Migrating from UNIX* to SUSE LINUX® Enterprise Server 9

A Novell® Migration Study

www.novell.com

SEPTEMBER 2004

Novell.

Addresses

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

Novell UK Limited
Novell House
1 Arlington Square
Downshire Way
Bracknell
Berkshire
RG12 1 WA

Compiled by

Novell Solution Creation and Marketing—Linux Team

Contributors

John Beuchert, Global Solutions Director
Kurt Brust, Global Solutions Manager
Justin Steinman, Solutions Manager, North America Solutions
Thomas DiNaro, Nterprise Consulting Technical Specialist
Nathan Wilkey, Solution Support Lead
Doug Clower, Global Solutions Manager
Johannes Meixner, Software Engineer, SUSE LINUX
Meike Chabowski, Product Manager, SUSE LINUX
Joyce Whiting, Solution Development Specialist, Novell
Erica Royer, Solution Development Specialist, Novell

Date

September 2004

Table of Contents

INTRODUCTION

If your organization is investigating whether to deploy Linux* and open source solutions—either now or in the future—you already know how much there is to consider. Decisions range from tactical deployments in edge-of-the-network or Web-serving functions to general infrastructure (like file and print) to enterprise use for application serving. Initially, you may be looking for advice or contemplating a limited proof-of-concept installation, or perhaps you've already made your business decision and are ready for an across-the-board deployment with Linux as the preferred platform.

At whatever point you are along the Linux migration path, you've probably read the hype—the claims and counter-claims—about whether Linux is or isn't ready for the enterprise in the data center or on the desktop. And frankly, the claims probably depend on which Linux migration paper you read most recently and who funded the research. But the fact is that Linux is now the fastest-growing operating system in the world with the fastest adoption in:

- Specific horizontal servers such as DNS/DHCP, Web servers, firewalls, e-mail servers and J2EE* application servers
- Lower-cost replacement of proprietary UNIX* systems such as AIX* on the IBM pSeries*, HP-UX* on the HP 9000, Solaris* on the Sun* Sunfire and various other implementations.

Organizations making the move to Linux usually do so for immediate and ongoing cost reduction with the migration typically keying off such factors as:

- Expiring leases on hardware
- Renewals of software contracts
- Budget cycles
- Compelling cost-saving opportunities

And, of course, migration is easiest when the applications and services running on UNIX are already available on Linux and when there are known real-world successes and documented migration methodologies.

This migration study assumes you've already decided that Linux is the right direction for your organization and focuses on *how* to make the move rather than providing reassurance about *why* you should. It provides insight into what you will be looking at to migrate edge-of-the network infrastructure and basic file, print and e-mail services to Linux—SUSE® LINUX Enterprise Server 9 in particular. The document is intended as a starting point in your discovery; it does not represent all of the options available to you. Other Linux migration scenarios—application migration, desktop migration, migration from other platforms (Windows* or Red Hat* to SUSE LINUX)—are addressed in companion Novell® Migration Studies. UNIX examples, usually based on Solaris, are provided throughout (UNIX and Solaris are used somewhat interchangeably) but can be adapted to other UNIX versions (HP-UX or AIX).

Note: Some of the information in this study is compiled from other sources. Where applicable, the original source is cited.

PLANNING THE MIGRATION

Because migrations can be problematic if not planned carefully, you'll want to make sure your technical staff has the necessary skills to implement and maintain a Linux environment. Novell offers help on both the planning and training fronts.

Planning

Novell Professional Services offers consulting engagements that range from Strategy and Discovery to Requirement Assessment to Planning and Design to Implementation. These offerings help you assess both current and future strategies, discover your readiness for moving to SUSE LINUX, understand 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

Training

Having technical staff trained on Linux can be critical to the success of your migration. We recommend that at least some of your technical staff be Linux certified (LPI level 1 or LPI level 2). Many third-party 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); CLP courses are the best place to start:
  - Course 3036: Linux Fundamentals
  - Course 3037: Linux Administration
  - Course 3038: Advanced Linux Administration
  - Course 301: 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:

www.novell.com/training

PREPARING FOR MIGRATION

Select Linux hardware

Servers and peripheral hardware [such as SCSI adapters, modems and CD-ROM drivers] all tend to be less expensive for Intel* and AMD-based machines than for their RISC counterparts. Maintenance costs are usually less as well.

But not all hardware drivers—particularly SCSI adapters and drivers for graphics, sound, video and network cards—are Linux-compatible. Sometimes the same computer make and model is shipped with a slightly different driver configuration, and these differences can take their toll. Having multiple SCSI adapters of the same make that need the same Linux driver can also cause problems because the machine recognizes only one device when it boots.

Although several large companies produce drivers specifically for Linux, many vendors leave this to the Linux community. Because the hardware market changes rapidly—almost daily—you'll want to monitor the market closely. Before you purchase a server, check with the vendor or the vendor's Web site to determine whether the hardware drivers you need for a specific adapter are available.

In addition, make sure you follow at least the minimum hardware guidelines (including processors, RAM and disk space) for the distribution you are using. Generally speaking, you will need a less robust processor on Linux than on UNIX; 1Ghz on a Solaris box is equal to approximately 750 Mhz on an Intel box because of the additional overhead imposed by the way UNIX code is handled.

Check the following sources for additional hardware information:

- The Linux home page at Linux online: http://www.linux.org
- The Xfree86 Project, Inc.: http://www.Xfree86.org
- The Linux Documentation Project on "Hardware Compatibility": http://www.tldp.org/HOWTO/Hardware-HOWTO/)
- "Migrating from Windows to Linux" (see "Hardware Compatibility") by Humphrey Cheung: http://www20.tomshardware.com/howto/20040329/win_linux-02.html

Make sure you have a backup

Make sure you have a backup to a tape or another hard drive. Disasters can happen—even with Linux.

Locate the documentation

Most of the basic Linux commands, system calls, libraries and system configuration files are documented in manual pages (as they are with Solaris), but don't ignore HOWTO and README files and GUI-based help programs. Additionally, many rpm packages install the source code documentation under /usr/share/doc. These are good resources for additional information.

You'll also want to refer to the *SUSE LINUX Enterprise Server 9 Installation and Administration* manual for detailed information about the services referred to in this paper. The manual can be downloaded from http://www.novell.com/documentation/sles9/index.html.

Make sure applications are Linux-compatible

Make sure a Linux version or alternative is available for any third-party applications you will be migrating from Solaris to Linux. You'll also want to make sure you have the Linux version of the installation CDs (note that CD

installation is different than it is with Solaris).Test applications in a lab or pilot environment before rolling them into production.

Check port availability

Make sure you have the appropriate ports available on Linux for all the services you had running on Solaris; in some cases, Solaris services require different ports than comparable services on Linux. In addition, you may want to customize the port environment for security or other reasons. To find out which ports are in use, access /etc/services on the Linux machine. When you install a product, the installation writes to this services file.

INSTALLING SUSE LINUX ENTERPRISE SERVER 9

The installation process for SUSE LINUX Enterprise Server 9 is simple and GUI-driven; the basic steps are similar to those for the Solaris Operating Environment installation:

- Select the directory location for installation files
- Choose the geographical locale
- Select the software packages to install
- Set the configuration for the keyboard, video card and mouse for X Windows

In most cases, SUSE LINUX Enterprise Server 9 probes the hardware to discover which drivers are needed and prompts for the boot loader installation: LInux LOader (LILO) or GRand Unified Bootloader (GRUB).

Explanations for all steps are clearly documented in the left pane of the SUSE LINUX Enterprise Server 9 installation screens; the complete S*USE LINUX Enterprise Server 9 Installation and Administration* manual is downloadable from

http://www.novell.com/documentation/sles9/index.html

COMPLETING POST-INSTALLATION TASKS

Once installation is complete, you'll want to make sure your server is operating as expected and that you have a way to recover if necessary. As part of this checkpoint, you'll want to:

- **Verify network connectivity**. Make sure the computer is being recognized on the network by pinging the router, gateway or other computers.
- **Keep a system snapshot.** It's always good to keep a snapshot of your source system so you have a way to recover if necessary.
- **Create diskettes**. You'll want both boot and rescue diskettes. Test both to make sure you can use them to boot from Linux.
- **Check applications**. Check all major application packages to make sure they work.
- **Create a non-root account**. You won't want to log in as root unless you are completing tasks that can be done only as root. Use the non-root account for day-to-day activities so you won't inadvertently compromise your system.

