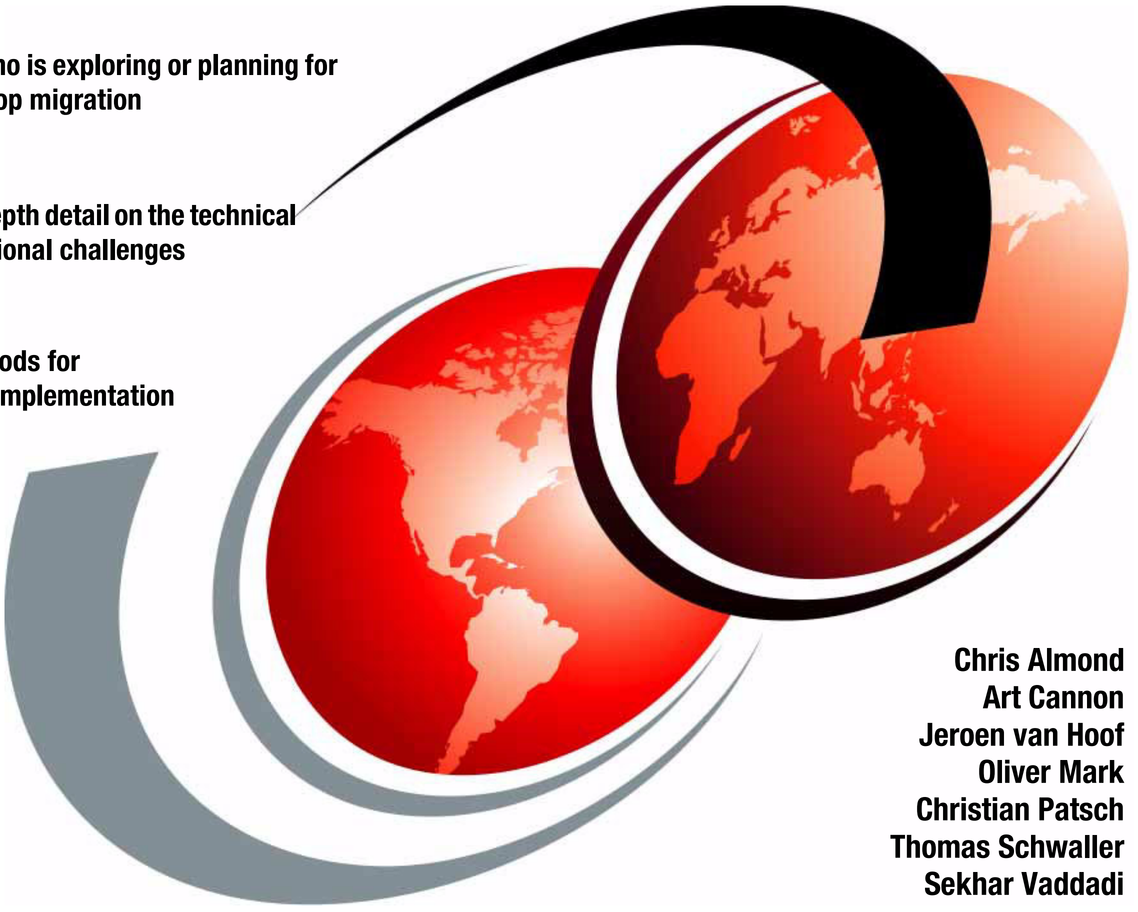
IBM

Linux Client Migration Cookbook

A Practical Planning and Implementation Guide for Migrating to Desktop Linux



For anyone who is exploring or planning for a Linux desktop migration

Provides in-depth detail on the technical and organizational challenges

Includes methods for planning and implementation

Chris Almond
Art Cannon
Jeroen van Hoof
Oliver Mark
Christian Patsch
Thomas Schwaller
Sekhar Vaddadi

Redbooks

ibm.com/redbooks

International Technical Support Organization

Linux Client Migration Cookbook - A Practical Planning and Implementation Guide for Migrating to Desktop Linux

December 2004

SG24-6380-00

Note: Before using this information and the product it supports, read the information in "Notices" on page xv.

First Edition (December 2004)

Contents

Figures

Tables

Notices

Trademarks

Preface

For several years now, many people involved with computing and the Internet have harbored hopes that Linux might become a viable end-user operating system choice for a broader segment of general purpose end users. At the same time there has been growing frustration with the problems and limitations of the current commercial desktop OS choices. In turn this frustration has fueled a greater need in the market for alternative desktop operating system choices. At the same time, Linux-based desktop-oriented distributions have improved tremendously as a result of the inclusive and open-ended dynamics of the open source development movement.

The goal of this IBM Redbook is to provide a technical planning reference for IT organizations large or small that are now considering a migration to Linux-based personal computers. For Linux, there is a tremendous amount of “how to” information available online that addresses specific and very technical operating system configuration issues, platform-specific installation methods, user interface customizations, etc. This redbook includes some technical “how to” as well, but the overall focus of the content in this book is to walk the reader through some of the important considerations and planning issues you could encounter during a migration project. Within the context of a pre-existing Microsoft Windows-based environment, we attempt to present a more holistic, end-to-end view of the technical challenges and methods necessary to complete a successful migration to Linux-based clients.

The team that wrote this redbook

This redbook was produced by a team of specialists from around the world working at the International Technical Support Organization, Austin Center.

Chris Almond is a Project Leader and IT Architect based in the Austin, Texas ITSO center. He has a total of 14 years of IT industry experience, including the last five with IBM. His experience includes UNIX/Linux systems engineering, network engineering, Lotus Domino-based content management solutions, and Websphere Portal-based solution design.

Art Cannon is a Consultant and Linux Team Leader in the IBM Global e-business Solution Center in Roanoke, Texas. His career with IBM began in 1977, and has included positions in manufacturing, development, education, and sales. In 1987 he joined the Sales & Distribution organization as an Area Designated Specialist for AIX. In 1993 he joined the Open Systems Center in

Roanoke, the precursor organization to the Global e-business Solutions Center. Since 1998 he has led the GeSC Linux team. He is a Certified Platform Specialist with both Linux and AIX.

Jeroen van Hoof is a Senior IT Specialist with IBM Global Services in the Netherlands. He has over 15 years of experience with Unix operating systems and over 10 years experience with Linux. Before joining IBM through PwC Consulting, he worked in high-performance computing at the Dutch National Supercomputing Centre as a parallelisation expert. His areas of expertise include Unix/Linux, Windows, TCP/IP networking (routing, switching), parallelisation (MPI, PVM), firewall security, programming (Fortran, C, Pascal, Perl), SAP Basis consultancy, and Web technologies (HTTP, HTML, CGI). He is a SAP Certified Technology Consultant and a Red Hat Certified Engineer. He holds a PhD in Theoretical Physics from the University of Nijmegen.

Sekhar Vaddadi is a Senior Software Engineer with IBM Software Labs in India. He has over nine years of experience with Unix and Linux operatings systems. His areas of expertise include Unix/Linux, Windows, networking, various programming languages, Web technologies, and transaction processing. He is a certified administrator in Linux and HP-UX. He received his Masters degree in Computer Science and Engineering from the Indian Institute of Technology, Mumbai, (IIT, bombay) India.

Christian Patsch is a Junior Consultant of the Linux-team at Becom Informationssysteme Gmbh a premium business partner of IBM in Germany. In April of this year, he got his degree in Business Informatics at the University of Applied Sciences Giessen-Friedberg, finishing a diploma thesis about migration scenarios for "linux on the desktop". He has five years of experience in the client/server area, including both Windows and Linux environments. His special areas of expertise in the Linux area are groupware solutions and client management.

Oliver Mark is a Certified Senior IT Architect and Principal at IBM Global Services Integrated Technology Services Germany. He has participated in a number of residencies including Warp, Linux, and other topics, and has published two external books. He has over 13 years of experience in the client/server area in various sectors, with a strong focus on Linux and OS/2-based end-to-end architectures. He is responsible for all IBM Global Services service offerings for Linux and OS/2 and its customers in the EMEA Central Region. He holds a degree in Economics and Computer Technologies. His areas of expertise include OS/2, Warp Server, IBM communication products, Windows, and Linux, as well as infrastructure and systems management architectures. As a member of the IBM Technical Expert Council, he has the opportunity to work on emerging topics, and gives IBM management advice on future directions. He has written extensively on migration strategy, assessments, and application selection.

Tom Schwaller is a Linux IT Architect and EMEA Linux Desktop Technical Leader living in Munich, Germany. He studied Mathematics and Theoretical Physics at the Swiss Federal Institute of Technolgy and been a research assistant at the Technical University of Munich in the field of high-performance computing (using Linux Beowulf clusters). After that he was editor in chief of the German Linux Magazine (from 1996–2000) and site manager of linux-community.de. He is also cofounder of the Linux New Media AG and joined IBM in 2001. He has worked with free software for more than 15 years and with Linux for more than 10 years. He is a big fan of scripting languages like Python, Ruby, Groovy, and Boo, and is a Linux hardware collector (Intel Dual Celeron and two ThinkPads, Xbox, AMD™ K6 and Athlon 64, several Alphas (DEC 2000/PC 150) AXPpci33, LX 164, Sun Ultra-5, Motorola Powerstack, Apple G3, IBM 43P-270, IBM B50, SGI Indy R4000, Cobalt Cube, HP PA-RISC 32/64, StrongArm PDA, Amiga, and hopefully some Power4/5 and zSeries machines in the not so distant future).

Acknowledgements

The following people provided significant support for this project:

Stephen Hochstettler, ITSO Project and Linux Team Leader: For his vision and support.

Greg Kelleher, Senior Program Manager Worldwide Linux Desktop Strategy and Market Development: For sponsoring the project and guiding development of the content outline and scope for this book.

Become a published author

Join us for a two- to six-week residency program! Help write an IBM Redbook dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You'll team with IBM technical professionals, Business Partners and/or customers.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you'll develop a network of contacts in IBM development labs, and increase your productivity and marketability.

Find out more about the residency program, browse the residency index, and apply online at:

`http://ibm.com/redbooks/residencies.html`

Comments welcome

Your comments are important to us!

We want our Redbooks to be as helpful as possible. Send us your comments about this or other Redbooks in one of the following ways:

- Use the online **Contact us** review redbook form found at:

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

- Send your comments in an Internet note to:

  `redbook@us.ibm.com`

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. JN9B Building 003 Internal Zip 2834
  11400 Burnet Road
  Austin, Texas 78758-3493

Part 1

Choosing Linux

1

Introduction

For several years now, many people involved with computing and the Internet have harbored hopes that Linux might become a viable end-user operating system choice for a broader segment of general purpose end users. At the same time there has been growing frustration with the problems and limitations of the current dominant commercial desktop OS offerings from Microsoft. In turn this frustration has fueled a greater need in the market for alternative desktop operating system choices. At the same time, Linux-based desktop-oriented distributions have improved tremendously as a result of the inclusive and open-ended dynamics of the open source development movement.

The goal of this redbook is to provide a technical planning reference for IT organizations large or small that are now considering a migration to Linux-based personal computers. For Linux, there is a tremendous amount of “how to” information available online that addresses specific and very technical operating system configuration issues, platform-specific installation methods, user interface customizations, etc. This redbook includes some technical “how to” as well, but the overall focus of the content in this book is to walk the reader through some of the important considerations and planning issues you could encounter during a migration project. Within the context of a pre-existing Microsoft Windows-based environment, we attempt to present a more holistic, end-to-end view of the technical challenges and methods necessary to complete a successful migration to Linux-based clients.

1.1 The migration landscape today

As this book was being written, the range of choices and capabilities of native Linux software was expanding rapidly. For those few nascient IT organizations that have the choice to grow an install base of Linux desktops organically (that is, they are starting a desktop IT infrastructure from scratch), then the Linux choice should provide a basis for all of their application needs today. But the majority of Linux desktop deployments will most certainly occur within the context of a migration. And, at an application's level, one of the most common and important migration challenges will likely be the feasibility of migrating users from the Microsoft Office productivity suite to a Linux-based equivalent.

Other high-profile migration challenges with heavy network infrastructure dependencies include messaging (Microsoft Outlook does not run natively on Linux), and to a lesser extent interaction with enterprise directory and authentication services. This book does include sections that discuss and demonstrate migration methods for Linux client integration into an existing Active Directory/Exchange-based network.

As for migration of office productivity suite applications, at this time we believe that the odds for migration success currently favor organizations or end users that do not rely heavily on use of advanced functions in Microsoft Office, or customized applications that integrate with or extend Office. We believe that greater odds for success currently favor the "fixed function" or "technical/transactional" usage patterns, as defined in 3.1, "Assessing usage patterns" on page 28, and 4.4.3, "Functional segmentation - Fixed function to general office" on page 65.

Important: In the near term, successful Linux client migrations will favor limited use clients. As Linux-based office productivity suite applications (and the content migration tools supporting those applications) mature, we expect greater frequency and success for migration of advanced office users.

Thus in the near term we believe that a significant share of desktop Linux deployments will be targeted at highly technical workers, students,and transactional workers. We expect that the public sector, especially outside of the US, will also be a major driver of desktop Linux adoption. Large deployments, ranging from 14,000 government desktops in Munich to 80,000 PCs for students in Spain, are already under way. Thanks to the unique nature of open source software, the lessons learned from these initial deployments are likely to rapidly result in even stronger Linux desktop offerings in the near future.

1.2 Identifying suitable environments

Some customers such as large banking and insurance companies, public administrations, and the retail sector are pushing towards an Open Source based solution not only on servers, but also on their corporate desktops. As with all products, technologies or solutions, a "one size fits all" approach to the open source desktop will not be feasible in all cases. Critical questions need to be asked:

- Is the customer's employee population strictly dependent on a third-party application, plug-in, or devices that are only supported on Windows?
- Has the customer intensively developed custom applications based on native Win32 APIs and/or programming environments, such as Visual Basic or other Windows Scripting languages?
- Is the customer's entire employee population dependent on Microsoft Office compatibility (for example, skills, file formats)?

If either answer is yes, then a Linux-based solution is probably a less-suitable alternative or a more complex (higher initial deployment cost) solution.

If your answer is no to the above questions, then the next step is to plan and execute a pilot migration (using this redbook as a planning guide). The results of the pilot migration should validate feasibility of production migration.

1.3 Strategic context

From a migration point of view, Linux is only one piece in the puzzle. Customers are faced with the problem of simplifying and optimizing existing end-to end IT infrastructures, including servers, databases, applications, networks, systems management processes, and clients with the goal of reducing costs and complexities resulting in a stable foundation for growth and new solution deployment.

All conversations should be about and around functions, not products or named features. Word processing has become a less-known word, compared to Microsoft Word; even so, Word is just one single product, currently available on the Windows and Apple Macintosh platforms, to provide this function. Several packages are available on every platform to choose from.

Next, a desktop is almost never used without a larger environment to operate in; it connects to servers, storage, and printers. It uses not only the operating system, but also a wide range of middleware products, application packages, and might even require custom application development. It has to be deployed, supported, and managed.

This redbook presents the task of migrating clients to Linux within an existing Microsoft Windows-based environment. In this redbook we do not present an entire network migration strategy to Linux-based clients and servers. But, it is safe to assume that any Linux client migration strategy will likely be part of a larger strategic IT infrastructure migration plan, where Linux plays a role in supporting enterprise network and application services. In Figure 1-1 we show an example infrastructure migration path for converting a Windows NT based infrastructure to OSS/Linux. This redbook focuses on the portions of this migration path that are within the dotted line box in the figure.

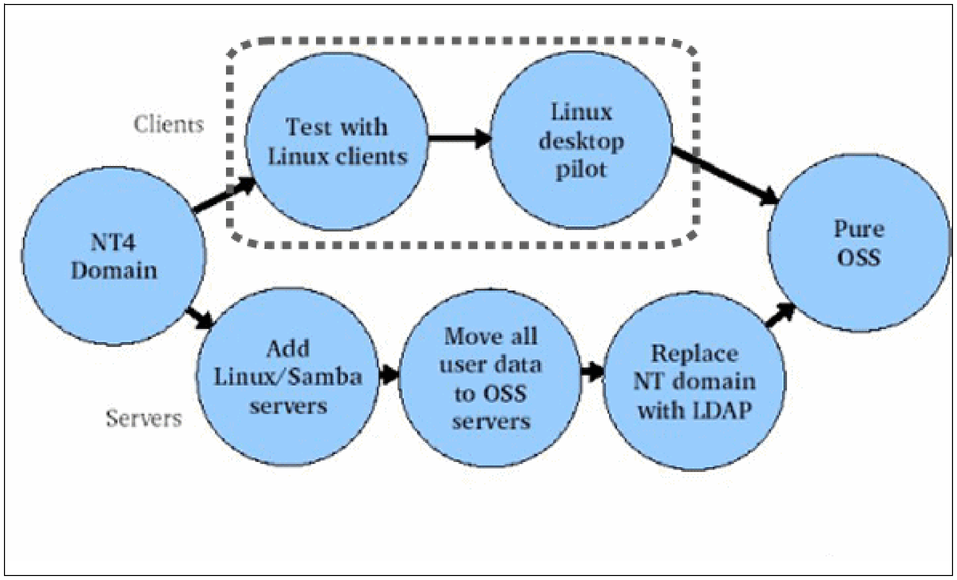


Figure 1-1 Migration route diagram[1]

[1] Source "The IDA Open Source Migration Guidelines", netproject Ltd © European Communities 2003

1.4 Client environments

The choice of an appropriate client platform for a particular set of users can depend on their functional role and the applications they must use to accomplish their objectives.

More and more line-of-business applications are being developed to depend less on the underlying operating system by taking advantage of open standards and pervasive technologies such as Web browsers. For economical reasons, many enterprises are quickly moving toward Service-Oriented Architectures (SOAs) that allow them to compose applications out of existing services. This allows and encourages the reuse of application logic and data across the enterprise and even between enterprises.

SOAs are often implemented through Web services. Web services is an emerging set of standards that ensure interoperability by using such common technologies as Extensible Markup Language (XML), Simple Object Access Protocol (SOAP), Hypertext Transfer Protocol (HTTP), and others.

Clients for applications based on Web services are often written in Java™ or based on Web browsers accessing portals. This makes the underlying operating system for clients transparent to the application and allows flexibility of choice, based on cost, support, flexibility, support for open standards, and so on.

This kind of environment makes for a good objective toward which many enterprises are moving. However, it will not happen overnight. There are still legacy applications that must be accessed and used as they are written today.

Therefore, it is important to understand what functions are required by a client platform to meet the needs of users today, while keeping an eye on the direction of technologies and enterprise architectures. This helps to ensure that the choices made today provide the capabilities that are required now and support the requirements of future architectures.

1.5 Why Linux

Linux has evolved into a powerful desktop operating system that can run on already existing hardware. In many cases, it requires less memory and processing power than other alternatives to provide similar performance on the client.

Because of its core design and open nature, Linux can be easily customized. Linux is available under the GNU General Public License[2] (GPL) agreement and can be obtained for free. However, most enterprises buy a Linux distribution to take advantage of the bundling features and support that accompanies them. The openness and flexibility of Linux, not the price, is becoming the driver for many organizations to migrate to this operating system. Its functionality, stability, scalability, and support have been key factors that have expanded the use of Linux from academic areas to the enterprise.

With support from such companies as IBM and others that deliver key client platforms, such as Lotus® Notes®, the Mozilla Web browser, open office suites, and Java desktops, Linux is gaining momentum as a desktop operating platform.

Linux supports the Portable Operating System Interface (POSIX) standard that defines how a UNIX-like system operates, specifying details such as system calls and interfaces. POSIX compliance has made it possible for developers to port many popular UNIX applications and utilities to Linux.

Linux also provides a complete implementation of the TCP/IP networking stack. A full range of clients and services are supported, including a standard socket programming interface so that programs that use TCP/IP can be easily ported to Linux.

Linux supports the standard ISO-9660 file system for CD-ROMs, printing software, multi-media devices, and modems. In short, it provides the facilities to support the requirements of a wide range of client application types.

1.6 Linux overview and distribution choices

In 1984, the Free Software Foundation (FSF), started by Richard Stallman, began the GNU project to create a free version of the UNIX operating system. This system can be freely used, but even beyond that, the source code can be freely read, modified, and redistributed. A number of components were created, including compilers and text editors. However, it lacked a kernel. In 1991, Linus Torvalds began developing an operating system in a collaborative way. All information was made available for anyone on the Internet to improve the operating system that was called Linux. Linux was exactly the operating system kernel the FSF needed.

In the Linux community, different organizations have created different combinations of components built around the kernel and made them available as a bundle. These bundles are called *distributions*. Some of the most well-known distributions include Red Hat, SuSE, Debian, Mandrake, etc.

[2] Copies of the GNU GPL Licenses can be found at http://www.gnu.org/licences/licenses.html. The GNU project is supported by the Free Software Foundation (FSF): http://www.gnu.org/fsf/fsf.html

Linux is a UNIX-like, POSIX-compliant operating system distributed under the GNU software license. This means that the operating system can be distributed for free. Linux supports all the major window managers and all the Internet utilities, such as File Transfer Protocol (FTP), Telnet, and Serial Line Internet Protocol (SLIP). It provides 32-bit and 64-bit multitasking, virtual memory, shared libraries, and TCP/IP networking. It is coupled to a native POSIX thread library for high-performance multithreading, symmetric multiprocessing (SMP) up to 16 logical CPUs or eight hyperthreaded CPU pairs, and massive parallel processing (MPP) up to 10000 AMD Opteron processors in a new Cray computer under development.

Linux has been developed to run on the x86, Itanium®, AMD64, and IBM @server zSeries®, iSeries™, pSeries®, and S/390® architectures. A common source code base is used for all of them.

1.6.1 Licensing

Linux is distributed under the GNU GPL agreement. The complete license agreement can be accessed at:

http://www.gnu.org

This section briefly summarizes the terms of this license:

- Free redistribution of the software is allowed. There are no restrictions in regard to selling or giving away the software.
- Modifications and derived works are to be distributed under the same terms as the original software.
- Modified source code may have distribution restrictions. The license should explicitly allow the redistribution of derivative works. This *cannot* include the patched code, since patched code may not be considered derivative work.
- The license does not discriminate against any person or group of persons.
- The rights that are attached to the program, when the license is distributed, apply to all to whom the program is redistributed.
- The license must not restrict other software that can be distributed along with the licensed software.

