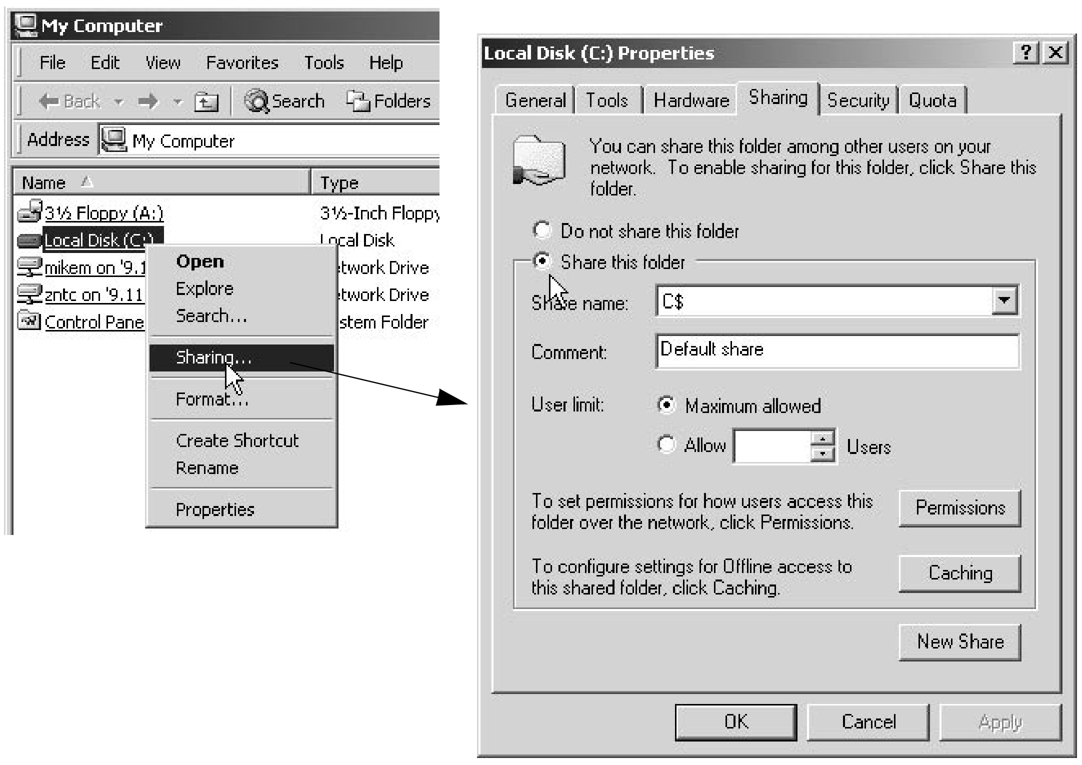
```
http://www.onlamp.com/pub/a/onlamp/excerpt/samba_chap7/index.html
http://www.onlamp.com/pub/a/onlamp/excerpt/samba_chap7/index2.html
```

1.2 File Serving function

Windows servers can easily share files and folders (directories), and can perform many of the ancillary functions associated with file serving. All Windows operating systems include both SMB client and server function.

1.2.1 Basic server function

When a folder is "shared", the computer sharing it becomes an SMB server. All logical drives are shared by default with the share name being the drive letter followed by a '$'. To view the sharing attributes of a drive, you can right click the drive in Windows Explorer and choose **Sharing** as is shown in Figure 1-1 on page 5.



You can share this folder among other users on your network. To enable sharing for this folder, click Share this folder.

( ) Do not share this folder
[✓ Share this folder]

Share name: C$

Comment: Default share

User limit: [✓ Maximum allowed] ( ) Allow Users

To set permissions for how users access this folder over the network, click Permissions.

To configure settings for Offline access to this shared folder, click Caching.

Figure 1-1 Sharing out a file resource in Windows

Disk types and sizes

An important aspect of file serving is the type and size of disks that are used. The choices are extremely varied.

On PC servers, the most affordable disks, or “hard drives” are EIDE (Extended Integrated Drive Electronics), while the more expensive and better performing disks are SCSI (Small Computer System Interface). A server can support more SCSI disks than EIDE, the throughput is faster for SCSI (typically a maximum of 160MB/s for SCSI vs. 66MB/s for EIDE), and the SCSI protocol is more sophisticated than EIDE. SCSI disks have a higher reliability and can be connected with longer cables (although the Serial ATA, SATA effort is trying to change this). Still, EIDE disks can be relatively reliable and are the lowest cost option.

More sophisticated servers will use hardware RAID. Typically RAID-5 or RAID-6 is used which will allow one or two disks to fail, without losing any data. After a disk fails, the array can be rebuilt while still online. A RAID controller, additional software and firmware is needed to maintain these systems. Each RAID array looks like a larger single physical disk to Windows.

Microsoft servers can also do RAID in software, but the performance of hardware RAID is superior, so software RAID is not as common.

Two more sophisticated and modular storage systems are becoming popular. The two systems are Storage Area Networks (SANs) and Network Attached Storage (NAS). Both systems have the advantage of off-loading processor requirements from the server to the specialized hardware.

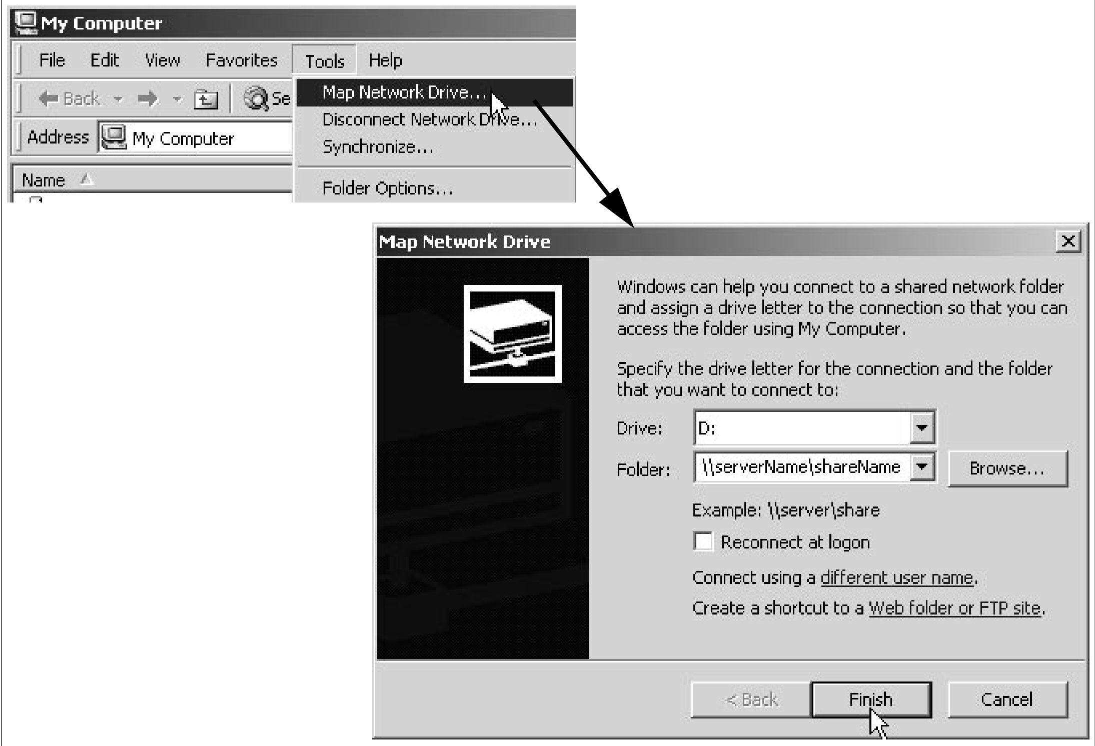
SANS are composed of specialized hardware that forms a separate storage network. The storage is not physically associated with a server, as is conventionally the case with PC servers. Rather, SANs allow data to be shared among servers. Additionally, SANs typically have their own network physically separate from the conventional TCP/IP network. This storage network allows *LAN-free backups* to be performed which is appealing in some shops because it takes the load off the LAN.

Network Attached Storage (NAS) is hardware that has a concept similar to SANs however it does not use a physically separate network. Rather, it uses the existing TCP/IP network and assigns a TCP/IP address to each of the NAS devices. NAS hardware has a small operating system running that is usually either Windows or Linux based. Generally they run well, however upgrading the OS for possible security exposures can be an issue.

Any of these options can offer large disk sizes by today's standards. 72GB, 144GB or larger disks are not uncommon, and SANs are often measured in terabytes.

1.2.2 Basic client function

Once a resource is "shared out" it can be access from an SMB client. On Windows, you can access a shared drive by using the DOS `net use` command, or more commonly, via the **Map Network Drive** function under the **Tools** menu of Windows Explorer:



Windows can help you connect to a shared network folder and assign a drive letter to the connection so that you can access the folder using My Computer.

Specify the drive letter for the connection and the folder that you want to connect to:

Drive: D:

Folder: \\serverName\shareName

Example: \\server\share

☐ Reconnect at logon

Connect using a different user name.

Create a shortcut to a Web folder or FTP site.

Figure 1-2 Accessing a shared folder from Windows

Once the drive is mapped, it appears to the user to be a local drive if adequate bandwidth is available. With a small network bandwidth, drives can still be mapped, however, poor performance can prevent the solution from being usable.

1.2.3 Distributed File System (Dfs)

A feature that was added to Windows 2000 is the Microsoft Distributed File System (the abbreviation is Dfs to avoid confusion with DCE DFS). Dfs allows dispersed shared resources on many servers to be arranged in a hierarchy. Organizing shared resources in this fashion is similar to the Internet's DNS and some UNIX file systems such as AFS. Users can get a single Dfs share and have access to all other shares on many servers. Dfs becomes beneficial in large organizations with many SMB file shares.

One of the main benefits of Dfs is location transparency. The users don't have to know the final server that holds their files. File servers can be relocated without requiring the user to reconfigure.

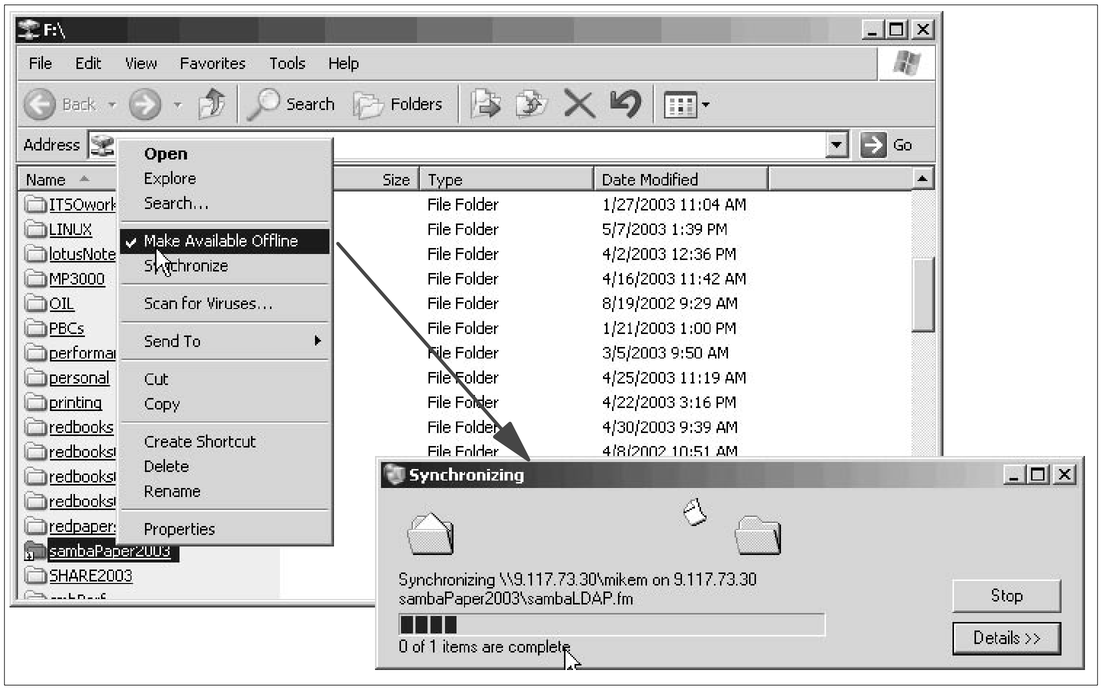
Additionally, fault tolerance is added when Active Directory is used in conjunction with Dfs. So if there are Active Directory replicas, the users don't even need to know the location of the Dfs root.

1.2.4 Offline files/client-side caching

Offline files, which is sometimes referred to as *client-side caching* is a new function that was added with Windows 2000. As the name implies, it is a client-side function that allows you to coordinate network files while working without a network connection.

By default, this function is not enabled, though it is easy to do. Any file, or more typically, any folder that is normally accessed via the network can be enabled. When you are working off-line, only those folders and files that are marked as off-line will appear.

To make a folder available offline, right click it and choose **Make Available Offline**. The folder will synchronize, by copying it to the folder named `offlineFolders` in your profile.



Synchronizing \\9.117.73.30\mikem on 9.117.73.30
sambaPaper2003\sambaLDAP.fm
0 of 1 items are complete

Figure 1-3 Making a folder available offline

When the network drive that contains the folder is not available, the folder will still be available. If changes are made while it is being accessed offline, it will have to be synchronized. If a network copy of a file and the offline copy of the same file are both changed, the synchronization process will detect this and one of the two copies must be used; there is no merge function.

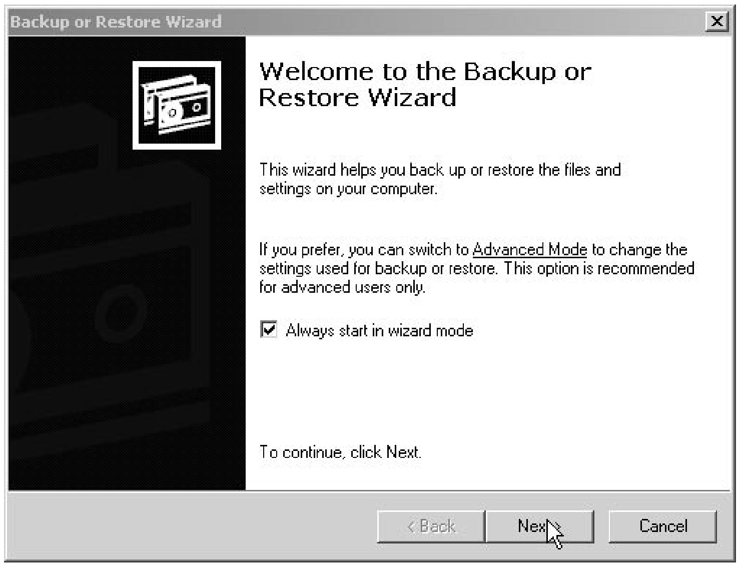
1.2.5 Encrypted File System

One of the features added to Windows 2000 is the ability to encrypt data stored on an NTFS partition. The Encrypted File System (EFS) creates a layer of encryption between the storage media and the operating system's method of storage, the file system. A key or certificate is needed to encrypt/decrypt the data. This is used commonly on storage that can be lost or stolen such as a hard drive on a laptop.

While this encrypts data between the operating system and the physical device, the data does not appear encrypted to users who have the proper key. Therefore, EFS does not address encryption over the network.

1.2.6 Backup and restore

Windows operating systems come with backup and restore tools. If you choose **Start** -> **Program Files** -> **Accessories** -> **System Tools** -> **Backup**, you will be presented with a backup wizard.



Welcome to the Backup or Restore Wizard

This wizard helps you back up or restore the files and settings on your computer.

If you prefer, you can switch to Advanced Mode to change the settings used for backup or restore. This option is recommended for advanced users only.

☑ Always start in wizard mode

To continue, click Next.

Figure 1-4 Windows Backup wizard

This built-in software addresses basic backup and restore needs. However, maintaining tape libraries, which becomes necessary as the volume of data to be backed up increases, is not included, so this is a basic backup and restore system by enterprise standards.

To address more sophisticated, enterprise-ready backup architectures, there is much third-party software available. Addressing these solutions is beyond the scope of this paper.

1.2.7 Anti-virus software

Viruses have become a problem as direct access to the Internet is now a business requirement. Slammer, sircam, Code Red and Nimda are just a few of the viruses that have infected millions of PCs, mainly via Microsoft IIS. Therefore, anti-virus software is usually a requirement. There is no anti-virus software shipped with Windows servers, but there are many vendor solutions. A couple of the more popular solutions are:

– Norton Anti Virus - `http://www.symantec.com/`
– Sophos Anti-virus - `http://www.sophos.com/products/software/`

1.2.8 Quotas

Windows NT 4 added the ability to set quotas, for the amount of disk space a user was allowed to use. A usability restriction is that quotas cannot be set on folders, rather, they must be set up on volumes (or partitions) for all users. Therefore, all users sharing the same volume must have the same quota.

Also on Windows NT it was agreed that quota management was difficult. Windows 2000 added simple quota management tools which made maintaining quotas more usable.

1.3 Print Serving function

Print serving is much more complicated than file serving because there are so many more variables with printers than with disks.

zSeries specific: Printers can be attached in at least three different ways:

- Physically to a PC via the parallel or USB port - for Linux on zSeries, this is not an option because the mainframe has no parallel or USB port. Linux on Intel supports these connections but is outside the scope of this paper.
- To the mainframe - currently there are no drivers written for Linux on zSeries to support channel-attached printers (though the Unit Record (UR) driver described in the redbook *Linux on IBM eServer zSeries and S/390: Large Scale Linux Deployment*, SG24-6824 could be used). These printers are supported on an z/OS (or OS/390) system. The z/OS *IP Printway* product has the ability to present channel-attached printers as traditional LPD UNIX printers.
- Via the TCP/IP network - these printers are becoming more common. This method of attachment is the only one addressed in this paper.

Windows servers can easily share printers. It should be noted that a “printer” in Windows terminology is software. The physical printer itself is referred to as a “print device”.

1.3.1 Basic server function

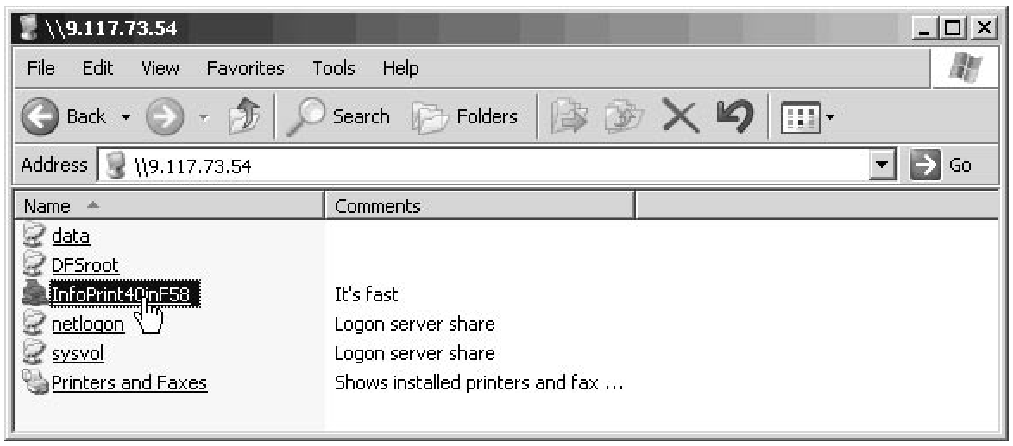
Printers can be “added” to Windows desktop systems. When the print device is physically attached to the desktop, or when an individual is only using one network-attached print device and has the necessary drivers accessible, it is simple to add the printer directly to the desktop system, and there is no need for a “print server”. However, print servers are often convenient for the following reasons:

- Multiple printers, often in many different geographic locations, need to be utilized.
- Access to correct print device drivers can be difficult or cumbersome for users to locate.
- Windows clients do not have the authorization to add printers directly.
- Configuring each individual printer to utilize characteristics of print devices can be difficult.

For these reasons, print servers are usually used. It is easy to make a Windows server a print server. Printers are simply added, with the additional step of adding drivers for download, and the Windows server is a print server.

1.3.2 Basic client function

From Windows desktops, it is very easy to add printers maintained on print servers. Find the printer you want to add in a browse list. If it is shared and you are authorized, it can be added by just double-clicking on the printer.



Name	Comments
data	
DFSroot	
InfoPrint40inF58	It's fast
netlogon	Logon server share
sysvol	Logon server share
Printers and Faxes	Shows installed printers and fax ...

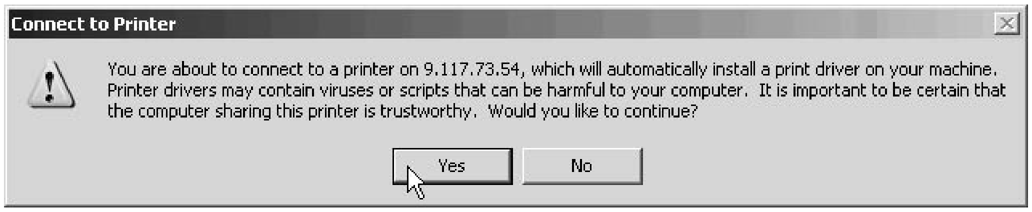
Figure 1-5 Adding a printer on a Windows print server

Alternatively, you can add a printer via the *Add Printer Wizard* from the Printers dialog.

When you add a printer directly, that is there is no print server in between, you will be prompted for the printer drivers. They may exist on your Windows system or CD, or you may provided them via various media. But using a print server enables you to bypass this step as is described in the next section.

1.3.3 Uploading and automatic downloading of printer drivers

One of the useful features of Microsoft print servers is the ability to automatically download printer drivers. When printer drivers are stored on the Windows server for a particular printer, and a user is adding that printer, they will be presented with the question “Before you can use the printer, it must be set up on your computer. Do you want Windows to set up the printer and continue this operation?” With Windows XP clients, the following slightly more ominous question is asked:



You are about to connect to a printer on 9.117.73.54, which will automatically install a print driver on your machine. Printer drivers may contain viruses or scripts that can be harmful to your computer. It is important to be certain that the computer sharing this printer is trustworthy. Would you like to continue?

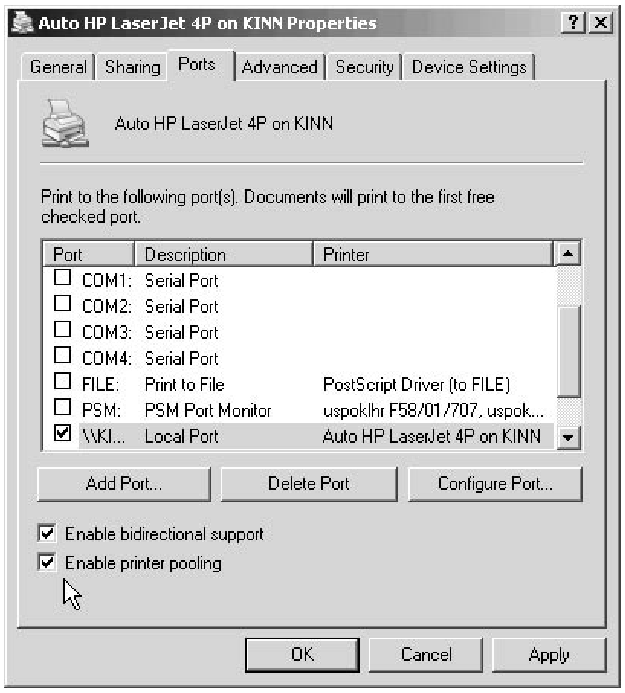
Figure 1-6 Windows XP automatic download of printer drivers slightly ominous question

Answering **Yes** to either question will add the printer without any further interaction. It should then be immediately available for use.

1.3.4 Printer pools

A *printer pool* is when multiple physical printers are connected to a single print queue. This can be a beneficial setup - if one printer goes down, print jobs will still go to other printers in the pool. Also, some printers or print rooms have a large volume of print jobs coming in. To distribute the workload equally, printer pools are often used.

Printer pools can be set up on Windows servers. This is done by checking the **Printer Pooling** check box on the **Ports** tab of the Printer's Properties dialog. Then select all ports for all printers that will be in the pool.



Print to the following port(s). Documents will print to the first free checked port.

Port	Description	Printer
☐ COM1:	Serial Port	
☐ COM2:	Serial Port	
☐ COM3:	Serial Port	
☐ COM4:	Serial Port	
☐ FILE:	Print to File	PostScript Driver (to FILE)
☐ PSM:	PSM Port Monitor	uspoklhr F58/01/707, uspok...
☑ \\KI...	Local Port	Auto HP LaserJet 4P on KINN

☑ Enable bidirectional support
☑ Enable printer pooling

Figure 1-7 Printer pooling check box

A requirement for Windows servers is that all printers in the pool must be identical.

1.3.5 Accounting

A function that is often requested for print servers is keeping track of how many pages each user has printed and making nice reports from the data. This function is often used to "charge-back" users for print services. Windows servers do not have a print server accounting function. This is an area where there are many vendor products available.

1.4 Time serving function

Setting up a time server and having clients set their clocks against a time server is an important, and often overlooked function. The Kerberos authentication protocol relies on accurate time of day clocks. In addition, many applications rely on synchronized clocks between the client and server, especially for file locking.

In a sense, there are very few time servers. Rather, most "time servers" are actually time clients of another time server which is a reliable source. The most accurate time servers are called stratum 1. When a client sets it's clock against a certain stratum time server, the client becomes a stratum n+1 client. It is common practice for a single server in an organization to set its clock against a stratum 1 or 2 time server, thus becoming a stratum 2 or 3 time server. Then many clients in the enterprise can set their clocks against this local time server.

Implementing this architecture is considered being respectful of network traffic to the accurate time servers on the Internet.

The time of day clock can be set accurately on a Windows server using the **`w32tm`** DOS command (try **`w32tm /?`**) and via the W32Time registry entries in:

```
HKEY LOCAL MACHINE\SYSTEM\CurrentControlSet\Services\W32Time\Parameters
```

There are many considerations to this function as it pertains to Active Directory that are beyond the scope of this paper. For more details see:

```
http://www.microsoft.com/WINDOWS2000/techinfo/howitworks/security/wintimeserv.asp
```

1.5 Authentication, authorization and related function

Microsoft operating systems have matured over the years in terms of authentication and authorization.