BEGINNING THE MIGRATION

General migration steps are noted here for moving typical edge (infrastructure and networking) services as well as Web-server, database, file, print and e-mail services to SUSE LINUX Enterprise Server 9. For additional detail about each service, refer to the corresponding sections below.

- Start by determining which services to migrate.

- Decide which source directories you are moving files from and which destination directories you are moving them to.
- Move the the designated files from Solaris to SUSE LINUX Enterprise Server 9 via secure copy protocol (SCP) or file transfer protocol (FTP); both FTP and SCP are included with SUSE LINUX Enterprise Server 9. Use FTP if you are transferring files internally or SCP if you are transferring files over the Internet.
- You will be moving all HTML files from the /etc./var/docs directory on Solaris to /srv/www/htdocs on SUSE LINUX Enterprise Server 9.
- Manually verify that all necessary files have been copied. If there were 640 files in one directory on Solaris, make sure 640 have been copied to SUSE LINUX Enterprise Server 9.
- Test the Web site if you are moving your Web site from Solaris. Note that there are programs (spiders) that test Web sites; these are run from a Windows machine and test every page on your Web site to make sure there are no errors. For additional information, refer to

  http://www.download.com

  or

  http://www.tucows.com
- Test the migrated services for a day or two in a pilot lab before cutting them over to production. If you will be running both the original Solaris and destination SUSE LINUX Enterprise Server 9 services on the same network, you will need to provide IP addresses—at least temporarily—for both systems. You may also need to tweak the DNS configuration if both services are running side-by-side.
- When you have verified that all services are working correctly, remove Solaris from service.

MIGRATING USER ACCOUNTS AND PASSWORDS

Moving user accounts from UNIX to SUSE LINUX Enterprise Server 9 is not an easy or straightforward task. You can use products such as Novell Account Management (NAM) 3.0, Pluggable Authentication Modules (PAM), LDAP redirection or PADL (PADL is recommended only for experts) to make moving accounts (identities) easier, but not foolproof.

Rather than managing individual accounts on each server, consider porting users to an enterprise directory (such as eDirectory) during the migration to centralize authentication and administration and to increase security.

User account locations

- On Solaris, user accounts are located in /etc/password; on SUSE LINUX Enterprise Server 9, user accounts are stored in /etc/passwd. Both include user name, password placeholder (for /etc/shadow), UID, GID, description, home directory location and default shell.
- Passwords are stored in /etc/shadow on both systems but are encrypted differently (Solaris uses a different algorithm) so they are not convertible.

Tools are available to convert the accounts in the Solaris /etc/passwd directory to LDIF format so they can be imported to an LDAP directory on SUSE LINUX Enterprise Server 9. Passwords are not converted and will need to be reassigned. For a useful /etc/passwd-to-LDIF conversion script, see

- Site: http://www.padl.com/OSS/MigrationTools.html
- Tool: migrate_passwd.pl (migrates users in /etc/passwd)

Other helpful migration tools are also available at this site.

MIGRATING AND CONFIGURING NETWORKING SERVICES

In most cases, migrating networking services (sometimes called edge services) from Solaris to SUSE LINUX Enterprise Server 9 is fairly straightforward because of the similarities in the two systems. Basic information about migrating primary services is included in the sections that follow.

DNS

You'll want to take inventory of the current file structure on Solaris so that you can either replicate it on SUSE LINUX Enterprise Server 9 or change it to better meet your needs. Solaris typically uses the following structure:

`/var/named/named.ca`—contains information about the root name servers

`/var/named/hosts`—contains the local server name and the IP address which may or may not be part of the DNS server

`/var/named/hosts.rev`—specifies one or more reverse domain files

`/var/named/named.local`—specifies the PTR record for the local loopback interface at the IP address 127.0.0.1

`/etc/resolv.conf`—reverses the domain name to the IP address

`/etc/dhcp/inittab`—stores initial information before you perform the implementation

Install DNS

Domain Name Service—along with other networking services—is installed as part of the SUSE LINUX Enterprise Server 9 if you select the LDAP server installation option. The name server BIND (short for Berkeley Internet Name Domain) is included and already configured, so it can be started immediately after installation. All the settings for BIND are stored in `/etc/named.conf`. However, zone data (such as host names and IP addresses) is stored in separate files in the `/var/lib/named` directory.

DNS can be configured with YaST, which provides both Wizard and Expert options. BIND runs as a pure caching-only name server until you configure its own zones.

To start the name server, enter the command `rcnamed start`(you must be logged in as root). If the name server does not start or behaves in an unexpected way, look for the cause in the /var/log/messages log file. Use `rcnamed status` to see whether the server is actually running.

Migrate DNS

This section includes instructions for manually migrating from UNIX DNS to Linux (BIND) using one of two options:

Option one—Create a secondary DNS: If you are currently running a primary DNS on Solaris, you can use the information in the secondary zone file on the Solaris server to create the primary zone on SUSE LINUX Enterprise server 9.

1. Complete a Zone Transfer by running the `rndc` command.
2. Use the secondary zone file on the Solaris server to create the primary zone file for SUSE LINUX Enterprise Server 9.

3. Change from the secondary to the primary using BIND config or change this in the `named.conf` file.

Option two—Replace the DNS server: If you are replacing the UNIX DNS server completely, use these instructions:

1. Create slave entries on the Linux server for each of the zones in your Solaris `named.conf` file as in the following example:

```
zone "example.org" {
        type slave;
        file "s/db.example.org";
        masters {
                10.11.1.3;
        };
        allow-query { any; };
};
```

2. Change the domain name, file path and master DNS server IP address to those for the Linux system. This will cause `named` to do a zone transfer of each of the domains into its respective files.
3. Change `slave` to `master` in `named.conf` (most often found in /etc/named).

   Note: You can also do a zone transfer using `named-xfer` for each of the x number of domains.
4. Edit each of the domain configuration files, changing the NS and SOA records to match the new nameserver.

   `ndc reload` and `named` will now act as the primary DNS server for these zones.

Use Novell eDirectory to host DNS

Novell eDirectory has traditionally used SAP and SLP to search for and advertise network services. DNS was added as a discovery protocol in eDirectory 8.7.1. This enhanced functionality means that if you ask for a tree name that eDirectory doesn't understand (either because you are communicating with a server that doesn't hold a copy of the tree or you are using a standalone application), the machine trying to do the discovery uses eDirectory's discovery protocols in the following order:

- DNS
- SLP
- SAP

Novell recommends putting the eDirectory tree name in DNS using an A, AAAA, or Service (SRV) resource record under the DNS domain the clients are going to use to resolve names. If you use A or AAAA records, the eDirectory servers must be running on the default 524 port. If the servers are using any other port, use an SRV record.

For complete information, see “How Novell eDirectory Works with DNS” in the *eDirectory Administration Guide* at

http://www.novell.com/documentation/lg/edir873/index.html?page=/documentation/lg/edir873/edir873/data/a2iii88.html

DHCP

To move from DHCP on Solaris, you will need to set up DHCP on SUSE LINUX Enterprise Server 9 and then follow the zone transfer information below to manually transfer the zones. Although this is a manual process, it should take only a half hour or so to complete. Once the zone transfer is finished, you'll need to shut down DHCP on Solaris.

DHCP servers (or daemons) provide clients with the ability to "plug and play" when connecting to any network. Using DHCP daemons provides a way to administer IP information without going from workstation to workstation to add it. The core of any DHCP system is the dynamic host configuration protocol daemon, which leases addresses and watches how these addresses are used according to the settings the administrator defines in /etc/dhcpd.conf.

Both a DHCP server and DHCP clients are available for SUSE LINUX Enterprise Server 9. The DHCP server available is dhcpd (published by the Internet Software Consortium).

Use the DHCP module in YaST to set up the DHCP server for the local network. The module can work in two different modes: initial and expert. Use the configuration assistant to walk through the configuration process. DHCP can be set up to store the server configuration locally (on the host that runs the DHCP server) or alternatively to have its configuration data managed by an LDAP server.

The DHCP daemon can be activated with `rcdhcpd start` and is ready for use immediately.

Use `rcdhcpd check-syntax` to check the syntax of the configuration file. If you encounter problems, check the information in the main system log, /var/log/messages, to identify the reason.