1.7 The rest of this book

The rest of this book consists of:

- Part 1, “Choosing Linux” on page 1
  - Chapter 1, “Introduction” on page 1
  - Chapter 2, “The case for migration” on page 11

    The next chapter in Part 1, “Choosing Linux” on page 1,” covers rationale and justification for a Linux client migration.
- Part 2, “Planning the pilot migration” on page 25
  - Chapter 3, “Organizational and human factors planning” on page 27

    This chapter provides a discussion of human and organizational factors that are important considerations for transition management, and the importance of gathering non-technical data about the “as is” client environment.
  - Chapter 4, “Technical planning” on page 37

    This chapter provides background, analysis, and planning guidance for the various technical topics that need to be considered as part of planning for a Linux client migration.
- Part 3, “Performing the pilot migration” on page 87
  - Chapter 5, “Migration best practices” on page 89

    Some technical methods that can be used to enhance the quality of your migration are explained in detail.
  - Chapter 6, “Client migration scenario” on page 115

    In this chapter we document steps necessary to complete our own sample migration.
  - Chapter 7, “Integration how-tos” on page 137

    This chapter provides detailed procedures describing how to complete some common migration configuration requirements.
- Part 4, “Appendixes” on page 163
  - Appendix A, “Using enterprise management tools” on page 165

    This appendix provides an overview and assessment of the currently available enterprise desktop management tools provided by Novell and Red Hat.
  - Appendix B, “Application porting” on page 213

    A complete Windows-to-Linux desktop migration may required that you re-program some applications to run natively on windows. This appendix provides a brief overview of porting considerations and some application porting development tool vendors.

- Appendix C, “Desktop automation and scripting” on page 215

  In this appendix we present details on some scripting and automation capabilities of Linux/Windows applications and desktops. These capabilities allow users and administrators to fully control their environment and automate tedious desktop tasks.

- Appendix D, “Client personalization” on page 221

  In this appendix we compare the methods for storing client-side personalization data between Windows and Linux. As this book focuses on migrating clients from Windows to Linux, migration of client-side personalization data could also be an important topic for planning and implementation phases of a project.

2 The case for migration

In this chapter we discuss the rationale behind and the justification for the client migration.

We will not go into detailed TCO or ROI issues in this chapter. This chapter will also not try to “sell” a migration. In this chapter we do want to discuss why you may want to migrate clients to Linux.

Since this book describes a migration to Linux clients from Windows clients, a comparison with the Windows platform is unavoidable. However, we primarily want to introduce the reasons for migrating on their own merits. A comparison with the Windows platform has been presented in various articles, papers, and reports many times, from various sources that are available on the Internet; and most of those sources contradict each other anyway.

The sections in this chapter are:

- 2.1, “Why migrate” on page 12

  Reasons for considering a migration.
- 2.2, “When to migrate - Or not to migrate” on page 20

  Deciding factors on actually performing a migration.
- 2.3, “Migration goals” on page 21

  What are the deliverables we want from the migration?

2.1 Why migrate

A migration to a different desktop client is triggered by an alternative being available to what is being used. A decision to migrate is based on the alternative having properties that compare favorably to the present situation. In this section we talk about the properties of the Linux desktop.

Looking at the Linux desktop, the properties that are most generally used to base decisions on are:

- Desktop security
- Total cost directly linked to the desktop client
- Managebility of the Linux client solution
- Client customization
- OSS philosophy
- Ease of use of the client

We look at each of these properties is detail.

2.1.1 Desktop security

Even though Linux has many security features or properties, we concentrate on the ones relevant to a Linux client. Since on a Linux client there is interactive user activity, the security issues we look at reflect that. We focus on the features most related to this interactive user activity and some others that are also relevant:

- Browser security
- Messaging-client security
- User fencing/security
- Bugfix response time
- Modularity of the operating system
- Firewalling

Browser security

The browsers used with a Linux client are mostly open source software. Some of these browsers are also available for other operating systems. In a Linux client the browser is not part of the operating system—it is a stand-alone application hosted by the operating system. For Linux platforms there are a number of browsers to choose from. The most commonly used are:

- Mozilla
- Opera
- Konqueror

Because the browser is not closely built into the operating system, it is more difficult to use the browser as an entry point for circumventing operating system security. All of the browser software runs in user space and not in kernel space. This also means that the browser software can be installed as a non-root user.

Apart from the fact that security exploits of the browser do not immediately affect the operating system, the bug-fixing process for the most commonly used browsers is usually very fast, most notably within days (and sometimes hours) after discovery of the security problem. This “speed to fix” is a result of the active involvement of a large and growing community of open source developers, and again because only the application is being patched and tested to provide the fix, not the host operating system as well.

Another temporary advantage of open source browsers is their small marketshare relative to Internet Explorer on Microsoft Windows, thus making them smaller targets for exploitation. This advantage would diminish in the long run as more clients begin using alternative open source browsers.

Messaging-client security

A messaging client in this context is defined as an application that is used to communicate messages with another user of another computer over a TCP/IP network, including the Internet. This includes both e-mail applications as well as instant messaging applications.

Like browser applications, messaging applications for Linux are open source software, and they are stand-alone applications that are hosted by the operating system. This means that all advantages listed in the section about browser security are valid for messaging applications as well.

The open source messaging client can generally handle more than one messaging protocol. This means that a choice of a messaging application for the Linux client may still provide some flexibility in the choice of which server-side messaging application or protocol is being used. The choice of client application can actually be influenced by the choice of server-side application. Thus security considerations can be an influence when designing messaging system services in a Linux/OSS-based environment.

User fencing/security

User security has been an important part of Unix operating systems from their early beginnings. Since Linux, like Unix, is inherently a multi-user operating system, it is possible to use this core feature to separate different security roles on the client. Also the fact that there is only one user with all administrative rights (the root user) by default helps in keeping the client secure.

The security options in Linux that can be applied to users and groups are typically applied to “fence off” the login environment of the end user. The end user does not, by default, have administrative access to the operating system on the client.

Bugfix response time

A large factor in responding to security exploits is the time to fix. The risk related to an exploit is directly related to the time the security “hole” is available to people who want to exploit it. Since most if not all components in a Linux client are open source software, fixes will either come from the open source community or from an enterprise vendor that offers this kind of support. It has been shown that the time to fix security exploits in the Linux kernel is quite short.

Also, since the source is open it is possible for an organization to develop a bugfix on its own, test it and deploy it internally, and even share the modified source code with the OSS development community for consideration

Firewall support

The Linux kernel has firewall services built into the kernel. As of Version 2.4 of the kernel, this firewall is called iptables. Configuring iptables correctly enables the administrator to implement a network security policy on the Linux client. Iptables firewalling can be used to secure the Linux client in the following areas:

- ► Protection against worm viruses.
- ► Prevent end users from using protocols not allowed by policies.
- ► Secure against known exploits.

2.1.2 Costs related to Linux client

The different cost factors related to the client are fairly general and independent of the operating system, but the relative impact and size of these cost factors can be highly dependent on the client operating system. Cost factors to consider for a Linux migration include:

- ► License and support cost for the Linux distribution
- ► Hardware cost
- ► Application cost for the base desktop client
- ► Management and support costt
- ► Migration cost

License and support cost for the Linux distribution

The Linux kernel and most applications included in Linux distributions are open source and licensed under the GNU Public License (GPL). This means that the software is freely distributable. Therefore there are no license costs per se related to the Linux client.

However, distributions packaged by enterprise distributors are not free. There is usually a per-seat pricing model for enterprise distributions. And this fee usually includes a support mechanism for the installed machine for one year. It may also be possible to purchase extra levels of support from the enterprise distribution vendors.

Given the open source nature of the Linux operating system and related software, it is also possible to use it completely free of license and support costs. For support the organization is then completely dependent on the open source community and well-trained personnel within the organization.

Hardware cost

Most Linux distributions can run well on older hardware. Depending on your client application requirements, it is even possible to re-use hardware that has already been retired because it cannot support the latest Microsoft Windows performance requirements.

However, the newer enterprise distribution offerings for the desktop have minimum memory (MB) requirements that approach the same requirements of Microsoft Windows. Since this is a memory requirement only, you may still be able to use existing or even retired systems to support these distributions.

Application cost for the base desktop client

The Linux distributions include a large number of basic applications like editors, imaging software, browsers, e-mail applications, instant messaging applications, and even some office productivity tools like word processing, presentation, and spreadsheet applications. This means that the cost for these basic applications when using a Linux client is small to none.

Even if an application needed is not included in the distribution, chances are that there is an OSS equivalent that can be installed free of cost.

Management and support costs related to the client

Keeping any production desktop client operational and free of bugs and security exploits is usually one of the larger cost factors. This is also the case for the Linux client. The fact that the operating system is Unix-like enables a lot of cost-saving options. For example, since the client can be remotely connected to and managed using telnet or ssh protocols, it is possible to install scripts on the client that can easily be remotely executed.

Using remote scripts it is possible to monitor the clients for problems and to execute a task on all clients from a central server. For example, it is possible that a bugfix can be implemented on all clients by remotely executing a script on the client that is able to fetch a patch and install it, without end-user interruption.

There are also systems management tools available from enterprise vendors, for example, the Red Hat Network tools from Red Hat and the Red Carpet tools from Novell.

2.1.3 Manageability of the Linux client

One of the major issues related to client choice is the way the client can be managed. Another issue is the cost related to this management as stated in 2.1.2, “Costs related to Linux client” on page 14. Some of the inherent properties of the Linux operating system, as well as tools developed in either the open source community or by vendors, make the Linux client a very manageable client alternative.

The properties and tools we discuss in this section are:

- Modular structure of the operating system
- Update/patch mechanism
- Inherent remote access
- Remote management and provisioning tools

Modular structure of the operating system

The Linux kernel has, by design, a modular structure. This means that the kernel is not one monolithic binary. Instead, it consists of a central smaller kernel binary together with various kernel modules. The kernel modules can be loaded when needed. Some of the modules have to be loaded at boot up because these are needed to read file systems or other peripheral hardware.

Not only the kernel is modular, but the application framework around the kernel has a modular construction as well. The applications like scripting engines (Perl, PHP, Python) or editors (gedit, vi) are not integrated into the operating system and can even be replaced by others or new versions more or less independently.

The modular nature of Linux means that updates or patches only involve a small part of the operating system, for example, either a single application binary or library within the application framework or a single kernel module. This modularity is the main reason that an update or a patch almost always does not require a reboot on Linux.

Update/patch mechanism

The update/patch mechanism for Linux does not need to be central to the system. What we mean by this is that since updating or patching will not be destructive to the system state (in other words, leading to a reboot), it can be done while other applications are running. Unless the update or patch impacts the core kernel or a running application, it can be done online.

This means that the mechanism that checks for new updates/patches can be scheduled regularly and be allowed to implement the changes automatically.

Inherent support for remote access

The Linux operating system has inherent remote access through TCP/IP, like all Unix operating systems. This is a really powerful tool for systems management, because almost all management tasks can be done remotely, and since the advent of the secure socket layer can be done securely using ssh.

Remote access to the client is useful for two types of administrative tasks:

- Remote execution of administrative programs or monitoring scripts
- Access the client from a central location to give support on the client

Remote execution

Remote execution is used to execute applications or scripts from a central location, for example, to force all clients to update a virusscaner or fetch a critical security patch. This can also be used to execute monitoring scripts that will enable the administrator to do preventive work, for example, to clear a temporary directory before a file system fills up.

Remote support

Almost all support on the client can be provided by accessing the client from a remote location. The only types of problems that still require hands on interaction at the client are network problems. All other problems reported by the end user can be handled by logging onto the client remotely and studying the system configuration and the log files. This capability could improve the effectiveness of helpdesk staff significantly.

Remote management and provisioning tools

There are a lot of tools available in the open source community as well as from commercial parties for remote management or monitoring and provisioning. The most-used open source tools are:

- Webmin[1]
- Big Brother/Big Sister[2]
- Nagios[3]

Webmin is mainly used for remote management of a single machine. Big Brother and Nagios are mostly used to monitor run-time properties of a network of

[1] `http://www.sourceforge.net/projects/webadmin`; `http://www.webmin.com`
[2] `http://www.sourceforge.net/projects/big-brother`; `http://www.sourceforge.net/projects/bigsister`; `http://www.bb4.org`
[3] `http://sourceforge.net/projects/nagios`; `http://www.nagios.org/`

machines, then apply preventive administrative tasks when system properties go outside of established limits.

Commercial tools are available for these tasks as well. For example, IBM provides Tivoli Configuration Manager and Tivoli Monitoring software for automation of monitoring and remote management tasks.

Red Hat distributions can use the Red Hat Network (RHN) to remotely manage individual systems or groups of systems. The RHN also offers automatic provisioning services as described in “Administration of Red Hat Desktop” on page 73.

Novell/SuSE distributions use Red Carpet software to remotely manage or provision systems. After Novell acquired Ximian software, the SuSE distributions adopted Ximian’s Red Carpet product as the standard tool for their enterprise distributions. Before that SuSE used Yast online Update (YoU) as the tool to remotely manage systems.

2.1.4 Client customization

Sometimes it is useful to match the client to the role of the end user using it. If a standard client build contains every possible application provided the source distribution, then becomes not only large but also more difficult to maintain. Therefore client customization to task becomes an important goal in a Linux desktop deployment strategy.

In this section we look at the following features of the Linux client:

- Flexibility to add or remove components of the Linux installation.
- Prevent “bloating” of the client by maintaining package control.
- Availability of task-oriented distributions.
- Desktop design and session manager flexibility.

Flexibility to add or remove components

Starting during the installation it is fairly simple to add or remove components to the installation. Most installers will supply a default installation, with the option to change which components to add or remove. This is again a result of the modular structure of the Linux operating system. This flexibility enables the construction of customized installations for the Linux client.

Prevent bloating of the client

Using the flexibility to add or remove components it becomes possible to prevent the client from “bloating”. Because customized Linux clients can be constructed for different end-user roles, it is not necessary to put all applications in one client image. This ensures that the client size will not grow out of control.

Linux distributions have been constructed as small as a single floppy disk (1.4 Mb). The enterprise distributions will install to several Gb in a default installation.

Task-oriented distributions

Because of the freedom to include and remove components and the absolute freedom to add other applications from the open source community, it becomes possible to create special task-oriented distributions. It is possible to create a distribution (and a Linux client based on it) for audio/video work or for signal processing. These task-oriented distributions are useful when there are some very specialized desktop client requirements in an organization.

Desktop design

One of the most obvious differences between the design of a Microsoft Windows client and a Linux-based client is the freedom of choice you have in selecting desktop session managers and window managers in Linux. Also, most session managers (certainly KDE and GNOME) also provide extensive sets of theme options that can be used to further customise the look and feel of the desktop user interface. These themes are essentially a collection of icons, colors, windowframes, backgrounds, and widgets, which combine to provide the selected look and feel.

2.1.5 OSS philosophy

In some cases the fact that Linux is open source software is reason enough for the migration. This is mainly because organizations want to avoid vendor tie-in. Another reason that organizations want to use open source software is that the source is available and can be studied or adapted to their specific needs when necessary.

For these reasons a lot of governemental organizations, especially in Europe, are studying moving to OSS and Linux-based clients.

2.1.6 Ease of use of the client

Because nowadays desktop clients are used by a majority of workers in many enterprises, it is a useful property of a client to be easy to use. A desktop client that needs extensive training before it can be used leads to enormous investments in training of end users.

2.2 When to migrate - Or not to migrate

This book focuses on methods for migrating Microsoft Windows-based clients to Linux-based clients within a mainly Windows-based enterprise. But in general the client migration is almost always part of a larger migration to open source software within the enterprise. This has to be taken into account when planning a client migration.

Even though the new Linux desktop might have properties (as indicated in 2.1, “Why migrate” on page 12) that are in favor of a migration, the total end result of the migration must have advantages as well. The total end result of the migration is not just a number of clients running Linux. Because of infrastructure and application environments, the architecture after the migration is usually more complex.

In this section we look at some circumstances that favor a decision to migrate to a Linux client. These circumstances can mostly be described as an absence of a complicating factor. We describe some in detail.

2.2.1 Client roles fit thin/slim client model

Clients within a client/server model are generally called either fat, slim, or thin. This indicates where the majority of the application used actually runs. For example, in a thin client only the presentation part of the application runs locally (for example, in a browser), and the rest of the application runs on the server. These different roles are described in more detail in 4.4.2, “Logical segmentation - Thin, slim, or fat” on page 64.

One of the complicating factors in the migration is a large number of fat clients. A fat client is a client with a large number of applications locally installed. The larger the number of applications then the greater the chance that some of those applications will be difficult to migrate to the Linux client. Some of them may even be unmigratable, which leads to a more complex infrastructure integration, as described in 4.7, “Un-migrateable applications” on page 81.

If the current client is thin, this means that migrating the client to Linux does not involve the complicating factor of a complex infrastructure integration. In the idealized case of the end user only using a browser to connect to a portal for all of their application needs, then migration becomes almost trivial.

The more the client fits a thin or slim client model, the easier it is to migrate to a Linux client. This leads to most early migrations only including the thin clients in an organization and leaving fat clients in their present state. It may also bring organizations to move to a thinner client prior to migrating to an alternative client OS altogether.

2.2.2 High number of migratable applications

An application is termed *migratable* when there is a direct way to migrate the application to a Linux client, either by using a Linux version of the application or using a Linux-based alternative.

As described in more detail in 4.7, “Un-migrateable applications” on page 81, applications that do not migrate well to Linux complicate the migration to a large extent. This also means that more migratable applications will lead to easier migration.

Before performing a migration and handling the unmigratables, some organizations will try to move away from applications that do not migrate easily. The best way to do this is to move the application to a portal-based application. This will not only facilitate moving to a Linux client, but then the application will also not be a problem in any future migrations.

2.2.3 Organizational readiness

Even if all other factors in the migration are favorable, an organization that is not ready for a migration will still be unsuccessful. An organization that is ready can be defined as:

- Having end users that are ready and able to move to another client
- Having administrators and support staff that are enthousiastic about the migration and knowledgeable about the technology you are migrating to
- Having procedures in place that will streamline the handling of problems during and after the migration

These factors are discussed further in Chapter 3, “Organizational and human factors planning” on page 27.

One of the most important factors is that the migration has to be carried by the administrators. The people who have to manage the clients and infrastructure after the move can make or break the migration. Getting system administrators and support staff behind the migration is one factor that can become very important in the decision process.

2.3 Migration goals

Not all migrations are the same. While eventualy migrations will have a goal of replacing all desktop clients in an organiztion with Linux clients, most early migrations will serve as pilot projects and only migrate a part of the desktops.

In this section we discuss both goals in more detail. We indicate how the goals of both types of migration are different. Details for planning either a partial migration, pilot migration, or full migration are found in the rest of this book.

2.3.1 Pilot migration

The actual goal of the pilot migration is not that some client will be running the Linux operating system. The main goal of a pilot migration is answering the following question: How can we deploy accross the organization with confidence?

A pilot migration has to target all types of usage models that will be included in an eventual full migration. All applications that are going to be considered in the migration have to be included in the pilot as well. This way the pilot can be seen as an information-gathering exercise for the full migration.

The pilot migration should give answers to most of the following questions:

- Can all applications still be used after migrating?
- Are end users able to use the infrastructure of the organization as before the migration?
- Are administrators still able to perform all their tasks after the migration?
- Do the new infrastructure components, like terminal servers or consolidated Windows desktops, function as planned?

2.3.2 Full migration

In a full migration the migration methods that were tested in the pilot phase are used to successfully complete migration of all targeted desktops to the new configuration.

Goals of a full migration could include the following:

- Increase the level of desktop client security.
- Improve managebility of desktop clients.
- Lower the overall total cost of ownership (TCO) of desktop clients.
- Decrease dependency on a single software vendor.
- Improve the life cycle of client hardware.
- Comply with govermental regulations and/or strategies (for example, China has declared use of OSS as a strategic imperative).
- Extent usage of Linux from servers to desktop to leverage existing experience and skills.

- An excuse for change (for example, to clean up existing problems and start new with a standardized client implementation strategy).

Good planning is a major part of the migration. If a migration is started from a technical perspective it is very easy to start with the technical challenges. That is certainly a part of a succesfull migration. But to complete a succesfull migration, management support on all levels is certainly just as important.

In the remainder of this book we discuss how to plan a migration, both human factors and from a technical point of view. We then document an example of migration, including technical *how to* steps for solving some of the issues involved.

Part 2

Planning the pilot migration

3

Organizational and human factors planning

This chapter provides useful information regarding non-technical issues that correlate with a client migration. In contrast to migrations in a data center, you have to consider many more human and organizational factors—justified by the fact that a migration will affect the daily work of the end users.

The sections in this chapter are:

3.1 Assessing usage patterns

When planning a pilot migration, one of the most important steps is to perform an as-is analysis of your current IT environment that focuses on client application usage patterns. From this analysis you should be able to derive a segmentation of role-based usage patterns. A very helpful tool for gathering this information is an end-user survey. Practical methods for performing this survey are provided in 4.1, “Assessing the client IT environment” on page 39.

3.1.1 Role-based client segmentation

While doing the analysis, it is necessary to focus not only on the client computers and the software packages that are installed on them, but also check your staff on occurring usage patterns. If you are able to group them into defined usage patterns, then you can use those groupings to help decide which groups are most suitable for a pilot desktop migration project.

Table 3-1 demonstrates segmentation into five different usage patterns. Below each pattern it shows example applications that fit the pattern. Note that this example segmentation also demonstrates a progression of thin (Fixed Function) to thick (Advanced Office) client requirements. More detailed definitions of each segment follow the table.

Table 3-1 Role-based client segmentation

Fixed Function	**Technical Workstation**	**Transactional Workstation**	**Basic Office**	**Advanced Office**
Limited use of business applications		Applications that drive business processes		
Limited office productivity	Simple office productivity			Advanced office productivity
No e-mail	Simple e-mail			Advanced e-mail
No instant messaging	No instant messaging			Instant messaging
Simple browser access to intranet, portals				Advanced browser access to Internet
File/print system management, network access, host emulation				

Fixed Function

This provides limited use of productivity applications and no e-mail or instant messaging, and simple browser access to specified portals. Examples include:

- Kiosk-style terminals
- Retail POS systems

Technical Workstation

This provides imited use of business applications, but simple office and mail tasks, and usually special applications needed to support specific work functions. Examples include:

- CAD/CAM workstation
- Software Development workstation

Transactional Workstation

This provides typical use of business applications, but still simple office and mail tasks, and limited browser access. Examples include:

- HelpDesk
- Order entry
- Travel agents

Advanced Office

This provides a knowledge worker that utilizes office productivity suites and advanced messaging tools to create and distribute content. Examples include:

- Project managers
- Engineers
- Marketing and sales

The Advanced Office presents potentially the greatest challenge in a desktop migration due to the challenges in maintaining similar application functionality and data fidelity when using Linux-based equivalent applications.

3.1.2 Surveying user data

Assuming that a group or role-based segment of users is identified for migration, then the next step would be to collect data (a technical survey) that captures all aspects of the workstation usage profile. This profiling data should include all applications that are being used, all application dependencies (for example, database servers), all required file types that are used, as well as client hardware data, peripheral dependencies, etc.

As part of this survey, you will need to pay careful attention to evaluating hardware and peripheral device compatibility. Another topic that needs special attention is the range of file types that are used company wide. In order to avoid the possibility of inaccessible files or content, it is very reasonable to make a list of all types, and then identify applications or migration paths that guarantee the usability on the new system.

3.1.3 End-user survey

The task of an end-user survey can take a lot of time, but is also very reasonable, as you get insights into the user point of view. This can help you discover how end users use the systems and what is important to them.

