

Migrating from IIS Web Server* to Apache SUSE LINUX® Enterprise Server 9

A Novell® Migration Study

www.novell.com

SEPTEMBER 2004



| | | |
|--------------|--|---|
| Disclaimer | Novell, Inc. makes no representations or warranties with respect to the contents or use of this document, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. | |
| Trademarks | <p>Novell, the Novell logo, and ZENworks are registered trademarks and eDirectory and Nterprise are trademarks of Novell, Inc. in the United States and other countries. Versora is a registered trademark of Versora. SUSE is a registered trademark of SUSE LINUX AG, a Novell business.</p> <p>* Linux is a registered trademark of Linus Torvalds. Sun, Solaris, and Java are registered trademarks of Sun Microsystems, Inc. IBM, AIX, S/390 and zSeries are registered trademarks and POWER is a trademark of IBM Corporation. Windows and Windows NT are registered trademarks and Internet Information Server is a trademark of Microsoft Corporation. Intel and Intel Itanium are registered trademarks of Intel Corporation. AMD is a trademark of Advanced Micro Devices. JBOSS is a registered trademark of Marc Fleury. MySQL is a trademark of MySQL AB. All third-party trademarks are the property of their respective owners.</p> | |
| Copyright | Copyright 2004 Novell, Inc. All rights reserved. No part of this publication may be reproduced, photocopied, stored on a retrieval system, or transmitted without the express written consent of Novell, Inc. | |
| Addresses | Novell, Inc. 404 Wyman Street, Suite 500 Waltham, MA 02451 USA | Versora 226 East Canon Perdido St. Santa Barbara, CA 93101 USA |
| Prepared By | Versora and Novell Solution Creation and Marketing - Linux Team | |
| Contributors | John Beuchert, Global Solutions Director, Novell Mike Sheffey, CEO, Versora Nick Lassonde, Software Architect, Versora Meike Chabowski, Product Management, SUSE LINUX Peter Poeml, Software Engineer, SUSE LINUX Kurt Brust, Global Solutions Architect, Novell Joyce Whiting, Solution Development Specialist, Novell | |
| Date | September 2004 | |

Contents

| | |
|--|----------|
| Migrating from Microsoft IIS Web Server to Apache on SUSE LINUX | 2 |
| Business Issues to Consider..... | 3 |
| Security..... | 3 |
| Scalability..... | 4 |
| Open Source..... | 5 |
| Financial Considerations..... | 5 |
| Total Cost of Ownership..... | 5 |
| Licensing Costs..... | 5 |
| Management Costs..... | 6 |
| Soft Costs..... | 6 |
| Planning Your Migration..... | 6 |
| Consulting | 6 |
| Training and Certification..... | 7 |
| Migration Considerations..... | 8 |
| Web Development Language Considerations..... | 8 |
| Technical Considerations..... | 10 |
| Configuration Examples..... | 10 |
| Passwords..... | 10 |
| Password verification..... | 11 |
| File System Hierarchy..... | 12 |
| Move files..... | 12 |
| Set up a virtual host..... | 13 |
| Align file permissions..... | 14 |
| Additional Configuration..... | 16 |
| Contact Us..... | 16 |

Migrating from Microsoft IIS Web Server to Apache on SUSE LINUX

In today's Internet world, the ability to provide reliable Internet communication and collaboration with customers is critical to every company's success. If much of your infrastructure has been built on Microsoft* technology, you are probably serving Web pages via your intranet or publicly over the Internet using Internet Information Server* (IIS). Perhaps with the latest attacks on and vulnerabilities discovered within IIS, you are considering a more reliable Web server platform. Ask yourself these questions:

- Have your expectations been met with IIS?
- How much time do you spend patching your IIS server?
- Has your IIS server been a frequent target of attacks or viruses?
- How often do you have to reboot your server?
- Are there other important tasks that you could be focusing on rather than supporting and maintaining your Web server?