When PCs and LANs were first moving into the enterprise, authentication was not a big issue as a small number of trusted users were physically attached to the LAN. The concept of workgroups were added, but this was more to address the size of browse lists than for security. As TCP/IP became a worldwide networking standard, increased authentication and authorization became a necessity. There have been two major progressions of the security infrastructure on Windows operating systems - *NT Domains* and *Active Directory*.

Real security began with Windows NT and domains. Windows NT 4 has the concept of Primary Domain Controllers (PDCs) and Backup Domain Controllers (BDCs). A domain could only reasonable manage about 5000 users. Novell Directory Services (NDS), based on the emerging Lightweight Directory Access Protocol (LDAP), was the first leader in directory service software. NDS was considered superior to NT domains in terms of architecture

To catch up to Novell NDS, Microsoft came out with Active Directory in Windows 2000. Active Directory still has domains, however they are different from NT domains. The Active Directory namespace is hierarchical, with names separated by periods the same as DNS. An Active Directory domain can manage 1,500,000 users or more.

1.5.1 NT Domains

When all of the users in an organization could fit into a single NT 4 PDC, the system was usable, but when multiple domains were required for political or size reasons, NT domains became more complex. Also, Windows NT 4 was prone to crashing. The fact that “BSOD” became a recognized acronym for *Blue Screen Of Death* lends credence to this. The instability of Windows NT hurt it as a central authentication and authorization solution. The stability of Windows 2000 and Windows XP has increased.

1.5.2 NT Domain trusts

When there are multiple domains in an enterprise, there are multiple Security Account Managers (SAMs). In order for users of one domain to use resources of another domain (in order for the SAMs to share user data), trusts must be set up between the domains. Trusts work in only one direction, therefore, a trust must be set up from each domain to the other for both domains to share resources. To complicate matters, one domain must *allow* the other to trust it for the trust to work. Therefore, four operations must be performed. As the number of domains grow, the complexity grows quickly.

1.5.3 Active Directory

To address the deficiencies with NT domains just described and to address the growing popularity of Novell's NDS (now called eDirectory), Microsoft introduced Active Directory with Windows 2000. Active Directory adopted the emerging standard of LDAP v3, however, retained features that considered it to be proprietary. While Microsoft claims it supports LDAP v3, it has fallen short in the following areas. For example:

- Active Directory requires API developers to perform external application integration that a pure LDAP server would handle.
- Active Directory has limited schema support within directory structures.

To address these issues, Microsoft will be introducing a new version of Active Directory called *Active Directory Application Mode* (ADAM) in Windows Server 2003 which is effectively an LDAP-only version of Active Directory. See:

```
http://www.microsoft.com/windowsserver2003/techinfo/overview/adam.mspx
```

Active Directory has moved away from the basically flat namespace of NetBIOS and WINS, and moved to the hierarchical, Internet standard, DNS. The concept of PDCs and BDCs was removed in favor of Active Directory Domain Controllers (DCs) which can be *trees* and *forests*.

Each Active Directory domain, one per DC, is a domain tree. Multiple Active Directory domains can be joined to form a domain forest. The data for each domain tree on the domain controller is read-write. A read-only copy of the domain forest called the global catalog can be made available to each domain controller. With Windows 2000 and later, user logons, DNS, Exchange e-mail and SQL Server directory data all must be stored in Active Directory if those services are to be used.

When Active Directory is set up on Windows 2000, it must run in one of two modes:

- Native Mode - only Windows 2000 (or later) Domain Controllers can participate in the Active Directory tree
- Mixed mode - Windows NT 4 PDCs and BDCs can also participate

1.5.4 Permissions and Access Control Lists

There have been two main progressions of file and folder permissions on Windows operating systems: DOS attributes and NTFS security.

DOS attributes

There are four DOS attributes which can be assigned to files and folders.

Read Only	File cannot be written to
Archive	File has been touched since the last backup
System	File is used by the operating system
Hidden	File is relatively invisible to the user

These attributes apply to FAT, FAT32 and NTFS file systems.

NTFS security

There are 13 basic permissions which are rolled up into six permission groups. These apply only to the NTFS file system, not FAT nor FAT32. The six permission groups are:

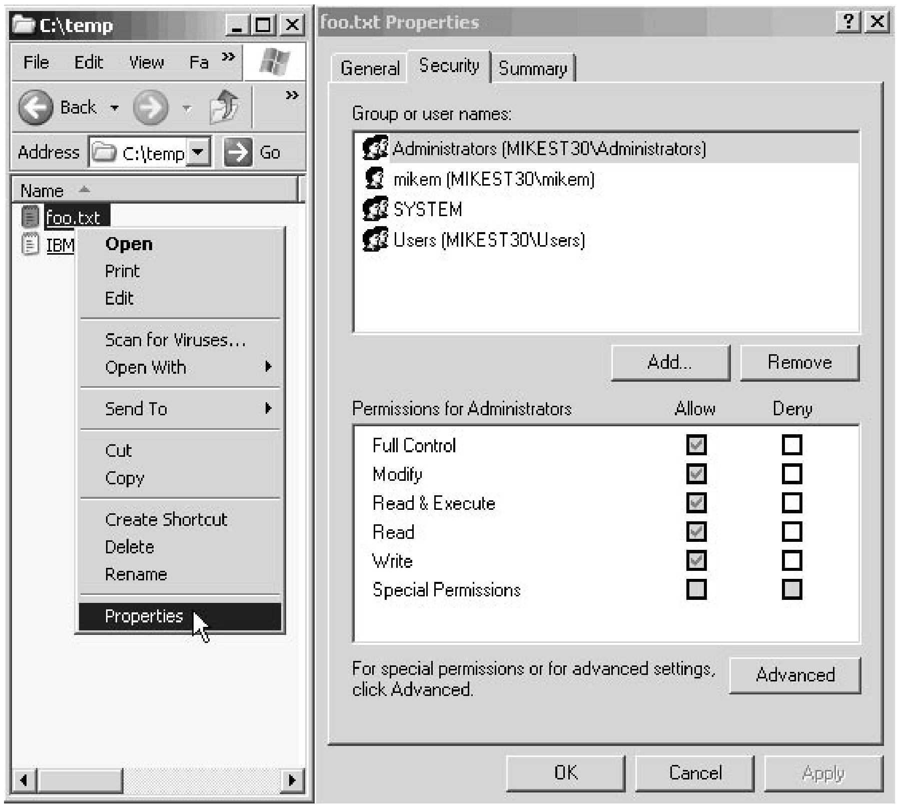
Full Control	Allow all 13 basic permissions

Modify	Allow all permissions except **Delete subfolders and files**, **Change permission**, and **Take ownership**
Read	Allow **List folder**/**Read data**, **Read attributes**, **Read extended attributes** and **Read permissions**
Write	Allow **Create files**/**Append data**, **Write attributes**, **Write extended attributes**, **Delete subfolders and files**, and **Read Permissions**
Read and execute	Allow all that the Read permission group allows plus **Traverse Folder**/**Execute File**
List folder contents	This is for folders only, not files. It is the same as Read and Execute for files

The 13 basic permissions are the following; some of them differ depending on whether they apply to folders or files:

- Traverse folders (for folders only)/Execute file (for files only)
- List folder/Read data
- Read attributes
- Read extended attributes
- Create files/Append data
- Write attributes
- Write extended attributes
- Delete subfolders and files
- Delete
- Read permissions
- Change permissions
- Take ownership

To access the permission groups, right-click on any file or folder in Windows explorer, choose the **Properties** menu item and then choose the **Security** tab as shown in Figure 1-8.



Group or user names:

- Administrators (MIKEST30\Administrators)
- mikem (MIKEST30\mikem)
- SYSTEM
- Users (MIKEST30\Users)

Permissions for Administrators	Allow	Deny
Full Control	✓	
Modify	✓	
Read & Execute	✓	
Read	✓	
Write	✓	
Special Permissions		

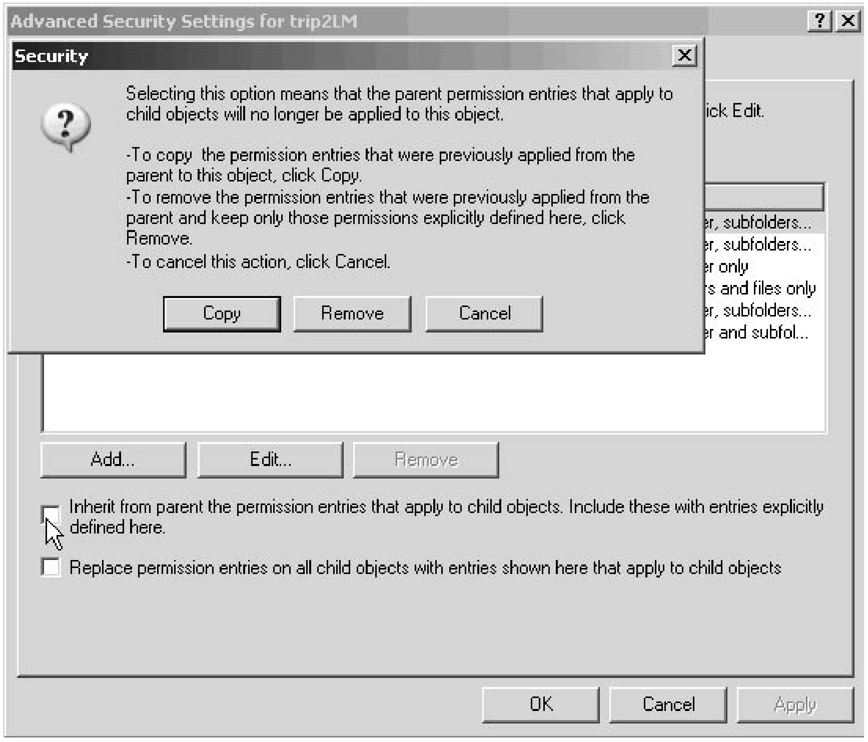
For special permissions or for advanced settings, click Advanced.

Figure 1-8 NTFS folder and file permissions

To see the 13 basic permissions, click the **Advanced** button at the bottom of the Security tab, then click **Effective Permissions**.

Inherited permissions

With Windows 2000, files and folders can be set to inherit permissions. By default, all files and directories in NTFS file systems have inherited permissions set. When a file or folder inherits permissions from the directory it is in, the permissions attributes check boxes are grayed out. You can remove the inherited permissions from a file or folder and will be presented with the question shown in Figure 1-9, “Removing inherited permission bits” on page 16.


Advanced Security Settings for trip2LM

Security

Selecting this option means that the parent permission entries that apply to child objects will no longer be applied to this object.

-To copy the permission entries that were previously applied from the parent to this object, click Copy.
-To remove the permission entries that were previously applied from the parent and keep only those permissions explicitly defined here, click Remove.
-To cancel this action, click Cancel.

Copy | Remove | Cancel

Add... | Edit... | Remove

☐ Inherit from parent the permission entries that apply to child objects. Include these with entries explicitly defined here.

☐ Replace permission entries on all child objects with entries shown here that apply to child objects

OK | Cancel | Apply

Figure 1-9 Removing inherited permission bits

When you remove the permission inheritance from a file or folder, you have to start with a new set of specific permissions. You can start with none (by selecting **Remove**) or you can start with the set which were previously inherited (by selecting **Copy**).

1.5.5 Group policies

Group Policies are a set of Windows 2000 Desktop management features which include:

- User profiles and logon scripts (including local and roaming profiles)
- Folder redirection
- Logon hours
- Software distribution, RIS and Intellimirror
- Desktop configuration control (desktop lockdown)
- Rights management and other security-oriented features

Group Policies enable desktop lockdown, rights management, roaming users and other security-oriented features. Group policies are complex and encompass many aspects of Windows management beyond the scope of this paper. Windows 2000 has about 400 group policies and Windows XP has about 600. However, some of the more commonly used function that are under the Group Policy umbrella are addressed in the next sections.

Windows NT did not have group policies, but had two system policy editors - one for NT clients and one for Windows 95/98 clients. They performed similar function, but did so by making registry changes on clients that added restrictions to the Windows Explorer user interface. The policy editor did not have to run locally on each of the clients, rather, policy files (`CONFIG.POL` for 9x clients and `NTCONFIG.POL` for NT clients) were put in the NETLOGON share and when the clients did a network logon the policies were applied. Once these policies were applied, they could not easily be removed, even by the `Administrator` on the local machine, as applying the policies modify the client’s registry. This architecture was perceived as poor and has been improved with Active Directory.

1.5.6 User profiles and logon scripts

A user profile is a group of settings that defines the environment that is loaded when a user logs on. It includes all the user-specific configuration settings, such as program files, Start menu, desktop settings, screen colors, network connections, printer connections, mouse settings, etc. There are three types of profiles:

Local User Profile	Is created the first time that a user logs on to a Windows client and is stored on the local hard disk. Changes are specific to the computer on which they are made.
Roaming User Profile	Is copied to and stored on a network share. This profile is downloaded every time a user logs on to any computer. Any changes made to a roaming user profile are synchronized with the server copy when they logoff.
Mandatory User Profile	A special type of profile that administrators can use to specify particular settings for users. Only system administrators can make changes to mandatory user profiles. Changes made by the user to desktop settings are lost when the user logs off.

User profiles can consume a lot of network bandwidth to synchronize, so there may be performance issues with logging on and off.

Logon scripts are files that get executed, usually batch (.bat) or command (.cmd) files, immediately after a user logs on.

1.5.7 Folder redirection

Folder redirection simply redirects the path of a folder to a new location. Typically the new location is a directory on a network share. Users have the ability to work with documents on a server as if the documents were based on the local drive. In the Group Policy *User Configuration* settings many aspects of each user’s computing environment can be set to follow them around from computer to computer. There are five folders that can be redirected:

Application Data	Application-specific user information
Desktop	Commonly used shortcuts seen when all windows are minimized
Start Menu	Program groups and shortcuts to invoke applications
My Documents	Directory of user-specific files and folders
My Pictures	Directory of user-specific pictures

This feature is convenient for users who frequently log in from different computers. Think of folder redirection as an extension to roaming user profiles. However, network and disk resources can be comsumed quickly with folder redirection as more data is stored in these folders.

1.5.8 Logon hours

By default, users can log on to NT Domains and Active Directory all hours of every day. There is a setting on the **Account** tab of User properties dialog to permit logons only during certain hours of certain days.

1.5.9 Software distribution, RIS and Intellimirror

Group policies can be used to allow the domain administrator to publish or update software to users or machines. Remote Installation Services (RIS) allows you to install a Windows server and clone it to other machines. Intellimirror works with RIS. If a user deletes or moves an operating system file, Intellimirror automatically repairs the machine. RIS and Intellimirror work together to detect the missing file and copy it automatically from RIS image.

1.5.10 Desktop configuration control

Group policies can be used to limit the functions that users can perform. With home PCs you can perform any function that is part of the operating system. Many IT shops have the same philosophy: leave the PC *wide open*. However, other IT shops want to prevent users from modifying the PC's configuration so as to keep the clients more consistent and stable. This can be done with *desktop configuration control*. Some times this is referred to as *desktop lockdown*.

1.6 Client access issues

In many shops the majority of desktops are running Windows operating systems. Because the Microsoft business model is built on frequent desktop operating system upgrades, there are a lot of versions of Windows.

There have been two main *lines* of Windows Operating systems. The *9x line* usually refers to Windows 95, Windows 98 and Windows Millennium Edition (ME). This is often shortened to 95/98/ME, or just 9x. This line died when Windows XP became available in 2002.

The *NT line* usually refers to Windows NT 4, 2000 and XP. Probably due to the instability of the 9x line, Windows 2000was much more widely adopted in the enterprise.

There are some differences between the two lines addressed in the next two sections. Additionally, non-Microsoft desktop operating systems are addressed.

1.6.1 Microsoft Windows 95/98/ME

One notable difference with Windows 95 is that you cannot use DNS names nor IP dotted decimal addresses in the server portion of the UNC. For example, the UNC `\\10.1.2.3\myShare` is not valid on Windows 95 clients because 10.1.2.3 is expected to be a NetBIOS name.

Another difference is that network logons are different than with NT/2000/XP clients. Joining these clients is different because the networking interfaces are different than on Windows NT/2000/XP clients. The Windows 2000 server CD has an executable in the `Clients` directory which installs the *Directory Services for Windows*. This allows older clients to "see" Active Directory resources.

1.6.2 Microsoft Windows NT/2000/XP

While many functions have been added since NT 4, file and print serving from a client point of view has not changed much until today's Windows XP SP2.

There are two Windows XP client operating systems: *Professional* and *Home*. The Home version **cannot join or access a domain-based network**. Therefore only XP Professional can be used. Think of Windows XP Home edition as the continuation of the 95/98/ME line, except that the code is based on the NT line.

XP Professional has additional security protocols not present in any of the older Windows operating systems - specifically the *sign or seal* function.

1.6.3 Other clients

The six Windows client operating systems addressed usually account for 90 - 100% of desktops in organizations today, so other desktop operating systems are secondary. Still, some are addressed here.

Older Microsoft clients

Windows 3.0 or older, Windows for Workgroups are not addressed in this paper.

Microsoft Terminal Services (Citrix)

Terminal Services clients should behave identically as other Windows clients.

Linux

Linux desktops can mount SMB file systems with the **`mount -t smb`** command. While this is not done in a pure Linux world, it can be done when the shared data resides on Windows or Samba servers. For example, from a Linux machine, a share named data on a Windows 2000 server at IP address 9.117.73.54 can be mounted on Linux with the following command:

```
# mount -t smbfs -o username=mikem //9.117.73.54/data /mnt
...
Password: ********
# ls /mnt
foo.txt*  newDocFromMyOffice.txt*
```

Section 2. Equivalent Samba function

This section mirrors Section 1: look to it for a description of the Microsoft Windows function and look to this section to see how, or if, the function is emulated by Samba. As a rule, Samba can replace the majority, but not all, of the function that Windows file and print servers provide.

Samba version 3 (or simply Samba-3) will address many of the shortcomings of the current Samba version 2. For more details, see 2.7, "Samba-3" on page 40.

When you are considering a Samba solution, you should follow two rules regarding Windows clients:

- **Rule 1**: Windows clients should not have to be modified in any way.
- **Rule 2**: When changes are needed to Windows clients, see Rule 1.

The fact that Samba is running on the server should be transparent to the users. Samba behaves like a Windows NT 4 file and print server, but cannot behave completely as a Windows 2000 or XP server, especially with regards to Active Directory. Another area of difference is that Samba does not support Kerberos authentication the way Active Directory does (while Kerberos originated on UNIX, Microsoft has *embraced and extended* the standard such that the Microsoft implementation is not compatible with UNIX/Linux).

Managing Samba from the command line

When Samba processes (**smbd, nmbd, winbindd** and **smbpasswd** for example) are started, a single configuration file is read: `smb.conf`. This file is comprised of sections, parameters and values and is fashioned after a Microsoft .ini file with the following format:

```
[section]
    parameter = value
```

Usually this file is in the `/etc/samba/` directory, but sometimes it is found in `/etc/` or `/usr/local/samba/lib/`. The latter case exists when Samba is built from source code - the default location for all Samba executables, configuration files and log files is under the directory `/usr/local/samba/`.

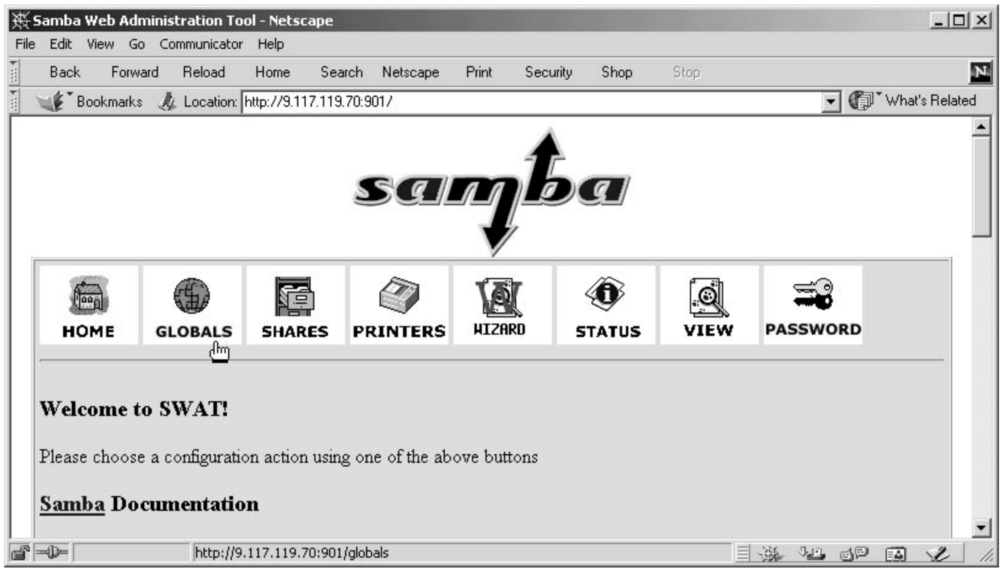
Samba services are started, stopped and queried from the *service scripts*, in the directory `/etc/init.d/`. In Red Hat 7.2 and SuSE SLES-7 there is a single script, **smb**, which starts both Samba daemons, **smbd** and **nmbd**. The new daemon **winbindd** is only needed when using it for authentication. It is probably for this reason that SuSE split out each Samba daemon into its own service script in the latest distribution, SLES-8. Now there are three scripts **nmb**, **smb** and **winbind**.

The **chkconfig** command is also supplied in SLES-8 and Red Hat 7.2 (SLES-7 includes a **chkconfig** command that performs no operation). It is used to check and set whether a process will start at boot time. For example, the following shows that **nmb** is not set up to start at boot time, but this is changed by including the **on** parameter:

```
# chkconfig nmb
nmb  off
# chkconfig nmb on
# chkconfig nmb
nmb  on
```

Managing Samba from a browser

Samba includes a management tool called the Samba Web Administration Tool (**swat**). It is a mini-Web server listening on port 901. Setting up **swat** is described in 3.1, "Setting up basic file serving" on page 43. Even if you do not use **swat** to maintain the `smb.conf` file, it can be very handy for looking up the description of parameters. A screen shot of **swat** is shown in Figure 2-1, "SWAT screen shot" on page 22.


Samba Web Administration Tool - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Shop Stop
Bookmarks Location: http://9.117.119.70:901/
What's Related
samba
HOME
GLOBALS
SHARES
PRINTERS
WIZARD
STATUS
VIEW
PASSWORD
http://9.117.119.70:901/globals

Welcome to SWAT!

Please choose a configuration action using one of the above buttons

Samba Documentation

Figure 2-1 SWAT screen shot

Samba and Webmin

Webmin is an open source, Web-based system administration tool. It consists of many Perl scripts. Webmin is available at:

`http://www.webmin.com`

In addition to the many types of system administration that Webmin can do, it also has a Samba module, and thus allows Samba management through a browser. An interesting feature of webmin is that communications can be encrypted with a secure socket layer while **swat** cannot.

2.1 Browse lists

The Samba **nmbd** daemon provides the SMB browse list function. The `smb.conf` parameters that are directly related to browse lists are `os level`, `preferred master`, `local master` and `domain master`. These pertain to whether the Samba server will be elected as the master browser and the domain master browser. Setting the `os level` parameter to a high value will ensure that it wins all browser elections.

The browse list determines which computers and resources are seen in the Windows *Network Neighborhood* or *My Network Places*.

zSeries specific: There is an issue with Samba servers running on zSeries not showing up in the browse lists. Often the primary network interface is a point-to-point connection such as virtual channel-to-channel (ctc) or IUCV. Even if a broadcast could be done on these connections, it would only be to the other side, z/VM. Therefore the Samba server will not show up in browse lists. There are a couple of workarounds to this - use an LCS or OSA connection where broadcasting will work as expected, or, if WINS servers are in place, point Samba to a WINS server with the "`wins server = <server>`" parameter.

2.2 File Serving function

Samba can provide the majority of the file server function that Windows servers can.

2.2.1 Basic server function

The Samba **`smbd`** daemon provides the equivalent SMB function. This daemon is started and stopped via the **`/etc/init.d/smb`** script. When supported, the **`chkconfig`** command can be used to query and set whether the **`smb`** script is started at boot time. One ‘master’ copy of **`smbd`** process runs, and another copy is forked for each user that is sharing a file or print resource.

An example of setting up basic file serving function is shown in 3.1, “Setting up basic file serving” on page 43.

Disk types and sizes

zSeries specific: zSeries or S/390 disk storage is referred to as DASD (pronounced “dasdy”, an acronym for Direct Access Storage Device). Typically it is allocated in approximately 3GB chunks often referred to as 3390-3s, or simply “mod 3s”. After formatting for Linux 2.3GB is usable. Because of this small size by today’s standards, a volume management system is needed. LVM is commonly used and is recommended.

There are several volume management systems available for Linux today, notably:

- Logical Volume Manager (LVM) - commonly used on SuSE distributions
- RAID tools, sometimes called *mdtools* - commonly used on Red Hat distributions
- Enterprise Volume Management System (EVMS) - a new system with multiple interfaces and allowing “plug-ins” of other systems.

SLES-8 commonly uses LVM which has its own terminology that is briefly described here:

- A hard drive or zSeries DASD volume is called a *physical volume* (PV), because that’s the volume where the data is physically stored.
- The PV is divided into several *physical extents* (PE) of the same size. The PEs are like blocks on the PV.
- Several PVs make up a *volume group* (VG), which becomes a pool of PEs available for the *logical volume* (LV).
- The LVs appear as normal devices in `/dev/` directory. You can add or delete PVs to/from a VG, and increase/decrease your LVs.

See Figure 2-2, “LVM block diagram” on page 24 for a conceptual view of these pieces.

