

Migrating File and Print Services from Windows* to SUSE LINUX® Enterprise Server 9

A Novell® and Versora® Migration Study

www.novell.com

SEPTEMBER 2004



Disclaimer Novell, Inc. makes no representations or warranties with respect to the contents or use of this document, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose.

Trademarks Novell, the Novell logo, and ZENworks are registered trademarks and eDirectory and Nterprise are trademarks of Novell, Inc. in the United States and other countries. Versora is a registered trademark of Versora. SUSE is a registered trademark of SUSE LINUX AG, a Novell business.

Linux is a registered trademark of Linux Torvalds. IBM is a registered trademark of IBM Corporation. Windows and Windows NT are registered trademarks of Microsoft Corporation. Intel is a registered trademark of Intel Corporation. AMD is a trademark of Advanced Micro Devices. All third-party trademarks are the property of their respective owners.

Copyright 2004 Novell, Inc. All rights reserved. No part of this publication may be reproduced, photocopied, stored on a retrieval system, or transmitted without the express written consent of Novell, Inc.

Novell, Inc.
404 Wyman Street, Suite 500
Waltham, MA 02451
USA

Versora
226 East Canon Perdido St.
Santa Barbara, CA 93101
USA

Prepared by Versora and Novell Solution Creation and Marketing - Linux Team

Title Migrating File and Print Services from Windows to SUSE LINUX Enterprise Server 9 - A Novell & Versora Migration Study

Contributors Mike Sheffey, CEO, Versora
John Beuchert, Global Solutions Director, Novell
Nick Lasonde, Software Architect, Versora
Johannes Meixner, Software Engineer, SUSE LINUX
Kurt Brust, Global Solutions Manager, Novell
Meike Chabowski, Project Manager, SUSE LINUX
Joyce Whiting, Solution Development Specialist, Novell

Date September 2004

Contents

Migrating File and Print Services from Windows to SUSE LINUX Enterprise Server 9.....	1
Training and Certification.....	2
Support staff.....	4
Licensing fees.....	4
Transfer Files Using Samba.....	4
Restore Equivalent Rights and Ownership.....	5
Configure the Samba Server.....	6
Install CUPS.....	7
From Windows to UNIX/Linux printing.....	8
CUPS spooling.....	9
CUPS filtering	9
Printer protocols.....	11
Printer drivers.....	11
PostScript printer description files.....	12
Configure CUPS.....	12
With YaST.....	12
From the command line.....	13
Change Printer Configurations.....	16
Set Up a CUPS Administrator.....	16
Manage CUPS from the Web.....	17
Migrate from LPRng/lpdfilter to CUPS.....	17
CUPS Files and Commands.....	17
Appendix A: Linux File System	21
File Permission Scenarios.....	23
File Commands.....	24

Migrating File and Print Services from Windows to SUSE LINUX Enterprise Server 9

All businesses of any size must provide their users with the ability to print and to store and access files on network servers. Accessing and saving files to network drives and printing to shared network printers is probably something administrators don't spend much time thinking about unless it's to help users install and re-install print drivers or assist them in accessing shared network directories.

If your organization has moved to Linux* in the data center or in some of your business workgroups or is planning to install other services on Linux, migrating Windows* file and print services to Linux is a good next step. This paper will help you identify what needs to be done to accomplish such a move and notes some of the benefits of making the move to SUSE® LINUX Enterprise Server 9 file and print servers. The information provided here should be considered a place to start rather than a thorough discussion of all possible alternatives or a complete roadmap for the migration.

PLANNING THE MIGRATION

Because making a decision to change operating systems has so many ramifications for your business and isn't always straightforward, we suggest that you plan carefully so that such a transition addresses not only your immediate goals but your long-term strategy as well. You'll also want to make sure your technical staff has the necessary skills to implement and maintain a Linux environment. While UNIX shares many similarities with Linux, migrating from Windows to Linux is quite another matter. Many companies, Novell and Versora among them, provide excellent professional consulting and training services to help you make the move.

Novell Professional Services offers consulting engagements that range from Strategy and Discovery to Requirements Assessment and from Planning & Design to Implementation. These offerings help you assess both current and future strategies, assess your organization's readiness for moving to SUSE LINUX, provide information about how to best approach a migration, and finally, help you implement your migration plans.

For additional information about Novell Professional Services, refer to <http://www.novell.com/linux/migrate>.

Versora provides consulting services that focus on migrating and integrating Windows and SUSE LINUX. Versora migration services address operating systems, Web servers, database servers, collaboration and messaging servers, and desktops.

Versora can provide your company with a cost analysis and a project plan that present the business case for integrating SUSE LINUX into your current environment.

For additional information about Versora Professional Services refer to <http://versora.com/services/>

Training and Certification

You will probably want at least some of your technical staff to be Linux certified (LPI level 1 and/or LPI level 2). Many third-party training and Linux certification courses are available to meet this need.

In addition to Linux certification, we recommend SUSE LINUX-specific training. Novell offers a variety of instructor-led and self-study certification and training options including the following:

- Novell Certified Linux Professional (Novell CLP) or SUSE Certified Linux Professional (SCLP); these courses are the best place to start:
 - Course 3036 - Linux Fundamentals
 - Course 3037 - Linux Administration
 - Course 3038 - Advanced Linux Administration
 - Course 3019 - Migrating to SUSE LINUX (for experienced Linux administrators)
 - Novell Practicum
- Novell Certified Linux Engineer (Novell CLE); these courses build on CLP and SCLP training:
 - Course 3017 - Fundamentals of Novell eDirectory™
 - Course 3015 - Novell Nterprise™ Linux Services
 - Novell Practicum

Note: Only the Practicum exams are *required* for certification.