If the questions above prompt you to evaluate your use of Microsoft IIS Web server, and if you are planning to move other network services to Linux*, you might want to consider replacing your IIS Web server with an Apache Web server running on SUSE® LINUX Enterprise Server 9. There are good reasons for switching to Apache and good reasons for running Apache on SUSE LINUX Enterprise Server 9.

This paper, a joint venture of Versora® and Novell® examines a few of the reasons this combination makes good business sense. Issues in making the conversion are discussed as are some of the general processes for making such a move. This paper is not a comprehensive migration plan, but rather an exploration of some of the options that need to be considered. The major components are, of course, Apache and SUSE LINUX Enterprise Server 9.

Apache—a robust, commercial-grade and fully featured HTTP Web server—was developed in 1995 and has collaboratively evolved through the volunteer efforts of several thousand software engineers. Apache is now managed by the Apache Software Foundation (ASF), a not-for-profit corporation incorporated in Delaware, Maryland, USA. It is distributed with SUSE LINUX Enterprise Server 9 as a YaST (Yet another setup tool) package ready to be installed and configured.

Apache has a reputation for being a reliable and stable platform. As a result of its sophisticated features, excellent performance, and low price (it is freely distributed), Apache has become the world's most popular Web server. Several organizations monitor the global Web server market share as well as trends in that market share.

For example, Bath, England, based Netcraft, who has monitored Apache usage continuously for several years, reported in September 2004 that the Apache platform maintains a 67% market share, based on responses from 54,500,000 sites worldwide. Their survey also shows market share growing slightly compared with previous surveys.

You can reference their current survey on

<http://www.netcraft.com>

or

http://news.netcraft.com/archives/2004/08/31/september_2004_web_server_survey.html

The uptime statistics compiled by Netcraft are even more impressive. See

<http://uptime.netcraft.com/up/today/top.avg.html>

Securityspace.com statistics are also widely accepted. Check their latest survey at

http://www.securityspace.com/s_survey/data/200408/index.html

San Diego, California-based Port80 Software, a Microsoft-certified partner, (<http://www.port80software.com>) surveyed the top 1,000 U. S. corporations, reporting that 54% use Apache on their Web sites and have about 20% of their servers running on Apache. This survey indicates a growth rate in the use of Apache over the last year of 4.3 %.

SUSE LINUX Enterprise Server 9 is a scalable, high-performance foundation for secure enterprise computing. Backed by Novell, SUSE LINUX Enterprise Server 9:

- Delivers the first enterprise-class Linux server built on the new Linux 2.6 kernel
- Features advanced I/O schedulers for quick application tuning
- Offers support for the latest hardware and advanced network, storage, and connectivity features for redundancy across peripherals and the network
- Supports clustering for automatic fail-over
- Provides Hotplug services so hardware can be changed without system disruption

SUSE LINUX Enterprise Server 9 also offers rich software-development capabilities through built-in network services and protocols, including CUPS, DNS, DHCP, IMAP, NTP, SLP, Postfix, PXE, Proxy, Samba, SNMP, SMTP and many others. It also includes application and database services—such as Apache (the subject of this paper), JBOSS*, Tomcat, MySQL* and PostgreSQL—and supports popular solutions from hundreds of independent software vendors.

SUSE LINUX Enterprise Server 9 also supports numerous architectures, including x86, AMD64, Intel* EM64T, the Intel Itanium* Processor family, IBM* POWER*, IBM zSeries*, and IBM S/390*.

To access a document describing the numerous technical features of SUSE LINUX Enterprise Server 9, refer to

http://www.novell.com/products/linuxenterpriseserver/sles9_featurelist_technical.pdf

BUSINESS ISSUES TO CONSIDER

Security

Apache. One of the most important issues facing enterprises today is increasing the level of IT security. Billions of dollars are lost every year due to insecure computer infrastructure. The last serious security hole discovered in Apache occurred in January of 1997 and was promptly fixed by the Apache community. Security issues such as denial of service attacks and unauthorized filename listings have also been promptly addressed by the Apache

community (see “Comparing Microsoft IIS and Apache HTTP Server”, Dave Murphy, ISSN 1535-3613).