Another important issue you can learn about by this survey is the existence of applications or other items that are not listed in the software catalog of a company. Sometimes users have built templates or databases on their own, which are very useful to them, or they have installed applications that are necessary for their work requirements.

3.2 Establishing functional continuity

This chapter includes methods that help to retain the functions of existing applications. After the migration, users in most cases will have to switch to different but functionally equivalent applications. In order to bridge this gap, which can result in a loss of productivity, it is useful to develop a strategy in which users get accustomed to the new applications.

3.2.1 Bridging applications

Some applications that run natively on Linux are also available natively for Windows. These applications provide an opportunity to minimize the transistion effects and retraining requirements that are triggered by an operating system migration to Linux. Thus it is possible to migrate to applications that will be supported on the Linux plaform prior to actual migration of the OS itself.

The benefit of such pre-migration changes is that the users are allowed to get accustomed to the new applications before the actual migration of the client-OS is done.

After the new OS is installed, the users will not experience any change at all as far the applications are concerned, making the switch easier.

Table 3-2 on page 31 provides examples of applications that could be used as “bridging” applications between Microsoft Windows and Linux. Applications listed in the second column could be deployed on existing clients running MS Windows, thus allowing for application migration to occur prior to OS migration.

Table 3-2 Example bridging applications

Application used in Windows	Linux/Open Source equivalent
Internet Explorer	Mozilla.org/FireFox
Outlook (Express)	Mozilla.org/Thunderbird
MS Office Word	OpenOffice.org[a] Writer
MS Office Excel	OpenOffice.org Spreadsheet
MS Office Powerpoint	OpenOffice.org Impress
Paint Shop Pro	The GIMP[b]
Messenger-Client (MSN,Yahoo,ICQ,...)	GAIM[c]

a. http://www.openoffice.org
b. http://www.gimp.org
c. http://gaim.sourceforge.net

3.2.2 Functionally equivalent utility applications

The designs of utility applications such as file system browsers, archivers, and viewers are more closely tied to the host operating system. They cannot be bridged in the sense that we describe for the applications listed in the previous section. One of the reasons Linux is considered to be approaching equivalency with Windows is the availability of many choices for utility applications. In many cases, these applications may in fact have more powerful feature sets than the equivalent utility applications in Windows.

Table 3-3 provides some examples of equivalent utility applications available in Linux.

Table 3-3 Examples of equivalent utility applications in Linux

File managers (Windows Explorer)	Konqueror (KDE) Nautilus (Gnome)
Archivers (like WinZip)	Karchiver,ark (KDE) FileRoller (Gnome)
Viewers (like ACDSee)	Konqueror, KView(KDE) Nautilus (Gnome) Adobe Acrobat Reader for Linux
Multimedia players (like Windows Media Player)	xmms, xine, RealPlayer

3.2.3 Web applications

Unfortunately, it is not possible to find similar or bridging applications for all of what is needed. Applications for ERP, CRM, or Collaboration are especially likely to have thick client implementations for which there are not cross-platform equivalents between Windows and Linux. Enterprise application vendors are responding to this not by developing separate thick client implementations for each OS, but by focusing on the Web browser as the container for extending their client applications to alternative client platforms (Linux).

This strategy is a good one to rely on because once you have a Web-enabled application, it will never cause any dependency on a specific operating system.

A good example for this situation is Domino Web Access, which was formerly known as iNotes. Until Version 6.5, the design of iNotes limited its use to Internet Explorer only, and thus was only usable with a Windows-based client. The latest version of Domino Web Access now supports the Mozilla browser, thus making it a cross-platform client application. SAP provides another example of this. They provide a fully functional Web client and Java-based fat clients that can be run in platform-independent browsers.

In the case that such a solution is not available, the approach of bridging applications to the new desktop by first transistioning to a cross-platform Web client interface cannot be used. In this case you may have to migrate the application to a newer version that does support multi-platform clients, or you may have to consider switching to another software vendor that meets cross-platform client requirements.

3.2.4 Building bridges to the server

In the last three sections we showed ways to make the change-over easier for the end users by switching to ported applications or Web clients. Another possibility to bridge gaps regarding functionalities is moving the application from the client to the server, which means the implementation of terminal servers.

Server-based computing brings along many advantages, such as a centralized point for updates or a better server utilization, but at first it makes an abstraction from the client. Regardless which operating system is installed or which sort of client is in use, you always get the same user interface delivered in your terminal application. The only requirement is the availability of an application that makes a stable and secure connection to the Terminal Server.

In the case of Windows Terminal Server or Citrix Metaframe, the requirements are met—clients are available for both Windows and Linux. Native Windows terminal servers can be accessed from Linux using rdesktop, an open-source client for the remote desktop protocol (RDP). The URL for this useful tool is:

`http://sourceforge.net/projects/rdesktop/`

Citrix delivers with its MetaFrame Suite ICA clients for several operating systems, including an ICA Client for Linux.

Using this technique, many applications can be moved to the server-based model. Your migration path in this case introduces the possibility of moving from a fat to a thin client. This approach is especially worth considering for clients that fit into the Fixed Function and Transactional Workstation roles as defined in Table 3-1 on page 28.

A new approach to server-based computing is given by the developers of NoMachine NX, who are able to reduce network traffic on both X and RDP sessions enormously. Since the code for the NX server is under GPL, free implementations will be available soon and maybe give the possibility for an ubiquitous desktop. More information can be found on their Web site:

`http://www.nomachine.com`

3.3 Human factors

As a desktop client migration affects end users in a very direct way, considering human factors in the transition management strategy is very important.

You can expect that a radical change in the desktop interface from what users are accustomed to will cause different kinds of reactions: From enthusiastic acceptance to outright rebellion. Therefore it is important to keep the end users informed about the new developments in a clear and understandable way. A communications plan is key.

Figure 3-1 on page 34 provides an example of a new technology acceptance curve in an organization.

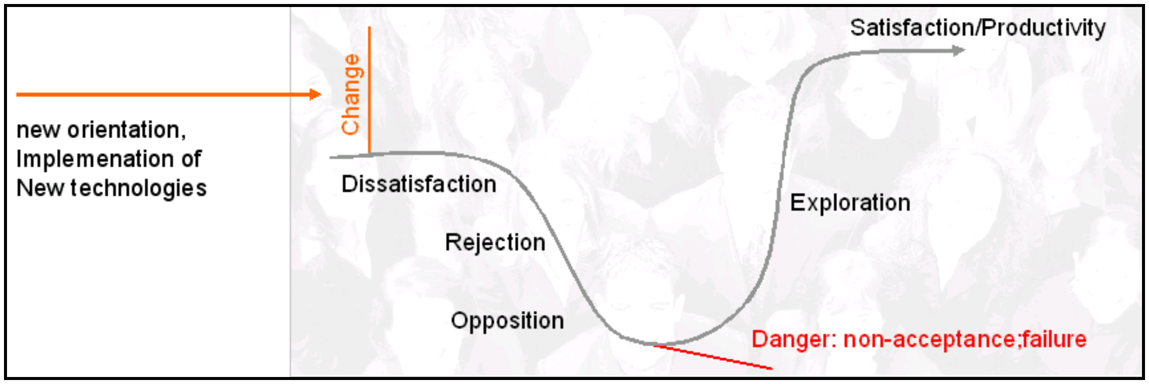


Figure 3-1 Impact of changes

The effect of a good communications plan will be to flatten out the negative aspects of the acceptance curve in Figure 3-1. A communications plan coupled with proper training on the new desktop applications should minimize the number of users that fall into the rejection/opposition mode, cause users to start out with acceptance instead of dissatisfaction as the initial response, and lead to a quick transition into exploration and productive use.

Regarding the IT support staff, these same issues have even more importance. A strategic decision to switch the OS and the way the clients are managed can cause the impression of disapproval of the work they have done so far. It could give staff the impression that their current skills are being devalued. It probably will not be easy to convince an administrator of Windows systems that migrating to Linux clients is a good strategy unless you can also convince them that the organization is prepared to make a significant investment in upgrading their skills as well.

Thus, two very important areas to focus on prior to starting a pilot migration are:

- Develop a communications plan.
- Select a pilot group that is best suited for the task.

3.4 Retraining considerations

In the course of the migration, retraining for users will be necessary in many cases. As classes are always cost-intensive due to the payment for a trainer and the non-productive time of the employees, one has to figure out ways that can reduce this amount.

3.4.1 Bridging applications can separate retraining from migration

Referring to the bridging applications mentioned in 3.2.1, “Bridging applications” on page 30, the strategy of replacing current applications with OSS equivalents that are available on both Windows and Linux could reduce training costs. The migration could be done in a smoother way, as users have had the chance to acclimate themselves to the same applications that will be used on a Linux-based client, prior to actual OS migration.

3.4.2 Learning a new look and feel

Another strategy to save costs is the effort to retain the look and feel of the current applications and the desktop. It is possible to customize certain aspects of the Gnome and KDE desktops to emulate the look and feel of the Windows Desktop and Windows-based applications. Many freely available themes are available for download and further customization. Examples for these can be found at:

```
http://www.kde-look.org
http://art.gnome.org
```

3.4.3 Familiar actions

Emulating actions is also a good idea. A good example is enforcing double-click instead of single-click as the open action for desktop icons in the window manager.

3.4.4 File systems: Everything has been moved

Windows users are used to a hierarchical file system based on partition mount points such as C:\, D:\, etc. The hierarchical file systems in Linux differ from this convention. Conventional mount points for file systems in linux include /usr, /home, etc. Migrated end users could encounter much confusion when trying to understand the new Linux file system hierarchy. To smooth this transistion, one recommended method is to migrate the entire contents of the user’s existing My Documents folder into a similarly named folder in the user’s Linux home directory. Inside of /home/user/My Documents, the content and structure will appear exactly the same as was present in the original folder in Windows.

3.4.5 Hands on Linux prior to migration

Many Linux distibution vendors are now providing live or bootable CD-ROM-based distributions. One notable live CD distribution is provided by Knoppix:

http://www.knoppix.com

The following description from Knoppix further explains what a live CD provides:

> KNOPPIX is a bootable CD with a collection off GNU/Linux software, automatic hardware detection, and support for many graphics cards, sound cards, SCSI and USB devices and other peripherals. KNOPPIX can be used as a Linux demo, educational CD, rescue system, or adapted and used as a platform for commercial software product demos. It is not necessary to install anything on a hard disk. Due to on-the-fly decompression, the CD can have up to 2 GB of executable software installed on it.

A live CD can be used to provide users with the ability to run a bootable Linux system on their desktop. They can use it to test and evaluate the UI, applications, and other facets of the Linux client, before an actual client migration occurs. And this can be done without harming the host operating system install on the local hard disk. Another benefit of using a live CD as part of a migration plan is for detection of hardware and device problems. The live CD should be able to help validate proper hardware detection and device support, and identify any issues prior to actual client migration.

4

Technical planning

In the first chapters we focused on the strategic and organizational part of the migration and discussed the issues of the migration in a more abstract way.

In contrast to this higher-level view, the following chapters explain more technical details that have to be considered when planning a migration from a Windows-based client to a Linux-based client.

Sections:

- 4.1, “Assessing the client IT environment” on page 39

  Before starting to look into technical details of a future Linux and Open Source-based environment, it is critical that you thoroughly assess the current status of IT infrastructure and the supporting processes.

- 4.2, “Integrating with existing network services” on page 45

  In this section we describe how to plan for integrating Linux clients in an existing network. We specifically look at how to incorporate Linux clients in a predominantly MS Windows-based network/domain.

- 4.3, “Standardizing the desktop” on page 51

  In this section we discuss some issues regarding a standardized Linux desktop. We discuss Linux standards in general, the different desktop environments with their underlying component models and lockdown mechanisms, and also more general topics like corporate identity guidelines, themes, application menu design and overall usability.

- 4.4, “Migrating applications” on page 63

  A migration path needs to be determined for each application that will be migrated to a Linux-based equivalent. Different migration paths will be necessary as not all Microsoft Windows-based applications have Linux-based equivalents. This section demonstrates methods for defining functional and logical client groupings based on what types of applications they run.

- 4.5, “Client administration planning” on page 68

  This section describes some methods for efficient administration of the Linux client.

- 4.6, “Desktop vs. notebook considerations” on page 76

  This seciton discusses additional considerations necessary for planning migration of mobile (notebook-based) clients.

- 4.7, “Un-migrateable applications” on page 81

  This section discusses issues related to applications that cannot be migrated to run directly on a Linux client. We propose alternate ways to support these applications.

- 4.8, “Post-migration troubleshooting and technical support” on page 84

  This section discusses some considerations for supporting a Linux client after migration.

4.1 Assessing the client IT environment

Before starting to look into technical details of a future Linux and Open Source-based environment, it is critical that you thoroughly assess the current status of the IT infrastructure and the supporting processes.

For the IT Infrastructure, many components will come to play, not only the core installation of a desktop installation, but also the any other infrastructure components around it. This might be servers providing file and print functions, databases, directory services, application servers (both native and Web based) up to mainframes, which will be accessed via host emulation or API-based communication. Some of the key considerations and methods in this area are discussed in more detail in 4.2, “Integrating with existing network services” on page 45.

Because we are focusing on migrating client workstations, two key areas in the client IT environment need to be surveyed:

- Human interactions: The end user application usage patterns
- Physical systems: Desktop system properties, special devices, and peripheral integration requirements

4.1.1 Assessing the context

Prior to starting with any technical aspects of a migration, what type of non-functional requirements are defined for this migration project need to be verified. A verification of those goals against the IT strategy is another critical success factor.

Questions to ask in this area are:

- What are the tactical goals of the migration?
- What measurements of success will validate achieving those goals?
- What are the financial aspects of the migration (budget, return on investment, justification, etc.)?
- What are the strategic goals for the migration? How does the migration fit into the overall IT strategy for the organization?

4.1.2 Assessing the client hardware

You need to complete a physical systems survey of all client systems targeted for migration. The results of this survey will enable you to identify any hardware support issues, and define rules for buying and replacing systems in the future.

This survey should ideally be done through an automated process on a regular base to allow history of data. It also can be done as a one-time scan for a migration purpose. IBM eGatherer technology and the Asset Depot offering can help implementation. For more details, see the IBM Redpaper *Using Asset Depot for Inventory Management:*

http://publib-b.boulder.ibm.com/Redbooks.nsf/RedbookAbstracts/redp3763.html?Open

Questions to ask in this area are:

- What is the hardware you are running on (vendor, type, model)? If this results in a large, diversified list, then consolidation is something to consider, regardless of which platform you are using.
- Is the hardware standardized? If all the machines are the same, then things like driver support and software distribution should be relatively simple already and will remain so under Linux.
- What are the local attached devices currently installed and required by users? This includes any type of printers, scanners, and special function keyboards required for business types of transactions. It would not include the convenience type of hardware an executive gets, just because she is important and a nice device needs to sit on her desk.
- Is support for Linux included in current requests or proposals to hardware vendors when buying hardware?
- Which key components of your hardware are currently required by users? For example, machines might have sound cards installed, but drivers are not installed, because users are not supposed to have sound. Therefore support on another operating system is not required as well.

IBM, for example, provides information on supported desktop hardware for their current models, so customers can check on both officially supported hardware as well as available drivers. Information can be found at:

http://www-306.ibm.com/pc/support/site.wss/MIGR-48NT8D.html

A sample client hardware survey table may look like Table 4-1 on page 41.

Table 4-1 Sample hardware survey results

Model	Type	RAM in GB	Disk in GB	Total	Attachments
2373MU4	T40	2	80	120	List of the various peripheral devices attached: Printers, scanners, digital cameras, USB devices, etc.
23668AU	T30	1	60	42	
2366MU1	T30	1	48	1023	
2366MU9	T30	0.5	48	311	
26473U5	T23	1	48	278	
26476RU	T23	0.5	30	67	
26478EU	T22	1	30	44	
264757U	T21	0.5	24	1	
2682MU3	R40	1	48	99	

4.1.3 Assessing the client software configuration

You need to complete a client software configuration survey of all client systems targeted for migration. The results of this survey will enable you to identify all applications, services, special configurations, and component support issues that need to be considered in the migration plan. This survey may shed light on specific technical support issues, such as hidden user IDs, scheduled tasks, etc.

Key survey questions:

- What is your set of native applications? This will result in a list of ISV applications including version used and potential fixes applied.
- What is your set of internal applications? This will result in a list of applications developed and maintained within the company.
- What is the set of applications requiring access to data external to the client? This will result in a list of applications accessing file servers, application servers, Web servers, databases, mainframes and other implementation of data processing. See 4.4, “Migrating applications” on page 63, for more details.
- Do you have groups of users defined? How are they characterized? This should give an overview as to whether there are some typical groups or users and, if so, how they are grouped. Grouping can be done by departments, applications used, type of work, or business responsibility. If a questionnaire is used interacting with end users, then this should become a kind of self-assessment.

- What are the security-related applications, processes, and regulations in place? This gives an overview of products used for securing the client, such as anti-virus, desktop safeguarding, port scanning, as well as rules and regulations of how those applications are installed, maintained, updated, and the user is enforced to use them. It also includes policies for security patches of any operating system component and installed application.

4.1.4 Assessing data dependencies

For most client/server applications, the availability of a functional replacement client-side application that runs natively in Linux is the only requirement. This is the case for applications such as a database client, which can be installed (assuming availability on Windows and Linux) on both platforms. Another example could be an application using a Web interface to access data stored on the server. As long as the Web interface can be run in a Linux-based browser, then client-side migration of that application becomes a non-issue.

For some applications, and those tend be local and native applications, data can be stored in a proprietary format and with file system dependencies that will require a data conversion process.

Applications in this category include native mail systems (that is, Lotus Notes) or productivity suites (that is, Lotus Smartsuite or Microsoft Office). Without discussing the actual technical migration, during the assessment, you are required to explore the current use of such applications.

Some example questions to ask in this area are:

- Do you use Microsoft Office? If so, which components and how often?
- Do you use macros in Microsoft Office? If so, which type of macros and for which components and how often?
- Do you use Microsoft Outlook? If so, which components and how often?
- Do you use Lotus Smartsuite? If so, which components and how often?
- Do you use Lotus Notes? If so, which components and how often?
- Do you use Microsoft Project? If so, which components and how often?
- Do you use Visual Basic to automate activities within or across applications?
- Do you share files with external organizations? If so, which formats and how often?

With some of the above questions, it becomes very clear that the underlying infrastructure also has to be reviewed. Using Microsoft Outlook on the client often leads into Microsoft Exchange on the server, while Lotus Notes on the client usually indicates Lotus Domino on the server. For both scenarios, such server installations have to be taken into consideration when designing a new client stack and migration of user accounts has to be planned.

A sample table showing application migration mapping from Windows to Linux equivalents is provided below. Note that the target applications list does not reflect all choices that are available as potential migration targets in Linux.

Table 4-2 Sample application survey

Current application	**#**	**Target application in Linux**
IBM Personal Communications	7	Exists for Linux
Intel PRO Ethernet Adapter and Software	7	Exists for Linux
Internet Explorer	7	Mozilla
Visual Basic 5.0	7	?
AT&T Network Client	6	Exists for Linux
Sametime Client v3.0	6	IBM Community Tools f
Lotus Notes 6.0.2	5	Lotus iNotes
Lotus SmartSuite - English	5	OpenOffice
Microsoft Office XP Standard	5	OpenOffice
Adobe Acrobat 5.0	3	Exists for Linux
DVDExpress	3	Xine
EasySync Pro	3	Evolution with Pilot-Link
RealOne Player	3	Exists for Linux
Shockwave	3	Exists for Linux
Windows Media Player 7.1	3	Xine
ADSM v3.1.0.7	2	TSM
Microsoft Visio Viewer 2002	2	Terminal Server
Adaptec DirectCD	1	xcdroast
Adaptec Easy CD Creator 4	1	xcdroast
Adaptec UDF Reader	1	xcdroast
Adobe Acrobat 6.0 Standard	1	Exists for Linux
Adobe Photoshop 7.0	1	gimp

Current application	#	Target application in Linux
Citrix ICA Client	1	Citrix for Linux
Java 2 Runtime Environment, SE v1.4.2_04	1	Exists for Linux
Macromedia Shockwave Player	1	Exists for Linux
Microsoft .NET Framework (English)	1	Mono
Microsoft Office 2000 Standard	1	OpenOffice
Nero 6 Ultra Edition	1	xcdroast
NeroVision Express 2	1	xcdroast
Netscape 6 (6.2.3)	1	Mozilla
Norton Speed Disk 7.0 for Windows NT	1	defrag
WinZip 8.0	1	Zip
Yahoo! Messenger	1	Exists for Linux

4.1.5 Assessing the infrastructure

Survey considerations in this area include:

- What is the network infrastructure your clients are connecting to?
- What is the topology of the network infrastructure? This includes a architectural overview of all local and remote connections including bandwidth and protocol conversion.
- What type of network protocols are installed and configured on the client in order to access any type of infrastructure components?
- What servers do exist and which services are provided from them? This will give a list of servers (physical or logical instances) and the services provided to the client. Such services include file, print, DHCP, Web content, dynamic content, applications, and others.
- What databases are required to be accessed and how is this access implemented? From this all connections to databases can be derived as well as the type of implementation, such as native client (that is, UDB client), API connection (that is, named pipes), message queueing (that is, MQ Series), and application access (that is, SAP client).
- What access to mainframes is used by workstations and how is it implemented? Accessing mainframes of different kinds can be done through

native clients (that is, Personal Communications), Web interfaces (that is, Host on Demand and Host Publisher), or API connections (that is, HLLAPI).

4.1.6 Assessing the user

Another key element of a migration of workstations is the end-user. In some cases, especially when talking about success and acceptance of a migration, this is the most important aspect at all. If users are committed to the change, looking forward to working in the new environment, or actively involved in development or deployment of the solution, then user expectation management issues can be minimized.

Questions to ask in this area are:

- What are the most often performed tasks for a user or group of users?
- Are there definitions of user roles? If so, how are they defined?
- What rules are defined, that users have to apply to?
- Are there any defined exemptions from these user roles or rules?
- Are there user-specific settings, which need to be migrated, such as browser bookmarks, wallpapers, etc.?
- Does any of the existing application uses plug-ins or cookies for operation?
- Are users involved in software selection process, and how?
- Do users have the option to pull software from an installation repository?
- Is there a process to request, install, or delete software?
- How often do users call the help desk and what are the average values of key measurements such as response time, re-occurred calls, and level of support involved?
- What is the skill level of users for the base operating system?
- What is the skill level of users for productivity applications (that is, word processing)?
- What is the skill level of users for business applications?
- What type of devices (local and remote) do you require access to?

4.2 Integrating with existing network services

In this section we describe how to plan for integrating Linux clients in a existing network. We specifically look at how to incorporate Linux clients in a predominantly MS Windows-based network/domain.