Novell certification and training options change periodically as new needs are identified and courses are developed. To learn more about these and other training options, visit the Novell training Web site at www.novell.com/training.

EVALUATING BUSINESS ISSUES

Linux is the fastest growing operating system in the world with some industry analysts predicting that Linux will enjoy 30-40% of the server operating system market by the end of 2007 (see http://www.theregister.co.uk/2003/06/16/massive_growth_in_linux/).

While many organizations have only experimented with Linux, others have bet-the-business. For example, Google is currently running 100,000 Linux servers. According to Forbes magazine, it would take twice as many machines running Windows to stay on par with Google's Linux infrastructure.

A 2004 Forrester Research study discovered that 46% of enterprises are already using some form of Open Source software (OSS). An additional 14% will begin implementing Open Source software within the next 9 months. More than half the enterprise IT

organizations will be using Open Source Software by 2005. Of those using or planning to use Open Source, 100% of the respondents use or will use Linux and 38% use or will use Samba. The respondents to this Forrester Research study are expecting to benefit in the following ways: (a) increased security, (b) better hardware platform support (c) lower total cost of ownership and (d) lower cost of software.

Since both Windows and Linux run on Intel*/AMD* hardware, the significant hardware savings that can be realized with a UNIX to Linux migration are not part of the equation. But there are other good business reasons to make the move: increased reliability, increased performance, less patch management, less server administration time, and lower overall operating costs—including licensing and administrative fees. According to Meta Group, Linux can drive a business's TCO down by 10-40% (Meta Group: *Linux World Journal*, August 2004).

A study conducted by Forrester Research found that two of the top five IT goals for organizations are to lower overall operating expense and improve work force productivity and processes (Ted Schadler Vice President, Software Research, Forrester Research). Here are some of the studies that suggest you can do just that with Linux:

- Printing via Linux servers is one of the most robust and scalable platforms available on the market. According to a *PC Magazine* lab test, Linux print serving provides up to 60% better throughput than printing under Windows. The same lab tests indicate, that from a performance perspective, Linux has more than 100% better average response time than printing under Windows (see <http://www.linuxjournal.com/article.php?sid=6729>).
- According to an *IT Week* lab test, Linux running Samba was twice as fast as a Windows 2000 Server and when tested against Windows 2003, 2.5 times faster (“Samba 3 runs rings around Windows Server 2003 in file serving performance” Roger Howorth, *IT Week* 12 Oct 2003).
- An IBM* study of a corporation with 1,100 users—and 140 file and print servers—migrating from Windows NT* to Linux, realized a 59% cost savings moving to Linux. Also noted was the ratio of Windows servers to server administrators—16:1. According to a study done by Robert Frances Group, that ratio on Linux servers is as high as 44:1 (“Total Cost of Ownership for Linux Web Servers in the Enterprise” September 2002, Robert Frances Group).
- The Robert Frances Group study also concluded that Linux is the least expensive operating system platform to deploy and manage. Linux was shown to reduce per server 3-year TCO from 100% - 700%.

To be fair, we need to mention that other testers and analysts have documented entirely different test and research results. As you analyze the performance studies for yourself, make sure any given study doesn't just support the interests of the funding organization be it Linux- or Windows-based.

Support staff

You will also want to carefully analyze your support staff before making the move to make sure initial training costs aren't prohibitive since there is a world of difference in the way Windows and Linux servers are managed. However, retraining key personnel is not as difficult or costly as some would lead you to believe.

Licensing fees

More and more companies are looking to reduce the cost associated with software licensing fees. According to a Forrester Research study conducted in 2004, reducing a company's overall operating costs is the top priority of 69% of IT organizations (Ted Schadler, Vice President, Software Research, 2004 Forrester Research).

Using SUSE LINUX Enterprise Server 9 can help in this regard since companies are not charged additional fees for client access licenses (CALs). This means that a company with 5,000 concurrent CALs can reduce licensing fees by \$250,000 moving to SUSE LINUX Enterprise Server 9.

MOVING FILES FROM WINDOWS

Transfer Files Using Samba

Samba is an Open Source implementation of the SMB protocol that runs on Linux and UNIX (renamed by Microsoft to Common Internet File System (CIFS)). The CIFS protocol is the foundation of Windows networking. Running Samba as a file and print server allows Windows clients (Windows 9x, NT, 2000, and XP) to print and access shared network resources.

The easiest way to transfer files from Windows to SUSE LINUX Enterprise Server 9 is via Samba. The default SUSE LINUX Enterprise Server 9 installation includes the Samba client, which is all that is necessary to browse Windows shares.

One of your first tasks should be setting up a location for storing files, usually in a location that normal users don't have rights to edit.

You can complete this file transfer using the graphical tools built-in to SUSE LINUX Enterprise Server 9.

1. Launch Konsole, and then type:

```
su
<the machine's root password>
cd /srv
mkdir samba
chmod 777 samba
```

This creates a directory (/srv/samba) to serve files from and gives everyone full permission to that directory.

2. Click on the desktop icon named Network Browsing, and then click on Windows Network.
You should see a list of domains.
3. Click on the domain for the source Windows server.
4. Click on the destination server (the one you are migrating to). Enter the username and password of an administrative user.
5. Open another Konquerer window (**Control-N**, or **Location->New Window**), and browse to the Samba server location (`/srv/samba`). (You can just type this path in the location bar.)
6. On the Windows server, highlight all the sharepoints you want to move and drag them to `/srv/samba`.