One of the reasons for Apache's excellent security record is that it is so well reviewed by the Apache and open source communities. Version 2.0 has been in use for a couple of years adding to the probability that major issues have been identified and resolved. For current information regarding Apache security, refer to the Apache ChangeLogs and reference the entries marked “security.” Relevant issues are nearly all marked. See

http://www.apache.org/dist/httpd/CHANGES_1.3

http://www.apache.org/dist/httpd/CHANGES_2.0

Another reason for this record is that a default Apache installation includes only the basic http server. Many additional modules are available and can easily be added. Conversely, Windows 2000 and IIS install several external .dll files and FrontPage server extensions by default, rather than allowing them to be configured as needed. This adds complexity that creates additional potential for vulnerability.

In addition, installed Apache components run as a “non-privileged user” which results in minimal damage if a buffer overflow occurs. IIS, on the other hand, gives root permission to any user thus allowing system-level access which can be catastrophic in the event of an attack on a Web server because it allows anyone who has root permission to access, change, and delete any file on the system.

SUSE LINUX Enterprise Server 9 is being evaluated for compliance with common criteria controlled access protection profile, Evaluation Assurance Level 4+ (CC-CAPP/EAL4+). Certification is expected shortly; its predecessor, SUSE LINUX Enterprise Server 8, achieved the highest level of security and operations certification ever achieved in the Linux market—Common Criteria Evaluation Assurance Level (EAL) 3+.

Other security features include:

- Rich user management, authentication, and access control
- Support for encrypted file systems
- Easy-to-use certificate authority and management
- Virtual private networking
- Integrated firewall and proxy services
- Automated monitoring and intruder detection

Scalability

Apache allows Web servers to run in a multi-threaded mode. Different multi-processor modules (MPMs) are available allowing some systems to scale better on high-end servers, while others are better suited to single processor machines. Apache is run by some of the most intensively accessed Web servers in the world including Google and the BBC. Its ability to scale can be illustrated with the hundreds of thousands of simultaneous connections to sites like these.

SUSE LINUX Enterprise Server 9 is the first enterprise-class Linux server built on the new 2.6 kernel which offers improved performance and scalability for large-scale Linux deployments. With advanced memory management and processor support, Native Posix Thread Library (NPTL), advanced I/O capabilities and the unique class-based kernel resource management (CKRM) feature, SUSE Linux Enterprise Server 9 offers the industry's best performance and linear scalability.

Open Source

SUSE LINUX and Apache are open source which means that companies can access the source code. The open source model is what has attributed to the wide array of Apache modules that are freely made available as part of the open source ideology; as a result, SUSE LINUX Enterprise Server 9 is more secure, more reliable, and more scalable.

While many companies may not need or choose to access source code, other developers will—in order to improve or customize the code for their own needs. Any improvements are then made available to the entire open source community. Developers intend to enhance open source technologies for as long as those technologies are relevant to the market place. Additionally, enterprises can contract vendors and developers to maintain and modify source code as part of a support agreement. The open source model increases innovation and reduces costs—two things that interest most corporations.

FINANCIAL CONSIDERATIONS

Total Cost of Ownership

Total cost of ownership (TCO) measures all of the costs associated with purchasing, deploying, and maintaining a system. The cost of hardware, software, consulting, training, and labor are some of the metrics used in this measurement.

While it may be difficult to quantify the TCO of your IIS servers, some studies have suggested some of the financial metrics for this equation. In a study done by the Robert Frances Group, the three-year TCO of a Microsoft Web server was \$190,000, while a Web server running on Linux reached a \$74,000 three-year TCO. Additionally noted was the Linux Web servers' "ability to massively scale horizontally without paying additional licensing fees (which) can yield significant cost savings over the long term" (see "How important is it for your business to be secure?" The Robert Frances Group, September 2002).