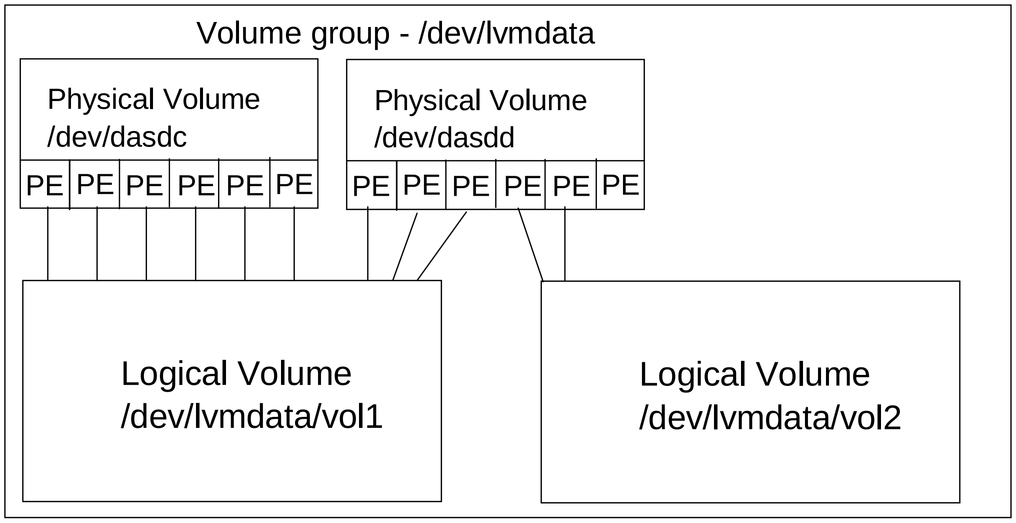

Volume group - /dev/lvmdata
Physical Volume /dev/dasdc
Physical Volume /dev/dasdd
PE
Logical Volume /dev/lvmdata/vol1
Logical Volume /dev/lvmdata/vol2

Figure 2-2 LVM block diagram

Logical volumes and striping

A logical volume can be set up to do striping. This can give the a nearly linear disk I/O performance increase with respect to the number of physical devices because it allows Linux to utilize multiple I/O paths concurrently (especially on zSeries hardware). With a striped logical volume, if you want to use all the space in all the physical volumes, you must use the same number of stripes as there are physical volumes. Also it is important to note that striped logical volumes cannot be extended with the **`lvextend`** command (the forthcoming LVM2 will allow striped logical volumes to be extended, though this function may not be available for some time).

A scenario for setting up a striped logical volume is described in 3.2, “Setting up a logical volume” on page 46.

zSeries specific: zSeries recently added support of the Fibre Channel Protocol (FCP). This allows attachment of devices that support Fibre Channel, such as the Enterprise Storage Server (ESS). See the redpaper *Getting Started with zSeries Fibre Channel Protocol* on the Web at:

```
http://www.redbooks.ibm.com/abstracts/redp0205.html
```

One possible downsize to FCP disks is that z/OS cannot vary them online. Therefore, an existing Disaster Recovery solution from z/OS to Linux cannot be used.

File system types

There are many types of file systems available on Linux. A broad classification is journalled or non-journalled. In general, a journalled file system is recommended because it will recover more quickly in the event of a system crash than a non-journalled file system will (where ext2 is by far the most common), There are many different journalled file systems including:

- ext3
- JFS

- Reiser
- XFS

All of these have different attributes and probably all are now considered stable and production ready. In SLES-8, Reiser and ext3 are the two built-in journalled file systems. The ext3 file system may have a small advantage over Reiser in terms of recoverability under extreme conditions, so it may be the best choice.

Other necessary attributes for a file system that Samba will utilize are POSIX ACLs and quota support. Without these, NTFS ACLs and NTFS quotas, respectively, will not work for Windows clients. In SuSE SLES-8, both ext3 and reiserfs support POSIX ACLs and quotas. If the Linux distribution you are working with does not have POSIX ACLs and quota support, you may have to rebuild the Linux kernel yourself. However, it is recommended to leave this job to the Linux distributor and to get support.

Also for Samba to utilize ACLs and quotas, it must be built with the **`configure`** parameters, `--with-quotas` and `--with-acl-support`. With SLES-8 these two features are compiled into Samba. Again, it is recommended that the Linux distributor do this work for you.

2.2.2 Basic client function

Windows clients pointing to a Samba server should behave almost identically as when they are pointing at a Windows server.

One area of difference is with Access Control Lists (ACLs). The Windows NT File System (NTFS) allows for individual attributes to be assigned to files and folders authorizing specific users, groups or a combination of both. See 2.5.4, “Permissions and Access Control Lists” on page 36 for details.

2.2.3 Distributed File System (Dfs)

Samba supports Dfs. To enable SMB-based Dfs for Samba, it must be configured with the `--with-msdfs` option. This option is used by SuSE and Red Hat in their recent distributions, therefore, Dfs should work by default.

Samba can become a Dfs server by setting the global `host msdfs` parameter. A share becomes a Dfs root via the share level `msdfs root` parameter. A Dfs root directory on Samba hosts Dfs links in the form of symbolic links that point to other servers. See 3.11, “Setting up a Dfs root on Samba” on page 70” for a scenario of setting up Samba as a Dfs root.

Similarly, Samba shares can be linked into a Dfs root on a Windows server.

2.2.4 Offline files/Client side caching

This function works with a Samba server. Because the function is implemented on the client side, there is really nothing that Samba has to specifically do to support it.

2.2.5 Encrypted File System (EFS)

An encrypted file system is possible with Linux between the Linux kernel and the physical device. Because authentication will be performed before users can get access to Samba shared resource, and because of the secure nature of the data center (where zSeries DASD is usually housed), you might question whether this function is necessary. However, there may be some data for which security is paramount and layers of authentication are desired. In this case, an EFS can be used.

A description of setting up an EFS is beyond the scope of this paper. For a good reference, see *Cryptographic Filesystems, Part One: Design and Implementation*, by Ido Dubrawsky, on the Web at:

`http://securityfocus.com/printable/infocus/1673`

2.2.6 Backup and restore

There are a number of ways of backing up and restoring Samba data. They can be divided into the following broad categories

- Using existing DASD backup
- Using a vendor product
- Using an open source software product

Using existing DASD backup

Many locations that have mainframes also have an existing OS/390 or z/OS backup regiment in place. If this is the case, Linux DASD can be added to the list of data to be backed up. To do this the compatible disk layout (cdl) format must be used when the DASD is formatted for Linux. This is the default value on the **`dasdfmt`** command with SuSE SLES-7 and SLES-8, Red Hat 7.2 and Debian 3.0. Additionally, the DASD must be offline it is being backed up.

Using a vendor product

Other enterprise-wide backup and restore solutions may be in place. There are several products available for use with Linux which will perform this function, including:

- Computer Associates BrightStor Enterprise Backup

  `http://www3.ca.com/Solutions/Product.asp?ID=251`

- IBM Tivoli Storage Manager (TSM)

  `http://www.tivoli.com/products/index/storage-mgr/`

- Innovation Data Processing FDR/UPSTREAM

  `http://www.innovationdp.fdr.com/ups.cfm`

- Legato Networker

  `http://www.legato.com/products/networker/index.cfm`

- SecureAgent's SecureBackup

  `http://www.secureagent.com/securebackup/`

- UTS Global Backup and Restore (BAR)

  `http://www.utsglobal.com/linuxprod.html`

Using an open source software product

There are many open source software solutions available for backup and recovery. Two are mentioned here - *Amanda* and *rsync.*

Amanda is an open source backup scheduler, originally developed at the University of Maryland for scheduling the backup of their computing facilities.

Amanda uses a client-server arrangement to facilitate the backup of network-attached servers. Using Amanda, it is possible to have a single tape-equipped server backing up an entire network of servers and desktops. Backups are scheduled on one or more servers equipped with offline storage devices such as tape drives. At the scheduled time, the Amanda server contacts the client machine to be backed up, retrieving data over the network and writing it to tape. The data from the client can be stored in a staging area on disk, which improves the performance of the tape writing process (or provides a fallback, in case of tape problems). Amanda can perform compression of the data being backed up, using standard Linux compression utilities (gzip, bzip). If network utilization is high, the compression can be done on the client to reduce the network load and potentially reduce backup times. This also lightens the load on the backup server, which may be processing many simultaneous backups. For more details on Amanda see:

```
http://www.amanda.org/
```

The rsync package is not a true backup system, but it has many interesting attributes. If the world of backup and restore can be divided into two halves - Disaster recovery and incremental backup, rsync can address the incremental backup half while a more specific system should be used for disaster recovery.

For an example of setting up rsync to do nightly backups of a Samba file system, see 3.10, “Setting up rsync for backup” on page 69. For more details on rsync see:

```
http://rsync.samba.org/
```

2.2.7 Anti-virus software

With Samba running, Windows data and executables are typically stored on Linux. There are two ways that this data can be interrogated for viruses:

- From the Windows client side looking at the shares
- From the Linux server side looking at the directories

From the Windows side, see the vendor solutions listed in 1.2.7, “Anti-virus software” on page 8. From the Linux side, there is one open source package shipped with SuSE SLES-8. It is samba-vscan and should be installed on your system:

```
# rpm -qa | grep vscan
samba-vscan-0.2.5d-58
```

There are also vendor anti-virus solutions for Linux:

McAffee McAffee virusscan

```
http://www.stalker.com/CGPMcAfee/
```

zGuard zGuard is a Linux based Internet Security Solution for S/390 and zSeries with firewall, IPsec-VPN, online virus scan of various protocols including Mail, HTTP, NNTP, and FTP, and basic IDS functionality. See:

```
http://www.zguard.de/
```

RAV RAV AntiVirus for MailServers provides AntiSpam, AntiVirus, Content Filtering and Message Stamping for Linux mailservers. RAV supports Sendmail, QMail, Postfix, Exim, CommuniGate Pro, Insight Server. See:

```
http://www.raeinternet.com/rav/ravzseries.html
```

Important: In June of 2003, Microsoft purchased RAV anti-virus.

The future of this company’s solution on Linux is now uncertain.

2.2.8 Quotas

Samba’s quota function is still considered experimental as of version 2.2.8a, however, many in the Samba community are using it, and it seems to work fine. Soft limits (resulting in warnings) and hard limits (which enforce the quotas) can be set both on the disk space used and on the number of files (inodes) by Linux user or group.

To enable Samba quotas, the majority of the work is done on Linux. It makes sense to use user quotas with “personal” Samba shares (`[homes]` for example) and to use group quotas with teams sharing files.

There are three Linux services associated with quotas:

boot.quota	Run the **`quotacheck`** command at boot time
quota	Designed for local file systems
quotad	Designed for NFS access to local file systems.

The **`quota`** and **`quotad`** services are not enabled when SLES-8 is installed, but turning them on is fairly easy. Once quotas are enabled on Linux, there is nothing special that Samba has to do to support them. For a scenario of setting up quotas on Linux and seeing them enforced via Samba see 3.14, “Setting up quotas on Linux then Samba” on page 75.

2.3 Print Serving function

Samba is not really a print server. You must set up Linux to be a print server, and then Samba can be a *middle-man* between the print server and Windows clients. There are three choices for the print server on Linux:

- LPD (Line Printer Daemon) and associated commands
- LPRng (LPR next generation)
- CUPS (Common UNIX Printing System)

It appears that CUPS is becoming the de-facto industry standard. One advantage that CUPS seems to have over LPRng is that it supports the new IETF Internet Printing Protocol (IPP). SuSE SLES-7 and SLES-8 come with CUPS installed and enabled. Red Hat switched from LPRng to CUPS. This section was found in the eWeek article *Red Hat Shows a More Limber Linux* by Jason Brooks:

> “LPRng print spooler software is among the software left out of Limbo or otherwise slated for removal from the next version of Red Hat Linux. LPRng has been scrapped in favor of CUPS (Common Unix Printing System). We've had good experiences with CUPS, and the preference for this single printing system should resolve confusion for users previously presented with a choice between the two mutually exclusive packages.”

In addition to the CUPS open source software package, there is an vendor product, *ESP Print Pro*, by Easy Software products that is a complete cross-platform printing system with over 3500 printer drivers for Linux and other UNIX operating systems. See:

```
http://www.easysw.com/printpro/
```

2.3.1 Basic server function

CUPS should be configured and tested on Linux first. When printing from Linux is working, then Samba can make the CUPS printers available to Windows clients.

CUPS and Samba on SuSE SLES-8 offers most SMB printing function, but not the following:

- Print pools (CUPS classes) - There appears to be some support in Samba 2.2.8a combined with CUPS 1.1.19. These levels are not available currently in SLES-8, but may be available in SLES-8 “SP2”, slated to be available in July 2003.
- Banner or separator pages - Adding separator pages could be done from Linux with CUPS, but could not be done from Windows clients. Details on the crux of the issue need to be investigated.

The next section lists CUPS directories and commands. There is a directory `/etc/cups/` with the following configuration files and directories:

`command.types`	MIME types file for the CUPS drivers
`classes.conf`	Definition of printer classes (or "printer pools")
`cupsd.conf`	The CUPS server configuration file
`mime.convs`	MIME type conversion file
`ppds.dat`	Data file of all PPD files under `/usr/share/cups/model/`
`client.conf`	The client configuration file
`mime.types`	MIME types file
`printers.conf`	Definition of printers
`ppd/`	Directory with PostScript Printer Description files for printers.
`certs/`	Directory with authentication certificate files for local HTTP clients
`ssl/`	Directory with SSL keys, certificates, etc.

CUPS can be manipulated via the Linux command line or via various GUIs. The most obvious graphical interface is the CUPS daemon itself. It is a mini-Web server similar to the SWAT architecture. It listens on well known IPP port 631:

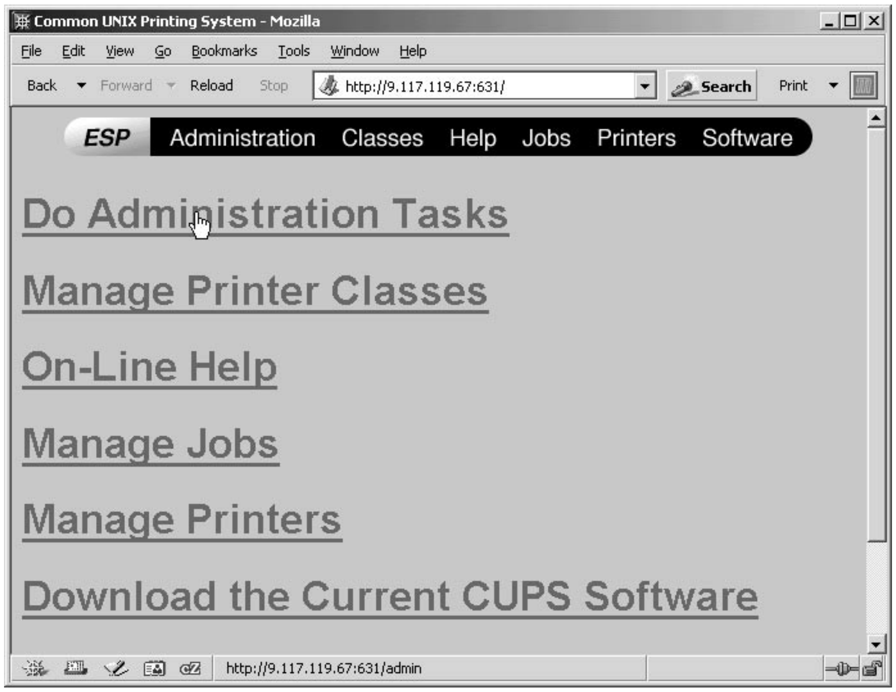

Common UNIX Printing System - Mozilla
File Edit View Go Bookmarks Tools Window Help
Back Forward Reload Stop
http://9.117.119.67:631/
Search Print
ESP Administration Classes Help Jobs Printers Software
Do Administration Tasks
Manage Printer Classes
On-Line Help
Manage Jobs
Manage Printers
Download the Current CUPS Software
http://9.117.119.67:631/admin

Figure 2-3 CUPS GUI management interface

In addition Webmin and YaST2 both have interfaces to CUPS. There is no replacement for knowing the CUPS commands, configuration files and data files.

The following system commands related to CUPS are in `/usr/sbin/`:

`accept`	Accept print jobs to the specified destinations.
`cupsaddsmb`	Export printers to samba for windows clients
`cupsd`	The cups daemon - Web browser listening on port 631 (ipp)
`lpadmin`	Set default, create or delete cups printers and classes
`lpinfo`	Show available devices or drivers
`lpmove`	Move a job to a new destination
`reject`	Symbolic link to **`accept`**
`lpc`	Line printer control program

The following user commands related to CUPS are in `/usr/bin/`:

`cups-config`	query various CUPS configuration values
`cancel`	Cancel jobs
`disable`	Symbolic link to **`accept`**
`enable`	Symbolic link to **`accept`**
`lp`	Print files
`lpoptions`	Display or set printer options and defaults
`lppasswd`	Add, change, or delete digest passwords
`lpstat`	Print cups status information
`lpq`	Show printer queue status
`lpr`	Print files
`lprm`	Cancel print jobs

For an example of using CUPS, see 3.6, “Setting up basic print serving with CUPS” on page 60 for a basic CUPS setup scenario.

Setting up Samba

Once printers are defined to Linux, it is relatively easy to add them to Samba. The following `smb.conf` parameters are important.

```
[global]
...
        printcap name = CUPS
        printing = CUPS
        printer admin = @ntadmin
...
[printers]
        comment = All Printers
        path = /var/samba/printers
        printable = yes
        create mask = 0600
        browseable = no
[print$]
        comment = Printer Drivers
        path = /var/lib/samba/drivers
        write list = @ntadmin root
        force group = ntadmin
        create mask = 0664
        directory mask = 0775
```

The `printcap name` and `printing` global parameters specify that CUPS is the print server. The default values are for lpd. The `printer admin` global parameter specifies the list of users that can administer printers via the MS-RPC interface.

The `[printers]` section is a special section analogous to the `[homes]` section for file serving. While the `[homes]` section makes a share out of all user’s home directories, the presence of the `[printers]` section in the `smb.conf` file causes all printers defined to the print server to be shared.

The `[print$]` section is another special section used to store printer drivers. As described in 3.8, “Enable automatic downloading of printer drivers” on page 64, one of the appealing features of Windows print servers is the ability to automatically download printer drivers. Beneath the directory specified by the `path =` parameter, the following directories must exist for each Windows client architecture that is to be supported:

`W32X86`	Windows NT/2000/XP x86
`WIN40`	Windows 95/98/ME
`W32ALPHA`	Windows NT Alpha_AXP

`W32MIPS`	Windows NT R4000
`W32PPC`	Windows NT PowerPC

All of these directories have been created on a SLES-8 distribution, though it is not common to see the last three architectures used.

2.3.2 Basic client function

The user should not see any difference when adding and using CUPS printers via a Samba server. One issue that has been identified is the inability to print banner or separator pages when using Samba.

2.3.3 Uploading and automatic downloading of printer drivers

Uploading of printer drivers can be done from a Windows client one printer at a time. For a manageable number of printers, this method is recommended because you can be fairly certain that the correct drivers are being uploaded. For a scenario see 3.8, “Enable automatic downloading of printer drivers” on page 64. For a large number of printers, some automation may be required.

There is an issue regarding the proper initialization of the printers. After drivers (DLLs) are uploaded to Samba no action takes place: Linux is not able to load a Windows DLL. So the driver is not given the opportunity to initialize the `PrinterDriverData` keys. Because the installation is not complete when a test page is printed, it fails. A workaround is to open the printer properties from a Windows client and set the page orientation. This will always initialize the driver. Then the page orientation can be changed back to the original value.

Setting up printer drivers can also be done on the Linux side. There is an executable named **`cupsaddsmb`** that is part of the CUPS package. It exports printers to Samba for use with Windows clients. There have been some quality concerns with this function, though it has worked fine for many.

2.3.4 Printer pools

A CUPS *class* is the equivalent of a printer pool. This function does not work with the default level of Samba and CUPS on SuSE SLES-8. CUPS 1.1.19 is necessary. However, this function may be enabled with the new packages on the SLES-8 SP2 CD which became available in June of 2003.

2.3.5 Accounting

CUPS accounting is rudimentary. CUPS does write to the file `/var/log/cups/page_log`, but the value of the data will be dependent on the backend filter. The following was found on the Internet at:

`http://printing.kde.org/faq/cups.phtml`

“CUPS passes nearly every job through the `pstops` filter; `pstops` does, amongst other things, the page counting. Output of this filter then may be piped into other filters or even a chain of filters (like "pstoraster --> rastertopcl") or sent to the printer directly (if it is a PostScript printer).

In any case, this works for network, parallel, serial or USB printers the same. For `pstops` to work, it needs DSC, Document Structuring Convention compliant PostScript (or near-equivalent) as input. This way it calculates the pages during filtering on the print server and writes info about every single page (what time, which user, which job-ID and -name, which printer, how many copies of which pages of the document, how many kilo-bytes?) into `/var/log/cups/page_log`. However, it is *not* giving correct results in the following cases:

- the printer jams and maybe therefor throw away the job (real life experience with real-life printers; or maybe throws away the job because of problems with the data format)
- jobs printed as “raw” are always counted as size of 1 page (and maybe multiple copies).

Therefore page accounting of CUPS is “only” an approximation (in many cases an excellent + good one, in others a quite poor one).”

2.4 Time serving function

Linux comes with a time client package named **`xntp`**. When **`xntp`** is used as a client to a reliable time source, the Samba server can then be used as a reliable time source. Then it can act as a time server to Windows clients. Windows clients can set their time with the command:

```
net time \\sambaServer /set /yes
```

Setting the `smb.conf` parameter `time server = yes` will cause Samba to advertise itself as a time server in the browse list through **`nmbd`**. Regardless of this setting, clients can still set their time with the **`net time`** command.

For a scenario see 3.9, “Setting up Samba as a time server” on page 68.

2.5 Authentication, authorization and related function

Samba can perform authentication and authorization many different ways. Clients can be configured via a registry setting to send clear-text passwords in which case the Linux `/etc/passwd` file can be used. Since about 1997 (Windows 95 OSR2 and Windows NT SP3), Microsoft operating systems encrypt passwords going over the network. Because Linux uses a different encryption algorithm, the encrypted password from Windows clients cannot be compared against the password in `/etc/passwd` (or `/etc/shadow`). Therefore the clients must be modified via a registry setting to turn off password encryption. This breaks Rule 1 (see Section 2., “Equivalent Samba function” on page 21) and is less secure.

Assuming encrypted passwords are to be accepted, users can be authenticated by Samba in the following ways:

- Via the Linux `smbpasswd` file
- Via a Windows domain controller and winbind
- Via a Samba server acting as a PDC
- Via an LDAP server

Authentication via the Linux smbpasswd file

This option is probably the easiest given the previous discussion. Each user must be added to the Linux system and typically their Linux and Samba passwords are kept in sync via the **`passwd`** and **`smbpasswd`** commands. SWAT also has an interface to modify passwords synchronously.

This option is good for small teams, but is often not viable for large consolidations of Windows servers. The user names and passwords are often already maintained on a domain controller, be it NT 4 or Active Directory. As such, it would be necessary to maintain three sets of passwords (`/etc/passwd`, `/etc/samba/smbpasswd`, and on the DC). When this is the case, one of the next three options is recommended.

Authentication via winbind and a Windows domain controller

A new function named winbind was added to Samba 2.2 to allow it to authenticate users against Windows domain controllers, both NT4 and Active Directory. Before winbind was added to Samba, there was a setup using the `smb.conf` parameter `security = server`. This solution is no longer recommended. The introduction of the winbind daemon has a much better Linux architecture.

The following `smb.conf` parameters are important to winbind:

```
security = domain
workgroup = <domain name>
password server = *
```

Consider the flow shown in the next figure:

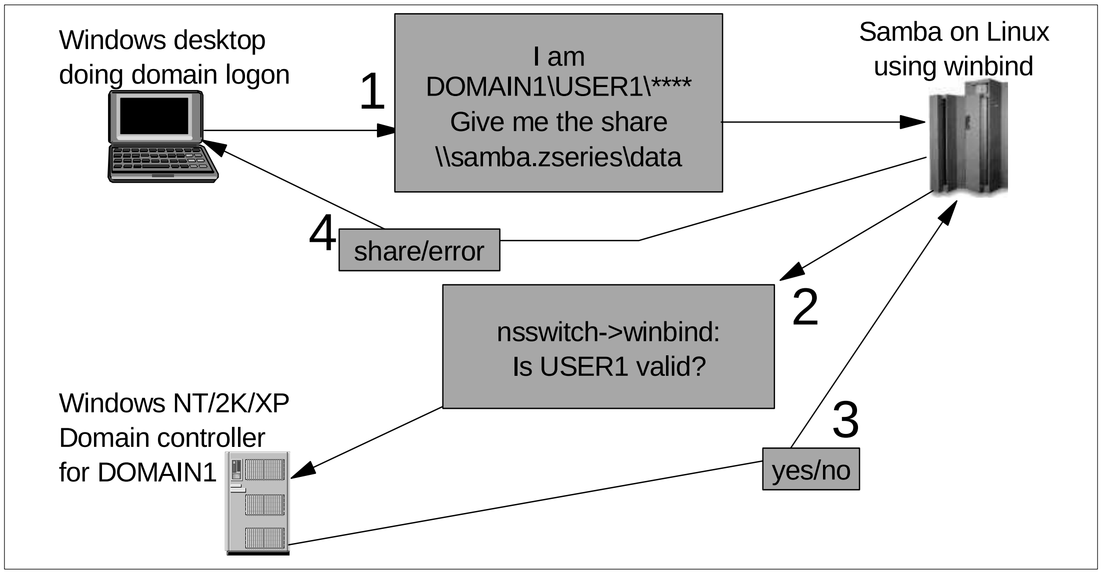

Windows desktop doing domain logon
1
I am DOMAIN1\USER1\****
Give me the share \\samba.zseries\data
Samba on Linux using winbind
4
share/error
2
nsswitch->winbind:
Is USER1 valid?
Windows NT/2K/XP Domain controller for DOMAIN1
3
yes/no

Figure 2-4 Authentication flow using password server parameter and winbind

As you can see, the password is not actually stored on Linux. If you are using the `[homes]` section the user's home directory is obtained from the `/etc/passwd` file. To automatically add an entry to the `/etc/passwd` file, the `add user script` parameter in the `smb.conf` file is often set to point to a script that will add an entry if it does not exist. Note that this happens *before* authentication is attempted, so the combination of these settings allow for a "self managing Samba", in terms of users and passwords.

The `password server = *` parameter tells Samba to locate the Domain Controller that will be doing the authentication. Often it is found by tying together the domain name specified in the `workgroup` parameter and the IP address in the file `/etc/samba/lmhosts`.

Winbind requires modification to the Linux Name Service Switch (NSS) - a function that allows easy modification of the type and number of authentication mechanisms. NSS is normally configured by editing the file `/etc/nsswitch.conf`. Additionally, winbind can be used to allow logins and other types of connections to be authenticated via the Pluggable Authentication Module (PAM), via configuration files in the directory `/etc/pam.d/`. Details on PAM are beyond the scope of this paper.