A popup menu will appear prompting for the next action; choose Copy Here to initiate the file copy.

Depending on the amount of information being moved, this may take a while.

Restore Equivalent Rights and Ownership

After the files are transferred, file permissions will need to be reset. Linux file system permissions are different from those used by Microsoft Windows systems and are not as granular.

This is particularly apparent for directories where multiple groups have access to the data within a file. On Linux file systems, this is not possible. You will need to find an alternate method to ensure that the proper permissions are set for transferred files.

Linux file systems keep track of permissions for user, group, and other. Permissions include read (r), write (w), and execute (x). A dash in the rights notation usually means that the permission is not set.

Consider the following permissions as a starting point.

Type of files	Permissions:	
	user, group, other	numeric value
Home directories (ex: <code>/home/<userid></code>)	<code>rwX --- ---</code>	700
User files (ex: <code>/home/<userid>/<myfile></code>)	<code>rw- r-- ---</code>	740
Share directories for a team (where the group is used for access)	<code>rwX rwx ---</code>	770
Shared team files (where the group is used for access)	<code>rw- rw- ---</code>	660

Refer to Appendix A beginning on page 21 for additional detail.

Configure the Samba Server

Complete the following to configure a Samba Server on SUSE LINUX Enterprise Server 9:

1. Launch YaST via the icon on your desktop.
2. From the left hand list, choose Network Services, and then scroll to the Samba Server option. A single click will bring up the preferences page and guide you through installing the Samba server if it hasn't been installed yet.
3. Configure the server.
 - a. Select the Identity tab to choose a domain for the server.
 - b. Under NetBIOS Host Name, enter a name to identify the server in the Windows network. We recommend using the same name as the host name.
4. Configure Shares.
 - a. Select the Shares tab, and then click Add.
 - b. Enter the name and description of the first share, along with the path (which should be `/srv/samba/<ShareName>`).
 - c. Repeat this step for each share.
5. Set share-level security.
 - a. Select a share and click Edit.
 - b. Select the Writeable option to specify whether users can write to this share.
 - c. Select the Users option to specify which users are allow to access this share.
 - d. Repeat this procedure for each share.

For additional information about Linux file permissions, refer to:

<http://www.linuxpowered.com/html/editorials/file.html>

<http://www.freeos.com/articles/3127/>

<http://www.wbglinks.net/pages/reads/linux/filepermissions.html>

SETTING UP LINUX PRINTING

Setting up printing on SUSE LINUX Enterprise Server 9 is best considered a configuration, not a migration; you will need to recreate the Windows printing environment on Linux rather than porting the printing environment from one system to the other.

Two primary printing options need to be considered with SUSE LINUX Enterprise Server 9: CUPS and Novell iPrint.

CUPS (bundled with SUSE LINUX Enterprise Server 9) - Most Linux distributions, including SUSE LINUX Enterprise Server 9, install CUPS (Common UNIX® Printing System), a UNIX-based standard for printing that is also used for printing under

Linux. The CUPS Linux print server uses the Internet Printing Protocol (IPP) to manage print jobs. CUPS provides network printer browsing and PostScript printer-based options for Linux. The LPD, SMB, and AppSocket/JetDirect protocols are supported.

An old-style BSD-like printing system, LPRng/lpdfilter, is also available (only with LPD support) which can be used instead of CUPS but LPRng/lpdfilter support is being phased out and, after SUSE LINUX Enterprise Server 9, cannot be configured with YaST (manual configuration will still be possible). You can use CUPS or you can use LPRng/lpdfilter—but using both together is not possible.

CUPS also supports printing on printers connected to Windows shares. CUPS uses Samba which supports the SMB protocol, making use of ports 137, 138, and 139.

Novell iPrint will be available on SUSE LINUX when Novell ships the Open Enterprise Server (OES) in late 2004. The iPrint version that ships with Novell Enterprise Linux Services 1.0 is compatible with SUSE LINUX Enterprise Server 8 but not with SUSE LINUX Enterprise Server 9.

While CUPS will meet many of the needs of small to medium printing environments, it does not scale to meet the needs of enterprise customers. iPrint, on the other hand, can host hundreds of printers and process substantial amounts of data that CUPS cannot approach. With CUPS, the conversion into the printer-specific format (the filtering) of all data to be printed takes place by default directly on the server. The advantage is that a print client is not necessary; the disadvantage is that the filtering process devours too many resources when a single server must support hundreds of printers. With iPrint, in contrast, the filtering takes place on the client system, but this requires the installation of printer drivers on the client.

If you are setting up printing in an enterprise environment, check on iPrint availability before making a final decision.

Install CUPS

Once the printer is connected to the network and the printer software installed, the printer needs to be installed on the SUSE LINUX Enterprise Server 9 operating system. Novell recommends using the YaST tools delivered with SUSE LINUX Enterprise Server 9 (or the command line) since third-party tools often have difficulties with SUSE LINUX Enterprise Server 9 security restrictions; using third-party tools often results in more problems than benefits.

CUPS is installed with SUSE LINUX Enterprise Server 9 as long as you select the Default or Full installation options—it's not installed with either of the "minimal" options. If a version of CUPS has been installed on your system previous to the SUSE LINUX Enterprise Server 9 installation, you need to be aware of the distinction between update and upgrade scenarios:

- **Update CUPS** - Software packages are updated but not the configuration files. Queues and the printer daemon cupsd will continue to behave as before, but

new SUSE LINUX Enterprise Server 9 features will need to be configured before they are available.

- **Upgrade CUPS** - Both the existing software packages and existing configuration files are replaced. All new features are immediately available.