Licensing Costs

According to a Forrester Research study conducted in 2004, reducing the company's overall operating costs is the top priority of 69% of IT organizations (Ted Schadler, Vice President, Software Research, 2004 Forrester Research). Moving Microsoft Windows Web servers to SUSE LINUX Enterprise Server 9 is one way to reduce overall operating cost as licensing fees are significantly less. When comparing the licensing costs of various Web server configurations running on Microsoft Windows, Microsoft IIS is at least three times as expensive.

Management Costs

With an array of unique management features, SUSE LINUX Enterprise Server 9 is easy to deploy, configure, and maintain across the enterprise:

- YaST enables easy installation and configuration of the operating system, network services, storage, clusters, and even applications.
- AutoYaST enables “no touch” deployment of Linux across the infrastructure.
- Open application programming interfaces (APIs) and support for the Common Information Model (CIM) standard allow integration with third-party management solutions.
- Class-based kernel resource management (CKRM) allows mainframe-like partitioning of large-scale servers.
- Directory-enabled configurations provide centralized management.
- Built-in support for Novell ZENworks® Linux Management offers an optimal solution for keeping Linux servers and applications up-to-date and secure.
- A common code base ensures the use of consistent management tools and automatic software updates—across all Linux deployments—for maximum flexibility and cost savings.

Soft Costs

There are other costs to consider that are difficult to estimate without organization-specific information. Your company will best be able to assess the dollar value implicit in the questions that follow:

- Will maintenance and support be more or less expensive?
- How much retraining will your technical staff need?
- How long will it take you to recoup the costs involved in making the migration?
- Do you have the staff expertise—and the time—to complete the migration in-house or will you need third-party help?
- Are there other “soft” costs that need to be factored into the equation (users' natural resistance to change, for example)?

PLANNING YOUR 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 SUSE LINUX environment as there are major operating system and administrative differences between Windows and SUSE LINUX. Novell and Versora offer help on both fronts: consulting and training.

Consulting

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

moving to SUSE LINUX, provide information about how to best approach a migration, and finally, help you implement your migration plans.

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

Versora Professional Services provides consulting services that focus on migrating and integrating Windows and SUSE LINUX Enterprise Server 9. These migration services address operating systems, Web servers, database servers, collaboration and messaging servers, and desktops. Versora can provide your company with a cost analysis and a project plan that present the business case for integrating Apache and SUSE LINUX Enterprise Server 9 into your current environment.

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

Training and Certification

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

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

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

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

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

MIGRATION CONSIDERATIONS

The remainder of this document highlights some of the areas you'll want to consider in making the move to an Apache Web server.

Web Development Language Considerations

Apache can take advantage of several development languages, varying from PHP to Perl, Python, and Java via JSP.

While many languages can be used to write a Web application for IIS, the vast majority are written in Visual Basic (VB) and are served via Active Server Pages (ASP). Microsoft hopes to change that with ASP.NET (see below), which works via the .NET framework and allows pages to be written in any .NET language (though VB.NET will most likely be the language of choice for most ASP.NET developers).

There are many limitations to ASP such as slow execution speed and poor ability to scale; generation of new code is slow and ASP is not cross platform (it runs only under Windows). These are among the reasons Microsoft has developed ASP.Net. But many Web developers have been formally trained in writing ASP, spending from 4-6 months to learn the development language and may be reluctant to change.

Since millions of Web sites have been written in ASP, it can be difficult for companies to move from this legacy environment to a new platform—whether another Windows platform or Linux—but there are good reasons for making the move:

- The most recent release, ASP 3.0, is the last version of the language. There won't be an ASP 4.0.
- Microsoft will discontinue support for Windows NT* at the end of 2004 as the operating system reaches “end of life.” Some IT Managers might ask, “Why do I need to upgrade; my NT server works just fine?” But the issue is that companies will be on their own to fix any problems—such as a new virus or worm or a vulnerability or security breach—no more Microsoft patch releases.