Authentication via a Samba PDC

Samba can be configured to run as an NT4 PDC. See 3.12, “Setting up a Samba PDC” on page 71. Many IT shops use Samba as a PDC with success. When Samba is acting as an NT 4 PDC, Windows clients can do network logons and therefore can utilize some of the more advanced Windows function such as startup scripts, roaming profiles and folder redirection.

There can be an issue with finding the Samba PDC. The following reference to XP clients is on the Web at:

> `http://www.microsoft.com/windows2000/dns/tshoot/`
>
> “Microsoft Windows XP Professional-based and Microsoft Windows 2000-based computers that are joining or operating in a Windows 2000 Active Directory domain search for a domain controller using a process known as the Domain Controller Locator. This process depends on locating certain DNS Resource Records (RRs) for domain controllers in the Active Directory domain namespace.
>
> Pointing domain members to the right DNS servers and ensuring that the DNS server contains the necessary records are critical aspects of troubleshooting domain operations.”

Authentication via an LDAP server

As more types of applications need to do authentication and authorization, there is a growing need for a central database of information. This can enable organizations try to attain the coveted *single sign-on*.

Using an LDAP server is recommended because it is the most open solution. Any of the following LDAP servers can being used. The first three are vendor products and include nice administration tools. The last one is a free, open source package, but is perhaps lacking in administration tools.

- IBM Directory Server
- Sun ONE - Open Network Environment (formerly iPlanet, Netscape Directory Server)
- Novell eDirectory (formerly NDS) - has been ported from NetWare to Linux. This is a good candidate for NetWare to Samba migrations
- OpenLDAP - open source software standard with most Linux distributions

LDAP architecture can get very complex. Consider the architecture shown in Figure 2-5, “A complex LDAP architecture” on page 35. Such architecture is beyond the scope of this paper. However, relatively simple scenarios are described in 3.4, “Setting up OpenLDAP” on page 52 and 3.5, “Setting up Samba to use OpenLDAP” on page 57.

Another option not described is to use the Perl module Net::LDAPapi rather than TCP/IP for LDAP communications. It makes authentication much easier as you just need to set the permissions on the UNIX domain socket therefore you don't need to store an LDAP administrator password anywhere. It requires the Samba and LDAP servers to be on the same machine.

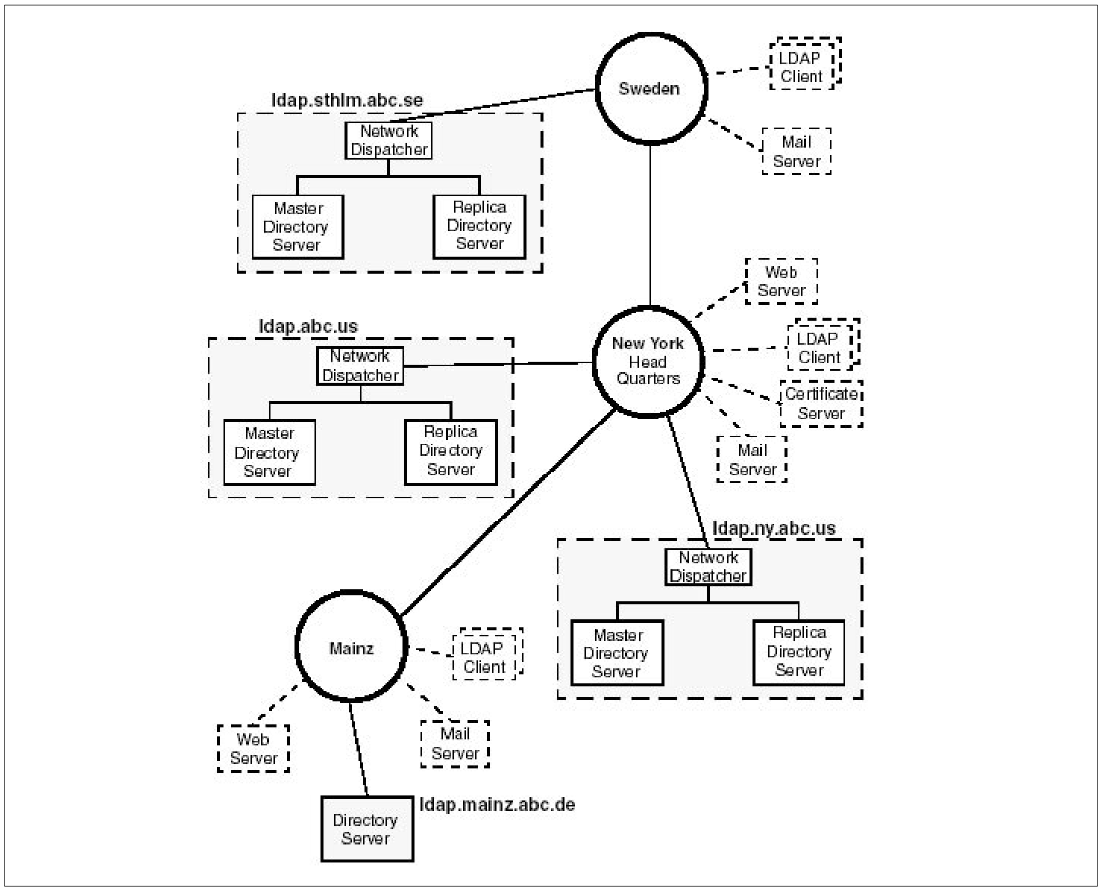


Figure 2-5 A complex LDAP architecture

2.5.1 NT Domains

Samba can act as an NT 4 PDC. See 3.12, "Setting up a Samba PDC" on page 71 for a description of how to do this.

Samba can emulate a BDC. Though this is not commonly done, it is described in chapter 10 of the Samba HOWTO collection.

2.5.2 NT Domain trusts

Samba does not support domain trusts (an offshoot of the Samba project called samba-tng, claims to support this, but this package cannot be recommended because it is not supported by the Linux distributors). Therefore, setting up Samba as a PDC is an all-or-nothing proposition when multiple NT domain controllers with trusts between them are in place.

Samba servers that have winbind set up and point to Windows domain controllers for authentication, can inherit the Windows server trusts via winbind (e.g. if there are two domain controllers trusting each other, winbind pointing at either one can authenticate users in either domain).

Samba-3 can participate in Samba-to-Samba as well as Samba-to-Windows NT4-style trust relationships. See chapter 16 of the Samba-3 HOWTO collection for details.

2.5.3 Active Directory

Samba2 can join a domain, but not as Samba can join an Active Directory domain as a member server. It cannot participate in an Active Directory domain as a domain server - that function will be available with Samba V3. For details see 2.7, “Samba-3” on page 40.

2.5.4 Permissions and Access Control Lists

Samba supports the four DOS attributes well, and many of the NTFS permissions and access control lists, but has some limitations supporting NTFS permissions.

DOS attributes

Samba can map all of the four DOS attributes. Because the three Linux execute bits are not specifically needed, the DOS attributes can be mapped to them. By default, the DOS `read only` bit is always mapped via the user read and write permission bits. The DOS archive attribute is mapped to the Linux user execute permission bit. The `system` and `hidden` attributes are not mapped, but can be. DOS attribute mapping is controlled with the following `smb.conf` parameters:

`map archive`	Map DOS `archive` attribute to owner execute bit (default = `Yes`)
`map system`	Map DOS `system` attribute to group execute bit (default = `No`)
`map hidden`	Map DOS `hidden` attribute to other execute bit (default = `No`)

See Figure 2-6, “Mapping DOS attributes to Linux permission bits” on page 36 to understand the mapping.

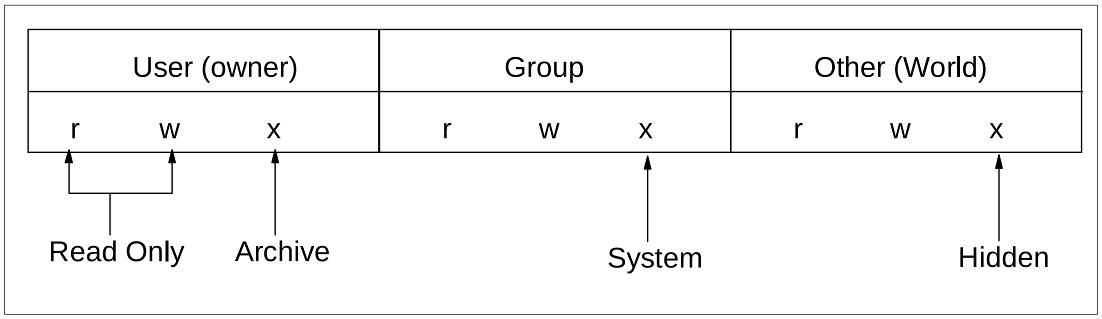


Figure 2-6 Mapping DOS attributes to Linux permission bits

NTFS security

The function to allow Windows clients to use their native security settings dialog box was added in Samba 2.0.4. At this time the default value for the `smb.conf` parameter `nt acl support` was changed from `false` to `true`. In order to correctly implement ACLs, they must be built into the Linux kernel and underlying file system. Even when they are built into a file system, they are not implemented by default. They can be implemented by changing the `defaults` keyword to `acl` (or adding `,acl` to existing parameters) for each file system they are to be used with in the `/etc/fstab` file.

Even when ACLs are implemented, there are the following limitations with respect to the function of NTFS file systems and the Microsoft security model:

- Samba cannot have multiple owners or groups
- Samba does not allow the *Take ownership* function
- Samba does not support all thirteen NT permissions

Samba cannot have multiple owners or groups

The Linux permission model is based on user and group ownership. A file or directory can have only one user owner and one group owner. These are called the file or directory's owner or group. Windows NTFS separates ownership from permission by using Access Control Lists (ACLs) that allow multiple users and groups, each with a unique set of permissions, to be associated with a file or folder. Linux file systems can also have ACLs, which have just recently become built into Linux distributions, but these are Posix ACLs and they differ from Windows ACLs.

Samba does not allow the Take ownership function

The **Take Ownership** button will usually not work on Samba as it does on NT. This is because the equivalent function on Linux, the **`chown`** command, can only be run as root.

Samba does not support all thirteen NT permissions

Linux only has read, write and execute bits, for user (owner), group and other (world). The NTFS permissions map to Samba as follows:

- The file owner **Read**, **Write** and **Read & Execute** permissions map to the Linux user (owner) `rwx` triplet.
- The file group **Read**, **Write** and **Read & Execute** permissions map to the Linux group `rwx` triplet.
- The NT Group `Everyone` **Read**, **Write** and **Read & Execute** permissions map to the Linux other (world) `rwx` triplet. This group is seen when using winbind authentication.

For example, consider a Samba setup using winbind to a Windows 2000 DC (domain name KGNLCC). A new file `foo` has the following permissions on Linux:

```
# ls -l foo
-rw-r--r--    1 KGNLCC+mikem KGNLCC+Domain Users       30 Jun 11 11:05 foo
```

To look at the permissions from a Windows client, map the drive, traverse to the file with Windows Explorer and click the **Security** Tab. If you wanted to give the world full control, you would select **Everyone** in the "Group or user names" section, click **Full Control** and then choose **Apply**:

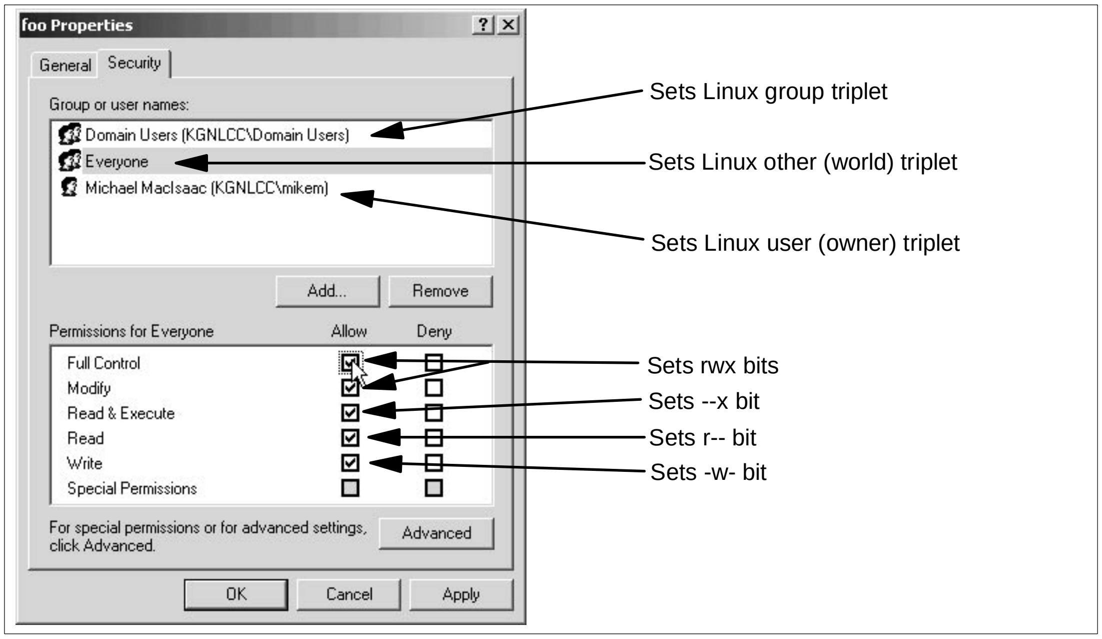


Figure 2-7 Setting permission bits from Windows to Linux via Samba

This will change the other (world) triplet to `rwx`:

```
# ls -l foo
-rw-r--rwx    1 KGNLCC+mikem KGNLCC+Domain Users       30 Jun 11 11:05 foo*
```

2.5.5 Group policies

Samba supports some group policies. To do so, it must be acting as a PDC, or using winbind that is pointing to a DC. This is because users must log on for the policies to be applied. See 3.12, “Setting up a Samba PDC” on page 71 and 3.13, “Setting up roaming profiles” on page 75 for these two setups.

Samba supports:

- Roaming profiles and folder redirection
- Logon scripts
- NT 4 System policies

Samba does not support:

- Software distribution
- Rights management and other security-oriented features
- Desktop configuration and control

Samba-3 will support many more group policies. See 2.7, “Samba-3” on page 40.

2.5.6 Roaming profiles and logon scripts

Roaming profiles are supported by Samba. The following `smb.conf` sections and parameters are required:

```
[globals]
...
        logon path = \\%L\profiles\%U
        logon home = \\%L\%U\.profile
        logon drive = H:
...
[profiles]
        path = /var/lib/samba/profiles
        read only = no
        browseable = no
        create mask = 0600
        directory mask = 0700
```

The `logon path` parameter specifies where the roaming profiles are to be kept for Windows NT/2000/XP clients. The `logon home` parameter is used with the **`net use /home`** DOS command.

The `[profiles]` section is where the roaming profiles data is kept.

The `[netlogon]` section is administrative tool used primarily for globally updating client machines with items like registry patches, anti-virus updates, program updates, etc. Anything you want to “push” out to the client, can be done via netlogon. In addition, you can use the share to enforce a system policy on a client or clients or perhaps backup a select group of files every time the user logs on. Any time a user logs onto the PDC and the `logon script =` option and `[netlogon]` share are present, Samba goes to the indicated path and executes the file referenced by logon script.

2.5.7 Folder redirection

Folder redirection is a function that is encompassed in roaming profiles.

2.5.8 Logon hours

This function is not specifically supported in Samba. A work-around is to set up cron jobs that disable and enable users via the **`smbpasswd`** command and the **`-d`** (disable user) and **`-e`** (enable user) flags at the appropriate times.

There are plans to support this function in Samba-3.

2.5.9 Software distribution, RIS and Intellimirror

Samba does not support this function.

2.5.10 Desktop configuration control

In general, Samba does not support this function.

A mandatory profile can be created on Windows and moved to Linux. Users do not have the ability to modify these settings. For Windows NT/2000/XP clients, renaming the profile from `NTUser.DAT` to `NTUser.MAN` makes it mandatory. For Windows 95/98/ME clients, renaming the profile from `User.DAT` to `USER.MAN` makes it a mandatory profile.

2.6 Client access issues

2.6.1 Windows 95/98/ME

Roaming profiles behave differently for Windows 95/98/ME clients. They key off the `logon home smb.conf` parameter.

When Samba is acting as a PDC, the [`netlogon`] share must be present for Windows 95/98/ME clients to do network logons.

2.6.2 Windows NT/2000/XP

When authenticating to Active Directory, Windows XP clients “sign or seal” the secure channel between the workstation and the domain controller. However, when Samba is acting as a PDC, the sign or seal function is not in place. The solution that is commonly recommended is to modify the XP client registry. The file `$SAMBA/docs/Registry/WinXP_SignOrSeal.reg` will modify the registry:

```
Windows Registry Editor Version 5.00
;
; This registry key is needed for a Windows XP Client to join
; and logon to a Samba domain. Note: Samba 2.2.3a contained
; this key in a broken format which did nothing to the registry -
; however XP reported "registry key imported". If in doubt
; check the key by hand with regedit.

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Netlogon\Parameters]
"requiresignorseal"=dword:00000000
```

However, this violates Rule 1(see Section 2., “Equivalent Samba function” on page 21)

2.6.3 Other clients

Linux

On Linux there is an **`smbclient`** command which is ftp-like command to interact with an SMB server, and there is the ability to mount SMB shares via the **`mount -t smb`** command. In general SMB is not the recommended protocol for file sharing, but it can be convenient in an environment with heterogeneous client operating systems.

2.7 Samba-3

Samba version 3 will add much new function. Following is an overview:

- Active Directory Services support. Samba-3 is able to join an Active Directory realm as a member server and authenticate users using LDAP/kerberos. It cannot be an Active Directory forest or tree.
- UNICODE and multi-byte character set support
- Full NT4 PDC support - will include two SAM solutions that will store the extended security information needed to implement a true replacement for MS Windows NT
  - `tdbsam` - stores all information stored in the smbpasswd file plus the extended Windows NT/2000/XP SAM information. It is recommended for sites with fewer than 250 users
  - `ldapsam` - works with OpenLDAP and the new Samba schema format.
- Better printing support including publishing printer attributes in Active Directory
- New **`net`** command modelled after the DOS command
- NT 4 style domain trust relationships
- Ability to map Windows groups to Linux groups using the **`net groupmap`** command.
- Net RPC vampire - will be able to obtain NT4 SAM accounts into it's own tdbsam or into an ldapsam database. Here are some scripts to get an idea of how it can be used:

vampire.sh

```
#!/bin/bash
smbpasswd -a root
/etc/samba/initGrps.sh
net getsid -S pdcmachine -w sargon -U Administrator%secret
net rpc join -S pdcmachine -w sargon -U Administrator%secret
net rpc vampire -S pdcmachine -U Administrator%secret
pdbedit -l
smbgroupedit -v
```

initGrps.sh

```
#!/bin/bash
net getlocalsid SARGON > /tmp/food
domsid=`cat /tmp/food | grep SID | cut -d: -f2`
echo $domsid
smbgroupedit -c $domsid-512 -u ntadmin
smbgroupedit -c $domsid-513 -u users
smbgroupedit -c $domsid-514 -u nobody
smbgroupedit -c S-1-5-32-544 -u root
smbgroupedit -c S-1-5-32-545 -u users
smbgroupedit -c S-1-5-32-546 -u nobody
smbgroupedit -c S-1-5-32-547 -u root
smbgroupedit -c S-1-5-32-548 -u sys
smbgroupedit -c S-1-5-32-549 -u bin
smbgroupedit -c S-1-5-32-550 -u lp
smbgroupedit -c S-1-5-32-551 -u daemon
smbgroupedit -c S-1-5-32-552 -u sys
```

Section 3. Samba scenarios

This section goes into detail by describing various scenarios that set up and implement many different Samba functions.

All of the examples were done on a Linux SuSE SLES-8 distribution running under z/VM on a zSeries z900 (2064). You should be able to recreate the following scenarios:

- Section 3.1, "Setting up basic file serving" on page 43
- Section 3.2, "Setting up a logical volume" on page 46
- Section 3.3, "Setting up winbind to use DCs for authentication" on page 50
- Section 3.4, "Setting up OpenLDAP" on page 52
- Section 3.5, "Setting up Samba to use OpenLDAP" on page 57 (builds on previous scenarios)
- Section 3.6, "Setting up basic print serving with CUPS" on page 60
- Section 3.7, "Setting up Samba to use CUPS" on page 63 (builds on previous scenario)
- Section 3.8, "Enable automatic downloading of printer drivers" on page 64 (builds on previous two scenarios)
- Section 3.9, "Setting up Samba as a time server" on page 68
- Section 3.10, "Setting up rsync for backup" on page 69
- Section 3.11, "Setting up a Dfs root on Samba" on page 70
- Section 3.12, "Setting up a Samba PDC" on page 71
- Section 3.13, "Setting up roaming profiles" on page 75 (builds on previous scenario)
- Section 3.14, "Setting up quotas on Linux then Samba" on page 75
- Section 3.15, "Setting up a z/VM VDISK swap space" on page 78
- Section 3.16, "Complete enterprise scenario" on page 78

3.1 Setting up basic file serving

The overall steps for basic file serving are as follows:

- Enable **`swat`** via **`inetd`**
- Add a user for Linux and Samba
- Add a share to the `smb.conf` file
- Start Samba
- Access the share from a Windows desktop

Enable swat via inetd

By default, the **`inetd`** "super-server" is not started in SLES-8. Change this with the **`chkconfig`** command:

```
# chkconfig inetd
inetd  off
# chkconfig inetd on
# chkconfig inetd
inetd  on
```

Now **`inetd`** will be started at boot time.

Verify that the swat service is defined at port 901:

```
# cd /etc
# grep swat services
swat           901/tcp         # CONFLICT, not official assigned!
```

In the default `/etc/inetd.conf` file with SLES-8, all lines are comments. Uncomment the **swat** line and start **inetd**:

```
# vi inetd.conf              # uncomment the swat line:
# swat is the Samba Web Administration Tool
swat    stream  tcp     nowait.400      root    /usr/sbin/swat  swat
# rcinetd start
Starting inetd                                                      done
```

SWAT should now be accessible via a browser at the URL:

```
http://<your server>:901
```

You will be prompted for a user ID and password. Use `root` and the root password. Figure 2-1, “SWAT screen shot” on page 22 shows an example.

Add a user for Linux and Samba

Because the `[homes]` section is present, there will be a share for each user. If you have not done so already, it is recommended that you first add a user ID and password to Linux that corresponds to the credentials you log onto the Windows desktop with. For example:

```
# useradd mikem
# passwd mikem
Changing password for mikem.
New password:
Re-enter new password:
Password changed
# mkdir /home/mikem
# chown mikemac.users /home/mikem
```

The **passwd** command creates a UNIX hash in the `/etc/shadow` file. To use Samba an “NT hash” is needed in the `/etc/samba/smbpasswd` file which is maintained with the command of the same name, **smbpasswd**:

```
# cd /etc/samba
# cat smbpasswd
# This file is the authentication source for samba. You add password
# information with the smbpasswd or smbadduser command.
#
# Cf. section 'encrypt passwords' in the manual page of smb.conf for
# more information.
# smbpasswd -a mikem
New SMB password:
Retype new SMB password:
Added user mikem.
# cat smbpasswd
# This file is the authentication source for samba. You add password
# information with the smbpasswd or smbadduser command.
#
# Cf. section 'encrypt passwords' in the manual page of smb.conf for
# more information.
mikem:500:F3265269D0AC8A0E944E2DF489A880E4:DF43D568E4C68049E43A6B09EBB041A6:[UX
]:LCT-3EA7D25B:
```

You should now have your user ID and password synchronized in three places:

- On the Windows desktop (either on the local machine or on a Domain Controller)
- On Linux in `/etc/passwd` and `/etc/shadow`
- In Samba in `/etc/samba/smbpasswd`

Add a share to the smb.conf file

Shares can added via the SWAT Web interface or by directly modifying the Samba configuration file `/etc/samba/smb.conf`. This example shows how to modify `smb.conf`. Add the share `[sharedoc]` to the default values that are set in SLES-8:

```
# cd /etc/samba
# cat smb.conf
# smb.conf is the main samba configuration file. You find a full commented
# version at /usr/share/doc/packages/samba/examples/smb.conf.SuSE
# Date: 2002-11-07
[global]
        workgroup = TUX-NET
        os level = 2
        time server = yes
        unix extensions = yes
        encrypt passwords = yes
        log level = 1
        syslog = 0
        printing = CUPS
        printcap name = CUPS
        socket options = SO_KEEPALIVE IPTOS_LOWDELAY TCP_NODELAY
        wins support = no
        veto files = /*.eml/*.nws/riched20.dll/*.{*}/
[sharedoc]
        path = /usr/share/doc/
[homes]
...
[printers]
...
[print$]
...
```

Start Samba

On past distributions “Samba” was started from a single script **`/etc/init.d/smb`**. With SLES-8, SuSE has split out the **`nmbd`** and **`smbd`** deamons to two scripts **`nmb`** and **`smb`**, and a third script **`winbind`**, if necessary. Verify that Samba is not running and then start it:

```
# rcnmb status
Checking for Samba classic NMB daemon                                    unused
# rcsmb status
Checking for Samba classic SMB daemon                                    unused
# rcnmb start
Starting Samba classic NMB daemon                                        done
# rcsmb start
Starting Samba classic SMB daemon                                        done
```

Access the share from a Windows desktop

Use the DOS **`net use`** command or the Windows Explorer *Map Network Drive* dialog to access the share named `sharedoc`. Note the large amount of Linux documentation this share makes available.

3.2 Setting up a logical volume

zSeries specific: Many of the commands and files are zSeries specific in this scenario. Linux running as a guest under z/VM is assumed.