Before you install SUSE LINUX Enterprise Server 9, make sure you understand how printing is handled by reviewing the information in the remainder of this section.

From Windows to UNIX/Linux printing

You can configure CUPS printing in many different ways. Only the most typical scenarios are covered here. It's important for those who will configure printing to understand the differences between Windows and UNIX/Linux printing systems, particularly differences in the way the two handle filtering and spooling.

With Windows, the most usual case is for client systems to convert the original data (e.g. plain text, MS-Office documents, or other proprietary formats) into printer-specific format and then send the printer-specific data to the print server (to a printer share via the SMB protocol); the print server then sends the printer-specific data to the printer. The print server does only the spooling. The filtering is done on the client system.

The client systems must care about the actual printer model (the printer drivers must be installed on the client systems). When printers are added or exchanged, additional drivers must be provided and users must download, install, or replace the drivers on their laptop or desktop to deal with this situation. End-users who connect their laptop or desktops to the network can't print until drivers are installed on the laptop or workstation.

With UNIX/Linux, the concepts of spooling (the plain data buffering) and filtering (the plain data transfer) are strictly separated:

- Network protocols are related only to spooling.
- Printer drivers are related only to filtering.

Both must cooperate to get a printout, but they are two different things and mixing the two causes many configuration errors.

With UNIX and Linux printing, client systems send the original data (plain text, PostScript or JPEG) to the print server, specifically to a print queue, via LPD or the IPP protocol. The print server then converts the data into printer-specific format (filtering) and sends the converted data to the printer. The print server does both the spooling and filtering. This means that client systems don't need to know about differences in printer models and don't need printer-specific drivers. The print server handles this information.

The advantage is that end-users can connect laptops and desktops to a network running a CUPS server, run their own cupsd on the laptop or desktop, and print immediately.

For additional detail, see "Intrinsic Design of CUPS for Printing in the Network" under

http://portal.suse.com/sdb/en/2004/05/jsmeix_print-cups-in-a-nutshell.html

If your existing network includes Windows client systems and you replace a Windows print server with a Linux print server, you will need to understand how CUPS handles spooling and filtering before setting up your printing environment.

CUPS spooling

As explained above, spooling is the plain data transfer from Windows client systems to the printer. With CUPS, spooling works without software (without a printer driver).

Two primary scenarios are possible:

Scenario 1: Each printer model must have a matching print queue on the Linux print server. Further, each print queue on the Linux print server must have a matching printer share so that the Windows client systems can send their printer data to the usual receiver. These printer shares are provided by Samba.

Scenario 2: Alternatively, Windows client systems can be configured to send their printer data, not to a printer share via SMB, but directly to a print queue via LPD or the IPP protocol. If all Windows client systems are changed this way, there is no need for Samba on the Linux print server. It is also acceptable for some Windows client systems to send their data via SMB/Samba to the print queue and to let others send their data directly to the queue via LPD or the IPP protocol.

CUPS filtering

Filtering converts original data into printer-specific format. In this case, software—a printer driver—is needed.

With CUPS, filtering can be done via the Windows client system or via the Linux print server, but is usually done on the server. It is possible to convert the original data into the printer-specific format on a Linux client system and let the Linux print server do only the spooling.

Note: It is also possible to convert to the printer-specific format on a Windows NT print server—not from the original data into printer-specific data—but from enhanced meta file (EMF) format. The drawback is that the EMF format is, to a certain degree, printer-dependent.

Filtering via Windows client systems

Windows client systems produce printer-specific data that is often random binary data. A normal print queue on the Linux print server does not accept random binary data because the Linux filtering system isn't able to convert it into printer-specific data (due to inherent design differences). Random binary data cannot be detected automatically.

If filtering is performed on Windows client systems, the Linux print server must be forced to send the data directly to the printer—and not attempt filtering. This is called

"raw" printing. The CUPS printing system can be forced to do raw printing by using the "-o raw" switch in the printing command (: `lp -d queue -o raw`).

Windows client systems can send raw print data in several ways:

- Via SMB/Samba

In this case, you can use an option in `/etc/samba/smb.conf` to enforce raw printing:

```
cups options = raw
```

This way, Samba, which gets the printer-specific data via SMB, forwards it in raw printing mode to the print queue and the CUPS printing system sends it directly (without additional filtering) to the printer.

- Via LPD to a CUPS print server

In this case, the raw option can be set in the configuration for the `cups-lpd` (the daemon that accepts data via LPD for CUPS) - see "man cups-lpd."

```
-o document-format=application/vnd.cups-raw
```

All data that has the MIME type "application/vnd.cups-raw" is not filtered but sent directly to the printer.

- Via IPP to a CUPS print server

If a raw option cannot be set in the Windows IPP software, the only reliable way to enforce raw printing on the CUPS server is to create an additional raw queue for each printer. (Printer manufacturers provide a "raw" option when adding a printer via the CUPS Web front-end.) The Windows client systems then send their data to this raw queue.

If only Windows client systems are used on the network, it is sufficient to have only raw queues.

Filtering via the Linux print server

When filtering takes place on the Linux print server, a pre-filtering process—into PostScript—must be completed on Windows client systems. Normally, the filtering system on a Linux print server cannot convert Microsoft Office documents or other proprietary formats into printer-specific data because an appropriate filtering program is not available.

Since the filtering system on a Linux print server accepts plain text, PostScript, JPEG and some other graphics formats, the Windows client systems must produce one of these accepted formats. As PostScript is the standard printing language under UNIX/Linux printing, the usual solution is to install a driver that produces PostScript on the Windows client systems.