If you are currently using ASP and are considering the use of a new development language, there are two obvious paths: PHP and ASP.Net.

PHP. PHP, which is open source and unencumbered by licensing fees, is, some believe, the superior choice of all Web-programming languages. The reasons are speed, excellent scalability, excellent memory management, integration with the MySQL database, no fees for adding additional modules (as there are with ASP), cross platform development portability, and open source code that can be modified to fix bugs or create new functionality to meet an organization's unique needs. Another significant advantage is the large number of ready-to-use solutions that can be found on the Web.

In addition, developing Web pages is quicker than with ASP (two-to-three times), and requires less investment in expensive server hardware. In addition, PHP requires no “formal” training, as with ASP. There are, however, numerous training sources for PHP from on-line tutorials to books available at local bookstores. Many industry experts suggest that the transition from ASP to PHP is much easier than from ASP to

ASP.Net since PHP is more like ASP than ASP.Net is, thus reducing the time and cost associated with a skills transfer.

ASP.Net. Microsoft will encourage a move to ASP.Net—and that's OK. ASP.Net is also an excellent language for developing Websites and runs well under SUSE LINUX Enterprise Server 9 running Mod Mono. While the names ASP and ASP.Net are very similar, they are significantly different languages. Proponents of ASP.Net like the flexibility to develop in several different languages, the GUI that provides a unified user interface, and Windows forms that expedite development.

Unfortunately, ASP.Net is not natively cross-platform. If you need to run ASP.Net code under SUSE LINUX Enterprise Server 9 or an operating system other than Windows, you need to use Mod Mono. Another shortcoming is that ASP.Net is a proprietary language. Any critical bugs discovered must be submitted to Microsoft for resolution.

When migrating an ASP site from IIS, you have a major choice to make. Should you continue to run ASP or convert to a different language?

If you decide to migrate to PHP, currently (September 2004), there's one way to ease this migration and that's using a program called asp2php, an open source project that seeks to automatically convert ASP sites over to PHP. While not perfect, this program makes many of the tedious changes that must be made when converting to PHP easier.

asp2php is available at
<http://asp2php.naken.cc/>

If you are not ready to switch to PHP because your developers are familiar with and prefer Visual Basic or because you don't want to invest time in sites that haven't been touched in years, you can still move to Apache and keep your existing ASP code. Two different commercial products are available to help in this transition:

- Sun Java System Active Server Pages, available at <http://www.sun.com/software/chilisoft/>
- Stryon's Instant ASP (iASP), available at <http://www.stryon.com/products.asp?s=1>

Both products let you host ASP on almost any platform on which Apache runs. However, you may need to make a few site-specific changes.

If you've already made the switch to ASP.NET, you can take advantage of the Mono project, an open source implementation of the .NET framework, including ASP.NET, that can be hosted via the Apache module mod_mono. This project is supported by Novell and has enjoyed significant participation from and has visibility with other Linux companies as well. VB.NET is still in its infancy with Mono version 1.0, but better support is planned for version 1.2, targeted for Q4 of 2004.

Information about the Mono project is available at
<http://www.mono-project.com/>

Technical Considerations

While Apache supports many of the same features as IIS—both use similar models of hosting multiple Websites on one Web server, both allow what appear to be Web subdirectories to be located at various places in the file system, and so forth—there are considerable differences in the way the two are configured. These differences are the result of differences in assumptions in the Linux and Windows' worlds (different focuses on security versus usability) to differences in the coding styles of the programmers involved.

The main difference is the way settings are configured.

- With IIS, all configuration is done via the management console and all changes are made with one set of graphical tools.
- With Apache, several different graphical tools are used. Some tools (such as a YaST2 module) plug in to KDE's Control Center while others (such as Webmin) are accessible via a Web interface.

However, some tools will not allow you to change every possible feature (YaST2, for example, has a very small subset of features), and other tools may even remove some settings you have configured manually (as with some versions of Webmin). For this reason, most experts suggest editing configuration files manually. If you go this route, the following directive is invaluable:

<http://httpd.apache.org/docs-2.0/mod/directives.html>

Configuration Examples

Several configuration examples are discussed below.