4.2.1 Setting the environment

Usually a Linux *client* will not be the first Linux machine added in an environment. Usually once you begin migrating clients to Linux, you have already accomplished Linux migration on back-end servers to some exent. This means that most descriptions of how to incorporate Linux clients in an existing network will describe the servers as being Samba servers running on Linux. Since this book is focused on client migration only, we decided to consider the scenario where Linux client pilot migration is occuring in an environment where back-end servers have not yet been migrated.

A Linux client migration project could occur within any of the following environments:

- NT4 domain with MS Windows NT4 PDC (and possibly BDCs)
- Active Directory domain with MS Windows 200x AD servers
- NT4 domain with Linux/Samba PDC and BDCs
- Other non-MS Windows domain based environment

The majority of environments will be one of the first two. The third option is becoming more wide spread, and most descriptions of client migration already assume the server backend to be migrated to a Samba domain.

In this book we concentrate on the first two types of environment (that is, a pure Windows environment) for two reasons:

- Most domains still fit this model.
- There are already a lot of descriptions on how to put Linux clients in a Samba domain.

Planning tip: If a server-side upgrade of the domain to either AD or Samba is already planned, take this into account in the client migration planning. Try to avoid integrating twice.

The tool of choice for integrating Linux clients with Windows domains is Samba. Samba is a very successfull Unix application that provides network services using the Server Message Block (SMB) protocol. This is the same protocol used by MS Windows operating systems to provide client/server networking services for file and printer sharing. Thus, Linux systems running components of Samba software can integrate seamlessly within a Windows domain for access network file shares, printing services, network browsing, etc.

For integation examples using Samba, see Chapter 7, “Integration how-tos” on page 137. Here we indicate technical issues that have to be taken into account when planning the migration.

4.2.2 Authenticating with a Windows domain

In this section we look at all technical issues that need to be considered when planning for authentication of Linux clients within an existing Windows domain.

Reasons for authenticating a Linux client in an existing Windows domain include:

- Network services that require domain authentication need to be accessed from the Linux client (network file servers, printers, etc.)
- Users will have only a single user name/password combination (network services single sign-on).
- Administrators will only need to administer a single user collection.

After deciding to authenticate with a domain, the following technically driven decisions have to be made:

- Are domain users created on all clients, or can we use winbind to enforce an unified login environment? (Using winbind you can force a Linux client login event to authenticate with a Windows domain. The end result of this is that the Linux client system becomes a full member of the Windows domain.)
- Using winbind means carefully choosing some parameters, specifically the winbind-separatort.
- Do we authenticate with an NT4 domain or natively with an Active Directory domain? In the latter case we also need Kerberos.

Choosing to create users locally on the client means extra administrative overhead. In this case when a user is added to the domain, the user ID has to also be added to any of the Linux clients that the user will be using to connect with that domain. Even though this process could be automated, it is really not necessary when using winbind.

Using winbind will lead to a choice for what is used as the winbind-separator. This is the character that will separate the domain name from the user name in the Linux user name. For example, AD6380+Administrator is the Linux user name of the user Administrator in domain AD6380 when the winbind-separator is a plus sign (+). The impact of the chosen character has to be studied in all applications and network services being used. Using the plus (+) character for separation generally is the best choice for most Linux shells and applications.

Planning tip: Plan and test winbind and the winbind separator extensively to validate the setting prior to migrating clients.

In the case of authenticating natively with an Active Directory domain, Kerberos has to be configured as well as Samba.

In summary, the applications that are going to enable us to authenticate with a Windows domain are Samba, Kerberos, and winbind.

4.2.3 File sharing using domain shares

Since the Linux clients are joining a domain and the majority of clients in the domain will probably still be Windows clients, the best way to share files is through Windows shares.

Mounting shares on a Linux client using **`smbmount`** gives the user file systems on the Linux client, which behave very much like any other file system. However, if the share is not open for everybody, a user name/password combination is needed, just like under Windows. When mounting a share by hand this can be input without problem. However, this might lead to several problems:

- Manually mounting shares using the **`smbmount`** command may not be an acceptable practice for all users. Users are not used to having to mount shares manually because under Windows this is done automatically when logging on.
- A way must be found to enable automatic mounting on login just like on Windows. In Linux, automatic mounting of SMB shares can be accomplished using entries in the /etc/fstab file.

A simple solution would be to mount shares in a logon profile. However, this needs a password, and the user is used to shares being mounted using the password supplied upon logon.

A Pluggable Authentication Module (PAM) exists to enable automatic mounting at logon, called pam_mount. Since this module is not completely mature yet, care has to be taken to include this in the planning. Currently it is the only useful way to incorporate some sort of single sign-on like under Windows.

When planning for this, extra time should be included to test the pam_mount module on the Linux client chosen for the project. When it works it is a very powerful method of mounting shares automatically.

Planning tip: Test and plan use of pam_mount extensively.

Home directories and shares

During planning it might be very tempting to push user home directories on Linux clients to Windows shares. This will enable a “thinner” client.

Putting the user’s home directory on a share would indeed enable logging on to any client and getting the same files present. Using a graphical logon, this would mean getting the same desktop and environment.

However, care has to be taken. Some graphical systems (most notably KDE) need to create symbolic links or sockets in the user’s home directory. One of the shortcomings of the SMB file system in Samba is not being able to create symbolic links or sockets in such a file system.

Planning tip: Using shares for home directories has to be planned and tested in detail.

4.2.4 Printing services in the domain

Of course it is possible to add printers directly to Linux clients. But this would create extra administrative overhead in a scenario where you are integrating Linux clients into an existing Windows domain that provides network printing services already. Since almost all Linux distributions now include the Common Unix Printing System (CUPS), this can be used in conjunction with Samba on the client to enable printing to the existing domain printers.

If you plan to use CUPS on the Linux clients to print to existing printers, some issues have to be taken into account:

- Is CUPS and Samba integration handled correctly?
- Do the printers need authentication? Domain passwords for shared printer access could be exposed.
- What are the advantages of using print servers versus direct printing to the printers network interface?
- Are your printer’s model and drivers available in CUPS?

CUPS and Samba integration

In most distributions, CUPS and Samba will be integrated properly. It needs to be checked, however, and this has to be taken into account when planning.

Most importantly you have to check if Samba is part of the CUPS backend. How to do this is described in 7.5, “How to use network printers in the domain” on page 151.

Printers and authentication

Printers in the domain may require domain authentication to be able to use them. This is possible using CUPS, by providing the user name/password in the URI of the printer. The consequence of this is that the password will be exposed in several CUPS configuration files.

While planning this has to be taken into account, the way around exposing passwords is either making printers unauthenticated (that is, available to everyone) or creating a special printer user for each printer and only incorporating this special user on those clients that need to print to the server.

Print server vs direct printing

Using CUPS it is possible to use the domain print-servers or print directly to the network interface of the printer (if available). Usually if all clients already in the domain use print-servers it is good to follow this principle for the Linux clients as well.

The advantages of using the print-servers are:

- Single type of prints. There is no difference between *Windows-prints* and *Linux-prints*.
- Administratively the printer is controlled from the print-server, so when it needs to be rerouted or disabled this is done for all clients

CUPS and model/driver for your printer

There is quite a big difference in the models and drivers included in CUPS in the different distributions. So this is one of the most important things to check.

Also, if your printer is old, or used by very few people, then decisions have to be made about how much to invest to get it working on Linux clients. The investment for a new printer that works “out-of-the-box” might be a lot less than to get older models to work.

4.2.5 DHCP and DNS configuration

In almost all cases, using DHCP and DNS from a Windows domain will work without problems. This is completely transparant to the end user. Once configured correctly for the Linux client, both protocols should work without any problems.

However, some care has to be taken when using DHCP and X11. The Linux client generally accepts host names from DHCP. If this occurs after X11 has started and the user is logged on, then new applications may have trouble opening Xauth issues.

Planning tip: Make sure a DHCP host name is accepted by the Linux client before starting X11 or logging on graphically.

4.2.6 Web proxy interface

The protocol that is used to talk to a Web proxy is independent of OS. This means that talking to a Windows Web proxy is as easy as setting the correct settings in the Web browser.

This means that interfacing the Linux client with an existing Web proxy is one of the easiest tasks of the migration.

4.3 Standardizing the desktop

In this section we discuss some issues regarding a standardized Linux desktop. We discuss Linux standards in general, the different desktop environments with their underlying component models and lockdown mechanisms, and also more general topics like corporate identity guidelines, themes, application menu design, and overall usability.

Related to desktop standardization is the topic of *client personalization.* In this context, *personalization* is a general term for all client-side data that is created and used to personalize the look and feel, presentation of functions, and other user interface customizations that are specific to the individual end user. See Appendix D, "Client personalization" on page 221, for more details.

4.3.1 Linux standards

In the past years a lot of effort has been put into standardizing Linux and higher-level subsystems. Organizations like the Free Standards Group, FreeDesktop.org, and the Open Source Development Labs (OSDL) started working on many different projects in this area. These projects use and extend existing standards like POSIX, the Single UNIX Specification, XML, DOM, CORBA, and many more. Some example projects are:

- File system Hierarchy Standard (FHS)
- Linux Standard Base (`http://www.linuxbase.org`)
- Internationalization (OpenI18N)
- Printing (OpenPrinting)
- Accessibility
- Clustering (Open Cluster Framework)
- Data Center Linux
- Carrier Grade Linux

Note: All major Linux enterprise distributions are Linux Standard Base (LSB) 1.3 certified. Also, the LSB 2.0 specification has been approved and published. Enterprise distribution compliance with the 2.0 standard should then soon follow.

Most of these standardization efforts were focused on low-level functionality and standardization of the file system layout, a new printing architecture, stabilization or creation of APIs for clustering and high availability, and to make sure that programs from independant software vendors (ISVs) are binary compatible between different Linux distributions.

With Linux entering the desktop marke, one has to go up much higher in the solution stack and consider user interfaces, multimedia (video, audio), mobile computing, security (data encryption, key management), calendaring, directory services, and also hardware issues. For example, the following HW/device standardization issues are currently presenting challenges:

- ACPI
- Bluetooth
- Infrared
- USB
- IEEE1394
- Wireless
- TV cards
- Smartcards

It can be quite difficult in some areas to accomplish this, since many hardware vendors do not publish their specifications or make Open Source drivers available (for example, for wireless or 3D graphics cards).

From a standardization point of view, the Linux desktop is currenlty a very fast moving target. A lot of development is occuring that can change the desktop “landscape”. For example:

- New X server extensions and modularization
- NoMachine (X protocol compression technology)
- Cairo (vector graphics system)
  - With X, OpenGL, Postscript/PDF output
- HAL (hardware abstraction layer)
- GStreamer (streaming media framework)
- D-BUS (message bus system)
- GNOME/KDE (component-based desktop environments)

4.3.2 Linux desktop environments

Compared to classical Windows or Mac systems, in "building" a Linux desktop you can be faced with multiple design decisions. In this building process, Table 4-3 demonstrates the choices that could be considered when designing each layer of the user interface.

Planning tip: Assuming that you are migrating to a Linux desktop distribution purchased from an enterprise vendor, then you will have the option of using the default pre-designed desktop environment as the basis for your clients. In that case, you do not need to design your standard desktop from the ground up, as discussed in this section. In fact, your selection criteria for choosing which enterprise vendor distribution to use could be based in large part on how closely the default desktop provided by that distribution meets your client-side functional requirements.

Table 4-3 Layers on top of the X Window system

Layer	Choice
GUI Toolkits	OpenMotif, GTK+, QT, WXWidgets, FLTK, FOX, Swing, SWT
Window Managers	FVWM, IceWM, WindowMaker, Metacity
Desktop Environments	XFCE, ROX, CDE, Jesktop
3D Desktop Environments	Looking Glass, Metisse
Component Desktop Environments	KDE, GNOME, GNUStep

Window managers (`http://xwinman.org`) are X client programs that control how other X clients are positioned, resized, or moved. They can also provide titlebars and other decorations to windows, handle window focus, and provide user-specified key and mouse button bindings. Example 4-1 shows how to start a nested X server with **`Xnest`**, which is an X server that is simultaneously an X client. This allows you to test other window managers or desktop environments in a window.

Example 4-1 Nested X server with Xnest

```
Xnest -fp `xset -q | grep fonts` :1 &
xterm -display :1 &
```

To simplify graphical programming, a lot of Graphical User Interface (GUI) toolkits have been developed on top of the basic X11 libraries. There are also many integrated development environments (IDEs) available that further simplify GUI programing:

- KDevelop/QTDesigner for KDE/Qt
- Anjuta/Glade for GNOME/GTK+
- MonoDevelop for Mono/C#
- Eclipse or NetBeans for Java/Swing/SWT

Desktop environments provide a much richer user environment than just simple window managers by providing standard productivity applications for:

- E-mail and calendaring
- Instant messaging
- Image manipulation
- Document editing and viewing
- Scanning
- CD/DVD burning, etc.

These applications are created with a standard GUI toolkit. They have a homogeneous look and feel, themeability, inter-application communication, drag and drop functionality, session management, and virtual desktop management.

GNOME, KDE, and GNUStep go one step further by using not just simple GUI toolkits and window managers but providing a full component programming model. As you can see in Table 4-4, there are many different component models available on Linux.

Table 4-4 Different component models

Component model	Used by
XPCOM	Mozilla
UNO	OpenOffice
Bonobo	GNOME
KParts	KDE
Eclipse Plug-in	Eclipse

Unification and standardization of the component model is a very difficult topic, but the applications can at least communicate with CORBA, XML-RPC, SOAP, or DCOP, which makes desktop application integration possible. Fortunately most recent Linux development activity is beginning to standardize around the GNOME and KDE desktop environments (and their respective toolkits). Some traditional UNIX vendors are also beginning to adopt these environments. We will use and discuss KDE and GNOME in this book.

KDE desktop

KDE (`http://www.kde.org`) provides a full-featured "Windows replacement" style desktop and application environment for Linux and other UNIX-like operating systems. KDE is based on the cross-platform Qt programming toolkit from Trolltech, and uses the KParts component model together with the DCOP IPC/RCP mechanism, which we describe in 5.1, "KDE Kiosk framework" on page 90.

With KDE you can place icons and folders on your desktop and start applications from the KDE pannel (Kicker), where you can also freely arrange KDE applets (time, battery and wireless status, virtual desktop chooser, etc.). SuperKaramba (`http://netdragon.sourceforge.net/`) might be an interesting option for even more fine tuning of KDE. Significant applications (`http://www.kde-apps.org`) for the KDE desktop include:

- *Kontact* PIM application with *Kolab* or *eGroupware* as calendaring server
- *KOffice* suite (KWord, KSpread, KPresenter, Kivio, Karbon, KChart, Kexi, etc.)
- *K3b* CD/DVD burning appplication
- *Digikam* digital photo management application
- *Scribus* desktop publishing system
- *Konqueror* Web and file browser

Note: Apple is using and extending the KHTML rendering engine in its Safari Web browser.

GNOME desktop

GNOME (`http://www.gnome.org`) is an open source desktop environment based on CORBA (ORBit2) and the GTK+ GUI toolkit. It is the default desktop for Red Hat and is also used by some traditional UNIX vendors including Sun. Significant GNOME desktop applications (`http://www.gnomefiles.org`) include:

- *AbiWord* word processor
- *Gnumeric* spreadsheet program
- *Evolution* e-mail and calendaring client

  With Novell Groupwise, OpenGroupware, Scalix, Bynari, Sun, ExchangeIT, or even Microsoft Exchange calendaring server as backends
- *GnomeMeeting*
- *GNUCash*
- *GTKam* (GUI for gphoto2)

- *Inkscape* scalable vector graphics (SVG) editor
- *Nautilus* file manager.

On Red Hat the GNOME desktop looks similar to the KDE desktop because of the cross desktop *BlueCurve* Theme. As with KDE, you can place icons and folders on your desktop and start applications from the GNOME pannel. On this panel you can also freely arrange GNOME applets (time, battery and wireless status, notofication area, virtual desktop chooser, etc.). For even more fine tuning of the GNOME desktop, *GDesklets* (`http://gdesklets.gnomedesktop.org/`) provides many options to add custom applications ("desklets") to the Gnome desktop.

4.3.3 Standardization issues

The standard Linux desktop installation process (such as that provided by Mandrake, Novell or Red Hat, etc.) provides the system administrator with many levels of choices for customizing the install process. This is not always what you want in an enterprise environment. But, one has to be very careful with hand-made changes, since future upgrades can result in a lot of unexpected work.

Logos and graphics formats

Nevertheless, one has to make changes to conform to corporate identity policies and guidelines. You probably want a company logo and other brand graphics to be part of your standardized desktop. Simple bitmap graphics are not really state of the art anymore, and should be kept to a minimum to eliminate display resolution problems.

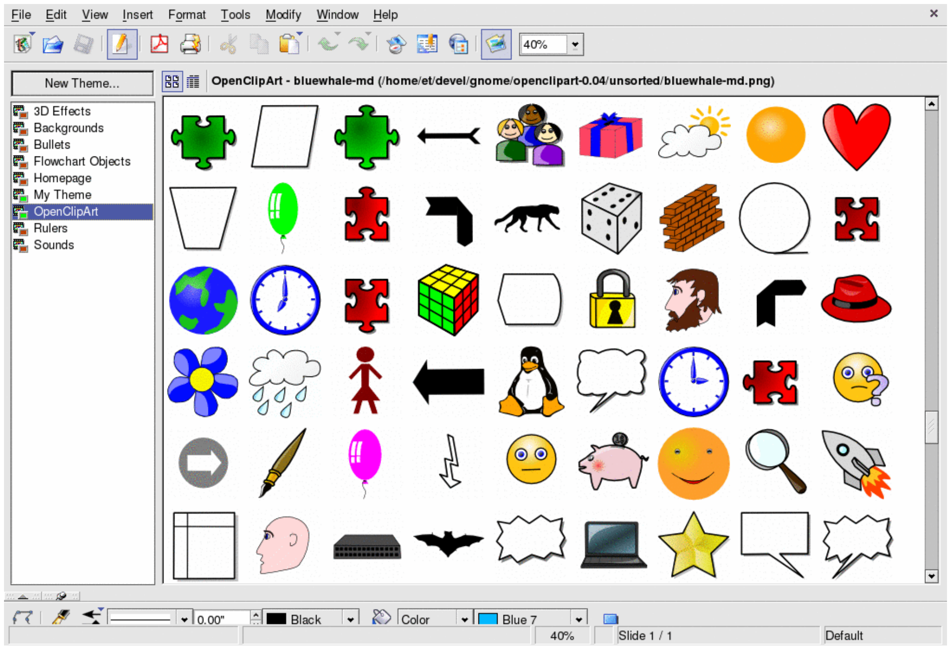

File Edit View Insert Format Tools Modify Window Help
40%
New Theme...
OpenClipArt - bluewhale-md (/home/et/devel/gnome/openclipart-0.04/unsorted/bluewhale-md.png)
3D Effects
Backgrounds
Bullets
Flowchart Objects
Homepage
My Theme
OpenClipArt
Rulers
Sounds
0.00"
Black
Color
Blue 7
40%
Slide 1 / 1
Default

Figure 4-1 SVG Clip Art Library

Figure 4-1 shows the SVG Clip Art Library (`http://www.openclipart.org`).

KDE and GNOME have very good Scalable Vector Graphics (SVG) support in the most current releases. You should use the W3 standard SVG (`http://www.w3.org/Graphics/SVG/`) when designing vector graphics logos and other graphics elements for your desktop. They just look better than bitmaps, are easily resizable, use a standardized format viewable by Web browsers, and are part of a general Linux desktop tend towards SVG-based themes.

Table 4-5 SVG libraries and programs

Library or program	**URL**
librsvg	`http://librsvg.sourceforge.net`
KSVG	`http://svg.kde.org`
Batik	`http://xml.apache.org/batik/`
Inkscape	`http://www.inkscape.org`
Sketch	`http://sketch.sourceforge.net`
KOffice	`http://www.koffice.org/karbon`

Library or program	URL
Mozilla	http://www.mozilla.org/projects/svg/
OpenOffice	http://graphics.openoffice.org/svg/svg.htm
Adobe SVG Plug-in	http://www.adobe.com/svg/

Themes

Images and logos are part ot the bigger topic of themes. The standard choices the Linux distributors or desktop projects make here (for example, Ximian Industrial, Red Hat BlueCurve, Mandrakegalaxy II or KDE Plastik theme) are quite reasonable and most users are comfortable with them.

Designing a new theme from scratch is probably not worth the effort and could be confusing to users reading standard documentation from the Linux distributors. If you really feel the need to design a corporate theme yourself then first visit the following Web sites to check if a theme that meets your needs is already designed and available:

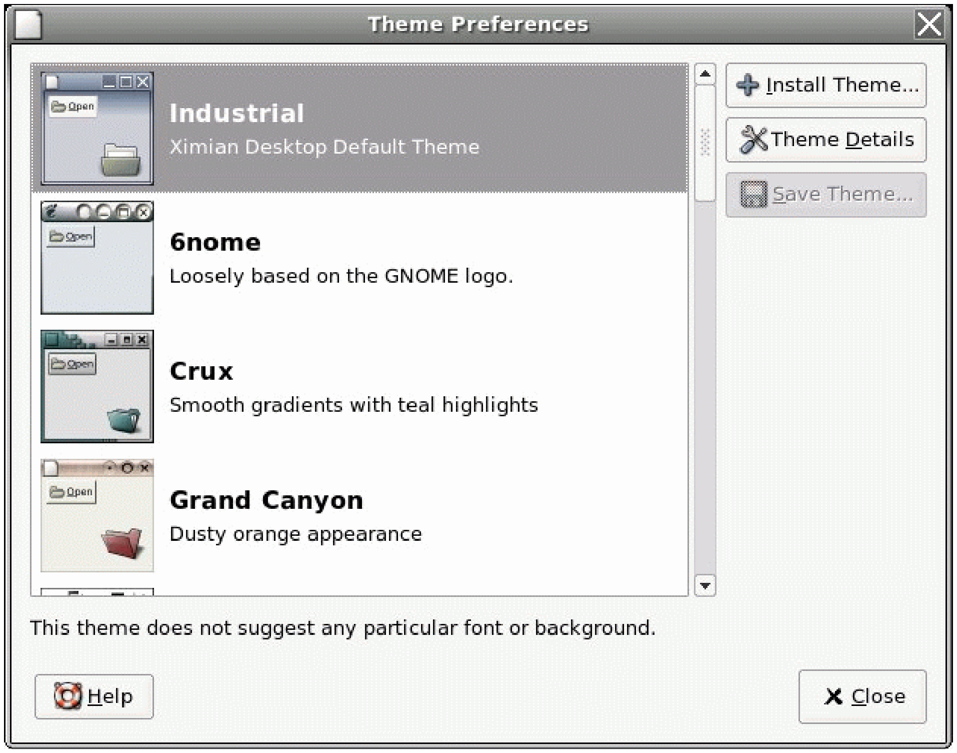
- http://themes.freshmeat.net
- http://www.customize.org
- http://www.kde-look.org
- http://art.gnome.org
- http://www.crystalgnome.org

Even when you decide to use a standard theme you can make a lot of small changes to refine the look even further, since not all graphics and theme developments (which is a very diverse area on its own) are adapted by the Linux distributors.