The Logical Volume Manger (LVM) allows you to make large volumes out of smaller 3390-3s. This scenario shows how an 11.5GB volume is made from five 3390-3s. The first two steps of this scenario assumes Linux running under z/VM which uses the USER DIRECT file to maintain user IDs. The overall steps for setting up a logical volume are as follows:

- Get DASD defined to the VM user ID
- Add the DASD in Linux
- Format and partition each DASD
- Initialize LVM then create and verify physical volumes
- Create the volume group and verify
- Create a striped logical volume using most of the volume group
- Create a journalled file system and mount the logical volume
- Make a Samba share of the directory
- Set the LVM to come up at IPL (boot) time

Get DASD defined to the z/VM user ID

DASD can be managed in a number of ways under z/VM. The most straightforward way is to add `MDISK` statements to the `USER DIRECT` file on the VM MAINT user ID, reserving cylinder 0 of each DASD for z/VM. Following is an example of a user directory entry for the VM ID `MP3KLNX6`. Five DASD with labels `VM210-VM214` will be assigned virtual addresses 201-204.

```
USER MP3KLNX6 PASSWORD 128M 512M     G
 INCLUDE LNXDFLT
 MDISK 100 3390 0001 3338 VM20F  MR RPASS WPASS MPASS
 MDISK 101 3390 0751 0100 VM218  MR RPASS WPASS MPASS
 MDISK 200 3390 0001 3338 VM210  MR RPASS WPASS MPASS
 MDISK 201 3390 0001 3338 VM211  MR RPASS WPASS MPASS
 MDISK 202 3390 0001 3338 VM212  MR RPASS WPASS MPASS
 MDISK 203 3390 0001 3338 VM213  MR RPASS WPASS MPASS
 MDISK 204 3390 0001 3338 VM214  MR RPASS WPASS MPASS
 MDISK 191 3390 0851 0050 VM218  MR RPASS WPASS MPASS
```

Add the DASD in Linux

If the `MDISK` statements were added while Linux was running, either Linux has to be shutdown and the VM user ID has to be logged off, or the CP LINK command can be used to link to the DASD. When you log back on, the VM ID will have access to the new DASD. However, Linux will not necessarily recognize them - the DASD has to be added dynamically to Linux. The **`dasd`** command is not standard with SLES-8. It is a small, useful script, introduced in the redbook *Linux on IBM eServer zSeries and S/390: Large Scale Linux Deployment*, SG24-6824-00, that allows you to dynamically add, delete and list DASD using a syntax that is easy to remember:

```
# cat /usr/local/sbin/dasd
#!/bin/sh
# dasd - simple utility for dynamic DASD management
if [ "$1" = "add" -a "$2" != "" ]; then
  echo "add range=$2" > /proc/dasd/devices
elif [ "$1" = "on" -a "$2" != "" ]; then
```

```
  echo "set device range=$2 on" > /proc/dasd/devices
elif [ "$1" = "off" -a "$2" != "" ]; then
  echo "set device range=$2 off" > /proc/dasd/devices
elif [ "$1" = "list" ]; then
  cat /proc/dasd/devices
else
  echo "Usage: dasd add|on|off vdev_or_range" 1>&2
  echo " dasd list" 1>&2
  exit 2
fi
```

With this script, the DASD added dynamically. The new DASD are assigned sequentially from `/dev/dasdc` to `/dev/dasdg`:

```
# dasd add 200-204
# dasd list
0100(ECKD) at ( 94:  0) is dasda : active at blksz: 4096, 2347 MB
0101(ECKD) at ( 94:  4) is dasdb : active at blksz: 4096, 70 MB
0200(ECKD) at ( 94:  8) is dasdc : active n/f
0201(ECKD) at ( 94: 12) is dasdd : active n/f
0202(ECKD) at ( 94: 16) is dasde : active n/f
0203(ECKD) at ( 94: 20) is dasdf : active n/f
0204(ECKD) at ( 94: 24) is dasdg : active n/f
```

Format and partition each DASD

Before the physical volumes can be created, they must first be formatted and partitioned for zSeries Linux. This is done using the **dasdfmt** and **fdasd** commands. A block size of 4KB is recommended using the `-b 4096` flag to the **dasdfmt** command. To create a single partition on each of the DASD, the **fdasd -a** flag is convenient. Because formatting DASD is a lengthy process, both commands for each of the five DASD is put into a bash **for** loop:

```
# for i in c d e f g
> do
>  dasdfmt -b 4096 -y -f /dev/dasd$i
>  fdasd -a /dev/dasd$i
> done
Finished formatting the device.
Rereading the partition table... ok
...
# dasd list
0100(ECKD) at ( 94:  0) is dasda : active, 2347 MB
0101(ECKD) at ( 94:  4) is dasdb : active, 70 MB
0200(ECKD) at ( 94:  8) is dasdc : active, 2347 MB
...
0204(ECKD) at ( 94: 24) is dasdg : active, 2347 MB
```

Initialize LVM then create and verify physical volumes

Initialize LVM with the **vgscan** command:

```
# vgscan
vgscan -- reading all physical volumes (this may take a while...)
vgscan -- "/etc/lvmtab" and "/etc/lvmtab.d" successfully created
vgscan -- This program does not do a VGDA backup of your volume group
```

Create physical volumes for each DASD with the **pvcreate** command. The shell regular expression `/dev/dasd[cdefg]1` can be used as a shortcut to address all five DASD:

```
# pvcreate /dev/dasd[cdefg]1
pvcreate -- physical volume "dasdc1" successfully created
pvcreate -- physical volume "dasdd1" successfully created
pvcreate -- physical volume "dasde1" successfully created
pvcreate -- physical volume "dasdf1" successfully created
pvcreate -- physical volume "dasdg1" successfully created
```

Verify physical volumes with the **pvscan** command:

```
# pvscan
pvscan -- reading all physical volumes (this may take a while...)
pvscan -- inactive PV "/dev/dasdc1" is in no VG  [2.29 GB]
pvscan -- inactive PV "/dev/dasdd1" is in no VG  [2.29 GB]
pvscan -- inactive PV "/dev/dasde1" is in no VG  [2.29 GB]
pvscan -- inactive PV "/dev/dasdf1" is in no VG  [2.29 GB]
pvscan -- inactive PV "/dev/dasdg1" is in no VG  [2.29 GB]
pvscan -- tot: 5 [11.46 GB] / in use: 0 [0] / in no VG: 5 [11.46 GB]
```

Create the volume group and verify

The volume group name `datavg` is created with the **vgcreate** command. When it completes, you can see it is added to the `/dev/` directory. The size of about 11.5GB is shown with the **vgdisplay** command:

```
# vgcreate datavg /dev/dasd[cdefg]1
vgcreate -- INFO: using default physical extent size 4 MB
vgcreate -- INFO: maximum logical volume size is 255.99 Gigabyte
vgcreate -- doing automatic backup of volume group "datavg"
vgcreate -- volume group "datavg" successfully created and activated
#  ls -ld /dev/datavg
dr-xr-xr-x    2 root     root     72 Jan 16 14:29 /dev/datavg/
# ls -l /dev/datavg
crw-r-----    1 root     disk     109,   0 Jan 16 14:06 group
# vgdisplay datavg | grep Size
MAX LV Size          255.99 GB
VG Size              11.43 GB
PE Size              4 MB
Alloc PE / Size      0 / 0
Free  PE / Size      2925 / 11.43 GB
```

If you need a logical volume greater than 256GB, you need to raise the default PE size of 4MB.

Create a striped logical volume using most of the volume group

Performance can be greatly increased with a striped logical volume. However, when striping is used, this logical volume cannot be extended via the **lvextend** command, so be sure to make it large enough up front.

```
# lvcreate --stripes 5 --size 11G -n lv1 /dev/datavg
lvcreate -- INFO: using default stripe size 16 KB
lvcreate -- rounding to stripe boundary size
lvcreate -- doing automatic backup of "datavg"
lvcreate -- logical volume "/dev/datavg/lv1" successfully created
# lvdisplay /dev/datavg/lv1
--- Logical volume ---
LV Name                /dev/datavg/lv1
VG Name                datavg
LV Write Access        read/write
LV Status              available
LV #                   1
# open                 2
LV Size                11 GB
Current LE             2818
Allocated LE           2818
```

```
Stripes                5
...
# vgdisplay datavg | grep Size
MAX LV Size           255.99 GB
VG Size               11.43 GB
PE Size               4 MB
Alloc PE / Size       2818 / 11 GB
```

Create a journalled file system and mount the logical volume

The **mkreiserfs** command makes a reiser journalled file system out of the logical volume. Then a mount point is created in the directory /data and the logical volume is mounted:

```
# mkreiserfs /dev/datavg/lvl
...
# mkdir /data
# mount /dev/datavg/lvl /data
# df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/dasda1           2.3G  1.3G  1.1G  55% /
shmfs                  62M     0   62M   0% /dev/shm
/dev/datavg/lvl      11.0G   33M 11.0G   1% /data
```

Make a Samba share of the directory

Add the following share to the /etc/samba/smb.conf file:

```
[data]
        path = /data
        read only = no
```

Set the LVM to come up at IPL (boot) time

Back up and modify the file /etc/zipl.conf to include the new DASD. Run the **zipl** command to write the new parameter file to the boot sector of boot (in this case, the root) file system:

```
# cd /etc
# cp zipl.conf zipl.conf.orig
# vi zipl.conf  --> add DASD 200-204
# cat zipl.conf
...
[ipl]
target=/boot/zipl
image=/boot/kernel/image
ramdisk=/boot/initrd
parameters="dasd=100-101,200-204 root=/dev/dasda1"
...
# zipl
```

Now the DASD at addresses 200-204 will be recognized at boot time and assigned to the files /dev/dasdc to /dev/dasdg. Modify the /etc/fstab file to mount the logical volume over the directory /data at boot time:

```
# cp fstab fstab.orig
# vi /etc/fstab                       # and add a line
# cat /etc/fstab
/dev/dasda1         /           reiserfs   defaults            1 1
/dev/datavg/lvl     /data       reiserfs   acl                 0 2
/dev/dasdb1         swap        swap       pri=42              0 0
devpts              /dev/pts    devpts     mode=0620,gid=5     0 0
proc                /proc       proc       defaults            0 0
```

Test a reboot with the **shutdown** command. The logical volume reiser file system should come up mounted over the directory /data/.

```
# shutdown -r now
```

More details on LVM can be found on the SuSE Web site in the paper *LVM - Logical Volume Manager* at:

```
http://www.suse.de/us/whitepapers/lvm/
```

3.3 Setting up winbind to use DCs for authentication

Winbind authentication is briefly described in 2.5, “Authentication, authorization and related function” on page 32. In this scenario, there is a Windows 2000 server on the LAN in mixed mode with the DNS name lccwin2k.pok.ibm.com and the domain name POKLCC. Samba is set up to authenticate users against it.

Stop Samba if it is running:

```
# rcwinbind stop
Samba WINBIND daemon not configured in /etc/samba/smb.conf.          skipped
# rcsmb stop
Shutting down Samba classic SMB daemon                               done
# rcnmb stop
Shutting down Samba classic NMB daemon                               done
```

Make a back up copy of the smb.conf file and modify the following parameters:

```
# cd /etc/samba
# cp smb.conf smb.conf.orig
# vi smb.conf  --> make the following changes:
workgroup = POKLCC               // the NT/AD domain name
netbios name = PBC99241          // the Linux DNS name (by convention)
security = DOMAIN
encrypt passwords = Yes
password server = *
winbind uid = 10000-20000
winbind gid = 10000-20000
winbind separator = +
```

Create an lmhosts file that associates the Windows 2000 server name (LCCWIN2K) and domain name (POKLCC) with its IP address.

```
# cd /etc/samba
# vi lmhosts  --> add two lines
9.117.73.54 LCCWIN2K
9.117.73.54 POKLCC
```

Verify the NSS winbind shared library is available:

```
# locate libnss_winbind.so
/lib/libnss_winbind.so
/lib/libnss_winbind.so.2
```

Backup and modify the /etc/nsswitch.conf file to use the shared library.

```
# cd /etc
# cp nsswitch.conf nsswitch.conf.orig
# vi nsswitch.conf  --> and add "winbind" to the passwd and group lines:
passwd: files winbind
group:  files winbind
```

Stop the name service caching daemon so it will not look for names in its cache. Winbind has its own cache, so **nscd** should not be run in conjunction with **winbindd**.

```
# /etc/init.d/nscd stop
Shutting down Name Service Cache Daemon                              done
# chkconfig nscd
nscd  on
# chkconfig nscd off
```

There has to be a machine (computer) trust created on the Windows server to allow the Samba server to join the domain. You can also create one on the fly with the `-U administrator` parameter to the **smbpasswd** command, but you must have the `administrator` password.

Join the POKLCC domain via the **smbpasswd** command with the **-j** and **-r** flags. Watch for the message **"**`Joined domain POKLCC`**"** (it is important to get this message. If you cannot join the domain, the computer may need to be reset on the Windows 2000 server)

```
# smbpasswd -j POKLCC -r LCCWIN2K
2002/07/30 09:55:56 : change_trust_account_password: Changed password for domain POKLCC.
Joined domain POKLCC.
```

When Samba joins the domain the following happens:

- The machine trust account password in the domain's SAM database is changed.
- A file called `secrets.tdb` is created that contains the machine SID for the newly joined samba server and the machine trust account password for the samba server.

Start Samba and verify it is running

```
# rcnmb start
Starting Samba classic NMB daemon                                    done
# rcsmb start
Samba SMB: Waiting for cupsd to get ready                            done
Starting Samba classic SMB daemon                                    done
# rcwinbind start
Starting Samba classic WINBIND daemon                                done
```

List the users and groups in the POKLCC domain with the command **wbinfo**:

```
# wbinfo -u
POKLCC+Administrator
POKLCC+Guest
...
# wbinfo -g
POKLCC+Domain Admins
POKLCC+Domain Users
...
```

Verify your trust with the Windows using the command **wbinfo -t**.

```
# wbinfo -t
Secret is good
```

Identify a domain user with the **id** command. For example:

```
# id POKLCC+user1
uid=10008(POKLCC+user1) gid=10001(POKLCC+Domain Users) groups=10001(POKLCC+Domain Users)
```

Winbind should now be working. You should be able to get a file share using the credentials of a user in the domain `POKLCC`.

3.4 Setting up OpenLDAP

This scenario shows how to do authentication with OpenLDAP that comes built in SuSE SLES-8. Integrating Samba with OpenLDAP is discussed in “Authentication via an LDAP server” on page 34 The overall steps are:

- Configure OpenLDAP
- Start OpenLDAP
- Create a Directory Information Tree (DIT)
- View the DIT via an LDAP browser
- Configure PAM to use LDAP for login and ssh
- Configure NSS to use LDAP
- Set up some LDAP wrapper scripts
- Use some LDAP client GUI tools to view the LDAP data

Configure and start OpenLDAP

Verify OpenLDAP is installed on SLES-8:

```
# rpm -qa | grep ldap
openldap2-client-2.1.4-26
nss_ldap-199-30
pam_ldap-150-44
openldap2-2.1.4-26
openldap2-devel-2.1.4-26
```

Configure the OpenLDAP server configuration file. First the file is backed up and (optionally) comments are removed via the **egrep** command:

```
# cd /etc/openldap
# cp slapd.conf slapd.conf.orig
# egrep -v '^$|^#' slapd.conf.orig > slapd.conf
# vi slapd.conf    --> add schemas, set suffix, rootdn and rootpw
# cat slapd.conf
include         /etc/openldap/schema/core.schema
include         /etc/openldap/schema/cosine.schema
include         /etc/openldap/schema/nis.schema
include         /etc/openldap/schema/misc.schema

pidfile         /var/run/slapd/slapd.pid
argsfile        /var/run/slapd/slapd.args

access to attrs=userPassword by self write
  by anonymous auth
  by * none
access to * by * read

database        bdb
suffix          "dc=poklcc,dc=ibm,dc=com"
rootdn          "cn=Manager,dc=poklcc,dc=ibm,dc=com"
rootpw          linux123
directory       /var/lib/ldap

index   objectClass     eq
```

Configure the OpenLDAP client configuration file, `/etc/openldap/ldap.conf`. Because it contains only comments, an empty file can be started with:

```
# mv ldap.conf ldap.conf.orig
# vi ldap.conf   --> add two lines
host    localhost
```

```
BASE    dc=poklcc,dc=ibm,dc=com
```

Start OpenLDAP

Start the OpenLDAP service. Note the files that get created in the directory `/var/lib/ldap/`:

```
# rcldap status
Checking for service ldap:                             unused
# ls /var/lib/ldap
# rcldap start
Starting ldap-server                                   done
# ls /var/lib/ldap
__db.001  __db.003  __db.005   id2entry.bdb
__db.002  __db.004  dn2id.bdb  log.0000000001
```

Set the LDAP service to restart at boot time:

```
# chkconfig ldap
ldap  off
# chkconfig ldap on
# chkconfig ldap
ldap  on
```

Create a Directory Information Tree (DIT)

Create a small LDAP Interchange Format (ldif) file with a top level structure and one users of class posixAccount named `ldapuser1`:

```
# cat initial.ldif
dn: dc=poklcc,dc=ibm,dc=com
dc: poklcc
objectClass: top
objectClass: domain

dn: ou=People,dc=poklcc,dc=ibm,dc=com
ou: People
objectClass: top
objectClass: organizationalUnit

dn: ou=Group,dc=poklcc,dc=ibm,dc=com
ou: Group
objectClass: top
objectClass: organizationalUnit

dn: ou=Users,dc=poklcc,dc=ibm,dc=com
ou: Group
objectClass: top
objectClass: organizationalUnit

dn: uid=ldapuser1,ou=People,dc=poklcc,dc=ibm,dc=com
uid: ldapuser1
cn: ldapuser1
objectClass: account
objectClass: posixAccount
objectClass: top
objectClass: shadowAccount
loginShell: /bin/bash
uidNumber: 1000
gidNumber: 100
homeDirectory: /home/ldapuser1
```

Add the entries in the ldif file:

```
# ldapadd -x -h localhost -D "cn=Manager,dc=poklcc,dc=ibm,dc=com" -f initial.ldif  -W
Enter LDAP Password:
adding new entry "dc=poklcc,dc=ibm,dc=com"

adding new entry "ou=People,dc=poklcc,dc=ibm,dc=com"

adding new entry "ou=Group,dc=poklcc,dc=ibm,dc=com"

adding new entry "ou=Users,dc=poklcc,dc=ibm,dc=com"

adding new entry "uid=ldapuser1,ou=People,dc=poklcc,dc=ibm,dc=com"
```

Look at the entries via the command line:

```
# ldapsearch -x
...
result: 0 Success

# numResponses: 7
# numEntries: 6
```

View the DIT via an LDAP browser

There are a couple of GUI browsers worth mentioning:

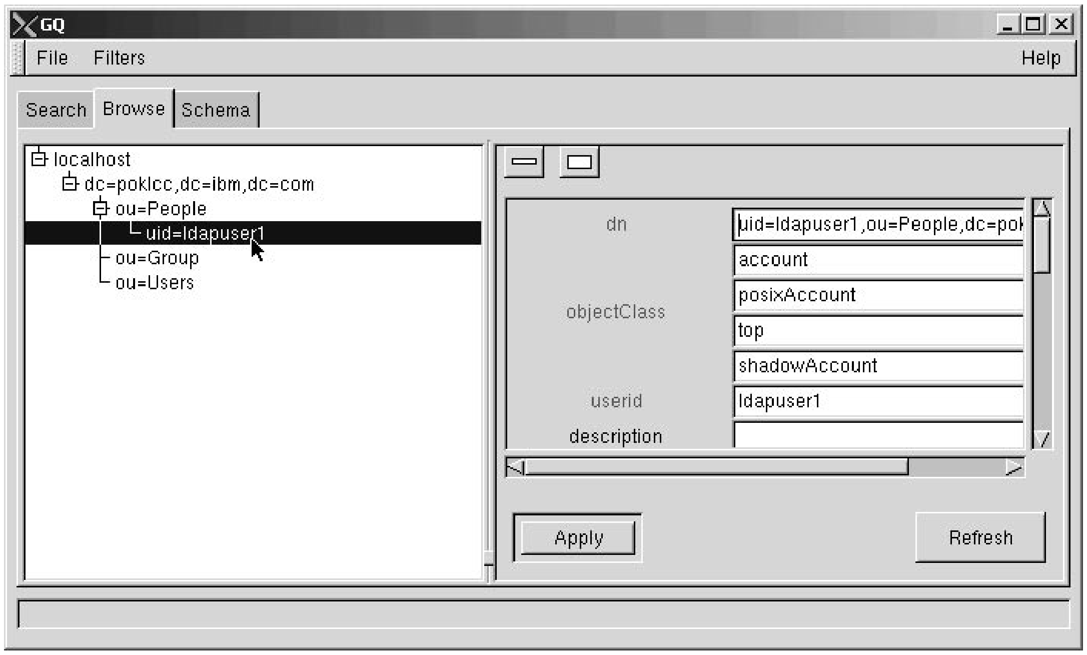
gq	LDAP browser built and shipped with SLES-8
web2ldap	A swiss-army knife for accessing/manipulating LDAP servers which must be built on SLES-8

Look at the entries via the GUI LDAP browser named **gq**:

- If you have a Windows desktop, start an X server (e.g Hummingbird eXceed).
- Set the `DISPLAY` environment variable and start **gq** which comes standard with SLES-8.

```
# export DISPLAY=<your.IP.address>:0
# gq &
```

Click on the browse tab to see a view of the DIT:



dn	uid=ldapuser1,ou=People,dc=pok
objectClass	account
	posixAccount
	top
	shadowAccount
userid	ldapuser1
description	

Figure 3-1 Screen shot of gq showing the DIT

Configure PAM to use LDAP for login and ssh

Warning - Modifying PAM configuration files can easily result in a system that you cannot log on to. By enabling access to both **telnet** and **ssh**, you reduce that chance. You may want **telnet** disabled for security reasons.

Enable telnet:

```
# cd /etc
# vi inetd.conf   --> uncomment the telnet line
telnet  stream  tcp     nowait  root    /usr/sbin/tcpd  in.telnetd
# rcinetd restart
Shutting down inetd                                             done
Starting inetd                                                  done
```

Test telnetting in with a non-root user ID.

Configure PAM so both local and LDAP users can telnet in:

```
# cd /etc/pam.d
# cp login login.orig
# vi login --> make the following changes
# cat login
#%PAM-1.0
auth    required        pam_securetty.so
auth    required        pam_nologin.so
auth    required        pam_env.so
auth    required        pam_mail.so
auth    sufficient      pam_ldap.so
auth    required        pam_unix2.so    nullok use_first_pass
account required        pam_unix2.so
password required       pam_pwcheck.so  nullok
password required       pam_unix2.so    nullok use_first_pass use_authtok
session required        pam_unix2.so    none     # debug or trace
session required        pam_limits.so
```

Before proceeding, test telnet to local users with no password, bad password and good password. Then test telnet to LDAP users with no password, bad password and good password.

Configure PAM so both local and LDAP users can ssh in:

```
# cp sshd sshd.orig
# vi sshd  --> make similar changes
# cat sshd
#%PAM-1.0
auth required   pam_nologin.so
auth required   pam_env.so
auth    sufficient      pam_ldap.so
auth    required        pam_unix2.so    nullok use_first_pass
account required        pam_unix2.so
account required        pam_nologin.so
password required       pam_pwcheck.so
password required       pam_unix2.so    use_first_pass use_authtok
session required        pam_unix2.so    none     # trace or debug
session required        pam_limits.so
```

Restart **sshd** to reread the `/etc/pam.d/sshd` file:

```
# rcsshd restart
Shutting down SSH daemon                                             done
Starting SSH daemon
```

Test ssh local users with no password, bad password and good password and then LDAP users with no password, bad password and good password.

Set the name service switch (NSS) to use LDAP

Backup and modify the file `/etc/nsswitch.conf` to use LDAP:

```
# cd /etc
# cp nsswitch.conf nsswitch.conf.orig
# vi nsswitch.conf   --> modify the next two lines
passwd: files ldap
group:  files ldap
```

Use the **id** command to query the one LDAP user that was added:

```
# id ldapuser1
uid=1000(ldapuser1) gid=100(users) groups=100(users)
```

Set up some LDAP wrapper scripts (optional)

Some wrapper scripts are written so LDAP commands such as **ldappasswd** and **ldapmodify** are easier to use. See Appendix , “Reference scripts” on page 84. The following scripts are written:

ldel	Invoke **ldapdelete** more quickly
ldifadd	Invoke **ldapadd** using an `ldif` file more quickly
ldifmod	Invoke **ldapmodify** using an `ldif` file more quickly
lpasswd	Invoke **ldappasswd** more quickly
luid	Invoke **ldappasswd** more quickly

Set the LDAP passwords

The `ldif` file that created the initial DIT did not have encrypted passwords. This can be seen that no password attribute is shown when a user is displayed with the **ldapsearch** command:

Set the password of `ldapuser1` with the **lpasswd** script:

```
# lpasswd ldapuser1
New password:
Re-enter new password:
Enter bind password:
Result: Success (0)

# ldapsearch -x uid=ldapuser1
...
userPassword:: e1NNRDV9QTJrbVgONC9xaVdDUWp4RVdlVGVWdTVJZnJFPQ==
...
```

Create a home directories for the user:

```
# cd /home
# mkdir ldapuser1
# chown ldapuser1.users ldapuser1
```

You should now have a minimal LDAP server set up.