On a default SUSE LINUX Enterprise Server 9 system, the DHCP daemon is started in a `chroot` environment for security reasons. The configuration files must be copied to the `chroot` environment so the daemon can find them. The files are copied automatically by `rcdhcpd start`.

To improve security, the SUSE LINUX Enterprise Server 9 version of the DHCP server comes with the non-root/chroot patch applied. This enables dhcpd to:

- Run with the permissions assigned to `nobody` (VSFTPD uses the Linux/UNIX `nobody` user as a part of the default configuration. On most Linux/UNIX operating systems, this user exists by default, but you can add it if not.)
- Run in a chroot environment (`/var/lib/dhcp/`)

To make this possible, the configuration file /etc/dhcpd.conf needs to be located in /var/lib/dhcp/etc/. The corresponding init script automatically copies the file to this directory upon starting. The server's behavior with regard to this feature can be controlled through the /etc/sysconfig/dhcpd configuration file. To continue running dhcpd without the chroot environment, set the variable `DHCPD_RUN_-CHROOTED` in /etc/sysconfig/dhcpd to `no`.

The DHCP server can also be set up at the command line using the following procedure:

1. Install DHCP on the SUSE LINUX Enterprise Server 9 machine (if not already installed) from the rpm package:

   ```
   # rpm -ihv dhcp-*.rpm
   ```

2. Edit the /etc/dhcpd.conf file on the SUSE LINUX Enterprise Server 9 server to modify the variables for your specific environment:
   At the Solaris box

   a. Check the /var/named/dhcptab file and note the IP zone range

   b. Check the subnet

   At the SUSE LINUX Enterprise Server 9 box,

   c. Add the correct IP subnet to the subnet `x.x.x.x`

   d. Add this range to the – `range dynamuc-bootp x.x.x.x  x.x.x.x`

   You can also obtain the lease time and DNSDAMIN values from this file.

   In the example below, the server is assigned an IP address of 10.0.0.1 and provides IP addresses for up to 253 clients.

   Sample /etc/dhcp.conf file

   ```
   #/etc/dhcpd.conf

   server-identifier   dhcp.clonedomain.com;

   default-lease-time  172800;

   max-lease-time      604800;

   option domain-name  "clonedomain.com";

   subnet 10.0.0.0 netmask 255.255.255.0

   range dynamic-bootp     10.0.0.2 10.0.0.254;
   ```

3. Start the DHCP server on SUSE LINUX Enterprise Server 9 by entering the following command:

   ```
   /etc/init.d/dhcpd start
   ```

4. Stop the Solaris DHCP server by completing the following process:

   a. Choose the Services menu in the main window.

   b. From the Service menu, Choose Start, Stop or Restart

   c. Click Stop to stop the DHCP or BOOTP daemon.

   d. Click OK to confirm your actions.

Note: If you plan to leave the Solaris DHCP server on the network, unconfigure DHCP so it won't be activated if the machine is inadvertently plugged in or repurposed without removing DHCP:

Type `/usr/sbin/dhcpconfig`

Choose Yes when prompted to unconfigure the DHCP server.

FTP

Note: The information in this section is abstracted from "Use VSFTP for a secure, reliable FTP server," by Scott Lowe, January 22, 2003. Read the entire article at

http://techrepublic.com.com/5100-6261_11-5034763.html

While there are many FTP servers to choose from, VSFTPD is considered one of the best in terms of stability, scalability and security. If you are using a different FTP server, we recommend that you consider VSFTPD as part of your overall migration effort.

If VSFTPD is not already installed on your system, you can install it using YaST from the SUSE installation media or download it from

http://vsftpd.beasts.org

Adding users

If you want to support anonymous FTP so users can download information from your servers without authenticating, you'll need to create an FTP user. Doing so reduces account administration overhead but also reduces the security of the server because anyone can access the files. To preserve the security of VSFTPD, the anonymous user's home directory must not be owned by the FTP user, and the user should not have any permissions for it. You can use the commands below to accomplish this.

FTP Commands

Command	Result
`mkdir/srv/ftp/`	Creates a dir named /srv/ftp
`/usr/sbin/useradd -d /var/ftp ftp`	Creates a user "ftp" with the home directory /var/ftp. On many systems, this user will already exist
`chownroot.root /var/ftp`	Changes ownership of the /var/ftp directory to the root user
`chmodog-w /var/ftp`	Removes the write permission from groups and others

Next, make sure the /usr/share/empty directory exists. If not, create it with the `mkdir` command.

Finally, install the executable file, help pages and other components not installed by default with VSFTPD. To do this, change to the directory in which you built VSFTPD and type `make install`. This installs everything you need to begin using VSFTPD except a configuration file. You can copy a sample configuration file (vsftpd.conf) located in /int /etc/vsftpd/ to the /etc directory by typing `cp vsftpd.conf /etc`.

VSFTPD modes

VSFTPD can be run in two modes: standalone and inetd/xinetd.

Running the product through the inetd (or xinetd) daemon gives you more control and is the recommended method. Additionally, you should note that as configured, VSFTPD will accept *only* anonymous connections, assuming that you created the FTP user as indicated previously. If you want to allow local users to authenticate, you will also need to configure Pluggable Authentication Modules (PAM). See the PAM section on page 17 for additional detail.

- **Standalone mode**: To run VSFTPD in standalone mode, add a single line to the end of the /etc/vsftpd.conf file that reads `listen=YES` and then execute `/usr/local/sbin/vsftpd &`. (The & tells the program to continue to run but brings you back to a command prompt. Assuming you get no error messages, you can now connect to the FTP server as an anonymous user and get directory listings and transfer files.)
- **Xinetd mode**: If you are running an xinetd machine, refer to the installation instructions included with VSFTPD. These can be downloaded from the following site:
  http://www.vsftd.org

Basically, you will need to:

- Go into /etc/xinet.d
- Edit the file vsftpd by changing the `disable= line` to `no`
- Restart the inetd daemon either via a reboot or `kill -SIGHUP {pid of inetd}`

xinet.d/vsftpd Parameters

Option	Default	Explanation
socket_type	stream	The type of TCP socket to use for this protocol (FTP is a TCP stream)
wait	no	Associated with the socket's ability to accept messages
user	root	The user who will launch this service (Note that VSFTPD reduces privileges as soon as possible after starting)
server/usr/sbin/vsftpd	`no`	The location of the server program associated with this configuration file Change this value to match if vsftpd is in a different location
nice	10	Modifies the default scheduling priority for the process; the range is between negative 20 (highest) and 19 (lowest)
disable	no	If a services is not disabled; it starts when xinetd starts
per_source	no	The number of concurrent connections allowed from the same IP address and is useful for limiting the number of connections from a single site
instances	no	Limits the maximum number of concurrent FTP connections to the server and is useful for limiting server load
no_access	no	Lists the IP addresses that are not allowed to access this service

Using the default configuration file, restart xinetd on the SUSE LINUX Enterprise Server 9 server by typing `/etc/init.d/xinetd restart` at the command prompt.

The edited file should appear as follows:

```
service ftp
{
  socket_type  = stream
  protocol  = tcp
  wait   = no
  user   = root
  server   = /usr/sbin/vsftpd
  disable   = no
}
```

Note: If you previously configured VSFTPD in standalone mode, you need to remove the line `listen=YES` from /etc/vsftpd.conf. Otherwise, `xinetd` will restart, but the VSFTPD service will not work.

You should now be able to connect to the VSFTPD server as an anonymous user, get directory listings and download files.

Enabling controlled access

Setting up an FTP server to distribute software to anyone who connects can be useful in many cases, but you may want to control access to the FTP resources. For example, suppose you want to set up a site just for your customers. You can do this with VSFTPD by making use of PAM. SUSE LINUX Enterprise Server 9 uses PAM for authentication.

VSFTPD comes with a sample PAM configuration. This file needs to be renamed and copied to the *pam.d* directory, and named either “ftp” or with the value specified by the “pam_service_name” parameter in /etc/vsftpd.conf; for example: cpvsftpd.pam /etc/pam.d/ftp.

Next, change the VSFTPD configuration to allow local user logins. To do this, edit the file /etc/vsftpd.conf and remove the comment from the line `local_enable=YES`. Now when you attempt to connect to the server as a Linux user, you will be placed in that user’s home directory.

PAM modules are shared libraries that allow the system administrator to choose how an application will authenticate users. These modules are referenced in the PAM configuration files, where they direct the authentication behavior of an application.