Take a look at KDE OpenOffice integration and Mozillux projects to see what is possible today.

- http://www.polinux.upv.es/mozilla
- http://kde.openoffice.org

Using a Windows XP theme with KDE or GNOME is probably not a good thing since users will expect exactly the same behaviour as with Windows, which of course is not the case using Linux and could be quite confusing.



This theme does not suggest any particular font or background.

Figure 4-2 Theme Preferences dialog in GNOME

Usability

Usability tests have shown that modern Linux dekstops like GNOME or KDE are very competitive with the Windows XP desktop. Wspecially new users (children, elder people, etc.) having no former experience with Windows or Mac user interfaces have no problems at all with a Linux desktop. They just have to get used to their new environment and computers in general, which is a normal process. In the following reports you can read more details about this success story of Linux usability on the desktop:

`http://www.linux-usability.de/download/linux_usability_report_en.pdf`

`http://www.userinstinct.com/viewpost.php?postid=gnome26review`

A lot of work in many different areas had to be done to achieve these results:

- Internationalization (UTF-8, Pango, etc.)
- Accessibility (for example, for blind people or users with physical disabilities)

Supporting assisting technologies like screen magnifiers, on-screen keyboards, keyboard enhancement utilities, screen readers, Braille displays, text-to-speech programs, speech recognition, etc.

- Programming standards
  - GNOME Human Interface Guidelines (HIG)
  - KDE User Interface Guidelines

Application menu design and desktop lockdown

Another desktop standardization topic to consider is the application menu design. In enterprise environments users do not need five different applications appearing in their menus that do basically the same thing. This approach has been taken by Ximian for quite some time, by simplifying the menu structure a lot (see Figure 4-3 for choosing the standard GNOME video conferencing application GnomeMeeting, using the Video Conferencing launcher).

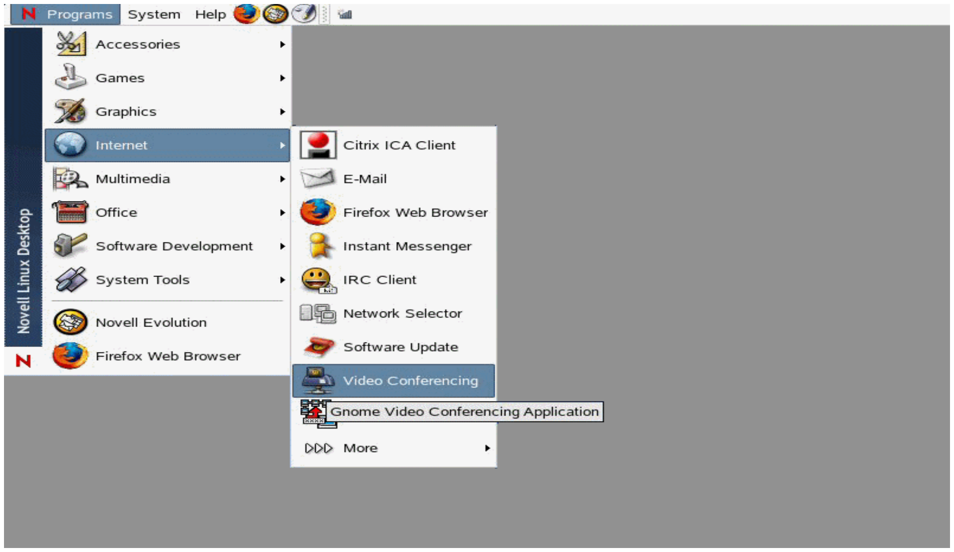


Figure 4-3 Simplified menu structure (Novell Linux Desktop)

In fully controlled environments you may want to reduce menu selections even further and lock them down so that the user cannot make changes to his configuration (like adding or changing panel applets, desktop launchers, menu entries, or background images). KDE provides the *Kiosk* framework to restrict the capabilities of the KDE environment. Although primarily designed for unattended operation of kiosk terminals, these features can also be very valuable in enterprise settings where a more controlled environment is desired. We will discuss the Kiosk framework in 5.1, “KDE Kiosk framework” on page 90.

Attention: Changing the ownership and read/write flags of configuration files is not necessarily a fool-proof lockdown mechanism, since the user can potentially rename the files and create new ones with the correct permissions.

When using a mixed GNOME/KDE environment one has to go one step further and unify the menu structure, configuration files, and lockdown mechanism. This is not a trivial exercise, since KDE and GNOME use different configuration file formats at the moment (KDE a .ini file and GNOME a XML-based one). A unified configuration system with flexible backends (for example, LDAP) and APIs for programmer are needed for larger Linux desktop deployments. There are some approaches like UniConf (`http://open.nit.ca/wiki/`) or even the Linux Registry Project (`http://registry.sourceforge.net`) that could be used for that, but these projects are not widely known yet. It remains to be seen if the OSDL, freedesktop.org, Novell, Red Hat, Mandrake, or other companies will produce new solutions in the future, since the KDE and GNOME projects have everything in place.

To keep things simple, companies doing a Linux client migration will probably stick to a single environment, so a cross-desktop menu configuration system is not really necessary. Using an LDAP-based configuration system with different user groups seeing different menu entries, directories, applications, etc. should be the goal in a larger enterprise environment though. We describe an innovative customer case in 5.3, “The self-managing Linux client” on page 111, where something similar has been realized with a special initrd file, which assembles all configuration data from many different sources and writes them into a read-only file system in memory at boot time.

In the last part of this section we discuss file system usage patterns and associated problems users face when switching from Windows to Linux.

File systems and their usage patterns

There are many different file systems, like ext2/ext3, ReiserFS, XFS, JFS, and GFS, available for Linux. If you use extended attributes (EA) and access control lists (ACLs), make sure that your backup program and other tools use them too, since these are new features for Linux. All these file systems are hierarchical and navigated from the top root node down to the directory you are looking for, that is, /home/amelie for the home directory of the user Amelie. And access controls by default would make this directory accessible only to the user Amelie, and system root.

Remember: Linux is a true multiuser system where many people can be logged into the same machine at the same time.

This means they do not have read, write, or execute access (cd /home/amelie will not work) to this directory, using the standard UNIX file system semantics (rwx for the user only but no permissions for groups and others).

In contrast to Windows, you have no drives (C:,D:, etc.) available on Linux (except when using Wine and its drive mapping features), since everything is just mounted into the file system. But you can easily add the old drive names in brackets when configuring desktop folder icons so the user recognizes his old environment. NFS, SMB, or even bind mounts (see Example 4-2) are very powerful mechanisms to build a file system hierarchy.

Example 4-2 Code

```
mount -o bind /usr/local/src /home/amelie/src
```

When a user understands this UNIX philosophy she will be very fast accessing files with a shell and TAB completion, but most Linux beginners will explore the file system using a file (and Web) browser like Konqueror or Nautilus.

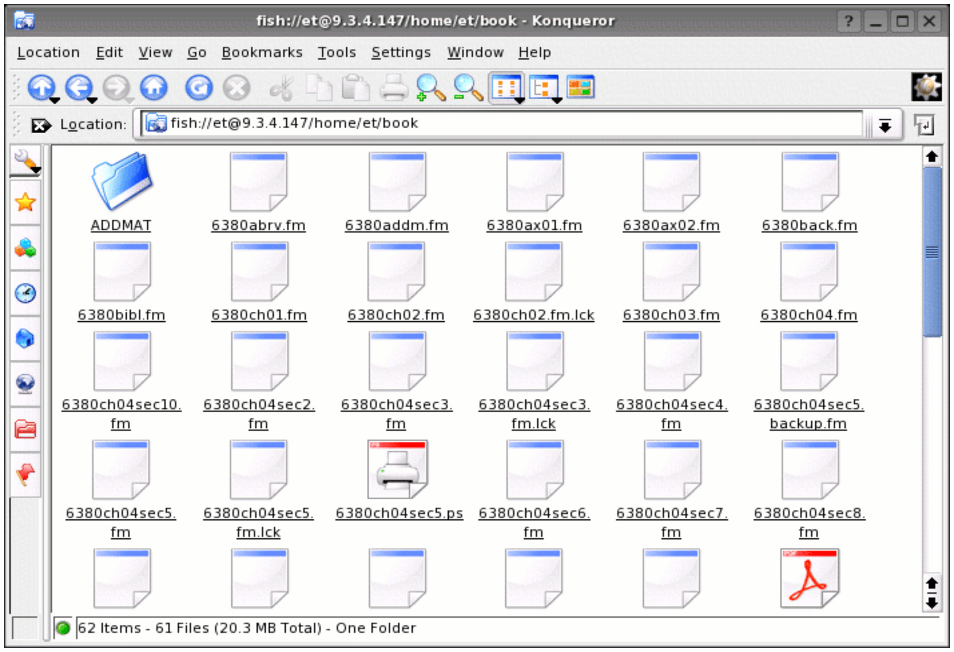


Figure 4-4 Accessing remote directory in transparent way with Konqueror (fish://)

These are very sophisticated tools, and by using virtual file system mechanisms (GNOME VFS, KIO_slaves) you can access files on remote servers or even physical devices like USB sticks, bluetooth devices, digital cameras, etc. in a transparent way by just using special URLs like ssh://, smb://, fish://, gphoto://, usb://, etc.

What the user also has to understand is that his home directory may not be locally available, but has been NFS mounted, so he has to synchronize some data when he wants to use them on the road with a mobile system (if this is possible) by using simple **`rsync`** mechanisms, special file systems like InterMezzo (`http://www.inter-mezzo.org`), or commercial programs like the Novell iFolder (the iFolder client is bundled with the Novell Linux desktop). If all data are kept locally (with regular backups to a central place) it could still be the case that e-mail data is stored on an IMAP server, so you have to make sure that your clients are supporting disconnected IMAP mode (for example, the Kolab client).

Synchronizing calendaring and other data with PDAs, Palms, or Smartphones is an another issue a system administrator has to consider. Programs available for this purpose are jPilot, kPilot, and gPilot. Also, USB sticks, bluetooth devices, and external firewire hard drives are becoming very popular these days, but from a security point of view one has to make careful decisions as to whether these devices will be allowed or deactivated, which is quite easy with Linux by deinstalling the corrsponding drivers or making driver loading impossible.

4.4 Migrating applications

A migration path needs to be determined for each application that will be migrated to a Linux-based equivalent. Different migration paths will be necessary, as not all MS Windows-based applications have Linux-based equivalents. Some example scenarios include:

- *Bridging* applications: These have native equivalents for both MS Windows and Linux. See 3.2.1, “Bridging applications” on page 30, for the definition and importance of using bridging applications if possible.)
- Similar applications: Providing same funtionality and data import capabilities, that is, OpenOffice.org provides Word, spreadsheet, and presentation capabilities and can import Microsoft office files. See 3.2.2, “Functionally equivalent utility applications” on page 31.
- Server-based applications: For an application that has no Linux-based equivalent. In this case, application servers provide some type of remote terminal service, and the Linux client then runs the application using a local remote desktop application interface. See 3.2.4, “Building bridges to the server” on page 32.
- New ISV application: This could be replacing an application with a new one, such as SAP replacing Navision.

- Web-based functional equivalent: Provide a platform-independent application interface that is browser based. See 3.2.3, “Web applications” on page 32.

4.4.1 Moving back to client/server computing

When planning for a Linux-based client migration, it is possible that you may find compelling reasons to consider moving to a client/server computing architecture for certain application services, as part of the migration. This consideration becomes especially evident when the Linux migration coincides with an application migration to a Web services based model (for example, moving from a PC-based application to an equivalent Web portal based solution).

Thus we need to consider patterns for logical segmentation of workstation types, as discussed in the following section.

4.4.2 Logical segmentation - Thin, slim, or fat

Generally speaking, a major shift from an existing client computing environment (Windows) to Linux can also trigger an expansion in workstation types that are used by the organization.

For the purpose of this logical segmentation discussion, we define three major “types” of workstations, as follows:

- Thin: Always connected; no local applications besides a displaying component, which could be either a Web browser or any other client component of a terminal services environment.
- Slim: Intermittently connected; some applications running locally (that is, not requiring synchronous server connectivity). This could be any type of implementation of such components ranging from self-written client/server to Eclipse-based Java components in a portal-based solution.
- Fat: Off-line capable; most applications are locally installed and run locally on the client.

Within a client/server environment, Figure 4-5 on page 65 shows the progression from thin to fat, where the important difference is in how many layers of the application logic are hosted locally on the client.

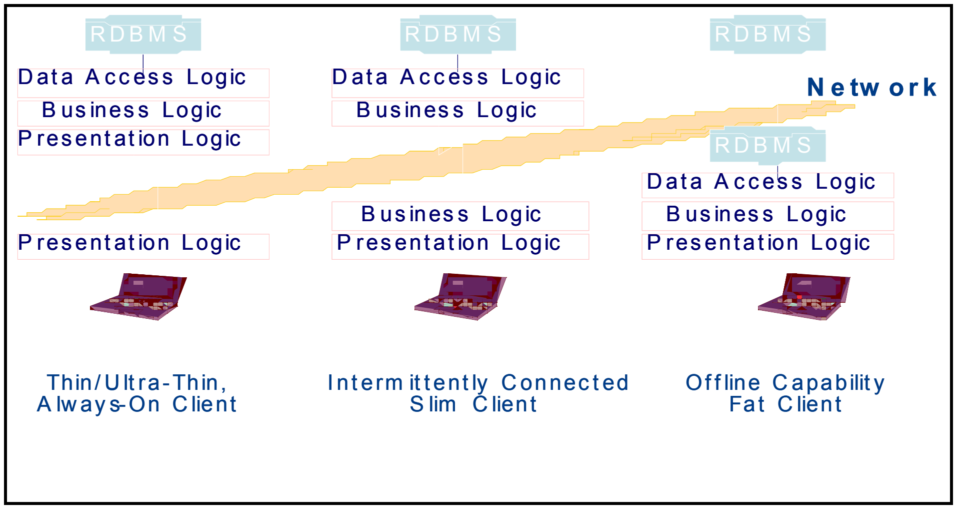
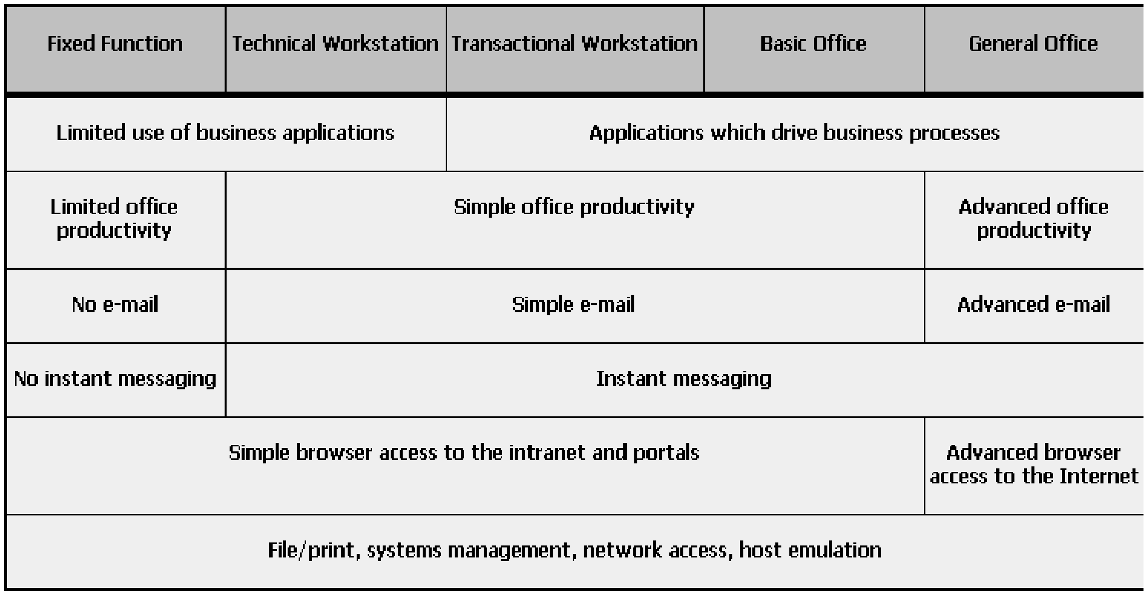


Figure 4-5 Thin, slim, or fat client

4.4.3 Functional segmentation - Fixed function to general office

When considering the range of applications hosted by a client (that is, what type of work is done at this workstation?), desktops in an enterprise or corporate environment can be roughly segmented into six distinct types, as shown in the top row of Figure 4-6 on page 66.



Fixed Function	Technical Workstation	Transactional Workstation	Basic Office	General Office
Limited use of business applications		Applications which drive business processes		
Limited office productivity	Simple office productivity			Advanced office productivity
No e-mail	Simple e-mail			Advanced e-mail
No instant messaging	Instant messaging			
Simple browser access to the intranet and portals				Advanced browser access to the Internet
File/print, systems management, network access, host emulation				

Figure 4-6 Client functional segmentation

The types are:

- Fixed function

  Users of these client machines run only a fixed and limited set of designated applications. Applications are customized for specific usage. For example, kiosk or point-of-sale terminal, portal-based application.

- Technical Workstation

  Users of these client machines work on industry-specific applications. It might require specific software packages, tailored to sector or problem domain, for example, Engineering applications (like CAD/CAM apps), entertainment applications (like movie animation).

- Transactional Workstation

  A client designed to run form-based applications to support transaction processing. Often additional functions required include access to an intranet and/or defined Internet sites, and simple e-mail (no attachments). Examples of transactional clients include travel agency workstations, bank teller workstations, and front office workstations in insurance agencies.

- Basic Office Workstation

  A client designed to run applications in support of a company's business processes. Support is required for ERP/CRM GUIs, intranet browsing, access to Internet sites, instant messaging, e-mail (with attachments), and the creation and viewing of simple documents (memos, letters, spreadsheets) within the company only. The level of Windows interoperability required will depend on the number of Windows clients deployed in the organization. The applications create files in portable formats (PDF, RTF, and HTML). Examples of basic office clients would include a loans officer workstation or an office administrator in a small business.

- General Office Workstation

  A client designed to run a broad suite of applications, including business process applications (ERP/CRM GUIs), complex compound document creation (word processing, presentation graphics, and desktop publishing), and collaboration (instant messaging, file sharing, workflow, and advanced calendaring). The level of Windows interoperability required will depend on the number of Windows clients deployed in the organization. Browsing of intranet and Internet sites is required, with support for a broad range of multimedia (streaming audio and video, shockwave, etc.). Examples of an advanced office client would include workstations to support sales and technical professionals, finance planners, and executive assistants.

Planning tip: You should expect that migration of clients that fit into the more advanced functional segments as defined in the right side of Figure 4-6 on page 66 will require more intensive application and data migration steps. In considering the overall cost of migration, it may be appropriate to identify and migrate only those workstation segments that fit into the left-hand side of this figure. As native Linux applications for the desktop mature and become more readily available, migration of the more advanced client functional segments, as shown in the right side of Figure 4-6 on page 66, should become more common.

4.4.4 Software solutions for Linux

There is growing ISV momentum around the Linux client, and the list of applications available on Linux is quickly growing every day.

Many software vendors now support Linux and are providing their applications natively under Linux, while others have adopted a Web-based model and have moved their application suites to Java-based implementations, which also make them run under Linux, assuming proprietary features such as ActiveX are not used.

More general use as well as industry-specific software applications that run natively under Linux are being released every day. The “list” of applications available today is very long, and very dynamic. Therefore, we do not think it is very useful to include a “snapshot” list of currently available applications. Instead, we direct you to learn more by visiting the Linux at IBM Solutions page:

http://www.ibm.com/linux/solutions

4.5 Client administration planning

This section describes some methods for efficient administration of the Linux client. For a corporate network with more than 20 clients or so it becomes impractical to install and maintain clients individually. An added advantage for maintenance of the Linux client is that remote logon is possible, either through ssh, telnet, or even an xterm connection.

The topics discussed in this section have to do with administering installed Linux clients after a migration. We do not discuss initial installation or roll-out of the Linux clients in this section. Topics in this section are relevant to migration planning, as it is important to consider the impact of post-migration administrative tasks.

At the end of the section we highlight two enterprise desktop distributions to show how the administration issues are solved using tools provided by those products.

Planning tip: Even though this section provides details on daily administration tasks, after a migration, these topics are still relevant. It is very important to consider the impact of post-migration administrative tasks during migration planning.

4.5.1 Operating system and vendor distribution updates

Keeping up with security concerns, and a constant stream of enhancements and bugfixes that are submitted by the OSS community as well as distribution vendors, means that there is an essentially continuous stream of updates and patches that are available. Methods for managing the update process include:

- Automation of all OS updates and patches.
- Facilitate the OS update process for the end user.
- Force critical OS updates and patches when necessary.

In the automatic update process the operating system will be updated at regular intervals. This means that a client will never be “out-of-date”. The process is completely out of the end user’s control.

When facilitating OS updates for the end user there needs to be an action of the end user to update. One possibility is to put updates on the corporate intranet and notify users to fetch and install them. This process can lead to clients with a varied status, some up-to-date and some out-of-date.

In both cases a mechanism has to be put in place that forces critical updates and patches. This can be an administrator running a script centrally, which contacts all clients remotely and forces an update using either the automated facility (casing it to run “now” instead of once a week) or the facilitated fetch and install method. This mechanism is needed in case of severe security problems where patches need to be applied immediately.

A typical Linux distribution consists of a lot of parts that all have independent updates. This means that there can be many different states of a client. Certainly a lot more than under Windows, which generally jumps with service packs (even Windows updates are generally small service packs and include changes for multiple parts of the OS). To be able to administrate a larger pool of Linux clients it is important to keep all at more or less the same level and version. This means that the automated update process is preferred over the end user driven process.

Most distributions have tools to do this update automatically. Examples are:

- Red Hat Network (up2date) tools on Red Hat; see “Administration of Red Hat Desktop” on page 73
- YaST online Update (YoU) on Suse
- Red Carpet on Novell; see “Administration of Novell Linux Desktop” on page 74

OSS alternative alternative update management applications include apt and yum.

Planning tip: Automate operating system updates and patching to keep all clients at the same level.

4.5.2 Application updates

If the application is included with and maintained as part of the distribution then the tools mentioned in the previous section can be used.

When the application is not part of the distribution there are several ways to approach the update:

- Use scripts to fetch updated files and configurations from a network server.
- For third-party applications, an update executable or script might be provided.

- Build a package that can be handled by OSS tools like apt and yum.
- If the change is extensive and the client is thin, replace the entire client image with a new one.

In early proof-of-concept tests and pilot migrations most of the changes can (and will) be done manually. When the size of the pool of Linux clients grows, this quickly becomes unfeasable, and more automated methods need to be put in place.

Planning tip: Plan to create an update mechanism during early proof-of-concept or pilot migration of the Linux client.