3.5 Setting up Samba to use OpenLDAP

This scenario builds on the previous two. The overall steps are:

- Add the samba schema to the LDAP server
- Configure the Samba library to use LDAP
- Modify the Samba configuration file to use LDAP

Add the samba schema to the LDAP server

Copy the file `samba.schema` to `/etc/opendlap/schema/` and add two schemata; `inetorgperson` and `samba`:

```
# cd /etc/openldap
# locate samba.schema
/usr/share/doc/packages/samba/examples/LDAP/samba.schema
# cp /usr/share/doc/packages/samba/examples/LDAP/samba.schema schema
# vi slapd.conf    --> add schemas, set suffix, rootdn and rootpw
# cat slapd.conf
include         /etc/openldap/schema/core.schema
include         /etc/openldap/schema/cosine.schema
include         /etc/openldap/schema/nis.schema
include         /etc/openldap/schema/misc.schema
include         /etc/openldap/schema/inetorgperson.schema
include         /etc/openldap/schema/samba.schema
...
```

Restart LDAP:

```
# rcldap restart
Shutting down ldap-server                                            done
Starting ldap-server                                                 done
```

Configure Samba library to use LDAP

In SLES-8, Samba is configured to run in one of two modes, LDAP or “classic”, which is the default. Change Samba to run in LDAP mode by modifying the file `/etc/sysconfig/samba`:

```
# cd /etc/sysconfig/
# vi samba  --> change "classic" to "ldap"
# grep SAM samba
#SAMBA_SAM="classic"
```

```
SAMBA_SAM="ldap"
```

Run **SuSEconfig** to pick up Samba changes:

```
# SuSEconfig --module samba
Starting SuSEconfig, the SuSE Configuration Tool...
Running module samba only
Reading /etc/sysconfig and updating the system...
Executing /sbin/conf.d/SuSEconfig.samba...
Finished.
```

Modify the Samba configuration file to use LDAP

There are a number of parameters specific to LDAP that must be set in the `smb.conf` file. Setting the variable `ldap server = localhost` means that Samba and LDAP are running on the same machine. Of course this does not have to be the case.

```
# cd /etc/samba
# cp smb.conf smb.conf.orig
# vi smb.conf  --> add the following parameters to the global section
#       LDAP parameters
        ldap admin dn = cn=Manager,dc=poklcc,dc=ibm,dc=com
        ldap server = localhost
        ldap suffix = dc=poklcc,dc=ibm,dc=com
```

The next two lines will change depending on whether you are using TLS security or not. If you are not using TLS, use the lines:

```
        ldap port = 389
        ldap ssl = no
```

If you are using TLS, use the lines:

```
        ldap port = 636
        ldap ssl = yes
```

Stop the two Samba daemons if they are running and restart them:

```
# rcnmb status
Checking for Samba classic NMB daemon                              running
# rcsmb status
Checking for Samba classic SMB daemon                              running
# rcsmb stop
Shutting down Samba classic SMB daemon                             done
# rcnmb stop
Shutting down Samba classic NMB daemon                             done
# rcnmb start
Starting Samba ldap NMB daemon                                     done
# rcsmb start
Starting Samba ldap SMB daemon                                     done
```

Using the **smbpasswd** command to create new users. Because the LDAP libraries are being used, the **smbpasswd** command sets the passwords in LDAP, not in the `smbpasswd` file.

First set the LDAP Manager password:

```
# smbpasswd -w linux123
Setting stored password for "cn=Manager,dc=poklcc,dc=ibm,dc=com" in secrets.tdb
```

Add two Samba users, smbuser1 and smbuser2 with the **useradd** command and create home directories (**Note**: to automate this, the global `add user script` parameter can be set in the smb.conf file and pointed to a script to add the user):

```
# useradd smbuser1
# useradd smbuser2
# cd /home
# mkdir smbuser1 smbuser2
# chown smbuser1.users smbuser1
# chown smbuser2.users smbuser2
```

Verify the entries were added:

```
# grep smb /etc/passwd /etc/shadow
/etc/passwd:smbuser1:x:1002:100::/home/smbuser1:/bin/bash
/etc/passwd:smbuser2:x:1003:100::/home/smbuser2:/bin/bash
/etc/shadow:smbuser1:!:12166:0:99999:7:::
/etc/shadow:smbuser2:!:12166:0:99999:7:::
```

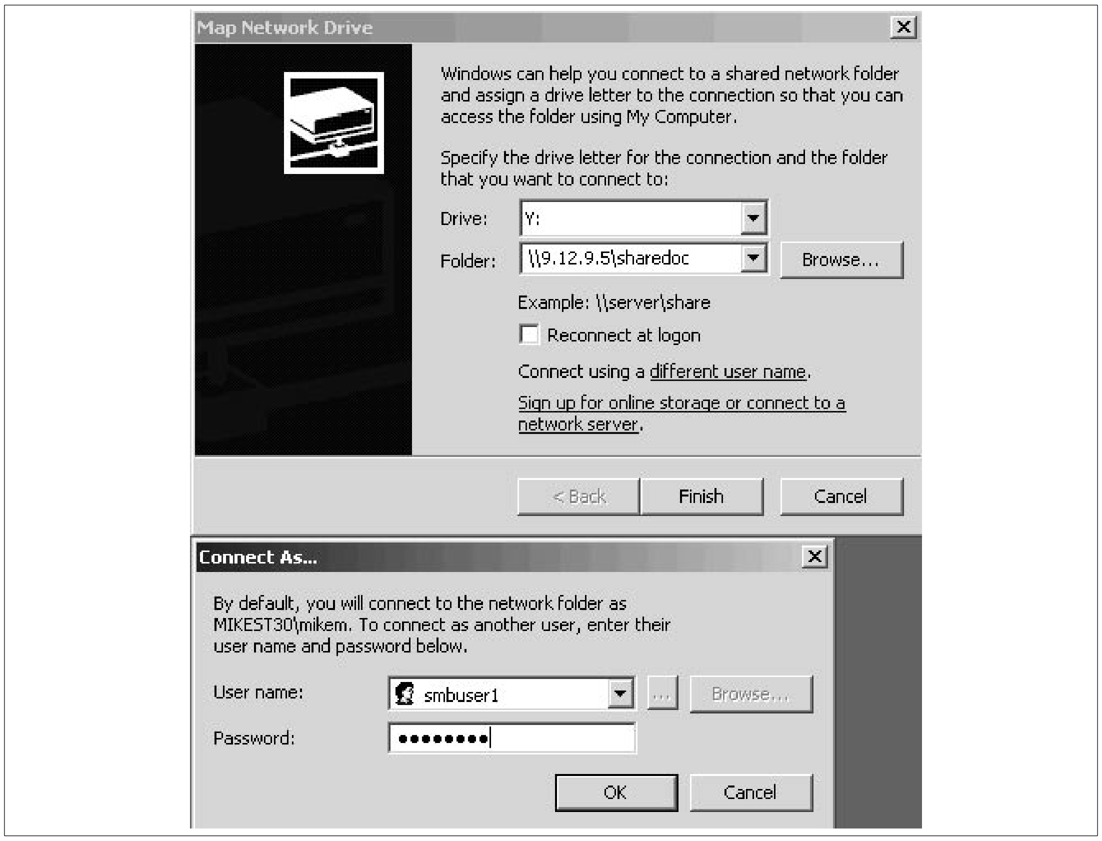
Add the users to LDAP:

```
# smbpasswd -a smbuser1
New SMB password:
Retype new SMB password:
LDAP search "(&(uid=smbuser1)(objectclass=sambaAccount))" returned 0 entries.
Added user smbuser1.
# smbpasswd -a smbuser2
New SMB password:
Retype new SMB password:
LDAP search "(&(uid=smbuser2)(objectclass=sambaAccount))" returned 0 entries.
Added user smbuser2.
```

Verify the entries were added:

```
# ldapsearch -x uid=smbuser1
...
dn: uid=smbuser1,dc=poklcc,dc=ibm,dc=com
uid: smbuser1
pwdLastSet: 1051207002
logonTime: 0
logoffTime: 2147483647
kickoffTime: 2147483647
pwdCanChange: 0
pwdMustChange: 2147483647
rid: 3004
primaryGroupID: 1201
lmPassword: C0470EE0F7D0C608C2265B23734E0DAC
ntPassword: E107A5C2EA2704CE18D2DA10537A02F5
acctFlags: [UX         ]
objectClass: sambaAccount
objectClass: account
...
# ldapsearch -x uid=smbuser2
...
```

Get a share from a Windows desktop. You can use the **Connect using a Different User Name** dialog as shown in to specify the `smbuser1` user.


Map Network Drive

Windows can help you connect to a shared network folder and assign a drive letter to the connection so that you can access the folder using My Computer.

Specify the drive letter for the connection and the folder that you want to connect to:

Drive: Y:

Folder: \\9.12.9.5\sharedoc

Example: \\server\share

Reconnect at logon

Connect using a different user name.

Sign up for online storage or connect to a network server.

Connect As...

By default, you will connect to the network folder as MIKEST30\mikem. To connect as another user, enter their user name and password below.

User name: smbuser1

Password: ••••••••

Figure 3-2 Mapping a drive and Connect using a different user name

3.6 Setting up basic print serving with CUPS

A print server must be set up before Samba can offer printers to Windows clients. On SLES-8 the default print server is CUPS. The overall steps are:

- Configure CUPS to start at boot time
- Modify the CUPS configuration file and start CUPS
- Add some printers to CUPS and test from Linux

Configure CUPS to start at boot time

Verify that CUPS is installed with the **rpm** command:

```
# rpm -qa | grep cups
cups-libs-1.1.15-40
cups-client-1.1.15-40
cups-1.1.15-40
cups-drivers-1.1.15-45
cups-drivers-stp-1.1.15-45
cups-devel-1.1.15-40
```

By default, CUPS is not started with the system. Set it to start at boot time with the **chkconfig** command:

```
# chkconfig cups
cups  off
# chkconfig cups on
# chkconfig cups
```

```
cups  on
```

Modify the CUPS configuration file and start CUPS

Make a backup of the file `/etc/cups/cupsd.conf` and optionally remove the 736 lines of comments (it's usually easier to read an uncommented configuration file than a heavily commented one):

```
# cd /etc/cups
# cp cupsd.conf cupsd.conf.orig
# egrep -v '^#|^$' cupsd.conf.orig > cupsd.conf
```

If you want to be able to access the CUPS Web server from any client, comment out the `Deny From All` lines. If you want better security, just allow those hosts from which you will be accessing the CUPS Web server:

```
# vi cupsd.conf   -->> comment out the "Deny from all" lines
# cat cupsd.conf
DocumentRoot /usr/share/cups/doc/
LogLevel info
Port 631
<Location />
Order Deny,Allow
#Deny From All
Allow From 127.0.0.1
Allow From 127.0.0.2
</Location>
<Location /admin>
AuthType Basic
AuthClass System
Order Deny,Allow
#Deny From All
Allow From 127.0.0.1
</Location>
```

Start CUPS:

```
# rccups status
Checking for cupsd:                                                  unused
# rccups start
Starting cupsd                                                       done
# rccups status
Checking for cupsd:                                                  running
```

Add some printers to CUPS and test from Linux

Printers can be added and maintained in at least three ways:

- The CUPS Web interface - a mini-Web server listening on well-known IPP port 631
- YaST2
- Command line - using the commands associated with CUPS

This example will use the CUPS commands. To add a printer, the **`lpadmin`** command is used. Before you do this, you need the following information:

- The printer's TCP/IP address
- The LPD queue name
- The Printer PostScript Definition (PPD) file under the directory `/usr/share/cups/model`

The TCP/IP address can be obtained from the printer itself or from an administrator. The LPD queue name can often be obtained from the printer's documentation. In this example the queue name for an IBM InfoPrint 40 printer is `PASS` which was obtained from the manual

Ethernet and Token Ring Configuration Guide for the InfoPrint 40. This manual was found on the IBM printing Web site:

```
http://www.printers.ibm.com/
```

The PPD file was also found on the IBM printing Web site after the driver package was downloaded and unzipped to disk. Before invoking the **lpadmin** command, the PPD file was copied and then compressed:

```
# cp IBM43322.PPD /usr/share/cups/model/IBM
# gzip /usr/share/cups/model/IBM/IBM43322.PPD
```

The file /etc/cups/ppds.dat must be rebuilt to pick up the new PPD file. This can be done by simply restart cups:

```
# rccups restart
Shutting down cupsd                                                   done
Starting cupsd                                                        done
```

The PPD file is now named `/usr/share/cups/model/IBM/IBM43322.PPD.gz` and is stored in /etc/Then the printer is added with the **lpadmin** command:

```
# lpadmin -p f58bydoor -E -v lpd://9.56.41.237/PASS -m IBM/IBM43322.PPD.gz
```

The flags used have the following meanings:

-p	Printer will be named `f58bydoor`
-E	Printer will be enabled (ironically the default is that a printer is disabled)
-v	Device URI to use the LPD queue //<host>/<queue name>. This means that the well-known lpd port 515 built into the printer's operating system will be used.
-m	The PPD file to use relative to the directory `/usr/share/cups/model/`

After the command completes, the `/etc/cups/printers.conf` file should be modified and a PPD file should be created in the directory `/etc/cups/ppd`:

```
# cat printers.conf
# Printer configuration file for CUPS v1.1.15
# Written by cupsd on Wed Apr 23 12:49:41 2003
<DefaultPrinter f58bydoor>
Info f58bydoor
DeviceURI lpd://9.56.41.237
State Idle
Accepting Yes
JobSheets none none
QuotaPeriod 0
PageLimit 0
KLimit 0
</Printer>
# ls /etc/cups/ppd
```

In this scenario, the PPD file was not created, so even though the printer was added, it will not have the proper attributes because it will not have a PPD file (this could be considered a bug that **lpadmin** does not report not copying a PPD file).

The problem was that CUPS was not restarted and thus the existence of the PPD file was not added to the small database in the file `/etc/cups/ppds.dat`. Note the file grows after CUPS is restarted:

```
# wc ppds.dat
     38    8698 1140784 ppds.dat
# rccups restart
Shutting down cupsd                                                   done
Starting cupsd                                                        done
```

```
# wc ppds.dat
     38    8701 1141440 ppds.dat
```

Deleted the printer with the **lpadmin -x** flag and add it back again:

```
# lpadmin -x f58bydoor
# lpq
no entries
# lpadmin -p f58bydoor -E -v lpd://9.56.41.237 -m IBM/IBM43322.PPD.gz
# lpq
f58bydoor is ready
no entries
```

Again check that the PPD file was added to the `ppd/` directory:

```
# ls /etc/cups/ppd
f58bydoor.ppd
```

This time it has been added. Note the file name corresponds to the printer name because each printer requires a PPD file to be associated with it.

Add a cover sheet attribute. This will cause a cover sheet to be printed at the start of your print job:

```
# lpadmin -p pok72far -o job-sheets-default=standard
```

If you are having problems with CUPS, the following command monitors the CUPS error log:

```
# tail --follow /var/log/cups/error_log
```

Now your printer should be set up on Linux. Print a file from Linux and verify it is working.

3.7 Setting up Samba to use CUPS

This scenario builds on the previous one. The following `smb.conf` parameters are pertinent to printing - they should all be set in the default `/etc/samba/smb.conf` file:

```
[global]
        ...
        printcap name = cups
        printer admin = mikem
        printing = cups
[printers]
        path = /var/lib/samba/printers
        create mask = 0600
        printable = Yes
        browseable = No
[print$]
        path = /var/lib/samba/drivers
        write list = mikem
        create mask = 0664
        directory mask = 0775
```

So by default, printing with CUPS is set up on SLES-8. All printers defined to CUPS should now be visible as resources through Samba. Type `\\<your.server.IP.address>` in a Windows Explorer dialog and click on `Printers`.

Start Samba if it is not already running:

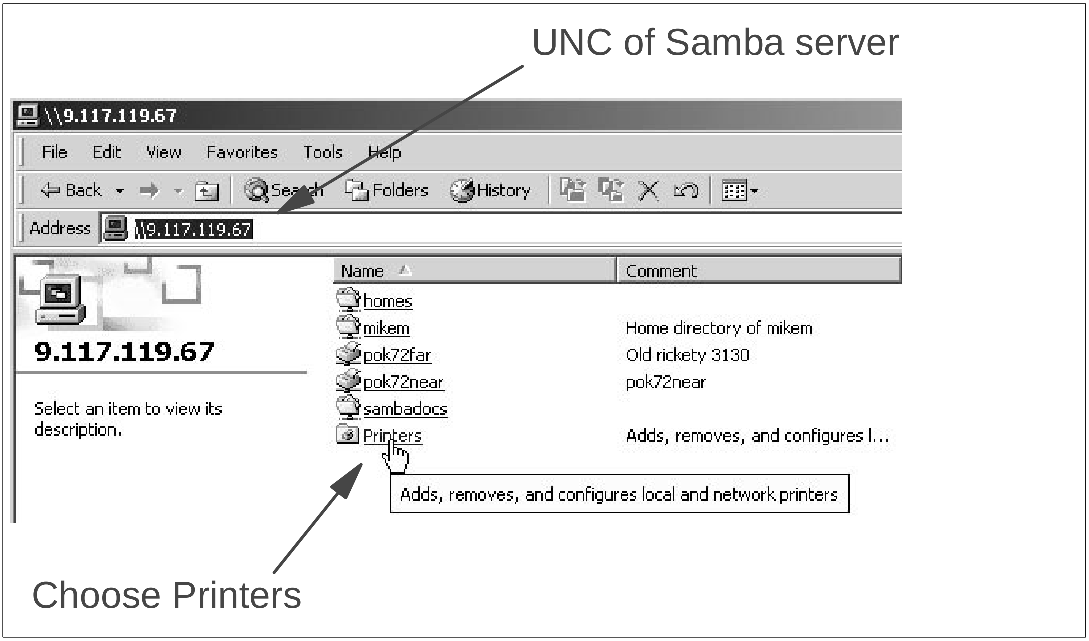
```
# rcnmb start
Starting Samba classic NMB daemon                                    done
```

```
# rcsmb start
Samba SMB: Waiting for cupsd to get ready                              done
Starting Samba classic SMB daemon                                      done
```

3.8 Enable automatic downloading of printer drivers

This scenario builds on the last two. You should have the drivers ready for the printer you want to make available.

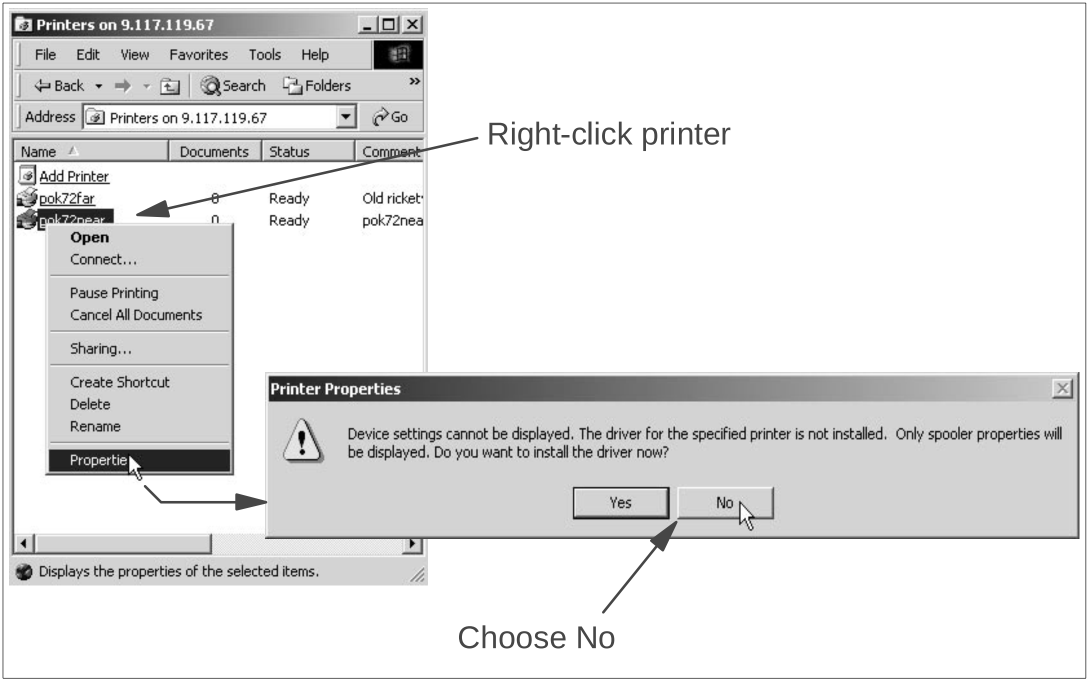
It is possible to set up the printer drivers from the Linux side. However, the scenario illustrates how to set up the drivers on Linux/Samba from the Windows side as this may have a better chance of getting the drivers set up correctly. Start by finding the resources that your Samba server has made available. Choose **Printers**.



Name	Comment
homes	
mikem	Home directory of mikem
pok72far	Old rickety 3130
pok72near	pok72near
sambadocs	
Printers	Adds, removes, and configures l...

Figure 3-3 Choosing Printers

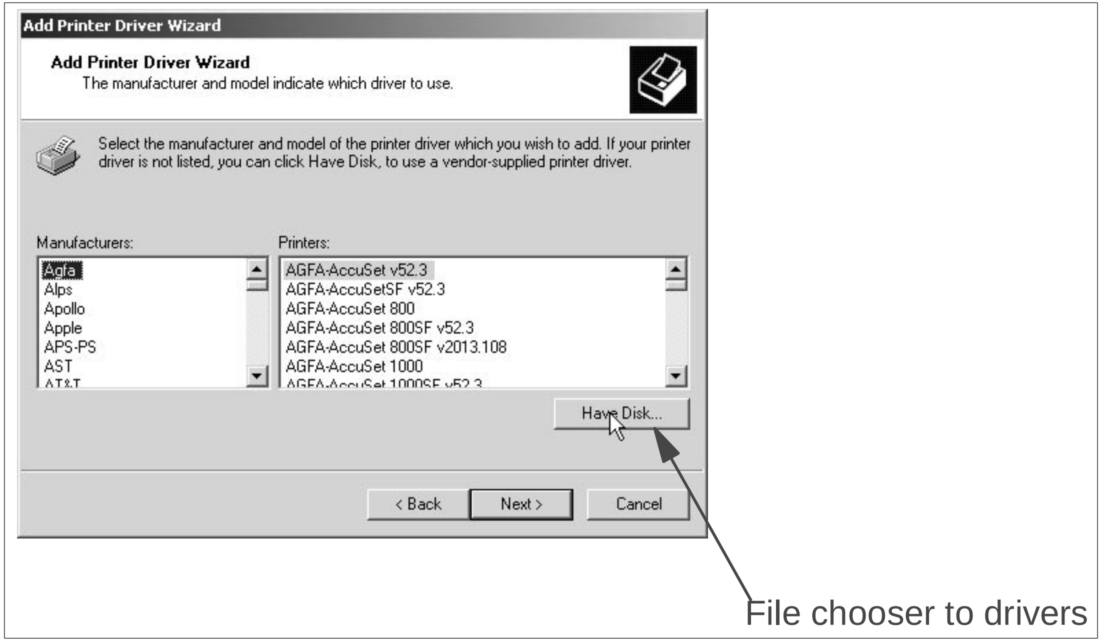
Assuming you have the special `[printers]` section, all printers defined in CUPS should be seen. Right-click the printer that you want to make drivers available for. Windows should detect that there are no drivers available for it. If you choose **Yes** to the question “Do you want install the driver now?”, the drivers will be installed on the local client. By choosing **No**, you start the *Add Printer Driver Wizard.*



Device settings cannot be displayed. The driver for the specified printer is not installed. Only spooler properties will be displayed. Do you want to install the driver now?

Figure 3-4 Choose properties of printer to upload drivers for

Choose **Have Disk** which will give you a file chooser so you can select the printer drivers



The manufacturer and model indicate which driver to use.

Select the manufacturer and model of the printer driver which you wish to add. If your printer driver is not listed, you can click Have Disk, to use a vendor-supplied printer driver.

Figure 3-5 Add Printer Driver Wizard

Windows will be looking for a .inf file with the printer model information.

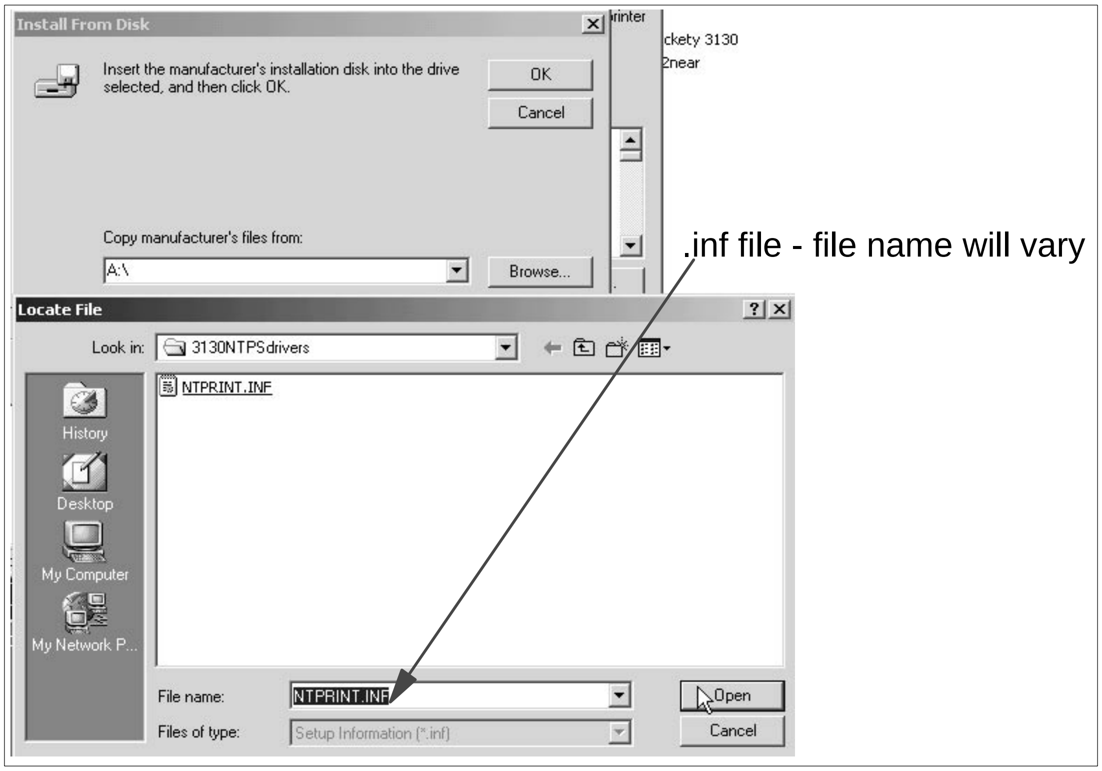
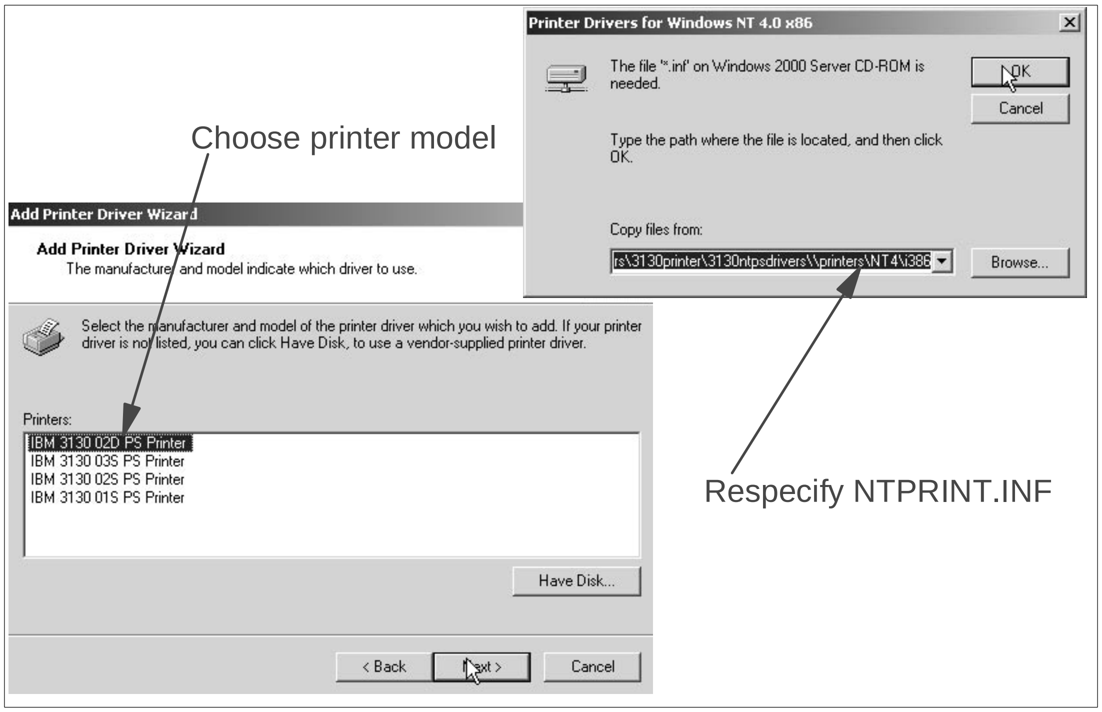

Install From Disk
Insert the manufacturer's installation disk into the drive selected, and then click OK.
OK
Cancel
Copy manufacturer's files from:
A:\
Browse...
Locate File
Look in: 3130NTPSdrivers
NTPRINT.INF
History
Desktop
My Computer
My Network P...
File name: NTPRINT.INF
Files of type: Setup Information (*.inf)
Open
Cancel
.inf file - file name will vary

Figure 3-6 Selecting drivers to upload

When you supply the correct .inf file, you should get a choice of your printer model(s).