Authentication modules are typically located in:

- UNIX: /usr/lib/security
- Linux: /lib/security

The PAM configuration files are:

- UNIX: /etc/pam.conf
- Linux: /etc/pam.d/<system-auth>

Note: Linux can also use /etc/pam.conf if it is present, but typically, the /etc/pam.d directory is used instead.

The Linux /etc/pam.d directory contains separate files for each application (instead of one configuration for all applications in /etc/pam.conf); these files are named for the application and control its behavior.

When you are converting an /etc/pam.conf file from Solaris to /etc/pam.d/<system-auth> files in SUSE LINUX Enterprise Server 9, it's important to remember that the module names will be different for each platform. For example, Solaris may use the usr/lib/ security/pam_unix.so.1, while Linux uses /lib/security/pam_unix2.so.

Note: The /lib/security path applies only to 32-bit, not 64 bit, architectures. If your server has a 64-bit processor, the path is already present; no path should be given to a PAM module. This way, PAM works correctly and finds the module itself.

SSH/VPN

Set up the VPN client

Before setting up the VPN client, you should meet the following prerequisites:

- You have installed OpenSSH on the office computer you will be connecting to. (OpenSSH is included with the SUSE LINUX Enterprise Server 9 distribution.)
- You have access to the source computer (on the LAN) that is running OpenSSH.
- You have an understanding of networking and TCP/IP in a Linux environment.
- You have an understanding of basic Linux commands.

Note: Edit the /etc/hosts.allow file to add IP addresses that are to have access.

To set up the client, complete the following:

1. Make sure necessary security precautions have been taken:
    - Turn off all unnecessary services on the Linux computer from which you are working.
    - Make sure security patches for the distribution you are using are up-to-date.
    - Use TCP wrappers to restrict the range of IP addresses that can access your computer.
    - Disable SSH root logins on both your computer and the office computer by editing the login file located in `/etc/pam.d`. Rem out the first `auth` line and save the file.
    - Disable password-only authentication for SSH connections.

    Note: If you are using NFS with SSH, disable this option in the /etc/ssh/ssh_config file instead.

    - Use a private/public key pair for authentication rather than a password.
    - Make sure the company firewall is configured to open only TCP port 22 to the source machine.
2. Log in with your regular user identity to the machine from which you are working.

3. Open a terminal session and create a key pair with the `sshh-keygen` command:

   `$ ssh-keygen -t rsa -f ~/.ssh/vpn-key Generating public/private rsa key pair`
   Enter passphrase. (Leave it empty for no passphrase, or press Enter to create a key with no passphrase.)

   Note: Using a passphrase in this instance doesn't add a significantly more security but does make your VPN more cumbersome to use. If you or your employer insist on having one, see man ssh-agent for some tips.
4. Make an SSH connection to the source machine. Hereafter, this will be referred to as session #1.

   Note: It's important to keep this session open since there is a possibility of locking yourself out of your source machine if you type the wrong information.
5. Once connected to the source machine, type `su -` to become root.
6. Make sure /etc/hosts.deny on both machines contains the line: `ALL: ALL`
7. Enter the following lines to `/etc/hosts.allow` on the source machine:

   ```
   ALL: 127.0.0.1

   sshd: a.b.c.d/255.255.255.x
   ```

   Substitute your own machine's IP address and sub-netmask for `a.b.c.d/255.255.255.x`.
8. Start a new shell session on your machine.

   Verify that you can still make an SSH connection to the source machine. If not, go back to session #1 and look at the logs to see what's wrong. To help in diagnosing problems, use

   ```
   man hosts.allow
   ```
9. When you have verified that session #1 is still running properly, close the second SSH session and return to session #1.

10. Working as root on the source machine, make sure the following lines in

    `/etc/sshd/sshd_config` are not commented:

    ```
    Protocol 2

    PermitRootLogin no

    PasswordAuthentication no
    ```

    Note: You may want to consider disabling SSH v. 1 for security reasons (there are numerous protocol vulnerabilities). Many Windows clients don't use SSH v.2, so you will need to change the client version as well.

    Note: See "Understanding and Implementing Security on SUSE Linux", a 2004 Brainshare tutorial, on the Novell innerweb at https://innerweb.novell.com/resourcecenter/item.jsp?itemId=12723

    This tutorial (TUT 303) may still be available on http://www.novell.com, but BrainShare files are often available only temporarily. Visit http://www.novell.com/brainshare/catalog/controller/catalog

11. Save any changes and type

```
/etc/init.dsshd restart
```

12. Return to the shell session on your PC and make sure you can start a new SSH session with the office machine. Again, if you have problems, check the logs to see if you can determine the cause.

13. Return to session #1 and create a non-root user on the source machine that you can use to run the PPP daemon:

```
# useradd vpn

# passwd vpn

Changing password for user vpn.
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
```

14. Configure the `sudo` command so that the `vpn` user is allowed to run the PPP daemon as root.

    Type `visudo` and add the following to the bottom of the file:

```
Cmnd_Alias VPN=/usr/sbin/pppd
Cmnd_Alias IFCONFIG=/sbin/ifconfig
Cmnd_Alias IPTABLES=/sbin/iptables
Cmnd_Alias PS=/bin/ps
Cmnd_Alias KILLALL=/usr/bin/killall
vpn ALL=NOPASSWD: VPN
vpn ALL=NOPASSWD: IFCONFIG
vpn ALL=NOPASSWD: IPTABLES
vpn ALL=NOPASSWD: PS
vpn ALL=NOPASSWD: KILLALL
```

15. Set the SSH options for the *vpn* user to allow login access from your PC:

```
# su - vpn
$ mkdir .ssh
$ chmod 755 .ssh
$ cd .ssh
$ vi authorized_keys
```

16. In the vi session (or text editor of your choice), paste in the contents of the `~/.ssh/vpn-key.pub` file from your machine.

    Save the file and set its ownership and permissions appropriately:

```
$ chown vpn:vpn /home/vpn/.ssh/authorized_keys
$ chmod 600 /home/vpn/.ssh/authorized_keys
```

17. From your network administrator, obtain a second, fixed, LAN IP address for the source machine. This interface will be used to NAT the traffic that travels through the PPP tunnel, allowing the PPP tunnel to connect your machine to the source LAN.

Set up the VPN script

Open a root shell on your remote machine and add the following to /etc/sysconfig/vpnopts:

```
# config file for VPN access to the office

# IP address of the remote machine to be connected

SERVER_HOSTNAME=mypc.officedomain.com

# username on the server that we run the tunnel as

SERVER_USERNAME=vpn

# use these IP addresses for the client and server ends of

# the PPP session

CLIENT_IFIPADDR=192.168.3.1

SERVER_IFIPADDR=192.168.3.2

# change these to match your office network

SERVER_LAN2=10.0.0.0

SERVER_LAN2_IF=10.0.100.0

SERVER_LAN2_MASK=255.255.255.0

# various SSH options for the client side

LOCAL_SSH_OPTS="-P -p 22"

LOCAL_VPNKEY=/home/phile/.ssh/vpn-rsa

# pppd options for the client and server

LOCAL_PPP_OPTS="updetach noauth passive ipparam vpn"

REMOTE_PPP_OPTS="nodetach notty noauth"
```

MOVING NFS TO LINUX

SUSE LINUX Enterprise Server 9 uses NFS 3.0, which supports TCP and larger block sizes. Support is achieved through the kernel configuration process. The nfsd server daemon is implemented similarly to Solaris nfsd in that it runs multi-threaded. Eight threads are started by default, but this number can be tweaked by modifying the startup script, `/etc/init.d/nfs`. This script starts both the client and the server processes for NFS. The authentication daemon `rpc.mountd` uses `/etc/exports` as its configuration file rather than `/etc/dfs/dfstab`. The syntax of /etc/exports is considerably different than /etc/dfs/dfstab in Solaris. Refer to the exports manual page for more information.