Special care should be taken to make sure that the desktop and menu structure are not adversely impacted when updating (or maybe even removing) applications. The menu should keep pointing to existing applications to prevent stale menu items. Icons on the desktop should point to applications still available, and to the right version if multiple versions exist simultaneously.

Important: Always include desktop and menu structure in changes related to application updates. A successfull update can appear failed because the users are left with stale icons or menu items.

4.5.3 Remote administration

The Linux client enables remote administration without having to install extra software. An administrator can log on remotely through one of the standard mechanisms, for example, ssh or telnet. This enables the administrator to analyse and fix problems remotely (except networking problems that would prevent remote access). It also enables the administrator to remotely run scripts to monitor clients and gain data about the state of the clients.

Monitoring of the clients is used to pro-actively prevent problems instead of reacting to users not being able to work. It is used to detect problems with disks, memory, and CPU usage, or even with certain applications. There are several products or solutions available to monitor systems, from commercial products like IBM Tivoli Software to OSS solutions like Nagios and Big Brother.

4.5.4 Roll-out of additional or replacement clients

In a steady state situation (that is, after the initial roll-out of Linux clients) there will still be on-going replacement and addition of client systems. For both types there should always be an up-to-date initial image to put on a new client. Some extra updates after installing the initial client should not be a problem, but a new client should not mean applying years of updates.

When replacing a client there are several things that have to be taken into account:

- In the case of a thick client, personalization data has to be transferred from the old replaced client or restored from backup.
- Configuration files need to be copied over.
- Care has to be taken that using the old name for the system will not lead to extra unnecessary problems.
- When using a new name on a replacement, make sure that the old name is removed from server-side configurations.

Important: In the case of winbind being used, take care to copy the winbind idmap file from the replaced client. Since this keeps track of domain user to uid/gid matching this is very important for file ownership.

On Red Hat systems the winbind idmap file is:

```
/var/cache/samba/winbindd_idmap.tdb
```

On Novell/SuSE systems the winbind idmap file is:

```
/var/lib/samba/winbindd_idmap.tdb
```

4.5.5 Backup of clients

Backup of the client is only important if there is changing data locally on the client. If all changing data is kept on a reliably backed up server, then replacing a broken client just means resetting with a clean initial client image or install.

If the client contains important application data, this has to be backed up. This is generally implemented in a client-server configuration where a client of the backup software is installed on the client that sends its data to the backup server. The backup server will then write the data to offline storage (tape). One such product that supports Linux clients for this purpose is IBM Tivoli Storage Manager.

However, as stated before, some configuration and cache files are important and do contain changing data even if no applications write data to local file systems. For example, end users can be allowed to make some changes to their desktop. Restoring these changes after a replacement of the client can be seen as a service to the end user. Otherwise, the end user might have to spend several hours getting it just right again.

We could make sure all the necessary files are kept by making an inventory of all these needed files during early migration or proof-of-concept and implementing a script that will collect all files in a zip or tar file and place them on a server before that server is backed up. Since these files are generally not that large, this method does not have a large impact on the backup.

Do remember to update the list of files as applications change or names of configuration files change with new versions.

Planning tip: Implement a method to back up key configuration and cache files from the client, to be used after client restore.

4.5.6 Virus mitigation

It is important to start protecting Linux clients from virus infection as soon as the clients become operational. Even though there are few viruses targetted at Linux, the number is expected to grow as the number of Linux clients grows.

Virus or worm protection is done in three ways:

- E-mail virus protection on the mail server
- Regular virus scanning on the client (at least once a day or once a week)
- Configuration of the client firewall to protect against worms

To scan for viruses on the mail server, several solutions are available, both OSS and commercial.

Virus scanning on the client should include all parts of the local file sysems. Several solutions, both OSS and commercial, are available. The virus definition files of the tool chosen should be regularly updated automatically. The end user should not be allowed to stop the scanning operation.

The Linux client firewall should be configured to defend the client against infection by worms. The firewall configuration should not be changeable by the end user and should be regularly, centrally updated.

Planning tip: Plan to start virus mitigation procedures on Linux clients early in the migration process. Linux viruses will appear eventually. Be ready.

4.5.7 Example of administration of enterprise distributions

In this section we highlight how two recent enterprise desktop distributions handle the issue of administration.

Administration of Red Hat Desktop

The Red Hat Desktop distribution is based on Red Hat Enterprise 3. Like all other products in this range, the mechanism for administration and update is Red Hat Network (RHN).

Red Hat offers three architectures for connecting to RHN: Hosted, Proxy and Satellite. Usually a client will connect directly to the RHN servers at Red Hat. This is called the Hosted architecture. Included in the Red Hat Desktop offering are the Proxy and Satellite architectures, where you can install a RHN proxy or RHN satellite server locally. The RHN proxy server caches traffic and content between the RHN servers at Red Hat and the local client. Using a satellite server it is possible to replicate a RHN server locally.

The proxy and satellite servers are useful when the number of clients increases and the traffic to Red Hat becomes large. Also these solutions will increase security because there is a limited connectivity to the Internet. An added advantage of the satellite server is that the RHN solution can be taken off-line (disconnected from the Internet) if desired.

Updating and patching the RHD client is done using up2date. The RHN alert notification tool is used to alert the end user that an update or patch is available. The update process can also be automated.

Using RHN there are three types of modules:

- Update
- Management
- Provisioning

These modules determine the service level you get from RHN.

The Update module only gives you basic update capabilites like:

- Access to a Web interface to manage the systems.
- Priority e-mail notification of new updates and errata.
- Errata information gives you a list of all available errata updates for each system.
- RPM dependency checking makes sure that every applied update gets all necessary dependencies.
- Auto update lets a system download and install errata and updates autonomically.

The Management module gives you all capabilities of the Update module plus you get extended management capabilities:

- You can group systems, to manage them as a group.

- Systems permissions can be assigned to administrators and groups of systems.
- Actions like updates can be scheduled for a system or group.
- Systems search allows you to search your systems and groups by package or errata or system specifications.
- You can compare packages between systems or build a package profile to compare systems against.

The Provisioning module is really useful when you use RHN to create new clients. Apart from everything in the Update and Management module, you also get:

- Bare metal provisioning - A tool to automatically provision a system using kickstart to deploy OS, packages, and activation keys.
- Exisiting state provisioning - Provision a system to take the state of an existing system of a predefined installation.
- A rollback functionality to roll back updates.
- Configuration management that can combine with kickstart actions to enable a comple provisioning action.
- Provision applications based on RPMs.
- Kickstart configuration writer.

All these options are available in the Hosted architecture. The Proxy and Satellite architectures add extras like custom channels, integrated network installs, and much more.

Using RHN it is possible to remote manage the RHD clients. For the Hosted architecture the management is limited to the OS and applications included in the distribution. Using the other architectures it is possible to create channels for third-party applications.

More information about RHN is available at:

`http://www.redhat.com/software/rhn`

More information is also in "Satellite server and Red Hat Network (RHN)" on page 166.

Administration of Novell Linux Desktop

The preferred method to update the Novell Linux Desktop is through the use of Red Carpet. Red Carpet Enterprise has become part of Zenworks. The new version Zenworks 6.5 includes a Linux Management part that is based on the old Red Carpet Enterprise code.

Zenworks offers a central system where client systems can be configured and where updates and patches can be pushed to the client. The central system resides inside the firewalls of the organization and can get its content (packages) from a variety of sources including Ximian servers, YaST online Update, and even Red Hat Network.

The clients connect to the server to update and patch using the rug or red-carpet programs. This connection uses the HTTP protocol. Administration is done on the Zenworks server using either the Web interface or command line commands.

The activation key that is used to activate the client actually specifies which administrators have access, which channels are available, and to which groups the client belongs.

Some of the features included in the new Zenworks 6.5 for Linux management are:

- Enhanced ACL - Closely related to Activation keys. Administrators, channels, groups, or machine notifications can be associated with an activation key.
- Check pending updates - Administrator can see which clients have updates pending and the importance of these.
- Package sets - Software packages can be grouped to a set. The sets behave as packages and can have a version and dependencies.
- Ad-hoc groups - Groups can be created whenever an Administrator needs one, for example, for all machines with a certain update pending.
- Enhanced transaction - Immediate or future execution scripts can be added or rolledback.
- Set client permissions - Client configuration files can be control edfrom the server.
- Server-side dry run - A transaction can be tested on the server to check dependency information without taxing the client.
- Machine comparisons - Tool lists differences between two or more clients.
- Choice of database backend - PostgreSQL or Oracle.
- Caching server - Server that caches software on different subnets.
- Enhanced mirroring tool - Tool generates package sets from product descriptions provided by mirrored servers.

The new client has the following additional features:

- Multiple server support - The client can access multiple Zenworks Linux management servers.
- Time-based rollback - Can be invoked locally and from the server.

- Expanded system information - The client can send hardware, software, and system information to the server.

Using Zenworks Linux management it is possible to remote manage software distribution to a large number of clients. Using the group and set functions, these client do not all have to be similar. Although a lot of the interface and commands still have Red Carpet Enterprise (RCE) embedded in them, it is clear that Linux management is going to be integrated in the Zenworks suite.

More information about Zenworks 6.5 Linux management can be found at:

`http://www.novell.com/documentation/zenworks65/index.html`

More information can also be found in “ZENworks Linux management” on page 190.

4.6 Desktop vs. notebook considerations

While looking at differences between desktops and notebooks, not only the included hardware pieces will differ, but the way of working might be different, too. Users of notebooks are considered to be working offline and therefore need to be able to synchronize their data once they are back online. Another topic in this relation is remote connectivity through wired or wireless networks.

But not only the hardware should support working in disconnected mode; the software should also have mechanisms necessary to adapt to the current connection type if necessary.

Planning tip: Notebook computer users are very likely to fall into the advanced end of the client functional segmentation shown in Figure 4-6 on page 66. Because of this, migrating notebook users will always require an extra level of technical planning, both for hardware compatibility issues as well as application migration.

4.6.1 Hardware considerations

Talking about hardware support in Linux, it is first important to state that this has improved a lot in the last years. More manufacturers see the rising distribution and use of Linux both in private and in corporate environments, and therefore start to deliver device drivers for Linux. Unfortunately there is still a lot to be done, especially in areas like wireless LAN device support, or specialities like Winmodems or GDI-printers. Also, the open source community runs several projects that develop device drivers, such as in the case of the WLan driver for the Intel Centrino™ chipset. A problem in this context is the firmware files, which are mostly closed source and so open source developers have to sometimes reverse engineer the device interfaces to achieve some level of functionality for native Linux device drivers.

Developments like this have even lead to ideas like the ndiswrapper project, which enables the loading of Windows ndis drivers in order to run network cards under Linux. Ndis is the Windows network driver API, and ndiswrapper allows use of this API as kernel modules in Linux. The URL of this project is:

`http://ndiswrapper.sourceforge.net`

Further great efforts in recent time were made in the kernel development by giving full support for all kinds of usb devices, Firewire and Bluetooth.

In conclusion of these facts, it is very important that the hardware types are determined during the planning stage of a client migration. Existing hardware that could need more effort to get it running than the current worth might have to be replaced with new pieces which are known to be compatible under Linux.

This rule applies especially for the selection of PC models. Almost every big PC vendor provides a model type that is suitable for Linux. IBM provides a Web site with information about supported models, which can be found on the following URL:

`http://www.ibm.com/pc/support/site.wss/MIGR-48NT8D.html`

Regarding desktop computers, it is quite easy to determine if Linux will run without problems; all the components, such as the graphics card, are known and can be checked for Linux support. Especially inboard components need special consideration because they likely cause problems; but even if this happens, it is possible to solve this by inserting a separate AGP card with a supported graphics chipset.

When considering notebook computers, such an exchange cannot be made, as the components cannot be removed from the mainboard. Thus it is necessary to check Linux support for Notebook models, and their features, before purchasing them.

Of special concern for mobile Linux-based computers is support for power management functions. The support of the standards APM and ACPI in Linux is still not the answer to all questions. Since ACPI is supported since Kernel 2.6, many functions can be used; but as the implementation in the hardware components is varying, not all problems are solved. For using standby in a stable way, for example, APM can still be the better choice for notebooks.

In conclusion of these considerations, it becomes clear that you have to pay special attention to the hardware that will be in use after the migration. Especially in the case of using notebooks, we highly recommend checking if Linux will run properly. Additional work might be required to have a customized version of a distribution that supports all components.

4.6.2 Peripheral extensions

When you look into the client landscape, you will recognize that many users not only have a desktop computer, but also different kinds of peripherals. Examples include:

- Locally connected printers
- Printservers
- Scanners
- Plotter
- Cardreaders
- Digital cameras

While planning the migration, it is important to assess this peripheral hardware in the same way that you assess the client hardware. Considerations on that are given in 4.1.2, “Assessing the client hardware” on page 39.

Especially in cases of cheap peripherals, we recommend building a list of approved and standardized devices and aligning this with your inventory. If some existing peripherals are hard to support in Linux it is probably cheaper buying new devices that are known to run under Linux.

Information about open source projects that support peripheral extensions can be found on the following Web sites:

- Common Unix Printing System (CUPS, see 7.5, “How to use network printers in the domain” on page 151, too):

  `http://www.cups.org`

- Scanner Access Now Easy (SANE):

  `http://www.sane-project.org`

- M.U.S.C.L.E. - Project for integrating SmartCards:

  `http://www.linuxnet.com`

- Unix SmartCard Driver Project:

  `http://smartcard.sourceforge.net`

- Project gphoto2 - Supporting digital cameras:

  `http://www.gphoto.org`

4.6.3 Connectivity options

In times of the Internet being everywhere, and everything “on demand”, connectivity becomes one of the most important features of a client. As the Internet was established once with UNIX-based servers, and Linux was created with the goal of having UNIX for PCs, it is evident that Linux can easily fulfill the requirements in this field.

Naturally, a lot of popular Ethernet network cards are supported, so in this area there should be no problems. Regarding wireless network connections, it is necessary to recognize some current limitations. Lots of the wireless network cards that are broadly available on the market are built for the SOHO-market, in which Linux only has a very small percentage. So the manufacturers of these cards want to save the money in order to be able to offer these cards very cheaply. As a result, many of these cards are hardly supported under Linux—choosing cards from better-known manufacturers is always good advice here, as, for example, Cisco offers Linux drivers for its wireless network cards.

As Internet connection via DSL or Cable modem operates via an Ethernet network interface, too, the same statement is valid here. With the aid of the PPP or PPPoE Protocol, establishing IP connections via these devices is known to be functional. Cable modem providers make this even easier by providing the IP adresses via DHCP.

A bigger problem is dial-up connections, especially in the case of the so-called winmodems, which are very common in notebook computers. These are modems that emulate the hardware components for the operating system, and so you need special software in order to use it. As software needs no resources and is cheaper to distribute, this might even be the better way. For winmodems, some Linux drivers are available for some models. Up-to-date information about this may be found at

http://linmodems.org

The support of these devices is dependent upon the distribution. Some have built in the winmodem support in their package; some have not. At least, it can be easier to use an external modem with a supported chipset than to spend a lot of effort on getting the internal winmodem working.

4.6.4 Work in offline mode

In this section we discuss working in offline mode.

Offline messaging

There are many full-featured desktop e-mail client applications available (MS Outlook, IBM Lotus Notes, etc.) that provide for offline access to a user’s e-mail.

In planning a Linux client migration for mobile (laptop-based) users, you can expect that selection of a Linux-based messaging client to replace a MS Windows-based client will be the single most important design decision in your mobile desktop application migration strategy. The challenge of providing for a functionally equivalent (or acceptable) messaging application that supports offline use is another reason that inclusion of notebook-based users can increase the complexity of a migration. (See the “Planning Tip” in 4.6, “Desktop vs. notebook considerations” on page 76.)

There are Linux-based messaging client options that provide for offline access modes. Mozilla Thunderbird is one example. Thunderbird supports both POP and IMAP protocols. More information about Thunderbird can be found here:

```
http://www.mozilla.org
```

Although Thunderbird will support offline messaging modes, it does not provide a way to integrate calendaring and scheduling functionality into the messaging client the way systems such as MS Exchange/Outlook and IBM Lotus Domino/Notes do. For this reason, we choose to demonstrate use of Novell’s Exchange Connector 2000 combined with the Novell Ximian Evolution e-mail client in Chapter 6, “Client migration scenario” on page 115. Extending the example migration scenario in Chapter 7, “Integration how-tos” on page 137, for the notebook user means that you will also have to test and evaluate the offline capabilities of that Novell-based solution.

Offline files

As every user to some extent creates and manages her own files and folders, it follows that the notebook user should be able to access copies of her files in offline/disconnected mode. And the primary storage location for those same files should be a regularly backed up and highly available server.

For example, Microsoft provides methods for automatic synchronization of files between servers and mobile clients. One of these methods uses the *Briefcase*, a special folder in which one had to put the files that should be synchronized. A second method was provided with the launch of Windows 2000/XP, where a new function called *offline files* was implemented. This mechanism is able to synchronize whole folders, and while the notebook is offline, the files are fully accessible. Changes are retransfered after connecting to the network automatically.

In Linux,there is no direct equivalent for this kind of server-to-client file syncronization built into the operating system itself. Instead, the task of developing and supporting Linux-based file synchronization services between client and server is left to ISVs. For example, Novell is offering a new solution for taking files offline and sharing them with other users. The product is called *iFolder* and is part of their Nterprise Linux Services product. More information can be found at:

http://www.novell.com/products/ifolder

Another possibility is to use the rsync protocol for this purpose. rsync is a powerful open source Linux utility that could be used for incremental file transfer. Used with the right options, it can fulfill the demand of syncing local and remote folders. As it builds checksums of the files and transfers only the missing parts, it can do this transfer very effectively over small-bandwith connections. The Web site of rsync is:

http://samba.anu.edu/rsync

By using rsync, you can develop a client/server file synchronization tool that could be run as both a scheduled or user-triggered service on notebook computers. But it is recommended that you develop a user-friendly wrapper (or client-side script) that controls this process, and thus hides complexity from the end user.

4.7 Un-migrateable applications

This section discusses issues related to applications that cannot be migrated to run directly on a Linux client. We propose alternate ways to support these applications for access from a Linux client.

We assume that the number of unmigratable applications is small and that a small percentage of end users are using these applications. If this is not the case, then the case for the client migrations has to be resonsidered.

4.7.1 What makes an application unmigratable

We define an application as unmigratable when one or more of the following statements about the application are true:

- A Linux version of the application or an alternative application does not exist.
- Porting the application to Linux is not feasible.
- License issues make a move to Linux impossible or highly expensive.

Once an application is designated as unmigratable there are several ways to migrate to a Linux client and solve the issues around this application:

- Investigate whether the application can run on a Windows server and be used through remote access mechanisms such as a Terminal Server, Citrix Metaframe, NoMachine, etc.

- Examine whether it is possible from a cost perspective to run VMware workstation on the client to create virtual Windows machines on those clients that still need to run the application natively in Windows.
- Create dual-boot clients if the application is not used very often.
- Leave some clients out of the migration and consolidate all unmigratable applications on those shared clients, for use by all end users who need to.

4.7.2 Terminal Server, Citrix Metaframe, or NoMachine solutions

If the application can run centralized on a server, solutions like Windows Terminal Server, Citrix Metaframe, or NoMachine can be used to access them remotely. Before an application can be moved to a central server some conditions have to be met:

- The application has to be able to run on a multi-user environment. A multi-user environment has several consequences:
  - Settings are stored in multiple places in the registry and file system.
  - More than one user can be running the application simultaneously.
- The application license must allow running on a multi-user environment.
- Resource needs of the application have to fit with more than one instance of the application running or with multiple applications running on the server.

Client applications are available for Linux for Terminal Server, Citrix Metaframe, and NoMachine. The client for Terminal Server is OSS. The Citrix client is available when you purchase Metaframe solutions. NoMachine has the NX client available for download.

4.7.3 VMware solutions

Using VMware (or any similar application that runs on Linux), it is possible to create a virtual Windows machine that the end user can use to work with applications that cannot be moved to Linux or a central server.

The virtual machine will be a completely functioning Windows machine and could be loaded with the image of a Windows client already in use before the migration to Linux. When using a domain this means that the virtual machine has to become part of the domain as well. The virtual machine has to be connected to the network via bridging or NAT.

This solution has several disadvantages:

- A full Windows license is needed for the virtual machine.
- VMware software needs a license also; this will lead to extra cost.

- Extra management tasks to keep VMs running on clients.
- Possible extra requirements on the client resources in terms of memory and diskspace.
- End user might be tempted to work in the virtual machine to avoid using the new Linux client.

Whether this solution is feasible with the extra cost depends on the number of clients involved and the cost of alternative solutions.

4.7.4 Dual boot solution

The solution to the unmigratable application problem using a dual-boot client is essentially the same as the VMware solution, except that in this case the end user can use either Linux or Windows and never both at the same time.

The disadvantages of this solution are:

- A full Windows license is needed to boot the client using a Windows operating system.
- Extra management and support tasks are needed for the Windows part of the dual-boot system.
- Extra complexity involved with dual-boot systems:
  - Unclear whether client is Windows or Linux; use different host names and IP addresses for both operating systems?
  - End user cannot switch quickly and will need an extra partition to work with files on both operating systems.
- End user may be tempted to boot to Windows and keep working using the old client.

Advantages over the VMware solution are that there are no VMware license costs, and no extra memory or CPU power is needed since only one operating system will be running at any one time. In this case extra diskspace is needed in the form of at least one partition. For both the VMware and the dual-boot solution it is best to make a minimal Windows client available. This way only the application that still needs to run under Windows is run under Windows.

4.7.5 What to do if all else fails

There may be applications left that do not lend themselves be moved off a Windows client using any of the mentioned methods. If these applications are to be used after the client migration because of economical, legal, legacy, or other reasons, a solution has to be found.

At present it is possible to use a Windows emulator (like Wine) to get an application running on the Linux client. This should, however, be seen as a temporary workaround. This is not a solution for the long run.

The most economical solution in the case of unmigratable applications might be to consolidate all unmigratable applications to a fixed number of Windows clients. These are then used by the entire end-user population to access the unmigratable applications. These central clients are used in a single-user mode with a single user sitting at the keyboard. Because if multiple users can access the application at once it is better to use a remote solution like in 4.7.2, “Terminal Server, Citrix Metaframe, or NoMachine solutions” on page 82. The centralized clients can be used remotely by tools that enable remote use by a single user. An example of a tool like this is VNC over a secure socket layer connection. This will remote the display, mouse, and keyboard to the VNC client on the Linux desktop. By using a secure socket connection we make sure that the remote connection is secure.

Of course, if the unmigratable application is a heavily used application, all these methods are unusable and a Linux client migration becomes extremely difficult.

4.8 Post-migration troubleshooting and technical support

After migrating a group of clients to Linux, the clients will have to be supported—not just in the normal day-to-day operations, but also for problems arising from the migration itself.

Not all of the support is of a technical nature. The usage patterns on the Linux client will be different from what the end user was used to on his previous client.