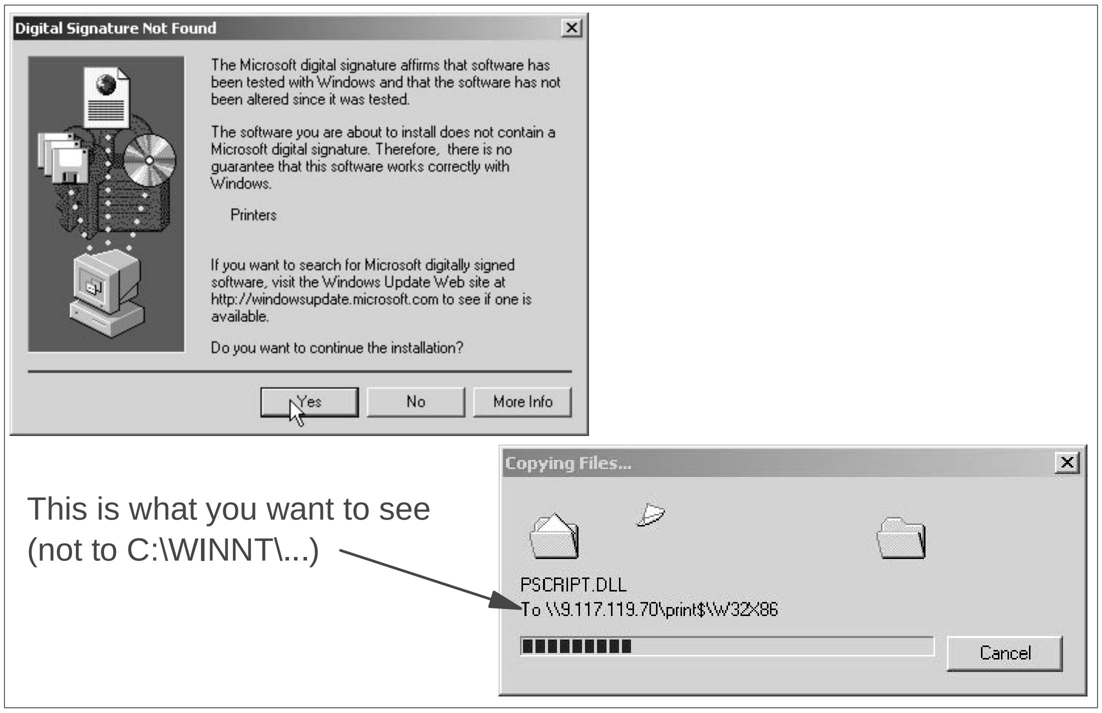
The file "*.inf' on Windows 2000 Server CD-ROM is needed.

Type the path where the file is located, and then click OK.

Select the manufacturer and model of the printer driver which you wish to add. If your printer driver is not listed, you can click Have Disk, to use a vendor-supplied printer driver.

Figure 3-7 Selecting printer model

When you select the printer model, you may get a complaint about a digital signature not being found. Select **Yes**. You should see the drivers being uploaded to the Linux server.


Digital Signature Not Found

The Microsoft digital signature affirms that software has been tested with Windows and that the software has not been altered since it was tested.

The software you are about to install does not contain a Microsoft digital signature. Therefore, there is no guarantee that this software works correctly with Windows.

Printers

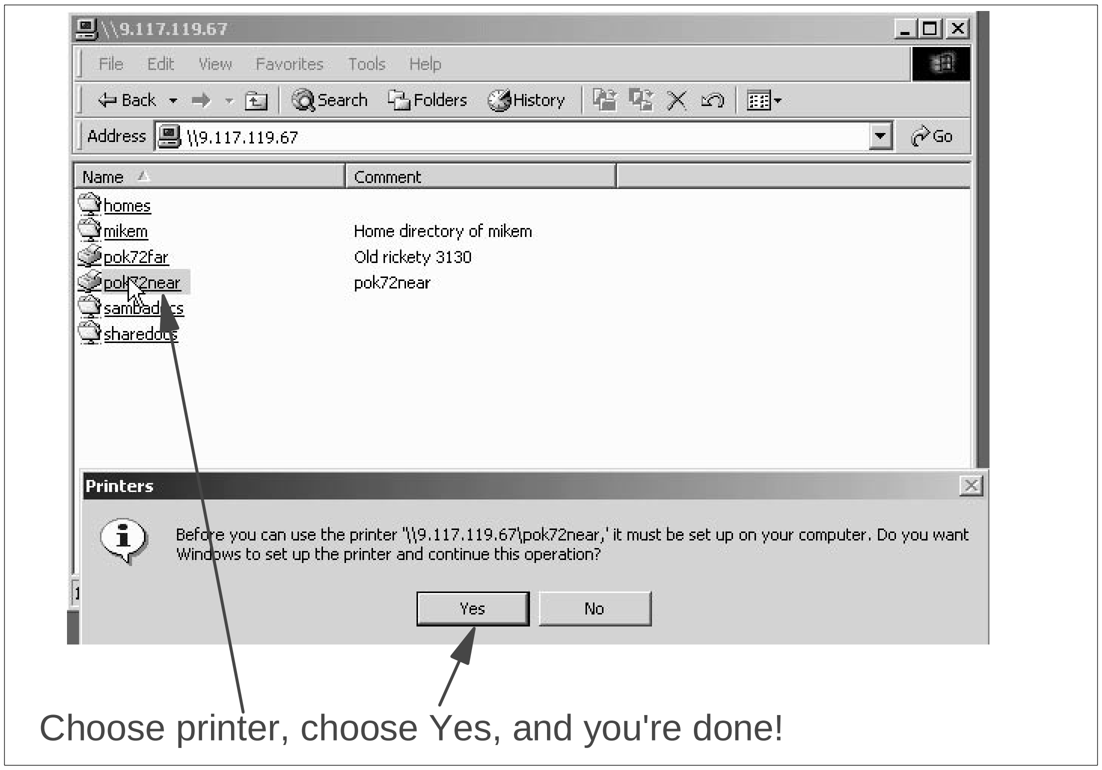
If you want to search for Microsoft digitally signed software, visit the Windows Update Web site at http://windowsupdate.microsoft.com to see if one is available.

Do you want to continue the installation?

PSCRIPT.DLL
To \\9.117.119.70\print$\W32X86

Figure 3-8 Drivers being uploaded

Now from any Windows client, you should be able to select the Linux/CUPS/Samba printer and get the following dialog. Choosing **Yes** to the question "*Do you want Windows to set up the printer and continue this operation?*" starts the process of automatically downloading printer drivers.


\\9.117.119.67
File Edit View Favorites Tools Help
Back Search Folders History
Address \\9.117.119.67
Go

Name	Comment
homes	
mikem	Home directory of mikem
pok72far	Old rickety 3130
pok72near	pok72near
sambadocs	
shareddocs	

Printers

Before you can use the printer '\\9.117.119.67\pok72near,' it must be set up on your computer. Do you want Windows to set up the printer and continue this operation?

Yes
No
Choose printer, choose Yes, and you're done!

Figure 3-9 Automatically downloading printer drivers

3.9 Setting up Samba as a time server

The Network Time Protocol (NTP) is commonly used to set the software clocks of operating systems very accurately. To be a time server, you only must set your clock accurately. When your server sets its time against a stratum 1 or stratum 2 time server, your server can become a stratum 2 or stratum 3 time server for clients. You will need access to the Internet on the well-known ntp port 123 to set the software clock. The SMB protocol allows clients to set the time, via the **net time** command, against an SMB server, such as Linux running Samba.

Verify NTP client, xntp, is installed:

```
# rpm -qa | grep ntp
xntp-4.1.1-53
```

Set up the NTP service, **xntpd**, to start at boot time:

```
# chkconfig xntpd
xntpd  off
# chkconfig xntpd on
# chkconfig xntpd
xntpd  on
```

Choose two stratum 1 or stratum 2 time servers near you. For reference, see:

```
http://www.eecis.udel.edu/~mills/ntp/clock1a.html
```

Back up the original NTP configuration file, `/etc/ntp.conf`, then set up a new one with two of the time servers. For example:

```
# mv ntp.conf ntp.conf.orig
# vi ntp.conf  --> add the following lines
cat ntp.conf
server    clock.llnl.gov
server    tock.usno.navy.mil
driftfile /etc/ntp.drift        # path for drift file
logfile   /var/log/ntp          # alternate log file
```

Start xntp with the **rcxntpd** command:

```
# rcxntpd start
Try to get initial date and time via NTP from  clock.llnl.gov tock.usdoneavy.mil
Starting network time protocol daemon (NTPD)                         done
```

Wait at least 64 seconds and check that your system is talking to the time servers with the **ntpq** command, **peers** subcommand:

```
# ntpq
ntpq> peers
     remote           refid     st t when poll delay   offset  jitter
======================================================================
*ntp1.usno.navy. .USNO.          1 u   36   64 21.098   -4.321   2.270
+clock.via.net   .GPS.           1 u   32   64 79.086   -5.215   0.785
+dns.cit.cornell ntp0.usno.navy. 2 u   43   64 19.957   -7.438   1.823
ntpq> q
```

- A * in the first column indicates the time server that is being used
- a + indicates a good fall-back connection.

You should always have one *, and one or two + entries. Check the state of your new time server:

```
# ntptrace localhost
localhost: stratum 2, offset 0.000040, synch distance 0.02551
```

```
ntp1.usno.navy.mil: stratum 1, offset -0.009210, synch distance
  0.00018, refid 'USNO'
```

Now your server's software clock should be *very* accurate. If Samba is running, Windows clients can set their time via the **`net time`** command. This command can be put in a startup script or can be run from a DOS prompt:

```
C:\>net time \\9.12.9.5 /set /yes
Current time at \\9.12.9.5 is 4/28/2003 2:07 PM

The command completed successfully.
```

Additionally, Samba can advertise itself as a time server in browse lists via the `time server = Yes` parameter in the `smb.conf` file.

3.10 Setting up rsync for backup

A convenient feature of rsync is that it can maintain a complete copy of a file system or share. If the rsync daemon is set up on the Samba server and an rsync client is run nightly from another backup server, a view of the entire file system can be available from the day before. This is often convenient when a file is accidentally deleted or corrupted.

On the Samba server side

Set up for the share `/zntc/` to be made available for rsync backup. This is done by modifying the file `/etc/rsyncd.conf` and the file with the rsync password, `/etc/rsyncd.secrets`:

```
# cd /etc
# cat rsyncd.conf
gid = users
read only = true
use chroot = true
transfer logging = true
log format = %h %o %f %l %b
log file = /var/log/rsyncd.log

[zntc]
path = /zntc/
comment = zNTC Linux team shared files
auth users = mikem
secrets file = /etc/rsyncd.secrets

# cat rsyncd.secrets
# user:passwd
mikem:secret
```

On the rsync client side

Set up a script on the machine where the Samba data is to be backed up in the directory `/etc/cron.daily/` Also, set up a password file. This script will run each night:

```
# cd /etc/cron.daily/
# ls -l run_rsync /etc/rsync.password
-rw-------    1 root     root            8 May  1 14:29 /etc/rsync.password
-rwx------    1 root     root           85 May  1 07:21 run_rsync
# cat run_rsync
rsync -azv --password-file=/etc/rsync.password mikem@9.117.73.49::zntc /backups/zntc
# cat /etc/rsync.password
secret
```

Every night the Samba share will be backed up. Only the changes made each day are moved across the network and even those are compressed. This solution is especially useful because an exact copy of the previous day's file system is accessible. If you accidentally delete or corrupt a file, you can ssh or **`ftp`** over to the backup server and restore the file.

3.11 Setting up a Dfs root on Samba

This scenario sets up a Dfs root on one Samba which has two Dfs links:

- One to a Samba/Linux SMB share
- One to a Windows SMB share

The overall steps for this scenario are:

- Set up SMB shares on Linux and Windows to be the Dfs links
- Set up Samba on a Linux image to be the Dfs root
- Set up symbolic links from Dfs root to target shares
- Start Samba on Dfs root
- Access the Dfs root

Set up a SMB shares on Linux and Windows to be the Dfs links

There is nothing special here so details are not supplied. Samba is running on a Linux image with IP address 9.117.119.67. A Windows server is running with IP address 9.117.73.54. Both file servers create a file share named `data`.

Set up Samba to be the Dfs root

The Dfs root will be set up in the new directory `/dfs/` on a Linux image with IP address 9.117.119.70. First stop Samba, make to Dfs root directory then modify the `smb.conf` file:

```
# rcsmb stop
Shutting down Samba classic SMB daemon                              done
# rcnmb stop
Shutting down Samba classic NMB daemon                              done
# mkdir /dfs
# cd /etc/samba
# vi smb.conf  --> add the following variables and share
[global]
...
        host msdfs = yes
[dfs]
        path = /dfs
        msdfs root = yes
```

Set up symbolic links from Dfs root to target shares

Special symbolic links are created on Linux using the keyword **msdfs** and a modified UNC format. The first symbolic link, named `linka`, points to the Samba share named data. The second symbolic link, named `linkb`, points to a Windows share named `data`. Even though the **`file`** command reports them as broken links, Dfs-aware clients will be able to resolve them through Samba:

```
# cd /dfs
# ln -s msdfs:9.117.119.70\\data linka
# ln -s msdfs:9.117.73.54\\data linkb
# ls -l
lrwxrwxrwx    1 root     root           23 May 12 10:18 linka -> msdfs:9.117.119.70\data
lrwxrwxrwx    1 root     root           22 May 12 10:40 linkb -> msdfs:9.117.73.54\data
# file *
```

```
linka: broken symbolic link to msdfs:9.117.119.70\data
linkb: broken symbolic link to msdfs:9.117.73.54\data
```

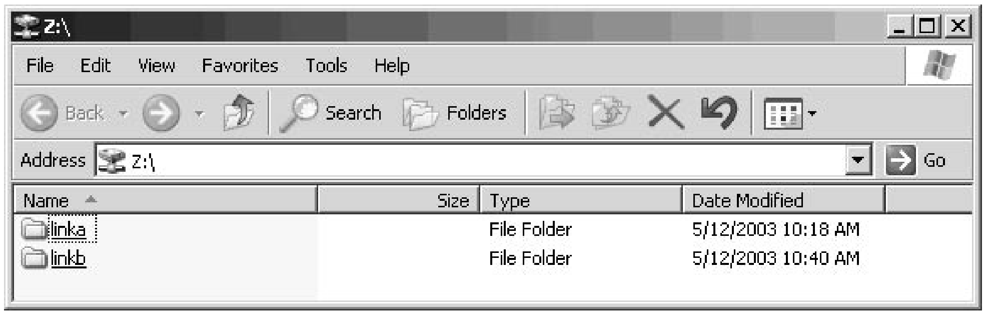
Start Samba

Samba is started again with the *rc scripts*.

```
# rcnmb start
Starting Samba classic NMB daemon                                   done
# rcsmb start
Samba SMB: Waiting for cupsd to get ready                           done
Starting Samba classic SMB daemon                                   done
```

Access the Dfs root

The Dfs root is accessed just as any other share - in this case the UNC `\\9.117.119.67\dfs` was given in the **Map Network Drive** dialog and the following share appeared. In a sense, this is a top level directory of SMB shares.



Name	Size	Type	Date Modified
linka		File Folder	5/12/2003 10:18 AM
linkb		File Folder	5/12/2003 10:40 AM

Figure 3-10 Dfs root on Samba from a Windows XP client

3.12 Setting up a Samba PDC

Important: winbind does not see new Domain users

When winbind is authenticating to a *native mode* AD DC, it does not see new users correctly. The following was appended to the Samba mailing list by a Samba team developer in June 2003:

> "Winbind in 2.2 doesn't obtain the sequence number in a native mode AD domain correctly. You must use an LDAP query to get the date. See the --with-*-ldap-hack in the configure options to enable the LDAP lookup (or wait for 3.0)."

The overall steps for this scenario are as follows:

- Upgrade Samba to samba-2.2.5-78
- Set up the `smb.conf` file
- Start Samba
- Add machine trusts manually or automatically
- Join the clients to the domain

zSeries specific: This scenario does not appear to work with a clean SLES-8 build which has a Samba RPM of `samba-2.2.5-60`. Trying to change the client's domain fails with the error "`A domain controller for the domain SMBLCC could not be contacted`". Upgrading to `samba-2.2.5-78` seems to be required, as the error is not encountered. To upgrade, the SLES-8 SP2 CD was mounted:

```
# rpm -qa | grep samba
samba-client-2.2.5-60
samba-2.2.5-60
# mount 9.117.99.3:/mnt/sles8sp2cd1 /mnt
# cd /mnt/suse/s390
# rpm -Uvh samba-2.2.5-78.s390.rpm
samba                      ##################################################
```

Set up the smb.conf file

The important `smb.conf` parameters are as follows:

```
[global]
        workgroup = SMBLCC
        netbios name = LINUX4
        encrypt passwords = Yes
        domain logons = Yes
        os level = 64
        preferred master = True
        domain master = True
        add user script = /usr/local/sbin/addSambaTrust %m
...
[netlogon]
        path = /var/lib/samba/netlogon
        write list = @ntadmin
```

The new domain will be named SMBLCC as specified in the `workgroup` parameter. The `netbios name` is the name of the server in the browse list. The `domain logons = Yes` parameter is the key to Samba acting as a PDC. It tells Samba to answer network logon requests. Setting the `oslevel = 64, preferred master = True` and `domain master = true` parameters ensure that Samba will become the domain master browser.

The `[netlogon]` section is a special section that creates a netlogon share. This is a share that Windows 95/98/ME clients must see in order to do a network logon.

Start Samba

Start Samba with the **`rcnmb`** and **`rcsmb`** commands:

```
# rcnmb start
Starting Samba classic NMB daemon                                    done
# rcsmb start
Starting Samba classic SMB daemon                                    done
```

Machine trust accounts must be added for each client. This is known as a *Computer Account* on NT. This can be done manually or automatically.

Add machine trusts manually

The account is created with the **`useradd`** command and it is verified that the account cannot be logged into. Then a corresponding Samba machine trust is created via the **`smbpasswd -m`** command:

```
# useradd -g 100 -d /dev/null -s /bin/false mikest30$
# smbpasswd -a -m mikest30$
```

```
Added user mikest30$.
```

Create a Samba password for the user with the **`smbpasswd -a`** command:

```
# smbpasswd -a mikem
New SMB password:
Retype new SMB password:
Added user mikem.
```

Add machine trusts automatically

This method is more secure because the client joins the domain immediately after the machine trust is created. You may have noticed the `add user script` previously in the `smb.conf` file:

```
        add user script = /usr/local/sbin/addSambaTrust %m
```

Let's look at a sample script:

```
# cat /usr/local/sbin/addSambaTrust
#!/bin/sh
# script to add a Samba machine trust (Windows "add computer") to /etc/passwd
# and /etc/samba/smbpasswd
# parm 1 - machine name
useradd -d /dev/null -g 100 -s /bin/false -M $1\$
echo "rc from useradd = $?" >> /tmp/foo
smbpasswd -a -m $1\$
echo "rc from smbpasswd = $?" >> /tmp/foo
```

You will probably want to write a more sophisticated script, but this one has the basics. Before adding the trusts, the *Samba user* root user must exist in the `smbpasswd` file. Add the Samba user `root` via the **`smbpasswd -a`** command:

```
# smbpasswd -a root
New SMB password:
Retype new SMB password:
Added user root.
```

If the Samba system administrator will be adding each client, the real root password can be used. However, the step of joining the domain might be delegated to individual users or to another administrator, in which case a different root password should be used.

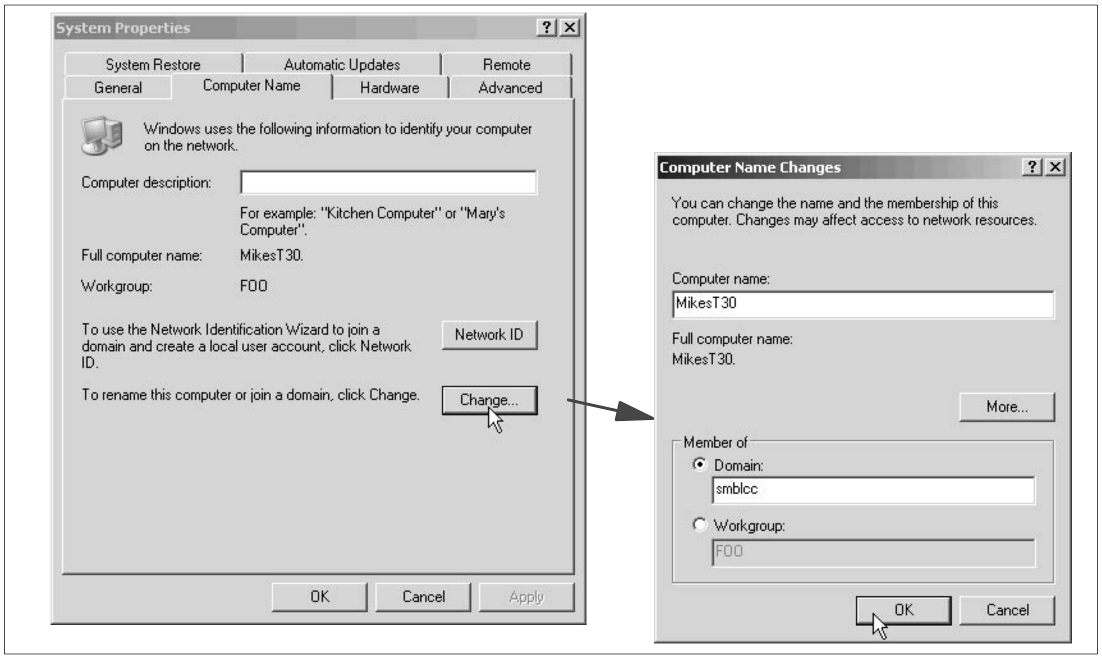
Join the clients to the domain

Before clients can join a new domain it has to be able to find the Samba PDC. Because it searches by domain name, this can be trickier than getting a share (which can use a DNS name or IP address). If the client and the PDC are on the same subnet, this will just happen with broadcast. If a WINS server is available, the domain name can be put in the WINS server and the PDC can point to it with the `wins server` parameter in the `smb.conf` file.

If neither of these options are available, the LMHOSTS file can be used. The LMHOSTS file maps IP addresses to NetBIOS names, and also allows for a domain name. The LMHOSTS file is in the directory `C:\WINNT\System32\drivers\etc` on Windows NT and 2000, or in `C:\WINDOW\system32\drivers\etc` on Windows XP.

```
C:\>cd WINDOWS\system32\drivers\etc
C:\WINDOWS\system32\drivers\etc>type lmhosts
9.12.9.5          linux4          #PRE #DOM:smblcc
```

The first time you join the domain, you have to specify a user ID with permission to let you join. Use `root` with the password in the `smbpasswd` file set earlier. To change the domain, open the **Control Panel**, open **System** then select the **Network Identification** (on Windows 2000) or the **Computer Name** (on Windows XP) tab. Then select the **Change** button and type in the name of the domain (the Network ID wizard will also accomplish the same):



Windows uses the following information to identify your computer on the network.

Computer description:

For example: "Kitchen Computer" or "Mary's Computer".

Full computer name: MikesT30.

Workgroup: FOO

To use the Network Identification Wizard to join a domain and create a local user account, click Network ID.

To rename this computer or join a domain, click Change.

You can change the name and the membership of this computer. Changes may affect access to network resources.

Computer name: MikesT30

Full computer name: MikesT30.

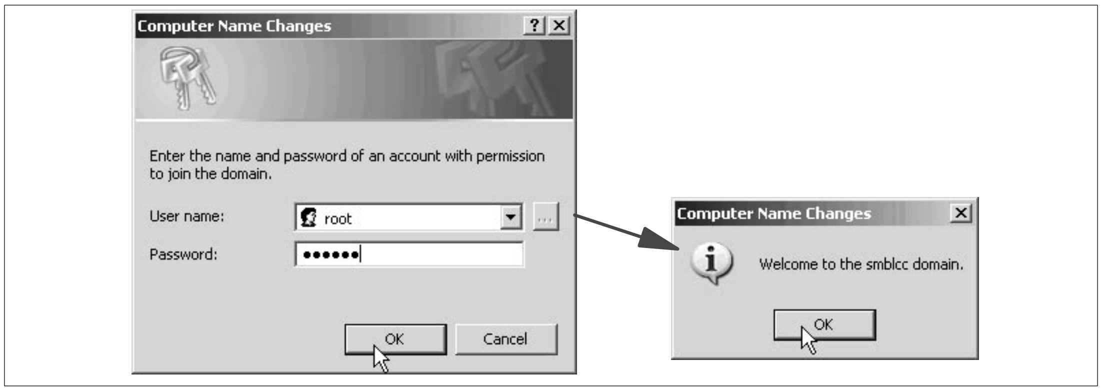
Member of

(●) Domain: smblcc

( ) Workgroup: FOO

Figure 3-11   Joining the Samba domain from a Windows XP client (1 of 2)

Then the root password is entered and the domain is joined (remember, the root password can be the same as the Linux root password, or it can be a special Samba root password just for joining domains).