Passwords

There are many ways to transmit passwords over the Web:

Basic Authentication. For years, the main option was what is now called Basic Authentication. The client machine sends the password as plain text to the server, which uses one of several available methods to validate the password against its password store.

Digest Authentication. Since sending clear text across the Internet isn't secure, Digest Authentication was developed, but wasn't implemented by every browser; instead, Basic Authentication over an SSL connection became the main source of security.

NTLM Authentication. One more form of authentication also gained acceptance around this time—NTLM Authentication. NTLM sends hashes of the password to the Web Server in the default Windows password format, in much the same way Windows networking does and without prompting the user for a password when run through Internet Explorer on the local area network. This format gained acceptance as the easiest way to provide Web security over Windows-based corporate networks. Originally, only Microsoft Internet Explorer and Microsoft IIS could take advantage of this technology, but in recent years Mozilla (and derivatives) and Apache (through add-on modules) also take advantage of NTLM.

Password verification

IIS verifies passwords in one of two ways: on either the local machine or against the domain controller the server is connected to. This poses a problem for migration as neither of these formats allows easy access to passwords, making them difficult to migrate. (This is really a good security feature, as strong passwords *should be* a one-way street. If the password can be retrieved, it can be hacked.)

Several vendors have created applications and tools to help in the password migration process as indicated below.

Local Machine

If passwords are stored on the local machine, a password hash can be retrieved with one of several tools.

BindView has created an application, `pwdump2` (available under the GPL), that allows administrators to get a dump of the users and their respective password hashes. These hashes can then be copied to the Apache server. You'll need some Apache modules that can verify different authentication schemes against these hashes. Apache's `mod_auth` provides Basic Authentication, but does so against crypt, MD5, or SHA format encryption.

At the time of writing, `pwdump2` is available at:

http://www.bindview.com/Support/RAZOR/Utilities/Windows/pwdump2_readme.cfm

Versora has released a modified module (`mod_auth_basic_ntlm`) that, instead, verifies the password against the NTLM hashes. Also from Versora, is `mod_ntlm_from_file`, which provides NTLM style authentication versus a file with stored NTLM hashes.

Additional information about Versora's Apache modules can be found at:

<http://www.versora.com/>

Domain Controller

If the passwords were previously verified against a domain controller, Apache can be made to do that too. If you decide to take this route, you need to add a user in only one location to update both domain and Apache users. Performing NTLM Authentication against a domain is as simple as installing and configuring the original `mod_ntlm`.

`mod_ntlm`. Since `mod_ntlm` authenticates against the domain controller for every request, there are some performance issues with using it. Also, `mod_ntlm` provides only NTLM Authentication. For additional information, see

<http://modntlm.sourceforge.NET/>

`mod_ntlm_wb`. A new fork of `mod_ntlm`, currently under development and tentatively titled `mod_ntlm_wb`, relies on `winbind` from the Samba project for authenticating against the NTLM backend. `Winbind` was designed so that other projects that need to communicate with a domain controller can share Samba's interface, thus allowing fixes and upgrades to be maintained in one place.

mod_ntlm_wb also plans to support Basic Authentication against an NTLM backend. For additional information, see

<http://source.grep.no/>

File System Hierarchy

One of the more obvious steps in migrating from IIS to Apache is moving Web documents to your new server. By default, IIS stores its documents in `c:\inetpub\wwwroot`. Apache on SUSE LINUX Enterprise Server 9 uses `/srv/www/htdocs` for the main server [in conformance with the Filesystem Hierarchy Standard (FHS)]; see <http://www.pathname.com/fhs/>.

Move files

Moving files from one server to another can be quite easy; using Samba, an open source implementation of Microsoft's file sharing protocol, is discussed here.