NFS and the automounter

Two different automounters can be used with Linux. The first, autofs, is the default that most Linux distributions—including SUSE LINUX Enterprise Server 9—use. The second, amd, came from BSD UNIX but has been ported to many operating systems, including Linux. Both are implemented somewhat differently than the Solaris automounter, but serve the same purpose. In addition to offering NFS, Linux autofs provides access to other types of network file systems and local hardware, such as diskette and CD-ROM drives. In this latter capacity, it serves a purpose similar to that of vold in Solaris. For useful configuration examples relevant to those familiar with the Solaris automounter, refer to "Autofs Automounter HOWTO" by Alvin Oga at

http://www.linux-consulting.com/Amd_AutoFS/autofs-5.html

NFS setup and configuration

The source of the information in this section is the "Linux NFS HOWTO," by Travis Barr and others, first published in August 2002 and currently hosted on

http://nfs.sourceforge.net/nfs-howto/server.html#CONFIG

The Linux NFS home page is hosted on sourceforge. Check there for mailing lists, for bug fixes and updates and to verify who currently maintains this document.

Note: As long as the UID is part of the Linux box, NFS capabilities are automatically included.

/etc/exports

This file contains a list of entries that indicate whether a volume is shared and how it is shared. An entry in /etc/exports typically looks like this:

```
directory machine1(option1,option2) machine2(option2,option3)share
```

Where

- Directory—Indicates the directory you want to share. It can be an entire volume, though it need not be. If you share a directory, all directories under it within the same file system will be shared as well.
- machine1 and machine2—Indicates client machines that will have access to the directory, listed by their DNS address or their IP address (test.novell.com or 10.10.0.8). Using IP addresses is more reliable and secure.
- Optionxx (option 1 and option 2)—Indicates the option listing for each machine and describes the kind of access the machine will have. Primary options are included in the following table:

/etc/export Options

Option	Description
ro	The directory is shared in read-only format; the client machine will not be able to write to it. This is the default.
rw	The client machine will have read and write access to the directory.
no_root_squash	By default, any file request made by user `root` on the client machine is treated as if the request has been made by user `nobody` on the server.

	Important: If `no_root_squash` is selected, root on the client machine will have the same level of access to the files on the system as root on the server. In spite of the security implications, using no_root_squash may be necessary if you want to perform any administrative work on the client machine that involves the exported directories. Use this option with caution.
no_subtree_check	If only part of a volume is exported, a routine called subtree checking verifies that a file requested by the client is in the appropriate part of the volume. If the entire volume is exported, disabling this check will accelerate the transfer.
sync	Specifies the use of async behavior, telling a client machine that a file write is complete—that is, has been written to stable storage—when NFS has finished handing the write over to the filesysytem. This behavior may cause data corruption if the server reboots and the sync option prevents this.

Examples

Typical exports file—If two client machines, `slave1` and `slave2`, have IP addresses 10.10.10.1 and 10.10.10.2, respectively, software binaries and home directories can be shared with these machines. A typical setup for /etc/exports looks like this:

```
/opt    10.10.10.1(ro) 10.10.10.2(rol
/home   10.10.10.1(rw) 10.10.10.2(rw)
```

This example shares `/opt` as read-only with `slave1` and `slave2` because `/opt` probably contains software; however, the benefits in allowing this may not outweigh the accompanying security concerns. On the other hand, home directories need to be exported as read-write if users are to save work to them.

Enterprise exports file—Use this option if you have a large installation with many computers on the same local network that require access to your server.

There are several ways of simplifying references to large numbers of machines. You can give access to a range of machines at once by specifying a network and a netmask. For example, if you wanted to allow access to all the machines with IP addresses between 10.10.0.0 and 10.10.0.255, use the following entries:

```
/opt    10.10.0.0/255.255.255.0(ro)
/home   10.10.0.0/255.255.255.0(rw)
```

Wildcards such as *.novell.com or 10.10. instead of hostnames are acceptable. However, any of these simplifications could cause a security risk if there are machines in your workgroup or local network that you don't trust completely.

Consider the following when determining what cannot (or should not) be exported:

- If a directory is exported, its parent and child directories cannot be exported if they are in the same filesystem. However, exporting both should not be necessary because listing the parent directory in the /etc/exports file causes all underlying directories within that filesystem to be exported.
- Using NFS to export a FAT or VFAT filesystem (such as MS-DOS or Windows 95/98) isn't recommended. FAT is not designed for use on a multi-user machine, and as a result, operations that depend on permissions don't work well. Moreover, some of the underlying filesystem design works poorly with NFS expectations.
- Device or other special files may not export correctly to any non-Linux clients on your network.

- Make sure you keep the UIDS and the GIDS the same to prevent users who may have the same UID/GID from viewing or editing each others' NFS-exported files.

`/etc/hosts.allow` and `/etc/hosts.deny`—These two files specify which computers on the network can use services on your machine. Each line of the file contains a single entry listing a service and a set of machines. When the server gets a request from a machine, it does the following:

- Checks `/etc/hosts.allow` to see if the machine matches a listed description.
- If so, the machine is allowed access.
- If not, the server checks /etc/hosts.deny to see if the client matches a listing in this file. If so, the machine is denied access.
- If the client matches no listings in either file, it is allowed access.

In addition to controlling access to services handled by inetd (such as telnet and FTP), this file can also control access to NFS by restricting connections to the daemons that provide NFS services.

You'll also want to restrict access to the portmapper. This daemon tells requesting clients how to find all NFS services on the system. Restricting access is the best defense against someone breaking into your system through NFS because completely unauthorized clients won't know where to find the NFS daemons if they can't access portmapper.

Caution: Restricting portmapper isn't enough if the intruder already knows how to find the daemons. If you are running NIS, restricting portmapper will also restrict requests to NIS. This is typically harmless since you usually want to restrict NFS and NIS in a similar way, but be aware of the implications. In general, it's a good idea with NFS (as with most Internet services) to explicitly deny access to IP addresses that don't need access.

Use hosts.deny—The first step in explicitly denying access is to add the following entry to `/etc/hosts.deny`:

```
portmap:ALL
```

Starting with nfs-utils, you can be a bit more careful by controlling access to individual daemons. It's a good precaution since an intruder will often be able to work around the portmapper.

If you have a newer version of nfs-utils, add entries for each of the NFS daemons:

```
lockd:    ALL

mountd:   ALL

rquotad:  ALL

statd:    ALL
```

Even if you have an older version of nfs-utils, adding these entries is at worst harmless (since they will be ignored) and will at best save you some trouble when you upgrade. Some administrators choose to put the entry `ALL:ALL` in `/etc/hosts.deny`, which causes any service that looks at these files to deny access to all hosts unless explicitly allowed. While this is the more secure behavior, you'll need to remember the restriction when you install new services or you won't be able to identify why they won't work.

Use hosts.allow—Add an entry to `hosts.allow` to indicate which hosts should be given explicit access. Entries in `hosts.allow` use the following format:

`service: host` (or `network/netmask`, `host` or `network/netmask`)

Here, host is the IP address of a potential client. It may be possible in some versions to use the DNS name of the host, but doing so is strongly discouraged.

If you have implemented the setup above and want to allow access to slave1.novell.com and slave2.novellcom, and the IP addresses of these machines are 10.10.10.1 and 10.10.10.2, respectively, you could add the following entry to /etc/hosts.allow:

portmap: 10.10.10.1 , 10.10.10.2

For recent nfs-utils versions, we recommend adding the following (again, if these entries are not supported, they are harmless):

lockd: 10.10.10.1 , 10.10.10.2

rquotad: 10.10.10.1 , 10.10.10.2

mountd: 10.10.10.1 , 10.10.10.2

statd: 10.10.10.1 , 10.10.10.2

If you are running NFS on a large number of machines in a local network, `/etc/hosts.allow` makes `network/netmask` style entries possible in the same manner as `/etc/exports` as explained above.

Once you have made changes to these files, you'll need to restart the NFS daemon and/or the Portmapper daemon using the following commands:

```
/etc/init.d/nfs restart

/etc/init.d/portmap restart
```

MOVING FROM SOLARIS APACHE TO SUSE APACHE

You can use either FTP or SCP to move from Apache on Solaris to Apache on SUSE LINUX Enterprise Server 9. If you are moving data from an internal location, use FTP. If you are transferring files across the Internet and need tighter security, use SCP. Make sure you have the FTP or SCP server set up correctly before you begin.

Basic instructions for both options are included here.

With SCP

If using SCP, complete the following:

1. Logged in as root, stop Apache on the Solaris machine using `apachectl stop`
2. At the SUSE LINUX Enterprise Server 9 machine, copy the documents to be moved to a destination directory.

```
cd /srv/www/

scp -rvp * www@Solaris.IP:/var/www/
```

3. Once the files have been copied, check and edit the permissions to replicate the configuration of the Solaris machine.