Enter the name and password of an account with permission to join the domain.

User name: root

Password: ••••••

Welcome to the smblcc domain.

Figure 3-12   Joining the Samba domain from a Windows XP client (2 of 2)

Now let's look at what happened in the background on Linux. The `add user script` is run and the return codes from **`useradd`** and **`smbpasswd`** show success. The machine trusts are added to the `/etc/passwd` and `/etc/samba/smbpasswd` files.

```
# cat /tmp/foo
rc from useradd = 0
rc from smbpasswd = 0
# tail -1 /etc/passwd
mikest30$:x:504:100::/dev/null:/bin/false
# tail -1 /etc/samba/smbpasswd
mikest30$:504:3007C3F6F74DCB0B478086620BEB4D1D:62C3580C1E684930F524F72C5C4392AB:[W
]:LCT-3EC297A6:
```

A good reference Web site for setting up Samba as a PDC is:

```
http://www.mandrakeuser.org/docs/connect/csamba6.html
```

3.13 Setting up roaming profiles

This scenario builds on the previous one. The following parameters and section are added to the `smb.conf` file:

```
[global]
...
        logon path = \\%N\profiles\%u
        logon script = startup.bat
        logon drive = H:
        logon home = \\homeserver\%u
...
[profiles]
        path = /var/lib/samba/profiles
        read only = No
        create mask = 0600
        directory mask = 0700
```

The `logon path` parameter determines where the user's profile is stored as viewed by the local Windows system. The `logon drive` and `logon home` parameters determine which what happens when a DOS **`net use /home`** command is performed. The `logon script` parameter allows you to specify a script that is run when a user logs on.

The special section `[profiles]` determines where the user profiles are stored in the Linux file system and which permission bits new files and directories will be assigned.

This is all that is needed to set up roaming profiles. Now when clients do network logons, their logon data is read from the profiles directory and when they log off, any changes are written to it.

For example, after a network logon with user ID `mikem` to the new `SMBLCC` domain, a new Windows desktop was shown. A DOS (Command) prompt and a calculator were dragged from the Start menu to the desktop. After a logout the following files were written to the profiles file system path (Windows Media Player was mysteriously added to the list):

```
# cd /var/lib/samba/profiles
# ls
mikem/
# ls mikem
Application Data/  Favorites/     NTUSER.DAT.LOG  Recent/      Templates/
Cookies/           My Documents/  NetHood/        SendTo/      ntuser.ini
Desktop/           NTUSER.DAT     PrintHood/      Start Menu/
# ls mikem/Desktop
Calculator.lnk  Command Prompt.lnk  Windows Media Player.lnk  desktop.ini
```

3.14 Setting up quotas on Linux then Samba

Quotas function must first be set up on Linux, then they can be enforced via Samba. Quotas are not installed with the default install of a SLES-8 system.

Install the quota package after mounting the SLES-8 cd1:

```
# rpm -qa | grep quota
# mount 9.117.99.3:/mnt/sles8cd1 /mnt
# cd /mnt
# find . -name "*quota*"
./suse/s390/quota-3.03-84.s390.rpm
# rpm -ivh suse/s390/quota-3.03-84.s390.rpm
quota                       ##################################################
```

There are three services associated with quotas:

boot.quota	Run the **quotacheck** command at boot time
quota	Designed for local file systems
quotad	Designed for NFS access to local file systems.

Because **quotad** is for NFS, it is not set to start on reboot. The quota script is not designed to be started interactively, rather, only at boot time:

```
# chkconfig | grep quota
boot.quota              on
quota                   off
quotad                  off
# chkconfig quota on
# chkconfig | grep quota
boot.quota              on
quota                   on
quotad                  off
```

To enable user and group quotas, the parameters `usrquota` and `grpquota` are set, respectively, in the `/etc/fstab` file. In this example, user quotas are set up for the file system mounted over `/home/`:

```
# cd /etc
# cat fstab
/dev/dasdb1         /                   ext3       defaults             1 1
/dev/dasda1         /home               ext3       defaults             1 2
/dev/dasdc1         swap                swap       pri=42               0 0
...
# cp fstab fstab.orig
# vi fstab  --> replace “defaults” with “usrquota”
# cat fstab
/dev/dasdb1         /                   ext3       defaults             1 1
/dev/dasda1         /home               ext3       usrquota             1 2
/dev/dasdc1         swap                swap       pri=42               0 0
```

Rebooting the system will both start the **quota** and **quotad** services:

```
# shutdown -r now
...
```

After the system comes up see the `usrquota` parameter on the mounted file home system and initialize the quota system with the **quotacheck** command.

```
# rcquota status
Checking for quota:                                              running
# mount | grep home
/dev/dasda1 on /home type ext3 (rw,usrquota)
```

A quota for an individual user can be changed with the **edquota -u** command:

```
# edquota -u mikem
Disk quotas for user mikem (uid 500):
Filesystem                   blocks       soft       hard     inodes     soft     hard
/dev/dasda1                       0          0          0          0        0        0
```

This brings you into a **vi** session where you can change the default quota of 0, which means there is no quota enforced. Quota units are kilobytes (KB). In this example, the soft limit is set to 10MB and the hard limit is set to 12MB (inodes or the number of files can also be limited, but they are not in this example):

```
# edquota -u mikem
Disk quotas for user mikem (uid 500):
Filesystem                  blocks      soft      hard     inodes     soft     hard
/dev/dasda1                  11884     10000     12000         29        0        0
```

Once one user's quota is set, it can be copied to all users with the **edquota -p** command. In this example an **awk** command is used to get users with a UID of 500 or higher from the /etc/passwd file:

```
# edquota -p mikem `awk -F: '$3 > 499 {print $1}' /etc/passwd`
```

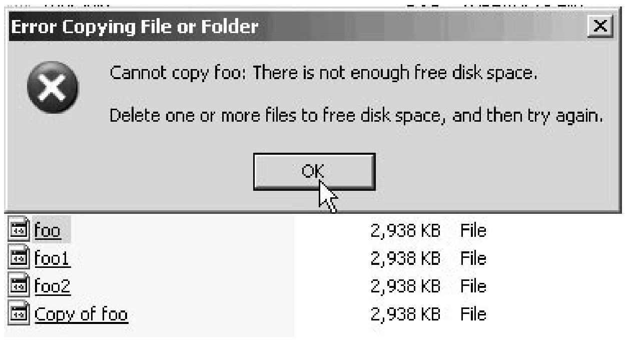
Now that quotas are enabled on Linux, they need to be tested. The quota for the user mikem is 10MB. A 3MB file can be copied two times, but a warning is issued after the third as the soft limit is reached. The fourth copy fails as the hard limit is reached:

```
$ ls -l foo*
-rw-r--r--    1 mikem    users     3008507 2003-05-20 11:11 foo
$ cp foo foo1
$ cp foo foo2
$ cp foo foo3
dasd(94,1): warning, user block quota exceeded.
$ cp foo foo4
dasd(94,1): write failed, user block limit reached.
cp: writing `foo4': Disk quota exceeded
$ ls -l foo*
-rw-r--r--    1 mikem    users     3008507 May 20 11:11 foo
-rw-r--r--    1 mikem    users     3008507 May 20 11:27 foo1
-rw-r--r--    1 mikem    users     3008507 May 20 11:27 foo2
-rw-r--r--    1 mikem    users     3008507 May 20 11:27 foo3
-rw-r--r--    1 mikem    users      122880 May 20 11:27 foo4
```

After the fourth copy fails, the **du** command shows that exactly 12MB (OK, 12000KB) are being used:

```
$ du -sk
12000 .
```

The next step is to test with Samba. First foo3 and foo4 are removed with the **rm foo3 foo4** command. Then Samba is started and the home share mikem is obtained from a Windows client. As from the shell, an attempt is made to make two more copies of the file foo. The first one succeeds without a warning about the soft limit. However, the next one fails as expected with the following error dialog:


Error Copying File or Folder

Cannot copy foo: There is not enough free disk space.

Delete one or more files to free disk space, and then try again.

```
foo           2,938 KB   File
foo1          2,938 KB   File
foo2          2,938 KB   File
Copy of foo   2,938 KB   File
```

Figure 3-13 Windows error dialog when Linux quota is exceeded

3.15 Setting up a z/VM VDISK swap space

zSeries-specific: An optimally performing paging (swap) area in a z/VM/Linux environment is a VM virtual disk. When paging is necessary, memory, not disk is accessed. z/VM will only allocate the memory when it is needed. Some Linux images are designed with small amounts of memory (e.g. 32MB) and much larger VDISK swap spaces (e.g. 128MB). This scenario is not yet documented in this paper, but a very good start is one the Web at:

```
http://linuxvm.org/Info/HOWTOs/vdiskswp.html
```

3.16 Complete enterprise scenario

A good reference is Chapter 12 of *Samba in 24 hours* on the Web at:

```
http://freebooks.by.ru/view/SambaIn24h/ch12.htm
```

Unfortunately, the meatier scenarios of migrating data at the enterprise level has not yet been documented. Both an LDAP and an Active Directory scenario need to be documented.

Following is an outline of the migration process:

- Create a sample Windows 2000 server AD + File server + Print server
- Create a Samba server
- Migrate directory information into LDAP
  - One Samba server is configured as a member server of the existing domain
  - Import of UNIX users and groups via winbind, Transfer to the LDAP directory
  - Gather the Windows-specific user attributes via **`rpcclient`** - Import into LDAP
  - Gather the password hashes via pwdump2 - Import into LDAP
  - Gather the machine accounts via rpcclient - Import into LDAP
- Migrate Data
  - Categorize the shares
  - Migrate them away from the DCs
  - Assign them to new Linux-based servers
  - Configure PDC and BDCs
  - Take-over of the SAM-database, LDAP-replication to the BDCs
  - Shut-down Windows-DCs and start the Samba-DCs **Attention:** All Windows-based DCs must shut down within a short time.
  - Migration of the shares with Windows-tools (**`scopy.exe`** or **`robocopy.exe`**)
- Migrate Printers
- Migrate ACLs

Section 4. Migration to zSeries considerations

This section addresses strategies for migration from Windows file and print servers to Samba and Linux. When migrating systems, it is always best to used a phased approach. At the end of each phase and before the start of another, the entire system should be working basically the same as, or better than, it was working before. Between each phase, a checkpoint can be made.

Given these considerations, a rough order of migration could be:

- Migrate file servers with user's personal shares. Use winbind to point to an existing Windows Domain Controller
- Migrate file servers with group or team shares.
- Migrate print Servers
- Migrate Domain Controllers, if necessary

One of the first questions to answer is what the architecture of authentication and authorization will be, for the short term and for the long term. The simplest type of Samba authentication is to use the `smbpasswd` file. To simply test Samba function, the `smbpasswd` file can be used, but this should only be for testing, or perhaps for very small teams using Samba. Usually, the first phase leaves authentication on Windows domain controllers. Setting up winbind can be tricky, but when it is set up correctly, it works fine. An interim step can then be to set up Samba to act as a PDC. With Samba-3, additional function is being added to bring it more in line with Active Directory. The long term goal should be to use LDAP services for a centralized, highly available authentication and authorization system. Using LDAP will not lock you into a single vendor's solution.

Another issue is how to architect a proper backup and restore. Typically a vendor product is already in place in a large enterprise. The issue then becomes how to fit the Samba architecture into the existing backup and restore solution.

Another issue to address is how to deal with centralized vs. distributed data. Often there is a requirement for file and print services in locations remote from the data center. Often the network bandwidth is small. Mapping a network drives is possible, but not usable due to the limited bandwidth. Usually, remote file and print servers on Intel PCs is the solution. What then needs to be addressed, is how these remote servers fit in with the authentication and backup and restore architecture that is agreed upon.

Once an overall architecture is agreed upon, pilot migrations can be started. Depending on the architecture, one rough order of steps required is:

- Set up Linux and Samba as decided upon.
- Extract user, group, share information from Windows (**`pwdump2`**, Samba-3 **`net rpc vampire`** command, etc.)
- Transfer the user, group, share information to Linux Samba (usually just FTPing one or more files)
- Recreate the user, group and share environment on Linux using scripts if necessary
- Copy user data from Windows servers to Linux. Copy ACLs associated with files and folders is an issue, as they are not always copied (**`robocopy`** from the Windows Resource Kit, or the freeware/product named **`xxcopy`** are often used)
- Test the pilot setup for integrity and performance.

4.1 zSeries performance considerations

Following are some performance considerations for Linux and Samba on zSeries:

- z/VM has a 3-10% performance penalty on average, but this can be offset by other function the z/VM affords, for example VDISK swap spaces.
- Use LVM with stripes - see 3.2, “Setting up a logical volume” on page 46
- Observations by the IBM Linux Scalability Center:
  - Network configuration for communications: Use direct connections from OSA gigabit or fast ethernet cards to each Samba server guest.
  - If the configuration must use connections to one or more Linux guests used as a router, the recommendations for routing are as follows:
  - Use VM guest LAN rather than VCTC to connect the guests to the Linux system that is providing routing capability, or
  - Use VM TCP/IP routing from the OSA card to the guest servers.
  - Recommended virtual memory size: 128 MB. If a large number of guests are to be implemented, the goal should be to keep the Samba server virtual memory size as small as possible while avoiding Linux guest paging.
  - Minidisk caching was of little value and used a large amount of VM storage.
- Recommendations by the IBM Linux Scalability Center:
  - With a single Gbe OSA card up to 25 guests with one concurrent request each and an aggregate throughput of 13.37 MB/second could be supported. Maximum OSA throughput was reached between 12-15 SMB processes.
  - With a single guest server and a single OSA card we were able to support up to 30 concurrent users at an aggregate throughput of 19.4 MB/second.
  - Summary of Native results vs. VM guest results. The cost in throughput between the 2.4.17 kernel in a native LPAR vs. running the timer change version of this kernel on z/VM is most significant with small numbers of guests.
    - Cost for the first 1-10 guests was 20-26% total.
    - Cost for 15 - 35 guest was only 9.7-16% total.
  - SuSE SLES 7 throughput is not as good as new internal kernel. At 35 connections the 2.4.17 Timer kernel produced up to 125% improvement over SuSE SLES 7. It is likely that significant improvement would be seen in SuSE SLES 8 vs. SLES 7.
  - The SuSE SLES 7 (2.4.7 kernel) QDIO communications provided significantly less throughput than the 2.4.17 kernel. With the changes provided in the 2.4.17 kernel and device drivers, the internal throughput improved by up to 100% in a heavily loaded guest, and by approximately 15% in a lightly loaded multiple guest configuration.
  - Download the latest QETH and QDIO device drivers supported for your kernel from IBM developerWorks to ensure that the installation is running at peak
- Recommendations from some unofficial testing. Try these parameters at your site to see if they give you better performance:
  - The following `smb.conf` settings may give you better performance.

```
max xmit = 8192
socket options = TCP_NODELAY IPTOS_LOWDELAY SO_SNDBUF=14596 SO_RCVBUF=14596
dead time = 10
```

4.2 Integrity issues

The quality and integrity of Samba should be perfect. There can be issues with Samba corrupting files, though this can be as a result of opportunistic locking and other factors designed into the SMB protocol

- Microsoft *.pst files can become corrupted with oplocks on. It is a safe practice to keep oplocks off for integrity, however, performance gains from oplocks are lost. See Chapter 14, *File and Record Locking*, of the Samba-3 HOWOTO collection. To turn oplocks off add the following global `smb.conf` parameters:

  ```
  kernel oplocks = no
  oplocks = no
  level2 oplocks = no
  ```

Section 5. Migration from Novell NetWare

In general there is less experience in migrating NetWare data than in migrating Windows data. However, the question of migrating from NetWare is often asked. There are two general approaches: convert the NetWare metadata to Linux or use NetWare 6.5 to move the data to Linux while keeping the NetWare directory and data intact.

5.1 Move NetWare data converting to Posix metadata

Following is a May 2003 append to the `linux-390` list server:

> “We created 4 files using NetWare commands containing group membership, directories, users, and rights. These files were read into a VB program we wrote to establish the copy statements and the Linux command file. After disabling logins to NetWare, we ran the copy statements to move the data to Samba, then ran the commands to establish the proper associations of users, groups, and directories. That's the thumbnail version of the process.”

In addition, some more details on creating the four files using NetWare commands were found on the Web:

First change context to root:

```
cx /R
```

Create file of group membership:

```
nlist user show “group membership” /C /S > D:\cv\user.txt
```

Create file of all users:

```
dir J:\USERS > D:\cv\nwdir.txt
```

Create file of directories:

```
dir J:\ /AD /X /S > D:\cv\nwdir2.txt
```

Create file of all rights:

```
rights J:\ /C /F /S > D:\cv\irights.txt
```

5.2 Move data to Linux via NetWare 6.5

Another approach to moving NetWare data to Linux is via Novell's NetWare 6.5. If the customer upgrades to NetWare 6.5, (which becomes generally available August 15, 2003) they get a free migration to NetWare 7 which will be the entire line of Novell services on Linux. This approach allows the directory, applications and file systems to remain intact. Novell will provide NetWare to Linux migration tools to make this seamless to the customer.

Section 6. Advantages of Samba and IBM zSeries

Samba and Linux can have many advantages. Most will agree that:

- Linux is a more reliable operating system than Windows.
- Linux has fewer licensing costs, and Samba has no Client Access Licenses (CALs).
- Linux does not lock you into Active Directory and frequently changing Microsoft protocols.

Samba and Linux on zSeries hardware can offer additional advantages:

- zSeries is arguably the most reliable computer hardware.
- z/VM is a virtualization layer that allows easy creation of virtual Linux servers and, with a proper architecture, easier system and network administration.

Samba can also be implemented as just part of the infrastructure, Enabling Samba services on Linux images that have a function other than file and print serving can aid Linux users. For example Web developers and programmers can move data to the Linux environment in ways that are already familiar to them and access them from a Windows environment.

Reference scripts

Following are the LDAP scripts used in 3.4, “Setting up OpenLDAP” on page 52. The *ldap commands* (**ldapadd**, **ldappasswd**, etc) require many parameters that are difficult to remember and also commonly require the same parameters. Simple wrappers are written around the ldap commands so a man page does not have to be referenced each time you want to use them.

The file `lvars` contains variables common to the scripts such as the LDAP host, suffix, manger, etc:

```
# cat lvars
LDAP_HOST=localhost
LDAP_SUFFIX=dc=poklcc,dc=ibm,dc=com
LDAP_MANAGER=cn=Manager,$LDAP_SUFFIX
LDAP_USERS=ou=People,$LDAP_SUFFIX
```

The script **lpasswd** invokes **ldappasswd** to change LDAP passwords:

```
# cat lpasswd
# lpasswd - a wrapper around ldappasswd to make it easier to use
#   variables are set in the file ./lvars
#   Hard code the parameters:
#     -x - use simple authentication instead of SASL
#     -h $LDAP_HOST - LDAP server IP@ or DNS name
#     -D $LDAP_MANAGER - rootdn set in lvars
#     -W - prompt for root pw
#     -S - prompt for new pw

scriptName=`basename $0`
if [ $# != 1 ]; then
# usage
  echo; echo "Usage: $scriptName <user_name>"; echo
  echo "Change the password on the LDAP server"
  echo "See this script, the file lvars and /etc/openldap/ldap.conf"
else
# do it!
  . /usr/local/sbin/lvars
  ldappasswd -x -h $LDAP_HOST -D $LDAP_MANAGER -W -S uid=$1,$LDAP_USERS
fi
```

Test the **lpasswd** script:

```
# lpasswd ldapuser3
New password:
Re-enter new password:
Enter bind password:
Result: Success (0)
```

The script **ldifadd** invokes **ldapadd** to add the contents of an ldif file:

```
# cat ldifadd
# ldiffadd - wrapper around ldapadd to make it add LDAP entries in .ldif files
#   Common variables are set in the file ./lvars
#   Hard coded parameters:
#     -x - use simple authentication instead of SASL
#     -D $LDAP_MANAGER - rootdn set in lvars
#     -h $LDAP_HOST - LDAP server IP@ or DNS name
#     -W - prompt for root pw
#     -S - prompt for new pw
scriptName=`basename $0`
if [ $# != 1 ]; then
```

```
# usage
  echo; echo "Usage: $scriptName <ldif_file>"; echo
  echo "Add the contents of an ldif file to the LDAP server"
  echo "See this script, the file lvars and /etc/openldap/ldap.conf"
else
# do it!
  . /usr/local/sbin/lvars
  ldapadd -x -h $LDAP_HOST -D $LDAP_MANAGER -W -f $1
fi
```

Test the **ldiffadd** command:

```
# cat test.ldifadd
dn: uid=ldapuser3,ou=People,dc=poklcc,dc=ibm,dc=com
uid: ldapuser3
cn: ldapuser3
objectClass: account
objectClass: posixAccount
objectClass: top
objectClass: shadowAccount
shadowLastChange: 12118
shadowMax: 99999
shadowWarning: 7
loginShell: /bin/bash
uidNumber: 1002
gidNumber: 100
homeDirectory: /home/ldapuser3
# ldifadd test.ldifadd
Enter LDAP Password:
adding new entry "uid=ldapuser3,ou=People,dc=poklcc,dc=ibm,dc=com"
```

The script **ldifmod** invokes `ldapmodify` to modify an entry based on an ldif file:

```
# cat ldifmod
# ldifmod - wrapper around ldapmodiy to save typing
#   Common variables are set in the file ./lvars
#   Hard coded parameters:
#     -x - use simple authentication instead of SASL
#     -D $LDAP_MANAGER - rootdn set in lvars
#     -h $LDAP_HOST - LDAP server IP@ or DNS name
#     -W - prompt for root pw
scriptName=`basename $0`
if [ $# != 1 ]; then
# usage
  echo; echo "Usage: $scriptName <ldif_file>"; echo
  echo "Modify LDAP entries based on the contents of an ldif file"
  echo "See this script, the file lvars and /etc/openldap/ldap.conf"
else
# do it!
  . /usr/local/sbin/lvars
  ldapmodify -x -h $LDAP_HOST -D $LDAP_MANAGER -W -f $1
fi
```

Test the **ldifmod** command:

```
 # ldifmod test.ldifmod
Enter LDAP Password:
modifying entry "uid=ldapuser3,ou=People,dc=poklcc,dc=ibm,dc=com"
```

The script **ldelete** invokes **ldapdelete** to delete an entry dn:

```
# cat ldelete
# ldel - a wrapper around ldapdelete to make it easier to use
#   variables are set in the file ./lvars
#   Hard code the parameters:
#     -x - use simple authentication instead of SASL
#     -h $LDAP_HOST - LDAP server IP@ or DNS name
#     -D $LDAP_MANAGER - rootdn set in lvars
#     -W - prompt for root pw

scriptName=`basename $0`
if [ $# != 1 ]; then
# usage
  echo; echo "Usage: $scriptName <dn>"; echo
  echo "Delete the <dn> from the LDAP server"
  echo "See this script, the file lvars and /etc/openldap/ldap.conf"
else
# do it!
  . /usr/local/sbin/lvars
  ldapdelete -x -h $LDAP_HOST -D $LDAP_MANAGER -W $1
fi
```

Test the **`ldelete`** command:

```
# ldelete uid=ldapuser3,ou=People,dc=poklcc,dc=ibm,dc=com
Enter LDAP Password:
# ldapsearch -x uid=ldapuser3
```

Resources

Books

Mastering Windows NT Server 4, Third edition, Minasi, et al, Sybex, ISBN 0-7821-1920-4
Mastering Windows 2000 Server, Fourth edition, Minasi, et al, Sybex, ISBN 0-7821-4043-2
Using Samba, 2nd Edition, David Collier-Brown, et al, O'Reilly, ISBN 0-596-00256-4
Teach Yourself Samba in 24 Hours, 2nd edition, Gerald Carter, et al, SAMS, ISBN 0-672-32269-2
Samba Essentials for Windows Administrators, Gary Wilson, Prentice-Hall, ISBN 0-13-040942-1
Samba Unleashed, Steve Litt, et al, SAMS, ISBN 0-672-31862-8

Using Samba, available in SWAT

Redbooks - free PDFs on the Web

- *Linux for S/390*, SG24-4987
  `http://www.redbooks.ibm.com/abstracts/sg244987.html`
- *Linux for zSeries and S/390: Distributions*, SG24-6264
  `http://www.redbooks.ibm.com/abstracts/sg246264.html`
- *Linux on IBM eServer zSeries and S/390: Large Scale Linux Deployment*, SG24-6824
  `http://www.redbooks.ibm.com/abstracts/sg246824.html`
- *Understanding LDAP*, SG24-4986
  `http://www.redbooks.ibm.com/abstracts/sg244986.html`

Web sites

- Network Time Protocol home page:
  `http://www.ntp.org/index.html`
- Samba home page:
  `http://www.samba.org`
- Linuxvm.org - *the* Linux on zSeries portal:
  `http://linuxvm.org`
- DeveloperWorks - Linux s390 code patches from IBM Boeblingen
  `http://www10.software.ibm.com/developerworks/opensource/linux390/index.shtml`
- Vendor applications for Linux on zSeries:
  `http://www.ibm.com/servers/eserver/zseries/solutions/s390da/linuxproduct.html`
- z/VM and Linux:
  `http://www.vm.ibm.com/linux`
- linux-390 archives:
  `http://www.marist.edu/htbin/wlvindex?linux-390`
- z/VM publications:
  `http://www.vm.ibm.com/pubs/`

Mailing lists

- linux-390 mailing list (subscribe at bottom of page)

  `http://www.marist.edu/htbin/wlvindex?linux-390`
- Samba mailing list (this host or other mirror)

  `http://us2.samba.org/samba/archives.html`