The administration of a Linux client also needs a new methodology for troubleshooting. Finding and solving problems under Linux is very different from Windows. Administrators have to adapt to the new troubleshooting methods for Linux clients. The first step in this process for the Linux client is not a reboot like it mostly is on the Windows client.

4.8.1 What to expect

To prevent a lot of post-migration troubleshooting the end users have to be prepared for the new Linux client. The expectations for the Linux client have to be managed very carefully.

Changes for the end user in migrating to the Linux client might be:

- The end user no longer has full control over file system contents.
- Even when the desktop presents a similar look and feel, the process of changing the desktop settings is completely different.
- Right-click menus are either gone or styled completely differently.
- The look and feel of favorite desktop applications has changed, or those applications have been entirely replaced with functional equivalents.

Most of these changes can be managed and anticipated by providing proper training and enablement materials. To prepare for this training a mapping has to be created that maps client tasks to applications on both the old and the new client. This mapping may be different for all roles in the organization. For example, a transactional worker using an application that migrates to a Linux version will have very little impact. If an application is replaced by another with a simular function the changes may be large.

4.8.2 How to handle the unexpected

While it is possible to fully prepare the end user in advance, some problems will still arise post-migration and during normal operation. Most of these problems are tasks for which support staff and administrators can be prepared and trained.

To tackle the unexpected problems, the support staff has to use an OSS/Linux-oriented approach to problem solving. To enable support staff to become used to this different approach, the differences have to be investigated.

In general, Windows operating system methodology for troubleshooting usualy starts with:

- Rebooting the system
- Checking eventlog
- Checking drivers and installing latest versions

The Linux operating system has more easily identifiable modules. Also there are lots of log files. This means that the methodology for troubleshooting starts with:

- Indentify the affected module of the Linux distribution.
- Check all involved system logs starting with the syslog.
- Verify configuration files do not contain errors or faulty settings.
- Restart services involved.
- Check errors generated by restart.

When using a Windows client, a reboot will solve more than half of the problems; when using a Linux client the problem usually remains after a reboot. This means that support staff has to learn how to approach the problem on the new client.

Generally, the best way to prepare support staff for Linux client problems depends on which level of support is given. In the case of helpdesk or level 1 support the best training is using a Linux client. When the support moves more toward administrative tasks, a basic Linux (or even Unix) training will have the best effect.

4.8.3 When to contact vendor enterprise support

Vendor support is usually contacted when the problem is found to be a bug or when software is not acting as stated in specifications or manuals. If the problem is not related to a vendor-specific part of the Linux OS or suite of applications, it is also possible to find a solution or a fix in the open source community.

The decision for searching community support has to be made for each issue. When a support contract with an enterprise distribution vendor is in place it is best to explore that avenue first, because trying to incorporate a community fix in a vendor distribution might lead to new problems.

Part 3

Performing the pilot migration

5

Migration best practices

In this chapter we describe some best practice methods you can use in your own Linux client migration projects. The topics covered are use of the KDE Kiosk mode, Gnome customization, and a general description of a self-managing Linux client solution based on Red Hat. Related automation topics are also discussed in Appendix C, “Desktop automation and scripting” on page 215.

The sections in this chapter are:

- 5.1, “KDE Kiosk framework” on page 90

  Methods for “locking down” the KDE desktop using the KDE Kiosk framework are described.
- 5.2, “GNOME customization” on page 104

  Methods for using the Gnome configuration system and gconftool-2 are demonstrated.
- 5.3, “The self-managing Linux client” on page 111

  Design strategies to deploy advanced self-managing Linux clients across a large enterprise are discussed.

5.1 KDE Kiosk framework

In this section we describe how to lock down your KDE desktop with the Kiosk framework that has been introduced in KDE3. We show you by editing the configuration files directly or using the very comfortable Kiosk Admin Tool[1] administration GUI tool how to map profiles to users and groups; mark configuration entries as immutable; and restrict actions, URLs, and resources.

For an introduction to how KDE user profile (personalization) data is contained see Appendix D, "Client personalization" on page 221.

5.1.1 Profiles

An out-of-the-box KDE installation gives the user a lot of power to change her settings according to personal preferences. In an enterprise environment this is probably not what you want, and one of the reasons why the Kiosk framework was introduced in KDE3. It allows you to disable certain KDE features to create a more controlled environment. It is built on top of KDE's configuration framework and adds a simple API that applications can query to get authorization for certain operations.

The KDE Kiosk framework should be used in addition to standard Linux security measures, that is, the files that disable certain features should only be writeable by the Kiosk administrator, which can be the root user or somebody else designated specifically for that task. The Kiosk framework uses the standard UNIX user and group semantics and adds a profile layer to it. Profiles are configuration sets that specify what can be done when working with KDE and are assiciated to single users or groups of users.

You have to tell KDE in the global configuration file /etc/kderc (another choice would be /usr/share/config/kdeglobals) which profiles are available and where to find the mapping file. In Example 5-1 we showcase the configuration with two different user profiles, which are internationalized (English and German [de]) and owned by the root user (ProfileInstallUser=root). The other three profiles look identical.

Example 5-1 Profiles and the mapping file are specified in /etc/kderc

```
[Directories]
kioskAdmin=root:
profileDirsPrefix=/etc/kde-profile/
userProfileMapFile=/etc/kde-user-profile

[Directories-default]
```

[1] http://extragear.kde.org/apps/kiosktool.php

```
ProfileDescription=Default profile
ProfileDescription[de]=Standard Profil
ProfileInstallUser=root
prefixes=/etc/kde-profile/default/

[Directories-ITSO]
ProfileDescription=ITSO Test Profile
ProfileDescription[de]=ITSO Standard Profil
ProfileInstallUser=root
prefixes=/etc/kde-profile/ITSO/
```

For our IBM Technical Support Organization (ITSO) migration we choose five different profiles specified by their unique names *default*, *Redbook*, *ITSO*, *Designer*, and *Administrator*, and corresponding configuration directories /etc/kde-profile/[Profile Name]. In Table 5-1 you can see the meaning of these five profiles we are using in our test scenario.

Table 5-1 Profiles and their meanings

Profile	Role
default	No specific role, default
Redbook	Redbook writers
ITSO	ITSO staff
Designer	ITSO staff with designer tasks
Administrator	Be nice to that person!

In the next step we have to assign profiles to our users and groups. You can use the graphical KDE Kiosk Admin Tool for that task (see Figure 5-3 on page 94) or just edit the configuration file by hand (see Example 5-2 on page 93):

http://extragear.kde.org/apps/kiosktool.php

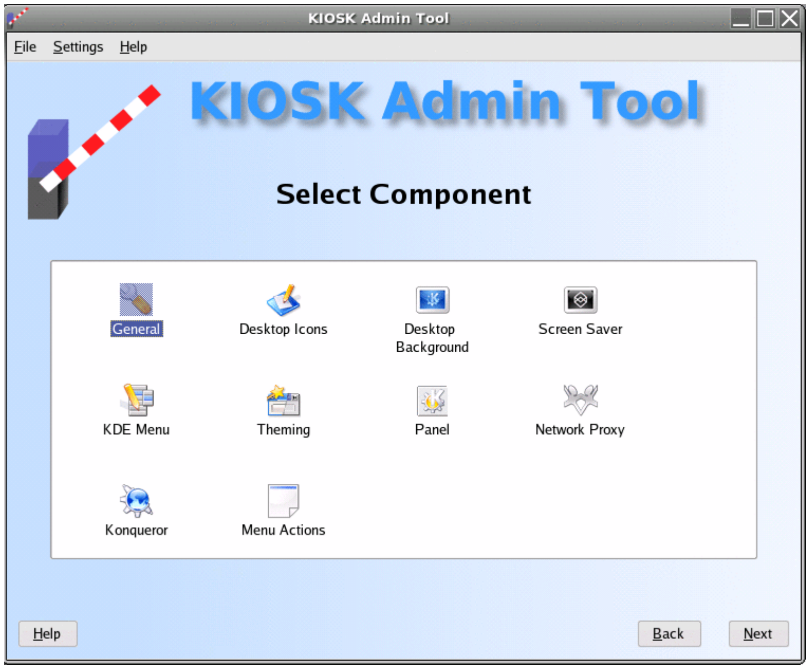
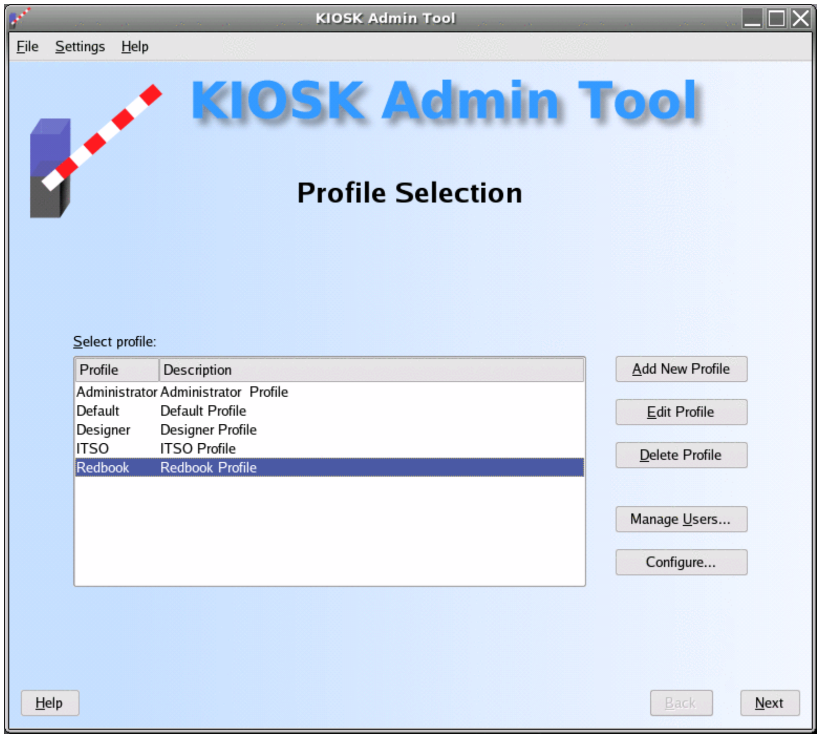


Figure 5-1 KDE Kiosk Admin Tool



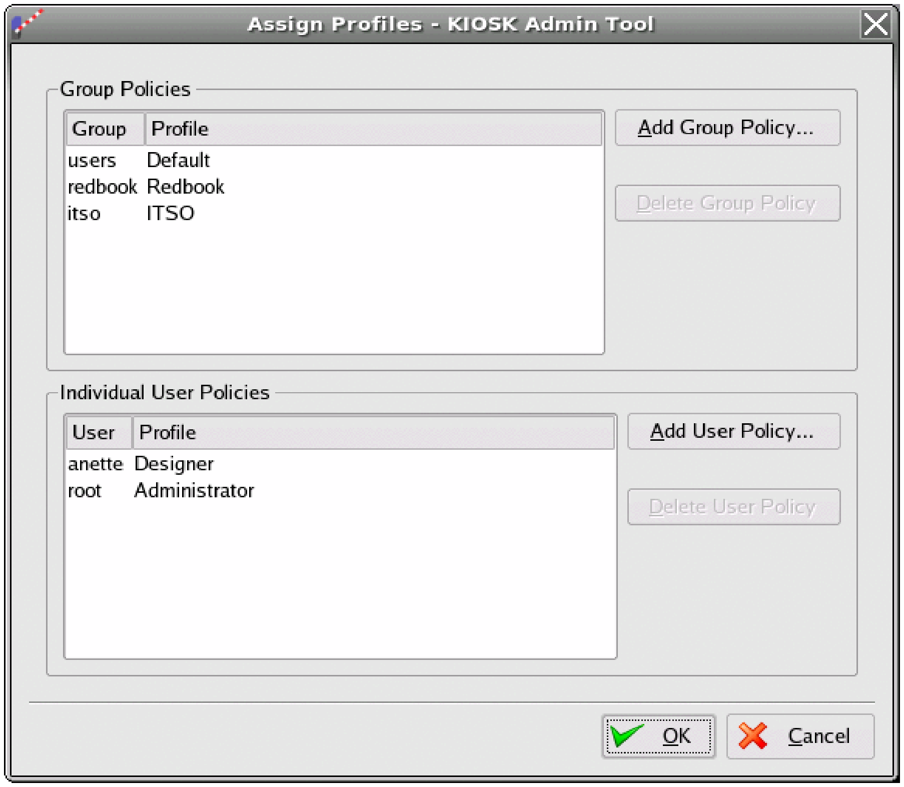
Profile	Description
Administrator	Administrator Profile
Default	Default Profile
Designer	Designer Profile
ITSO	ITSO Profile
Redbook	Redbook Profile

Figure 5-2 User pofile with the Kiosk Admin Tool

When a user has more than one profile (for example, amelie=Administrator,Designer) asossiciated then the settings of the profiles listed first have higher priority in conflicting situations. The same is true for groups in the Groups section (for example, partner=Partner,Redbook for the partner group, in which case the Partner profile will have higher priority).

Example 5-2 Mapping pofiles to users and groups in /etc/kde-user-profile

```
[General]
groups=itso,redbook,users

[Groups]
itso=ITSO
redbook=Redbook
users=default

[Users]
anette=Designer
root=Administrator
```



Group	Profile
users	Default
redbook	Redbook
itso	ITSO

User	Profile
anette	Designer
root	Administrator

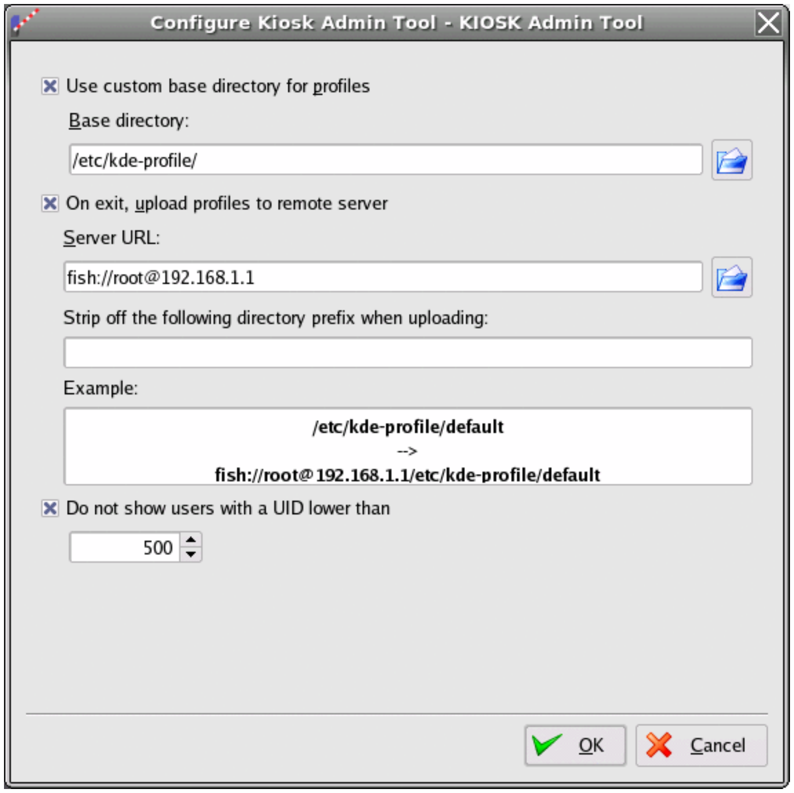
Figure 5-3 Configuring users and groups with the Kiosk Admin Tool

Another interesting case happens when a user not listed in the Users section is part of different UNIX groups that are mapped to different profiles in the Groups section. In that case the profiles listed earlier will have higher priority. Last but not least, if a user is listed in the Users section only the profiles in this entry are taken into account (that is, the UNIX groups the user belongs to and which are mapped to profiles play no role in that case).

Although all these rules are quite straightforward, a simple or even linear profile hierarchy will be suffient in most cases. For example, you can design the default user profile first, add some features for a group needing more privileges, and so on. If there are groups not fitting in that hierarchy (in our case this could be the designer group), build their profile separately and either merge it with a profile in the first hierarchy or use it on its own.

A nice feature of the Kiosk Admin Tool is its remote configuration ability. In Figure 5-4 on page 95 you see how to map a local directory to a remote one (you can even change the base URL path). Files are transferred, for example, by the

SSH protocol and the usual fish:// semantics or any other protocol that is supported by the KDE networking framework.


Configure Kiosk Admin Tool - KIOSK Admin Tool

☒ Use custom base directory for profiles

Base directory: /etc/kde-profile/

☒ On exit, upload profiles to remote server

Server URL: fish://root@192.168.1.1

Strip off the following directory prefix when uploading:

Example:

```
/etc/kde-profile/default
-->
fish://root@192.168.1.1/etc/kde-profile/default
```

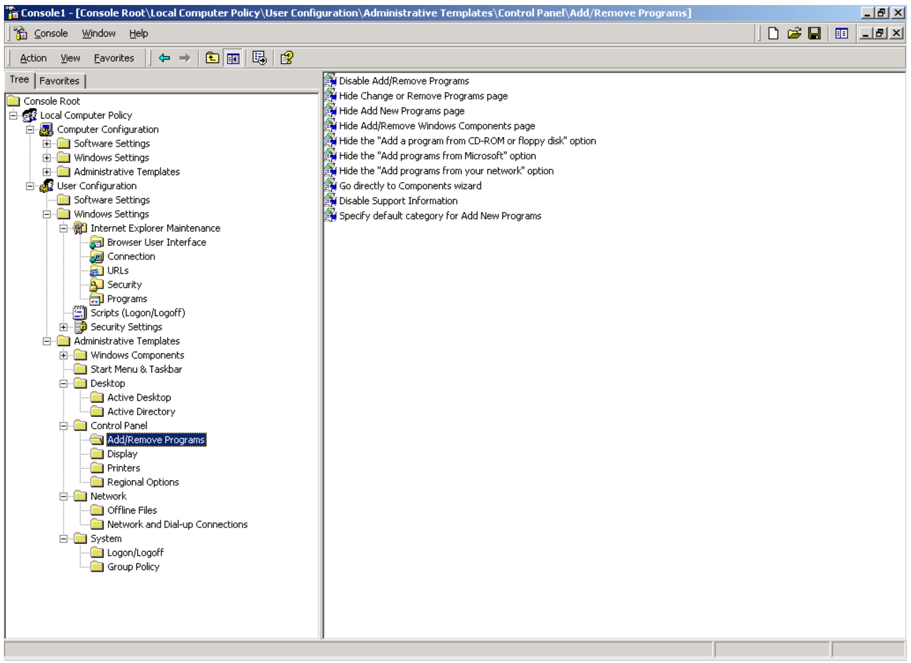
☒ Do not show users with a UID lower than 500

Figure 5-4 Remote configuration with the Kiosk Admin Tool

5.1.2 The Kiosk Administration GUI tool

While exploring the lockdown possibilities of the KDE Kiosk Admin Tool you will find some similarities with the Microsoft Management Console (MMC), which you can see in Figure 5-5 on page 96. The Kiosk framework not only visually hides entities and features of the desktop, but makes them really unusable. Even if the lockdown options of the Kiosk Admin Tool look quite simple, you have to know that under the hood you can configure and fine tune a lot more.

Disabling Alt+F2 (run command) and access to a command shell might be interesting options for a public terminal, but one has to be aware of the fact that it is quite easy to open a shell (xterm) with Mozilla, so one has to take care of that, too, by other methods than the Kiosk framework, which Mozilla does not use. When you disable the logout option you also have to disable shutting down the X server by pressing Ctrl+Alt+Return, or the whole machine with Ctrl+Alt+Delete (in (/etc/inittab); otherwise the whole exercise could be quite useless.

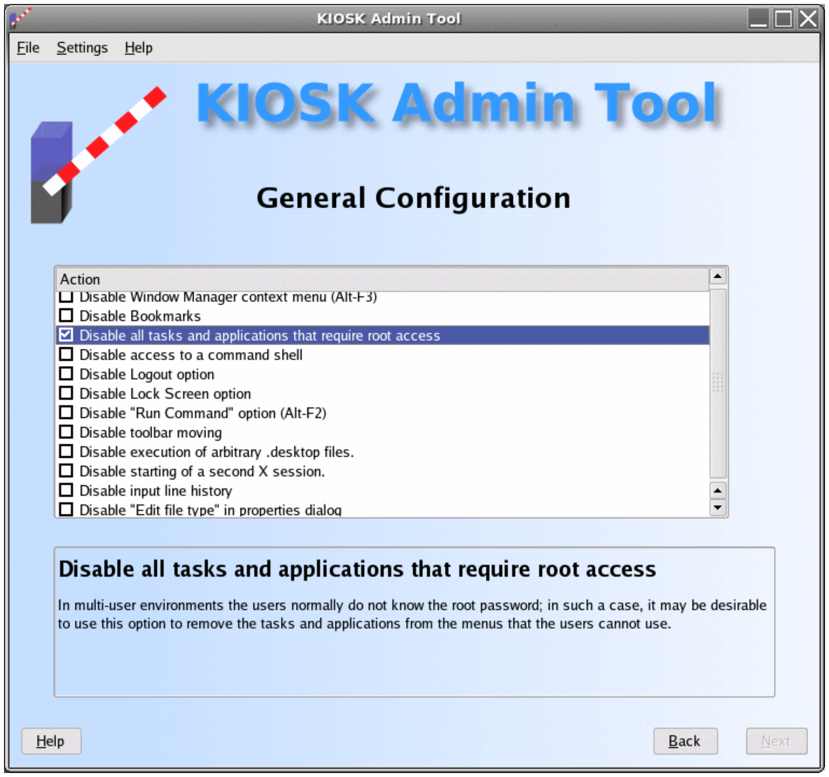


Disable Add/Remove Programs
Hide Change or Remove Programs page
Hide Add New Programs page
Hide Add/Remove Windows Components page
Hide the "Add a program from CD-ROM or floppy disk" option
Hide the "Add programs from Microsoft" option
Hide the "Add programs from your network" option
Go directly to Components wizard
Disable Support Information
Specify default category for Add New Programs

Figure 5-5 The Microsoft Management Console (MMC)

All these remarks show that one has to consider a lot of side effects to make the whole setup bullet proof, since so many Linux applications and subsystems do not use the KDE Kiosk (or the work-in-progress GNOME lockdown) framework. Let us take a look at all the available general options now. For the general configuration:

- Disable window manager context menu (Alt+F3).
- Disable bookmarks.
- Disable all tasks and applications that require root access.
- Disable access to a command shell.
- Disable logout option.
- Disable lock screen option.
- Disable Run Command option (Alt+F2).
- Disable toolbar moving.
- Disable execution of arbitrary .desktop files.
- Disable startig of a second X session.
- Disable input line history.
- Disable “Edit file type“ in properties dialog.