The two PostScript printer drivers most often used are:

- The CUPS Driver for Windows - see <http://www.cups.org/windows.php>
- The Adobe® PostScript® printer drivers - see <http://www.adobe.com/products/printerdrivers/main.html>

Printer protocols

Before you begin printer installation, you must determine which protocol the printer supports. If the manufacturer does not provide the needed information, you can use the `nmap` command (`nmap` package) to detect the protocol. The `nmap` command checks a host for open ports.

CUPS supports the following protocols:

- Socket (for example, port 9100 or 35)
- LPD (port 515)
- IPP (port 631)
- SMB via Samba (ports 137, 138, and 139); Samba supports printers connected to Windows shares

Note: Be aware that some manufacturers modify the standard protocol to deploy systems that have not implemented the standard correctly or to provide functions not available in the standard protocol. Unfortunately, these extensions (which run well on other operating systems) can cause problems on Linux. You might have to experiment with various options in order to achieve a functional configuration.

Printer drivers

Many printer manufacturers do not provide Linux drivers for non-PostScript printers. If this is an issue, check first with the printer vendor. If a Linux version isn't available, you may be able to use one of the common printer languages: PostScript, PCL or ESC/P. Printers usually support at least one of these languages.

To find out if your printer is supported by Linux, check the following sources:

- The SUSE LINUX printer database:
<http://cdb.suse.de/>
or
<http://hardwaredb.suse.de/>
- The printer database:
<http://linuxprinting.org>
- The Ghostscript Web page:
<http://www.cs.wisc.edu/~ghost/>
- Included drivers:

<file:///usr/share/doc/packages/ghostscript/catalog.devices>

PostScript printer description files

PostScript Printer Description (PPD) is the computer language that describes the properties (such as resolution) and options (such as duplex) of PostScript printers. These descriptions are necessary to make use of the various printer options in CUPS. During SUSE LINUX Enterprise Server 9 installation, many PPD files are pre-installed. In this way, even printers that do not have built-in PostScript support can be used.

The best approach in configuring a PostScript printer is to obtain a suitable PPD file and store it in the `/usr/share/cups/model/` directory or add it to the print system with YaST (preferred approach). You can then select the PPD file during installation.

Configure CUPS

As inferred in the previous discussion, there isn't one best way to set up CUPS printing. This discussion treats only a fraction of the possibilities. Before setting up your SUSE LINUX Enterprise Server 9 printing environment, you'll want to be thoroughly familiar with the documents listed at the end of this section.

During SUSE LINUX Enterprise Server 9 installation, many CUPS print options are activated by default. These can be modified later on a job-by-job basis, with YaST, from a terminal window, or with command-line tools.

CUPS can be installed and configured with third-party tools, but we recommend using the command line or the YaST tools delivered with SUSE LINUX Enterprise Server 9. YaST is ideal for facilitating the configuration and is best equipped to handle the SUSE LINUX Enterprise Server 9 security restrictions. Brief instructions for both command-line and YaST methods are included here.

To configure CUPS, you will need the following information:

- The printer's TCP/IP address - which can be obtained from the printer itself or from an administrator
- The LPD queue name - which can often be obtained from the printer's documentation
- The PostScript Printer Description (PPD) file under the directory `/usr/share/cups/mode`

With YaST

Complete the following to configure printing with YaST:

1. Log in as root to the KDE or GNOME desktop.

2. Start the YaST printer module either from the YaST Control Center or from a terminal window:
 - From the YaST Control Center select:
Start Applications > System > YaST
Then select Hardware > Printer.
 - From a terminal window enter:
`yast2 printer`
3. Create a configuration for the printer:
 - If the printer is listed, select it and then select Configure.
 - If the printer is not listed, select Other and then select Configure to set up the printer manually.
4. Follow the on-screen prompts to configure the printer.

Note: For additional configuration information, use the Help windows available with each configuration screen.

When the printer (or queue) is configured

 - The print queue is added to `/etc/cups/printer.conf`
 - A `ppd` file is created for the printer in `/etc/cups/ppd`
 - The name of the print queue is added to `/etc/printcap` (This file is created and updated automatically; avoid changing it manually.)

From the command line

Before configuring CUPS from the command line, make sure you understand how CUPS works in SUSE LINUX Enterprise Server 9. See http://portal.suse.com/sdb/en/2003/09/jsmeix_print-einrichten-90.html

To configure CUPS from the command line, complete these steps:

1. Make sure CUPS is running properly by typing the following:

```
lpinfo -v
```

The command should return information similar to the following:

```
network socket
network http
network ipp
network lpd
direct parallel:/dev/lp0
...
```

2. **Conditional.** If you are updating a previous Samba-client environment with a Windows share printer or if you need to print via Samba, you may need to configure CUPS to print to a Samba spool.

This requires a program called `smbpool`, which is in the Samba-client package that is installed with SUSE LINUX Enterprise Server 9.

Note: This link should already exist; however, it may be missing if you used the update option during installation and the Samba-client was not updated.

- a. As root, from an xterm window, type the following:

```
ln -s `which smbpool` /usr/lib/cups/backend/smb
```

This will link the program "smbpool" to the CUPS directory for the Samba backend.

- b. Restart CUPS with `/etc/init.d/cups restart`

3. Make sure a CUPS administrator has been created (see below) and that a CUPS-specific password has been set for the user root in `/etc/cups/passwd.md5` using the following command:

```
lppasswd -g sys -a root
```

Note: Earlier versions of SUSE LINUX, and other Linux distributions that use CUPS, use `/etc/shadow` to authenticate or verify passwords. With SUSE LINUX Enterprise Server 9, `cupsd` runs as the user `lp` which does not have access to `/etc/shadow`. Instead, the CUPS-specific authentication via `/etc/cups/passwd.md5` must be used as indicated above.