   Note: The httpd.conf on Solaris and SUSE LINUX Enterprise Server 9 are generally incompatible—even if the same version is used. This configuration file will have to be rewritten to include appropriate modules (PHP, mod_ssl, mod_perl), library paths, specific directives for SUSE LINUX Enterprise Server 9, appropriate server root directories and virtual hosts.

4. Test the Apache configuration for syntax by entering

   ```
   apachectl configtest
   ```

   `Syntax OK` should be the response.

5. Start the Apache server on SUSE LINUX Enterprise Server 9 by entering

   ```
   /etc/init.d/apache start
   ```

6. Tail the `/var/log/httpd/error_log` to make sure the configuration is correctly implemented.

7. Have someone familiar with the system test the result with a compliant browser.

With FTP

1. Since you will need root privileges to transfer the files, edit the /etc/pam.d/vsftpd file on the SUSE LINUX Enterprise Server 9 machine to allow root access:

   Place a # in front of the line that reads:

   ```
   auth required pam_listfile.so item=user sense=deny file=/etc/ftpusers
   onerr=succeed
   ```

   Note: You may also need to edit the /etc/vsftpd.conf file to enable PASV mode:

   ```
   #pasv_enable=NO to pasv_enable=YES
   ```

2. Stop and restart the service so any changes will take effect.

   ```
   /etc/init.d/xinetd stop
   /etc/init.d/xinetd start
   ```

3. Return to the Solaris server, log in as root and go to the /var/apache/htdocs directory.

4. Prepare the directory using GZIP:

   gzip -rc * > apache.gz

   Note: The -r gathers all the subdirectories and -c compresses the file to make it easier to transfer.

5. Use FTP to access the SUSE LINUX Enterprise Server 9 box:

   ```
   ftp <ip address of linux box>
   ```

6. Log in to SUSE LINUX Enterprise Server 9 as root.
7. At the ftp> prompt, type `binary` and select Enter.

   Note: You may need to switch to PASV mode by typing quote PASV.
8. Use FTP or SCP to transfer the apache.gz file from the Solaris box to the SUSE LINUX Enterprise Server 9 box:

   Use FTP

   ```
   put apache.gz /srv/www/htdocs
   ```

   Use SCP

   ```
   scp apache.gz webuser@linux.ip:/srv/www/htdocs
   ```

   Note: This may take a while, depending on the size of the file. You will be notified when the file has been transferred successfully.
9. Return to the SUSE LINUX Enterprise Server 9 server and CD to /srv/www/htdocs.

   Note: If you have anything else in this directory, you may want to remove it to reduce the chance for problems.
10. Uncompress the `apache.gz` file. Once this is complete, you should be able to view all your files and sub-directories.

    Note: You may need to stop and restart the http daemon:

    /etc/init.d/apache restart
11. Now that root no longer needs to access the FTP files, return to the SUSE LINUX Enterprise Server 9 box, access the /etc/pam.d/vsftpd file, and remove the # from the following line:

    ```
    auth required pam_listfile.so item=user sense=deny file=/etc/ftpusers
    onerr=succeedline
    ```
12. To view your Web site, open a Web browser and access the following location on the server:

    http://localhost

    Note: If you are running virtual servers, you will need to either transfer the `/etc/httpd/httpd.conf` file from Solaris (as long as the releases are the same) or manually edit the `/etc/httpd/httpd.conf` file on SUSE LINUX Enterprise Server 9 to add the IP addresses for all the virtual machines.

    If you are running anything more than a standard Web server (such as JAVA or PHP), additional configuration of SUSE LINUX Enterprise Server 9 may be necessary.

MIGRATING E-MAIL SYSTEMS

How you set up e-mail services on SUSE LINUX Enterprise Server 9 or port them from Solaris depends on whether you want just e-mail or a complete collaborative environment that includes e-mail, calendaring and scheduling.

If you only need to replicate e-mail services currently hosted on Solaris, you can port Solaris Sendmail to Linux Sendmail. Linux Sendmail is included with SUSE LINUX Enterprise Server 9.

If you need a collaboration environment, consider either Open Exchange or Novell GroupWise® 6.5 for Linux.

Note: This solution presupposes using Sendmail because there is no extra charge on SUSE LINUX Enterprise Server 9. However, if Exchange is being used on the Solaris server, the better option is migrating to Open Exchange.

Install Sendmail

Since e-mail is text-based on both UNIX and Linux systems, migrating Sendmail from Solaris is relatively straightforward. With SUSE LINUX Enterprise Server 9, the YaST mail server module is installed by default as part of the LDAP server. If you decide against using an LDAP server, the YaST mail server module will not work because it depends on LDAP functionality. You will need to set up a mail server with the help of the Mail Transfer Agent (MTA) module. For additional information, refer to "LDAP—A Directory Service" in the *SUSE LINUX Enterprise Server 9 Installation and Administration* manual:

http://www.novell.com/documentation/sles9/index.html

Configure Sendmail

Sendmail is controlled by a configuration file called `sendmail.cf`. Because Sendmail has to read this file to find its configuration every time it is called, the design of the file takes advantage of computer parsing.

Numerous example configuration files are distributed with the Sendmail source; tweaking any one of them will work for most purposes. File notation, while somewhat overwhelming initially, is actually relatively simple.

Generate the Sendmail configuration file

The best way to configure Sendmail is to use m4, a macro preprocessor shipped with Sendmail that cuts the configuration process down to several lines in a master configuration file (these files end in .mc). This master file is then used with m4 to generate a sendmail.cf.

The majority of the configuration is generic except for turning on some features appropriate for the system and tweaking a few options. In most cases, modifying one of the sample m4 master configuration files is all that's needed. You won't even have to look at the sendmail.cf; just create a short .mc file, run it through m4 to create a sendmail.cf, and install the resulting configuration file. Occasionally, complex configurations require additional work.

The following example .mc file is used in generating the sendmail.cf for a standalone machine:

```
include(`../m4/cf.m4')

        VERSIONID(`$Id: configuration.html,v 1.9 2004/07/24 19:45:27 brier Exp $')

        OSTYPE(unknown)


        FEATURE(always_add_domain)
```

define(`UUCP_RELAY', `smtp:uunet.uu.net')

define(`LUSER_RELAY', `smtp:anywhere.com')

MAILER(local)

MAILER(smtp)

The m4 macros in the .mc file look like this:

```
name(arg1, arg2, ..., argn)
```

If any of the arguments to the macro are strings, they must be surrounded by quotes. However, the quoting conventions are different than for most other situations. For example:

```
define(`LUSER_RELAY', `smtp:anywhere.com')
```

Carefully identify which characters are being used to generate the quotation marks. It's important to get this correct.

To generate a sendmail.cf with the above .mc file, simply issue a command similar to the following from the cf/cf subdirectory of the Sendmail distribution:

```
# m4 iu-standalone.mc >/tmp/sendmail.cf
```

Install the Sendmail configuration file

To install the new sendmail.cf, copy the new version into place (always backup the original first) and restart the Sendmail daemon. Many administrators keep a copy of the cf directory, so they can easily modify .mc files and re-generate sendmail.cf files as needed.

Copy users' mail from Solaris

Once Sendmail is set up, you are ready to copy users' mail.

1. Go to `/var/mail` on the Solaris box; each account name will be listed in this directory.
2. Copy users' e-mails to the `/var/spool/mail` directory on the SUSE LINUX Enterprise Server 9 mail server using either SCP or FTP.

   Since both e-mail servers use text-based messaging, you will not need to convert documents.

   Note: You can create a tool to automate this as a cron job since you are just moving mail from one server to another.

MIGRATING THE FILE SYSTEM

Set up the SUSE LINUX Enterprise Server 9 file system to mimic that on Solaris, or consider this an opportunity to consolidate and reconfigure the file structure.

There are three tested methods for transferring files from UNIX to SUSE LINUX Enterprise Server 9. They are listed in order of preference below:

- FTP or SCP
- NFS
- File copies through an intermediate device (such as a drive attached to a workstation or the workstation itself)

Only the NFS method maintains ownership and permissions on the files. The two other options lose this information so you will need to regenerate these values once the files have been relocated.

With FTP

One of the easiest ways to transfer files from Solaris to SUSE LINUX Enterprise Server 9 is using File Transfer Protocol (FTP). FTP transfers typically provide the greatest transfer rates, but at a cost: ownership and permission values will be lost.