1. Make sure file sharing is enabled on your Windows server, and then share out the folders you need to migrate (your document root, for example).
2. At the SUSE LINUX Enterprise Server 9 machine, log in as root so you'll have the permissions needed to edit the document root.
3. Temporarily grant Write access to the document root. Launch Konsole and type:

```
sudo chmod 777 /srv/www/htdocs
sudo mkdir /srv/www/vhosts
sudo chmod 777 /srv/www/vhosts
```

Enter the root password for the server if prompted.

Note: When you've completed this process, you'll need to reset the file permissions as needed.

4. Use the built-in graphical tools to copy the files:
 - a. Click on the desktop icon named Network Browsing and then click on Windows Network. You should see a list of domains.
 - b. Click on the domain for the server you are migrating from.
 - c. Click on the server you are migrating to.
 - d. Enter the user name and password of an administrative user.
5. Open another Konqueror window (Control-N, or Location->New Window), and browse to the Apache document root location (`/srv/www/htdocs` or subfolders of `/srv/www/vhosts/`). You can just type this path in the location bar.
6. At the Windows server window, highlight the IIS folders that you shared and want to move, and drag them over to `/srv/www/htdocs`.
7. At the popup menu that appears, choose Copy here.

Depending on the number of files being copied, this can take a while.

Set up a virtual host

This task discusses setting up a virtual host with a handful of virtual directories in order to get a feel for editing configuration files manually.

In order to save changes to the Apache configuration file, you'll need to edit that file as root. (For security reasons, we don't recommend logging in to your server as root unless you need to make changes.)

To manually edit `httpd.conf`, complete the following steps.

1. Launch Konsole, and type:

```
sudo kwrite /etc/apache2/httpd.conf
```

The highly modularized `httpd.conf` file included with SUSE LINUX Enterprise Server 9 will be displayed. These files are subdivided into much smaller components than is typical, making system updates easier.

The main configuration file includes detailed comments explaining each setting.

You'll see that to add a new virtual host, you simply need to add a configuration file for it in the `vhosts.d` directory.

2. Make a copy of the example template (with a command like: `sudo cp /etc/apache2/vhosts.d/vhost.template /etc/apache2/vhosts.d/migratedvhost.conf`).
3. Open the copy with `kwrite` (`sudo kwrite /etc/apache2/vhosts.d/migratedvhost.conf`).
 - Near the top you'll see `<VirtualHost *:80>`, which means that this host will listen on any incoming connection on port 80 on any IP address.
 - A few lines down is the most important directive, `ServerName`. This directive tells Apache which `VirtualHost` to use when making a connection. Change this to the first of the virtual hosts you wish to migrate (so it says something like `ServerName www.suse.com`).
 - Locate the `DocumentRoot` directive. This directive indicates which directory to serve this virtual host from.
 - Change this to wherever you copied the files (somewhere in `/srv/www/vhosts/`; for example, `DocumentRoot /srv/www/vhosts/www.suse.com/`)

4. For practice, add an additional directive (`DirectoryIndex`—one that isn't already in the template).

Each directive can be placed in only a limited number of locations in the configuration file; in this case, locate `DirectoryIndex` inside a virtual host or in a directory entry.

To change which document gets loaded by Apache when this virtual host is loaded, simply add, under the `DocumentRoot` directive, a new line saying: `DirectoryIndex default.htm` or whatever the default document was on your IIS machine.

If you used multiple default documents in different directories, you can add them in succession, such as: `DirectoryIndex default.htm default.asp index.php`. Replace the rest of the instances of `dummy-host.example.com` throughout the template with real paths of your migrated host.

5. Test the configuration by directing the SUSE LINUX Enterprise Server 9 computer to load from the new server instead of the IIS machine.

- a. Edit the hosts file (by running `sudo /etc/hosts`) and add a line at the end of the hosts file that says `127.0.0.1 www.suse.com`.

Once this has been done, any time you visit `www.suse.com`, you will connect to your local machine instead.