Without such a CUPS-specific password, no one—not even root—can authenticate to `cupsd` or log in at `http://localhost:631/admin` (see next step).

4. Add a printer.
 - a. Access `http://localhost:631/admin` and log in as root.
 - b. Select Add Printer.
 - c. Enter a name, location, and description of your choosing.

Example:

Name: hplj1200
Location: bldg J, floor 7
Description: HP Laserjet b-w

The name can contain letters, numbers, and underscores—but not special characters.

5. Choose the connection type for the device.

Examples:

- Parallel Port #1 or direct parallel:/dev/lp0
- AppSocket/HP JetDirect or network socket
- LPD/LPR Host or Printer or network lpd
- Internet Printing Protocol or network ipp
- Windows Printer via SAMBA or network smb

6. Choose the device.

If your printer is attached to a Windows box and you are printing from a Linux desktop or print server, choose Windows Printer via Samba.

Note: This item will not appear on the list if you did not complete step 2 above.

7. Choose the Device URI.

Examples:

- parallel:/dev/lp0
- socket://network-printer:port
(for example socket://192.168.101.202:9100)
- lpd://lpd-printserver/queue
(for example lpd://192.168.101.202/lpt1)
- ipp://cups-server/printers/queue
(for example ipp://192.168.101.202/printers/funprinter1000)
- smb://smb-server/printer-sharename
- smb://workgroup/smb-server/printer-sharename
- smb://username:password@smb-server/printer-sharename
- smb://username:password@workgroup/smb-server/printer-sharename

URI provides a Windows-type mapping to whatever computer you are connecting to. This can be tricky for Samba printing as there are several forms that URI can take (see smb examples above).

8. Select the make of your printer, select its model, and then choose a driver.

The PPD file contains information about the selected driver and the driver options that are available for your printer. Different drivers often have several available options which result in different printout speed and quality. Select a suitable PPD file for your printer.

In most cases, higher quality requires more system resources.

- For non-PostScript printers, use the "recommended" Foomatic PPD file.
- For PostScript printers, you can use a generic Foomatic/Postscript PPD file, but for best results, use the PPD file from the printer manufacturer.

For additional information, see the section "PostScript printer description files" and the section "General information on setting up PostScript printers" in

http://portal.suse.com/sdb/en/2004/03/jsmeix_print-einrichten-91.html

- For unlisted printer models, try using a PPD file or a compatible model from the same manufacturer or choose one of the "generic" makes and models.
9. Make sure the printer has been added successfully.
 - a. Print a test page.
 - b. View the print queue by clicking Printers and then selecting the printer's name. Any active jobs will appear in the queue.
 - c. View completed jobs by clicking Show Completed Jobs.
 10. Complete any configuration tasks for this printer (such as changing the output resolution to 600 dpi) by selecting
`Printers > Configure printer`

Change Printer Configurations

Printer configurations can be modified using:

- **The YaST Control Center:** Log in to the KDE desktop and select Start Applications > System > YaST and then select Hardware > Printer.
- **The command line:** List all options for a printer by entering the following command:

```
lpoptions -p queue-name -l
```

Change an option using the following command:

```
lpadmin
```

Set Up a CUPS Administrator

If you will be managing CUPS from the Web or through the printer administration tool in KDE, you must set up the user root (or any other user) as CUPS administrator with the CUPS administration group sys and a CUPS password. This can be done as root with the following command:

```
lppasswd -g sys -a root
```

If this is not done, administration is not possible via the Web interface or the administration tool, as the authentication will fail.

Manage CUPS from the Web

To manage printer classes, jobs, and printers from the Web interface, enter the following:

```
http://localhost 631
```

Migrate from LPRng/lpdfilter to CUPS

For instructions on migrating from LPRng/lpdfilter to CUPS, see the *SUSE LINUX Enterprise Server 9 Installation and Administration* manual available at <http://www.novell.com/documentation/suse.html>.

CUPS Files and Commands

CUPS file locations and common printer commands are noted below for your convenience. Note that this is *not* a complete list.

CUPS File Locations

File	Location
CUPS configuration directory	/etc/cups/
CUPS printer definitions	printers.conf
These printer definition files are analogous to the file /etc/printcap on UNIX and Linux systems that use the LPD print server.	
Class definitions	classes.conf
CUPS printer daemon (cupsd)	/etc/init.d/cups This script launches cupsd at system startup. The cupsd daemon administers the local queues and filters or converts data to a printer-specific format.
cupsd configuration file	/etc/cups/cupsd.conf
Jobs submitted for printing	/var/spool/cups The printer daemon cupsd collects the job, determines the type of the data, converts it to printer-specific format, and submits it to the printer.
CUPS logs	/var/log/cups/ The default log level is set to logLevel info in /etc/cups/cupsd.conf; set the level to logLevel debug to get more output for help with troubleshooting Every access to the CUPS daemon from a browser or a CUPS/IPP client is logged in /var/log/cups/access_log.

File	Location
PostScript Printer Description (PPD) files	/etc/cups/ppd/ One file for each printer defined to CUPS describing the printer's PostScript capabilities and physical attributes.