FTP is sufficiently robust to handle transfers of large amounts of data. Furthermore, there are a number of smart FTP clients [such as ncftp (interactive) and wget (command line)] that can resume a transfer where it left off. Both of these clients are included with SUSE LINUX Enterprise Server 9.

If you have problems using the -c flag—continue, wget offers similar functionality. There is also a good chance FTP services are already running on the server where the data resides. (If the service is not set up, however, it is trivial, in most cases, to do so.)

SUSE LINUX Enterprise Server 9 includes a basic FTP client and server. Novell recommends using a client (such as ncftp) that supports specifying directories as part of the transfer. Recursive directory retrieval is supported by wget as well. This facilitates retrieving large directory structures such as users' home directories.

Migrating files can be a time-consuming process; even on a fast network (100MB/sec and above), transferring large amounts of data can be the most time-consuming event in a migration.

The basic process for transferring files using FTP is as follows:

1. Where possible, position the two servers sharing the transfer as close together as possible.

   Both servers should be on the same subnet, and if possible, on the same physical network switch. This reduces network latency and increases security during the file transfer.

2. Confirm that the Solaris server holding the data has a configured FTP server. (You will need to know the IP address and login ID).

3. Make sure file and directory permissions for incoming data have been determined on SUSE LINUX Enterprise Server 9.

4. Start the FTP service on the Solaris server containing the files to be migrated.

   Note: FTP is often called ftpd on Solaris and is usually turned off by default for security reasons.

   The FTP service can be configured to run from the super daemon inetd or xinetd.

5. Log in to the Solaris server using the FTP client on the SUSE LINUX Enterprise Server 9 system.

6. Navigate to the data to be transferred.
7. Make sure the FTP client is prepared to place the incoming data in the correct directory (at the command line, use the “lcd” command; for GUI programs, navigate graphically to the correct location).
8. Begin the transfer. For large amounts of data this may take a while.

With NFS

NFS transfers retain permissions but sometimes drop file ownership. User and Group ownership may need to be reset.

To mount a remote NFS volume, issue the following command:

```
mount –t nfs –o rsize=8192,wsize=8192,hard <server:dir> <dir>
```

With a file dump

This method is particularly useful for older UNIX platforms that have neither FTP nor NFS functionality and can be accomplished a number of ways. Decide whether to make an intermediate copy of the data on an online or nearline storage device or to perform a direct copy facilitated by a workstation capable of connecting to both the source and the target file systems.

SETTING UP PRINTING

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

Printing options

Two primary printing options need to be discussed in relation to SUSE LINUX Enterprise Server 9: CUPS and 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 AppSoctket/JetDirect protocols are supported.

An old-style BSD-like printing system, LPRng/lpdfilter, is also available with SUSE LINUX Enterprise Server 9 (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. SMB uses ports 137, 138, and 139.

Novell iPrint

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 Nterprise Linux Services 1.0 is compatible with SUSE LINUX Enterprise Server 8 but not 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 either the command line or the YaST tools delivered with SUSE LINUX Enterprise Server 9 since third-party tools often have difficulties with SUSE LINUX Enterprise Server 9 security restrictions; using them 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 the CUPS update and upgrade scenarios:

- **Update CUPS**

  SUSE LINUX Enterprise Server 9 updates the software packages 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**

  SUSE LINUX Enterprise Server 9 replaces both the existing software packages and existing configuration files. All new features are immediately available.

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 in the way they handle filtering and spooling, so the print system can be configured appropriately.

- **With Windows**, the most usual case is for the client system to convert the original data (plain text, Microsoft Office documents or other proprietary formats) into printer-specific format and then send the converted data to the print server or 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.

With Windows printing, the filtering is done on the client, so printer drivers must be installed on the client systems; when printers are added to the network or exchanged, appropriate drivers must be provided. Users must then download the drivers and install them on their laptops or desktops before they can print to network printers.

- **With UNIX/Linux**, the concepts of spooling (plain data buffering) and filtering (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 distinct entities.

  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 filtering and spooling before setting up your printing environment.

CUPS spooling

Spooling is the plain data transfer from Windows client systems to the printer. With CUPS, as indicated above, spooling works without software, or in other words, 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 Window's 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 to 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

  You can also set the raw option in the configuration for the cups-lpd (the daemon that accepts data via LPD for CUPS):

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

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

  For additional information about sending raw data via LPD, refer to `man cups-lpd`.

- 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, then 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:

  http://www.cups.org/windows.php

- The Adobe® PostScript® printer drivers:

  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 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 directory /usr/share/cups/model/ 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 is no 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 TCP/IP address, which can be obtained from the printer 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 Control Center, select

     Start Applications > System > YaST, and 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 the steps below:

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 smbspool, which is in the Samba-client package 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 smbspool` /usr/lib/cups/backend/smb
      ```

   This will link the program smbspool 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/passwe.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, and lp 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 the computer you are connecting to. This can be tricky for Samba printing as there are several forms that URI can take (see the smb examples above).

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

   Often several drivers are available for each model. Select a suitable PPD file for your printer.

   The PPD file contains information about the selected driver and the driver options that are available for your printer. Different drivers often have several possible configurations, which produce variations in printout speed and quality. 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 the best results, use the PPD file from the printer manufacturer.
   - 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.

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

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

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 -1`
  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, print jobs and printers from the Web interface, enter: `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/sles9/index.html

CUPS files and commands

Some common CUPS file locations and 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 /var/log/cups/access_log logs every access to the CUPS daemon from a browser or a CUPS/IPP client.
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: `lpr -P queue file` System V: `lp -d queue file`
	See also `man lpr` or `man lp`, or view the following files: `/usr/share/doc/packages/cups/sum.html#USING_SYSTEM` or `/usr/share/doc/packages/cups/sum.html#STANDARD_PARAMETER`.
Display print jobs	Berkeley: `lpq -P queue` System V: `lpstat -o queue -p queue` If you do not specify a queue, all queues are displayed. To list active print jobs, use `lpstat`. See also `man lpq` or `man lpstat`, or view the file `/usr/share/doc/packages/cups/sum.html#USING_SYSTEM`
Accept print jobs (/usr/bin/accept)	`accept queue`
Cancel print jobs	Berkeley: `lprm -P queue jobnumber` System V: `cancel queue-jobnumber` See also `man lprm` or `man cancel`, or view the file `/usr/share/doc/packages/cups/sum.html#USING_SYSTEM`
Configure a print queue	All users can display queue options with the following command: `lpoptions -p queue -l`

Reject print jobs (/usr/bin/reject)	`reject queue` (to make the printer unavailable for an extended period of time—for repairs, for example)
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

- 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:
  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_

MIGRATING DATABASE SERVICES

Most vendors that support UNIX database servers also support Linux; for example, Oracle, DB2, Sybase and Informix all provide Linux versions. You may also want to consider migrating databases to an open source database platform such as MySQL.

As long as the database environment has been updated with current patches and is under an existing support agreement, you should have access to the tools and support needed for the migration.

More important than database migration is the consolidation effort. Simply changing the database operating platform rarely results in a significant business benefit. As you consider migrating from one operating system to another, you should also consider:

- Upgrading to the next version
- Consolidating on a single database vendor environment
- Implementing advanced redundancy features

You'll want to assess the size and complexity of the migration by taking account of current database instances, platforms and versions, size, connectivity and peak workload.

It is also important to accurately inventory the server equipment used to manage the databases to identify opportunities to reduce the number of database instances and decommission end-of-life equipment.

An excellent source for real-life scenarios representing all major databases is

http://www.dbforums.com/

See also

http://www5.experts-exchange.com/Databases/

http://shearer.org/en/writing/replacemicrosoft/replace-windows2000-howto.html

The following table lists some of the more highly used databases and provides links to an explanation of the vendor's Linux position and to vendor-supplied tools.