- b. Before connecting, you'll also need to restart Apache by running the `rcapache2` script:

```
/etc/init.d/apache2
```

As with older versions of IIS, Apache must be restarted for any changes to take effect (because the configuration file must be parsed, a task too time consuming to complete for each request; IIS changes settings directly in memory.)

- c. Launch a Web browser and visit `www.suse.com` (or whatever your migrated virtual host is).

You should see your default document (though perhaps not parsed correctly if it's ASP, PHP, or the like) from your virtual host.

This completes a very basic Apache setup.

Align file permissions

The security models used with IIS and Apache are vastly different. IIS is tightly integrated with Windows and uses the exact same authentication rights as Windows itself. While this leads to easier, central user management, it also leads to security problems.

For instance, if a user is created for Web access, by default that user will also have permissions to log on locally, connect over Windows networking, and possibly even connect directly to an SQL server database. These permissions must then be disabled for that user. Apache, on the other hand, by default stores its own user names, passwords, and permissions. A user created for Apache will be able to connect only via Apache. Keep in mind that it is possible to make Windows create new users without as many permissions, and it is also possible to configure Apache to handle all users from a central repository of users.

Another major difference in file permissions is the way a logged-in user can access a file. When a user logs on to IIS, IIS uses the user's access rights to access any file. Thus if user *joe* needs access to upload to a directory, then *joe* must be given write permissions to that directory. You will also need to allow Write access for the directory in the IIS console—before *any* user can upload. Apache, on the other hand, will always run as the user it's configured as (*wwwrun* on SUSE LINUX Enterprise Server 9. User *nobody* is not commonly used.)

In order to give *joe* access to upload to that same directory, you must set write permission for *nobody* on that directory. Doing so, however, gives other users complete Write access to the Apache directory.

Avoiding this situation is a bit complicated. Let's first see what the configuration should be and then subdivide the configuration into components. The example below can be used in any virtual host, directory, or file directive:

```
AuthType Basic
AuthName "Password Area"
AuthUserFile /etc/httpd/basic.passwd
AuthGroupFile /etc/httpd/basic.group
AuthAuthoritative On

<Limit GET POST PROPFIND OPTIONS CONNECT>
    Require valid-user
</Limit>

<Limit PUT DELETE PATCH PROPPATCH MKCOL COPY MOVE LOCK UNLOCK>
    Require user joe
    Require group uploaders
</Limit>
```

- The `AuthType` and `AuthName` directives turn on basic authentication. We've already seen a configuration like this when talking about `mod_ntlm_from_file`.
- The `Require` directive tells Apache which users or groups are required to access this page. `Require` commands are or'ed together, meaning that lines

```
Require user joe
Require group uploaders
```

will require either the user *joe* or any user in the group *uploaders*. The word *valid-user* is a keyword that means *any user at all*.

- The `Limit` command is the tricky part but allows us to have different requires per different types of access. Given the example above, we have two different groups: one for downloading and one for uploading.

Where do these names come from? The `Limit` directive works on HTTP methods and so we break those directives down into two different types. Note that unlike IIS, there is no way to limit whether a user can execute an application.

If you again look at the configuration, you can see an interesting tidbit. In order to set permissions in IIS, you set the general level access through IIS itself, and user level access on the files/directories in question. In Apache, you set the general level access on the file or directory, and the user level access through Apache's configuration file.

Additional Configuration

There's still a lot left to configure via Apache's configuration files. Depending on your needs, you may want to set up virtual directories or files, set timeouts, tweak connection settings, or change the error pages people receive when a error occurs. You will probably even want to set up PHP, mod_perl, JSP, or Sun Java System Active Server Pages 4.0 (so you can run your ASP pages from IIS on Apache). For information on configuring every setting possible in Apache, go to the Apache Documentation Project at

<http://httpd.apache.org/docs-project/>

Information about how to set up different languages or run time environments should be available from the same place you acquire those systems.

CONTACT US

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

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

1-801-861-7000 (worldwide)

Fax: 1-801-861-8473

E-mail: customer_response_center@novell.com”