CUPS printer commands

CUPS accepts both Berkeley3 and System V commands. System V commands can also be used to configure queues. Examples are provided in the following table:

Task	Example Command Format
Submit a print job	Berkeley: <code>lpr -P queue file</code> System V: <code>lp -d queue file</code>
	See also <code>man lpr</code> or <code>man lp</code> , or view the following files: <code>/usr/share/doc/packages/cups/sum.html#USING_SYSTEM</code> or <code>/usr/share/doc/packages/cups/sum.html#STANDARD_PARAMETER</code> .
Display print jobs	Berkeley: <code>lpq -P queue</code> System V: <code>lpstat -o queue -p queue</code> If you do not specify a queue, all queues are displayed. To list active print jobs, use <code>lpstat</code> . See also <code>man lpq</code> or <code>man lpstat</code> , or view the file <code>/usr/share/doc/packages/cups/sum.html#USING_SYSTEM</code>
Accept print jobs (<code>/usr/bin/accept</code>)	<code>accept queue</code>
Cancel print jobs	Berkeley: <code>lprm -P queue jobnumber</code> System V: <code>cancel queue-jobnumber</code> See also <code>man lprm</code> or <code>man cancel</code> , or view the file <code>/usr/share/doc/packages/cups/sum.html#USING_SYSTEM</code>
Configure a print queue	All users can display queue options with the following command: <code>lpoptions -p queue -l</code>
Reject print jobs (<code>/usr/bin/reject</code>)	<code>reject queue</code> (makes the printer unavailable for an extended period of time—for repairs, for example)

Task	Example Command Format
Disable a print queue (/usr/bin/disable)	disable queue (example: disable lj4050)
Enable a print queue	/usr/bin/enable queue

SUSE LINUX Enterprise Server 9 printer commands

Use the following to manage print jobs from the command line:

Printer Task	Example Command Format
Stop cupsd	rccups stop or /etc/init.d/cups stop
Start cupsd	rccups start or /etc/init.d/cups start
Save configuration changes to /etc/cups/cupsd.conf	rccups restart or /etc/init.d/cups restart Note: Earlier versions of cupsd used the Reload command. Reload cannot be used with SUSE LINUX Enterprise Server 9 because cupsd is running as lp. As soon as cupsd starts running as lp, port 631 cannot be opened; hence Reload is not possible. For complete information about changes to cupsd, see http://portal.suse.com/sdb/en/2003/09/jsmeix_print-einrichten-90.html
List/check printer options	lpoptions -p queue-name -l
Change printer options	lpadmin

Additional CUPS Information

For additional information about CUPS and the way CUPS is implemented in SUSE LINUX Enterprise Server 9, refer to the following:

- Common UNIX Printing System:
<http://www.cups.org>
- Chapter 13 (printing) in the *SUSE LINUX Enterprise Server 9 Installation and Administration* manual:
<http://www.novell.com/documentation/sles9/index.html>
- Information about the CUPS printing process, changes in the way the SUSE LINUX Enterprise Server 9 printer configuration works—particularly in relationship to cupsd—and changes that have been made to enhance printing security.
http://portal.suse.com/sdb/en/2004/05/jsmeix_print-cups-in-a-nutshell.html
http://portal.suse.com/sdb/en/2003/09/jsmeix_print-einrichten-90.html
http://portal.suse.com/sdb/en/2004/03/jsmeix_print-einrichten-91.html

- Information about printing via Samba Share or Windows Share
http://portal.suse.com/sdb/en/2003/11/jsmeix_print-smb-90.html
- Information about CUPS printing support in Samba 3.0; see both of the following:
<http://www.samba.org/samba/docs/man/Samba-HOWTO-Collection/CUPS-printing.html>
http://www.linuxprinting.org/kpfeifle/SambaPrintHOWTO/Samba-HOWTO-Collection-3.0-PrintingChapter-11th-draft.html#14_8_1

Appendix A: Linux File System

LINUX FILESYSTEM TYPES

There are four types of files in the UNIX/Linux filesystem.

Ordinary Files - An ordinary file may contain text, a program, or other data. It can be either an ASCII file, with each of its bytes being in the numerical range 0 to 127, i.e. in the 7-bit range, or a binary file, whose bytes can be of all possible values 0 to 255, in the 8-bit range.

Directory Files - Suppose that in the directory `x` I have `a`, `b`, and `c`, and that `b` is a directory, containing files `u` and `v`. Then `b` can be viewed not only as a directory, containing further files, but also as a file itself. File `b` consists of information about directory `b`; i.e., the file `b` has information stating that directory `b` has files `u` and `v`, how large they are, when they were last modified, etc.¹

Device Files - In UNIX/Linux, physical devices (printers, terminals etc.) are represented as ``files." This seems odd at first, but it really makes sense. This way, the same `read()` and `write()` functions used to read and write real files can also be used to read from and write to these devices.

Link Files - Suppose we have a file `X`, and type

```
ln X Y
```

If you then run `ls`, it will appear that a new file, `Y`, has been created, as a copy of `X`, as if we had typed

```
cp X Y
```

However, the difference is the `cp` command does create a new file, while `ln` merely gives an alternate name to an old file. If you create `Y` using `ln`, then `Y` is merely a new name for the same physical file `X`.

FILESYSTEM PERMISSIONS

Similar to PC operating system file structures, Linux files are organized into directories on the disk hardware. Each directory can contain any number of subdirectories, each holding files.

Every file and directory on a Linux filesystem contains information regarding permissions (rights) in its inode. The section of the inode that stores permissions is called the mode of the file and is divided into three sections based on the user that receives the permission to that file or directory:

- -User (owner) permissions
- -Group (group owner) permission
- -Other (everyone on the Linux system) permissions

In UNIX, all files are protected under some access control mechanism, so that the owner of a file can deny all other users access to his files. The first column of the long directory list shows the access characteristics of a file, in the form of 10 flags, e.g. `drwxr-xr-x`.