Database	Vendor Position on Linux	Vendor-Supplied Migration Tools
Oracle	Oracle fully supports a move to Linux. They have backed their position by introducing their "Unbreakable Linux" campaign. http://otn.oracle.com/tech/linux/htdocs/linux_techsupp_faq.html#Strategy	http://www.oracle.com/oramag/oracle/02-may/index.html?o32tools.html http://otn.oracle.com/tech/migration/workbench/index.html
DB2	IBM says "with the freedom on Linux and the power of DB2, you have a flexible and scalable solution." http://www1.ibm.com/servers/eserver/iseries/linux/pdfs/db2_linux.pdf http://www306.ibm.com/software/data/db2/linux/tools/	http://www.oracle.com/oramag/oracle/02-may/index.html?o32tools.html http://otn.oracle.com/tech/migration/workbench/index.html

Database	Vendor Position on Linux	Vendor-Supplied Migration Tools
MySQL	MYSQL develops and markets a family of high performance, affordable Linux database servers. http://www.mysql.com/documentation/mysql/bychapter/manual_Installing.html#Windows_vs_Unix	http://www.mysql.com/portal/software/convertors/index.html http://www.mysql.com/portal/software/item-124.html
SQL Server	http://download.microsoft.com/download/6/4/b/64b07be2-0912-4c71-9341-343fc67bec26/SQLLinuxDevCosts.pdf	
Sybase	http://www.sybase.com/detail?id=1028940 http://www.sybase.com/content/1026234/platforms9.pdf	http://www.sybase.com/content/1025612/ASE_Linux_Migration_wp.pdf

TROUBLESHOOTING

Troubleshooting is not much different than with Solaris. The netstat, rpcinfo, and nfsstat commands all perform similarly in both systems. The major difference is that instead of using snoop (as on UNIX), Linux uses a packet-sniffing tool called tcpdump. The tcpdump command provides similar information but has different command-line options and output formats. For example, the equivalent to `snoop -d <interface>` is `tcpdump -i <interface>`.

The full list of command-line options for tcpdump can be found in the manual page. For the latest tcpdump version and documentation, refer to

http://www.tcpdump.org

CASE STUDIES

Abstracts from several SUSE LINUX case studies are included here. While these are specific to SUSE versions previous to SUSE LINUX Enterprise Server 9, they should indicate what you can expect.

In addition to the summarized success stories below, other SUSE LINUX case studies (by industry, by solution or by region) can be downloaded from http://www.suse.de/en/company/customer_references/index.html

California State University, Chico

The College of Business at California State University in Chico replaced several small, individual applications with SAP*. This project started on donated HP-UX systems, but the program grew, their equipment became dated, and the technology bust happened. When the college outstripped the capacity of the RISC servers and their experience with Windows made it an unattractive option for them, they tried Linux.

"[Linux} has lowered our overall cost, and we need fewer students to help with the servers. ...[We are] more successful in delivering services to students and to other universities. My best experience with SUSE has been sleep. [Previously], I monitored the servers from home and had to get up at 6 a.m. to make sure things were still up. ... Our few Windows servers were compromised at least twice in the past year. ... [Linux] has not been hacked—period." Gino Edinger, manager, Chico SAP Help Center.

See the full story at: http://www.suse.de/en/company/customer_references/pdf/chico.pdf

Apollo-Optik

Apollo-Optik is one of the leading retail chains in the optical sector in Germany. In conjunction with partner Econtec GnbH, they replaced their UNIX-based server system with SUSE. Initially, Apollo-Optik moved all German stores to SUSE but has been so pleased that the company plans to expand the solution to their other European companies. Additionally, they anticipate migrating clients from Windows to the SUSE LINUX desktop.

“The advantages of a solution based on SUSE LINUX Standard Server are easy to communicate. Apart from excellent technical characteristics, this operating system delivers all the components needed for deployment in a professional environment: full support, systematic maintenance and investment security.” Peter Bartonik, Sales Representative, Econtec GmbH.

“Together with Fujitsu-Siemens hardware, we reduce the needed investment per store by 50%. This alone is stunning. Additionally, we save a substantial amount on support costs due to the stability and error-free operation of our solution.” Apollo-Optik

See the full story at: http://www.suse.de/en/company/customer_references/pdf/apollo.pdf

Higher Regional Court of Düsseldorf

During an investigation focused on upgrading their RISC hardware, the Higher Regional Court of Düsseldorf decided to investigate moving from UNIX to Linux. One imperative constraint was that their current software, including COBOL sector software (JUKOS), Oracle 9i, and ACUCOBOL-GT (compiler) had to be accommodated by the new Linux system. They chose a SUSE LINUX Enterprise server solution—and they haven't looked back.

“Of course the cost factor played a major role in our decision. In view of the increasing pressure on public budgets, we gladly opt for a solution that delivers excellent, substantially improved results with far less expensive hardware.” Eckhardt Kopatz, Project Manager, Higher Regional Court of Düsseldorf

See the full story at:

http://www.suse.de/en/company/customer_references/pdf/olg_nrw.pdf

ADDITIONAL READING

Research sites

- KnowledgeStorm Linux:
  http://linux.knowledgestorm.com/search/recentsearches/kslinux/12z
- IBM and Linux White Paper Library:
  http://www-1.ibm.com/linux/whitepapers/index.shtml
- IBM RedBooks:
  http://www.redbooks.ibm.com/
- IBM RedPapers:
  http://publib-b.boulder.ibm.com/Redbooks.nsf/redpapers/
- Sun BluePrints program:
  http://www.sun.com/blueprints/

UNIX to Linux migration

- "Getting Started on a Unix-to-Linux Migration," by Ken Milberg
  http://librenix.com/?inode=4616

  This article includes information about application porting and migration project management. Milberg suggests that "the complexity of what you are trying to do is directly related to the amount of system-dependent code that you have," and encourages consideration of several points before beginning:
  "Does the application use standard binaries, or do they depend somewhat on the hardware platform you are running? Is your application based on Java or C++? Are there third-party dependencies relevant to the application that may not even be available on Linux? This is the level of detail that you must drill down to."

- "Linux Overview for Solaris Users," by John Cecere, SunServices, Aug 2003
  http://www.sun.com/blueprints

  This paper provides an excellent overview of the differences and similarities between Solaris and Linux environments. The document discusses Linux terms, compares commonly used Solaris commands and their purpose with their Linux counterparts, discusses system and application libraries, provides reference tables for kernel-related parallels between Linux and Solaris, and gives suggestions for rebuilding the kernel to remove unneeded drivers. Comparisons of partition tables, file systems, boot processes, and system and log files are also included

- "UNIX System Management and Security: Differences between Linux, Solaris, AIX and HP-UX," by Haral Tsitsivas, February 18, 2003
  http://www.giac.org/practical/GSEC/Haral_Tsitsivas_GSEC.pdf

  This paper explores security differences among the several versions of UNIX, including Red Hat Linux. Applicable open source software and management and security tools are also evaluated.

- "RedHat ES 3.0 vs. SUSE Server 8.0: Battle for the Enterprise," Joshua D. Drake, April 25, 2004.
  http://www.devx.com/opensource/Article/20840/0/page/1

  Drake compares the Enterprise Server versions of Red Hat and SUSE. After noting both features and problems with the installation programs, administration and included software, Drake concludes: "This is what a commercial Linux is about—added value. SuSE appears to have a firm grasp on this notion. I typically don't like to provide free PR for companies, but I was honestly impressed with the SuSE offering and was quite surprised with the completeness of the product."

- "Successfully Migrating to Linux: Business and IT Considerations," IBM Global Services, June 2004
  http://www-1.ibm.com/services/us/its/pdf/g510-3885-00-linux-migration-wp.pdf

  This IBM Global Services white paper provides an excellent discussion of the business reasons for moving to Linux and also suggests that "migration is not simply the rollout of new hardware, software and applications. To make it as safe and cost-effective as possible, considerable preparation is required to properly plan for, design, test, optimize, and measure the new system. ... Failure to properly conduct a migration to Linux can, at best, lead to greater costs. At worst, it could put mission-critical computing tasks at risk due to mismanagement and reduce the credibility of an organization's overall Linux initiative."

Application porting

- Technical guide for porting applications from Solaris to Linux:
  http://www-106.ibm.com/developerworks/eserver/articles/porting_linux

- Linux porting and migration tools for various platforms (UNIX, Windows, NetWare and Java applications):

http://developer.novell.com/linux/tools.html#sol

- "Migrating Red Hat applications to SUSE —Technical Overview," by Paul MacKay and Arun Singh, Senior Software Engineers, Novell, Inc., June 2004
  http://www.novell.com/coolsolutions/cooldev/features/a_red_hat_migration_cdev.html

  Although this paper focuses on moving Red Hat applications to SUSE, it contains good information for porting UNIX (and other) applications as well. The article points out that "although the core components of Linux are the same across distributions, for technical, philosophical and business reasons there are differences that developers need to understand in order to "migrate" an existing application from one Linux distribution to another."