The meanings of the flags are shown below:

Position 1	File type:	d (directory) - (ordinary file) l (symbolic link)	
Position 2-4	Permissions for the owner:	r (read) w (write) x (execute)	binary 4 binary 2 binary 1
Position 5-7	Permissions for other users in the same group:		
Position 8-10	Permissions for all other users:		

Note that a hyphen ('-') denotes lack of the given permission type. For example, `r-x` means that read and execute permission are granted, but not write permission.

Three regular permissions can be assigned to each of these users as indicated in the following table:

Permission	Definition for Files	Definition for Directories
Read	Allows a user to open and read the contents of a file	Allows a user to list the contents of the directory (if they've also been given execute permission)
Write	Allows a user to open, read, and edit the contents of a file	Allows a user to add to or remove files from the directory (if they've also been given execute permission)
Execute	Allows a user to execute the file in memory (if it is a program file) and shell scripts	Allows a user to enter the directory and work with directory contents.

- In order to remove a file, you must have write permission to it.
- In order to view the contents of a directory, i.e., see what files are there, you need read permission for that directory.
- In order to read from a file, write to it, or execute it, you need execute permission for the directory.
- The root user receives the permissions available to all three categories of users.
- Only the owner of a particular file or directory and the root user have the ability to change ownership (`chmod`) of the file or directory.

FILE INFORMATION

The 'a' ("all") and 'l' ("long") options of the `ls` command provide a lot of information about files in a specified directory (if a directory is not specified, then the current directory is assumed). Here is a sample output from typing `ls -al`:

```
drwxr-xr-x 6 ecs4005 1024 Apr 22 13:30 ./
drwxr-xr-x 74 root 1536 Mar 24 12:51 ../
-rw----- 1 ecs4005 188 Apr 13 15:53 .login
-rw----- 1 ecs4005 6 Mar 24 11:29 .logout
-rw----- 1 ecs4005 253 Apr 10 12:50 .xinitrc
-rw-r--r-- 1 ecs4005 516 Apr 10 13:00 .twmrc
-rw-r--r-- 1 ecs4005 1600 Apr 22 10:59 test2.out
```

The output is separated into six columns:

- 1st column - access permissions (see below)
- 2nd column - number of file entries (in the case of directory files)
- 3rd column - owner
- 4th column - size in bytes
- 5th column - date and time of last modification
- 6th column - name

File Permission Scenarios

Situation

A directory has permissions of `777`.

When user `root` goes to this directory and creates a file, based on `root`'s `umask`, this file now has permissions of `644` (`_rw_r_r_`).

Many assume that irrespective of what the parent directory permissions are, users in the `Other` category can only read the file created by `root` and cannot write to it or delete it. Unfortunately, in this scenario, only the parent directory's permissions are evaluated. The permissions set at the file level are over-written. As a result, anyone can delete or write to the file created by `root`.

Solution

If users (owner, group, or other) have write rights to the directory, they can delete any files in that directory—regardless of the file level rights.

To avoid this problem, turn on the sticky bit. (In this case, this means setting permissions of `1777` on the parent directory.) This prevents users from deleting files unless they are also the owner of the file.

File Commands

A few of the most often used Linux commands are discussed briefly in this section.

chmod

You can use this command to change the access permissions of any file for which you are the owner. The notation used is:

u	User (i.e. owner)
g	Group
o	Others
+	Add permission
-	Remove permission
r	Read
w	Write
x	execute

For example, the command

```
chmod ugo+rw .login
```

would add read and write permission for all users to the `.login` file of the person issuing this command.

In some cases, it is useful for a user to deny himself/herself permission to write to a file; e.g., to make sure he/she doesn't remove the file by mistake.

du and df

- The `du` command displays the size in kilobytes of all files in the specified directory along with the total of all those sizes; if no directory is specified, the current directory is assumed.
- The `df` command displays the amount of unused space left in your disk systems.

diff

This command displays line-by-line differences between two ASCII files. If for example, you have two versions of a C source file but don't remember how the new version differs from the old one, you could type

```
diff oldprog.c newprog.c
```

Wild Cards

There are two wild-card characters in UNIX, `*` and `?`.

- The wildcard `*` matches with any string of characters. For example,

```
rm *.c
```

would delete all files in the current directory whose names end with `.c`.
- The wildcard `?` will match with any single character. For example,

```
rm x?b.c
```

would delete all files whose names consisted of five characters, the first of which was 'x' and the last three of which were 'b.c'. Example: `rm prog?.c` will delete all the files (in the current directory) The files `x3b.c` and `xrb.c` would be deleted, while the file `xuvb.c` would not.

In addition,

- `[0-9]` matches character from '0' through '9'
- `[a-z]` matches character from 'a' through 'z'

For instance,

```
rm test[1-3].c
```

would remove `test1.c`, `test2.c` and `test3.c` but not `test4.c`.

ADDITIONAL READING

Many books that explain the Linux filesystem are available. Refer to them for detailed information and interpretation. Some of the information above was abstracted from "Exploring Linux Filesystems" in *Linux+ Guide to Linux Certification*, by Jason W. Eckert and M. John Schitka (ISBN 0-619-13004-05)

CONTACT US

For additional information about Novell, please contact your local Novell Authorized Reseller or system house or call, fax, or email Novell at:

1-888-321-4272 (U.S. and Canada)

1-801-861-7000 (worldwide)

Fax: 1-801-861-8473

E-mail: customer_response_center@novell.com