- ☐ Disable Window Manager context menu (Alt-F3)
- ☐ Disable Bookmarks
- ☑ Disable all tasks and applications that require root access
- ☐ Disable access to a command shell
- ☐ Disable Logout option
- ☐ Disable Lock Screen option
- ☐ Disable "Run Command" option (Alt-F2)
- ☐ Disable toolbar moving
- ☐ Disable execution of arbitrary .desktop files.
- ☐ Disable starting of a second X session.
- ☐ Disable input line history
- ☐ Disable "Edit file type" in properties dialog

Disable all tasks and applications that require root access

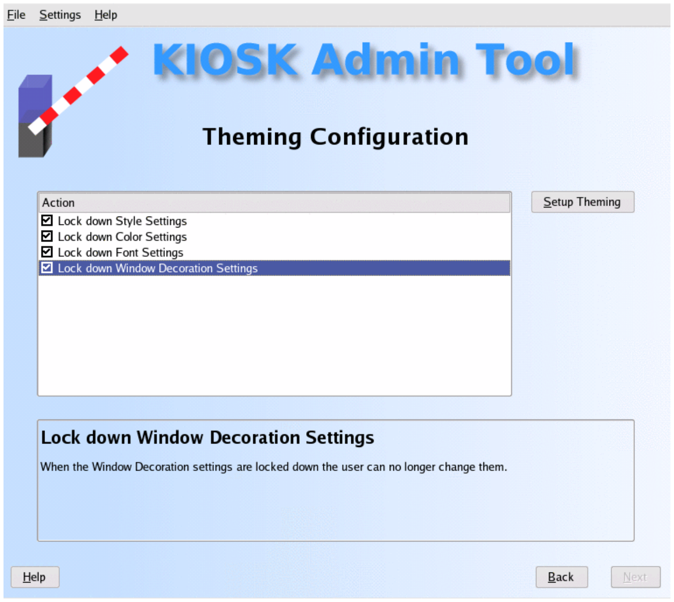
In multi-user environments the users normally do not know the root password; in such a case, it may be desirable to use this option to remove the tasks and applications from the menus that the users cannot use.

Figure 5-6 Genereal configuration with the Kiosk Admin Tool

The following sections can be configured live, that is, you can add applets to a second Kicker pannel, change the background image or color, manipulate menu entries, and add icons on a second desktop and see the results immediately.

- Desktop icons configuration
  - Lock down desktop settings.
  - Disable context menus.
  - Lock down all desktop icons.
  - Lock down system-wide desktop icons.
- Desktop background configuration
  - Lock down desktop background settings.
- Screen saver configuration
  - Lock down screen saver settings.
  - Disable OpenGL-based screen savers.
  - Discrete screen savers only.

- KDE menu configuration
  - Disable all tasks and applications that require root access.
  - Disable menu editing.



KIOSK Admin Tool

Theming Configuration

Action
☑ Lock down Style Settings
☑ Lock down Color Settings
☑ Lock down Font Settings
☑ Lock down Window Decoration Settings

Lock down Window Decoration Settings

When the Window Decoration settings are locked down the user can no longer change them.

Figure 5-7 Themeing configuration with the Kiosk Admin Tool

- Theming configuration
  - Lock down style settings.
  - Lock down color settings.
  - Lock down font settings.
  - Lock down window decoration settings.
- Pannel configuration
  - Lock down panel.
  - Disable context menus.
  - Disable menu editing.

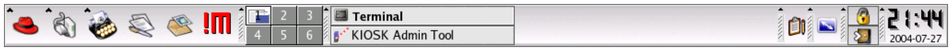
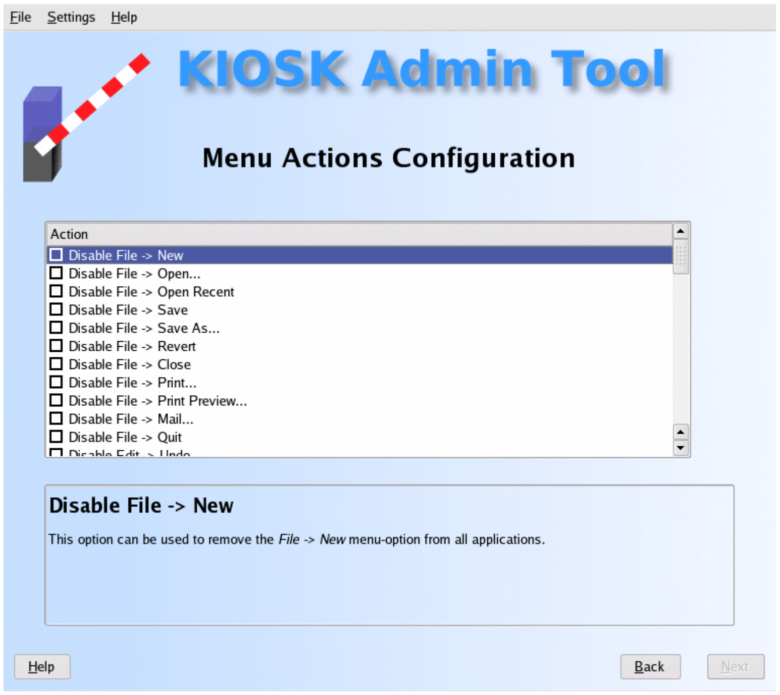


Figure 5-8 Standard KDE pannel configured with the Kiosk Admin Tool

- Network proxy configuration
  - Lock down proxy settings.

The disabling of actions in the next two sections is an art itself since there are so many of them. We look at that in more detail in 5.1.4, "Action restrictions" on page 100.

- Konqueror configuration
  - Disable properties in context menu.
  - Disable Open With action.
  - Disable Open in New Tab action.
  - Disable file browsing outside home directoy.
- Menu actions
  - Disable **File** → **New**.
  - Disable **File** → **Open**.
  - Disable **File** → **Open Recent**.
  - Disable **Help** → **About <Application>**.
  - Disable **Help** → **About KDE**.
  - Etc.



Action
- [ ] Disable File -> New
- [ ] Disable File -> Open...
- [ ] Disable File -> Open Recent
- [ ] Disable File -> Save
- [ ] Disable File -> Save As...
- [ ] Disable File -> Revert
- [ ] Disable File -> Close
- [ ] Disable File -> Print...
- [ ] Disable File -> Print Preview...
- [ ] Disable File -> Mail...
- [ ] Disable File -> Quit
- [ ] Disable Edit -> Undo

Disable File -> New

This option can be used to remove the *File -> New* menu-option from all applications.

Figure 5-9 Menu actions configuration with the Kiosk Admin Tool

5.1.3 Immutable configuration file entries

As we have seen, it is quite easy to generate lockdown profiles with the graphical Kiosk Admin Tool, but what does this program really do behind the scenes? Starting with KDE 3, configuration entries can be marked immutable, and once such a value has been read its value cannot be changed any more via KConfig or user entries in $KDEHOME (normally $HOME/.kde).

Entries can be marked immutable on an entry-by-entry, group, or file basis by adding a [$i] at the right places (see Example 5-3). If KDE does not have write access to the user's configuration files they will automatically be considered immutable and the user will be warned about that fact. If you do not like this behavior, add a warn_unwritable_config=false to the KDE Action Restrictions section in /etc/kderc (or kdeglobals on the gloabal, profile, or user level) to disable this warning for all applications. As we already mentioned, non writable user configuration files are not a foolproof lock down mechanism since the user can potentially rename these files and add new ones according to his taste. Consider the file system mechanisms an add-on to the much more sophisticated KDE Kiosk framework.

Example 5-3 Immutable entries by the [$i] magic

```
[ScreenSaver]
Enabled[$i]=true

[Desktop0][$i]
Wallpaper=/usr/share/backgrounds/images/default.png
WallpaperMode=Scaled

[$i]
[Applet_1]
ConfigFile=kminipagerappletrc
DesktopFile=minipagerapplet.desktop
FreeSpace=0
WidthForHeightHint=92
```

5.1.4 Action restrictions

Using the Kiosk Admin Tool, we already encountered action restrictions that are configured on a profile level in the kdeglobals file in the KDE Action Restrictions section. For our ITSO profile take a look at Example 5-4 to see what kind of entries have been generated

Example 5-4 Action restrictions in /etc/kde-profile/itso/share/config/kdeglobals

```
[KDE Action Restrictions][$i]
action/kdesktop_rmb=false
```

```
action/kicker_rmb=false
action/menuedit=false
editable_desktop_icons=false
editable_system_desktop_icons=false
movable_toolbars=false
run_command=false
user/root=false
```

There are many actions available and a lot more will be added in the future by the different applications using the KDE framework. You have actions that refer on a global level to menu and toolbar entries (for example, action/file_new) and general actions for:

- Printing (print/options)
- Screensavers (opengl_screensavers)
- Desktop (logout, lock_screen, movable_toolbars, start_new_session)

There are also actions related to standard KDE programms like:

- Konqueror (action/openintab) and KDesktop
- KWin (action/kwin_rmb)
- Kicker (action/kicker_rmb)
- Konsole (action/show_menubar)

There are also a lot of additional application actions that can be explored as usual with the **dcop** command on the command line or with the KDE program kdcop in an easy-to-use graphical way. Enter, for example, one of the following commands after starting kmail to see what actions kmail offers.

```
% dcop kmail qt objects | grep KActionCollection | cut -d '/' -f 3
% dcop kmail kmail-mainwindow actions
```

There are also actions that refer to applications that need to be run as a different user. They are prefixed with a user/ followed by the user name (for example, user/root=false to disable all application entries that require root access). If you are interested in even more subtleties check the Kiosk readme:

http://webcvs.kde.org/cgi-bin/cvsweb.cgi/kdelibs/kdecore/README.kiosk

5.1.5 URL restrictions

It is possible to restrict URL-related actions with the Kiosk framework based on the action, the URL in question, or in some cases the referring URL. As you can see in Example 5-8 on page 98, the general syntax is quite long, so we will explain some simple cases.

Example 5-5 URL restriction: General syntax

```
[KDE URL Restrictions]
```

```
rule_count=<N>
rule_1=<act>,<ref_proto>,<ref_host>,<ref_path>,<proto>,<host>,<path>,<enabled>
...
rule_N=<act>,<ref_proto>,<ref_host>,<ref_path>,<proto>,<host>,<path>,<enabled>
```

In the first part of Example 5-6 you can see how to disable browsing with KDE's file dialogs outside the $HOME directory by using the list action. The first rule disables browsing any directories on the local file system, while the second enables browsing in the $HOME directory, which is exactly what we want. You can also see that KDE expands the environment variable $HOME. Usually one has to put [$e] after the entry name, or even [$ei] to prevent the application replacing the entry with the actual environment value's value after saving, but this is not necessary everywhere. The second part of Example 5-6 shows how to allow the user to open files in the $HOME and $TMP directories but nowhere else on the file system (opening files from the Internet is still possible though).

Example 5-6 URL restriction examples

```
[KDE URL Restrictions][$i]
rule_count=2
rule_1=list,,,,file,,,false
rule_2=list,,,,file,,$HOME,true

[KDE URL Restrictions][$i]
rule_count=3
rule_1=open,,,,file,,,false
rule_2=open,,,,file,,$HOME,true
rule_3=open,,,,file,,$TMP,true
```

You can also use shell commands in KDE configuration files, as in Example 5-7, but do not overuse this feature.

Example 5-7 Using shell commands in KDE configuration files

```
Host[$e]=$(hostname)
...
[Icons]
Theme[$e]=$(source /usr/share/config/kdeglobals.defaults && echo $Theme)
```

5.1.6 Resource restrictions

The last lockdown capability we will discuss is KDE resource restrictions, which make it impossible for users to override file lookup outside $KDEHOME/share (where most KDE applications put their resources like HTML documentation, sound files, etc.) with resources lying in $KDEHOME/share. In the first part of

Example 5-8 we do this for all resources, while in the second part we use a more fine granular approach by just restricting sound, localization, and wallpaper files.

Example 5-8 Resource restriction covering all resources

```
[KDE Resource Restrictions][$i]
all=false

[KDE Resource Restrictions][$i]
sound=false
locale=false
wallpaper=false
```

All the entries we could have used are

- data (share/data)
- html (share/doc/HTML)
- icon (share/icon)
- config (share/config)
- pixmap (share/pixmaps)
- apps (share/applnk)
- xdgdata-apps (share/applications)
- sound (share/sound)
- locale (share/locale)
- services (share/services)
- servicetypes (share/servicetypes)
- mime (share/mimelnk)
- wallpaper (share/wallpaper)
- templates (share/templates)
- exe (share/bin)
- lib (share/lib)
- all (share/)
- data_<application name> (share/apps/<application_name>)

This is somewhat confusing since not all directory names are mapped to the same names in the configuration file. Example 5-9 shows how to use the Control Module and Custom Restriction sections in the kdeglobals file.

Example 5-9 More KDE restriction options

```
[KDE Control Module Restrictions]
kde-background.desktop=false
kde-colors.desktop=false
kde-fonts.desktop=false
kde-kwindecoration.desktop=false
kde-proxy.desktop=false
kde-style.desktop=false
```

```
[KDE Custom Restrictions]
restrict_file_browsing=true
```

5.1.7 Summary

After reading this section you should have quite a good understanding of the KDE Kiosk framework and be able to use it in your own Linux client migrations. Do not overuse its features though, since the more you configure the more you have to maintain later. You certainly know this effect from using the Microsoft management console, which gives you a lot of possibilities too. Windows profiles cannot be migrated feature-wise one-by-one, so you have to mimic your old settings in the new environment; but KDE (and GNOME) offers a lot more (and ever expanding) configuration choices.

5.2 GNOME customization

We will customize a GNOME desktop that will be targeted for use by a very light office user. The applications we provide are:

- x3270
- Mozilla

For an introduction to how Gonme user profile (personalization) data is contained see “Desktop personalization: GNOME containment model” on page 224.

Folders are created as a directory structure and are not centralized in one place since Linux is a true multi-user operating system. In GNOME, the desktop directory is hidden and renamed as $HOME/.gnome2.

We manipulate the desktop structure in the following subdirectory:

```
$HOME/.gnome2/Business Applications
```

Icons are handled in the same way as they are in KDE. This means that they are ASCII text files. You can save them in the $HOME/.gnome2 directory to populate the desktop.

5.2.1 GNOME configuration system

The accessibility configuration of the GNOME desktop differs heavily from KDE, which uses simple and easy-to-edit ASCII files. The GNOME desktop uses a database, which is represented by Extensible Markup Language (XML) files to build and customize all of its settings. These files are numerous and can be found under the /etc/opt/gnome/gconf/gconf.xml.defaults/desktop/gnome/

directory for the Novell Linux Desktop (for the Red Hat Workstation 3 you have to replace the base path /etc/opt/gnome by /etc).

This directory contains subdirectories for each portion of the GNOME desktop environment, as shown here:

```
/etc/opt/gnome/gconf/gconf.xml.defaults/desktop/gnome/accessibility
/etc/opt/gnome/gconf/gconf.xml.defaults/desktop/gnome/applications
/etc/opt/gnome/gconf/gconf.xml.defaults/desktop/gnome/background
/etc/opt/gnome/gconf/gconf.xml.defaults/desktop/gnome/file_views
/etc/opt/gnome/gconf/gconf.xml.defaults/desktop/gnome/font_rendering
/etc/opt/gnome/gconf/gconf.xml.defaults/desktop/gnome/interface
/etc/opt/gnome/gconf/gconf.xml.defaults/desktop/gnome/lockdown
/etc/opt/gnome/gconf/gconf.xml.defaults/desktop/gnome/peripherals
/etc/opt/gnome/gconf/gconf.xml.defaults/desktop/gnome/sound
/etc/opt/gnome/gconf/gconf.xml.defaults/desktop/gnome/thumbnailers
/etc/opt/gnome/gconf/gconf.xml.defaults/desktop/gnome/typing_break
/etc/opt/gnome/gconf/gconf.xml.defaults/desktop/gnome/url-handlers
```

Within each of these subdirectories is a file called %gconf.xml (Example 5-10).

Example 5-10 %gconf.xml file

```
/etc/opt/gnome/gconf/gconf.xml.defaults/desktop/gnome/lockdown

<?xml version="1.0"?>
<gconf>
   <entry name="disable_print_setup" mtime="1089888947"
          schema="/schemas/desktop/gnome/lockdown/disable_print_setup"/>
   <entry name="disable_printing" mtime="1089888947"
          schema="/schemas/desktop/gnome/lockdown/disable_printing"/>
   <entry name="disable_save_to_disk" mtime="1089888947"
          schema="/schemas/desktop/gnome/lockdown/disable_save_to_disk"/>
   <entry name="disable_command_line" mtime="1089888947"
          schema="/schemas/desktop/gnome/lockdown/disable_command_line"/>
</gconf>
```

5.2.2 gconftool-2

The GNOME desktop ships with a tool called gconftool-2. You can use it from the command line to save edits and to commit changes manually to the XML files. You can set mandatory values that users are unable to change and that are used for new users created on the system.

Note: Each of following examples of **`gconftool-2`** is a command. These examples work with GNOME 2.6.

The following resets the default background and makes it a mandatory change:

```
gconftool-2 --direct --config-source
xml:readwrite:/etc/gconf/gconf.xml.mandatory --type string --set
/desktop/gnome/background/picture_filename
/usr/share/backgrounds/images/corporate_logo.jpg
```

The second example is to set the number of workspaces available to the user. In this instance, we set two workspaces, since this user only has access to and uses two applications. With one in each workspace, each application can be run in its own work space.

```
gconftool-2 --direct --config-source
xml:readwrite:/etc/gconf/gconf.xml.mandatory --type int --set
/apps/metacity/general/num_workspaces 2
```

The number of configuration options that are available to the gconftool-2 tool are vast and too new numerous to list in this book. To display the current options that are set for a desktop, enter the following command:

```
gconftool-2 -R /desktop
```

Although we do not cover all options, we show how we customize our desktop and explain each of the commands that are used. To list all the settings used on the default panel, enter the following command:

```
gconftool-2 -R /apps/panel
```

The following output shows the global variables for the GNOME panel settings. Instead of listing all settings as we did in the previous command, enter the gconftool-2 to look at specific areas, for example:

```
gconftool-2 -R /apps/panel/global
```

This displays the following global default values:

```
screenshot_key = Print
panel_hide_delay = 500
enable_animations = true
drawer_autoclose = true
keep_menus_in_memory = true
window_screenshot_key = <Alt>Print
panel_minimized_size = 3
tooltips_enabled = true
menu_key = <Alt>F1
confirm_panel_remove = true
panel_animation_speed = panel-speed-medium
highlight_launchers_on_mouseover = true
run_key = <Alt>F2
enable_key_bindings = true
panel_show_delay = 300
```

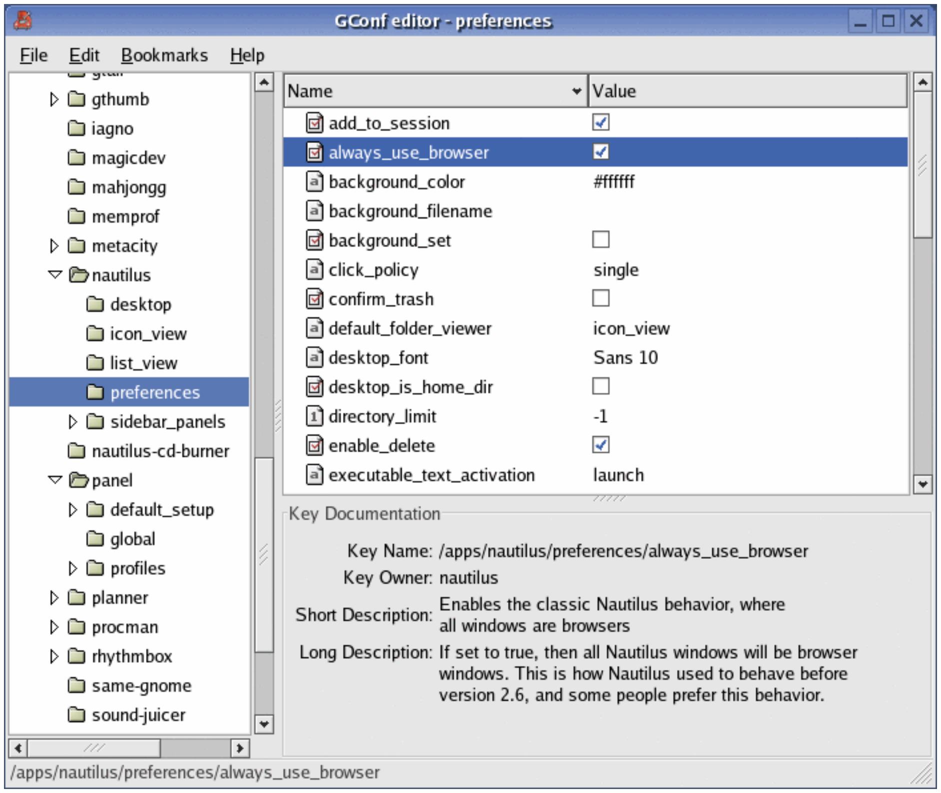
The previous values are edited by using gconftool-2. The following code disables the run_key:

```
gconftool-2 --direct --config-source
xml:readwrite:/etc/gconf/gconf.xml.mandatory --type bool --set
/apps/panel/global/run_key false
```

If the following command is issued to display all the global values, the change was made to the run key. It is now disabled for all users and they are unable to change it. This is because the gconf.xml.mandatory directory was used. If we used the gconf.xml.default directory, it would set a system default for all new users, but would be configurable by the users.

```
gconftool-2 -R /apps/panel/global
   screenshot_key = Print
   panel_hide_delay = 500
   enable_animations = true
   drawer_autoclose = true
   keep_menus_in_memory = true
   window_screenshot_key = <Alt>Print
   panel_minimized_size = 3
   tooltips_enabled = true
   menu_key = <Alt>F1
   confirm_panel_remove = true
   panel_animation_speed = panel-speed-medium
   run_key = false
   highlight_launchers_on_mouseover = true
   enable_key_bindings = true
   panel_show_delay = 300
```

While gconftool-2 is very useful for scripting there is of course also a graphical tool called gconf-editor available. You can see it in action in Figure 5-10 on page 108 while setting the navigational mode of Nautilus back to its old behavior (that is, disabling the spacial mode introduced in GNOME 2.6).



gthumb
iagno
magicdev
mahjongg
memprof
metacity
nautilus
desktop
icon_view
list_view
preferences
sidebar_panels
nautilus-cd-burner
panel
default_setup
global
profiles
planner
procman
rhythmbox
same-gnome
sound-juicer

Name	Value
add_to_session	✓
always_use_browser	✓
background_color	#ffffff
background_filename	
background_set	☐
click_policy	single
confirm_trash	☐
default_folder_viewer	icon_view
desktop_font	Sans 10
desktop_is_home_dir	☐
directory_limit	-1
enable_delete	✓
executable_text_activation	launch

Key Documentation

Key Name:	/apps/nautilus/preferences/always_use_browser
Key Owner:	nautilus
Short Description:	Enables the classic Nautilus behavior, where all windows are browsers
Long Description:	If set to true, then all Nautilus windows will be browser windows. This is how Nautilus used to behave before version 2.6, and some people prefer this behavior.

/apps/nautilus/preferences/always_use_browser

Figure 5-10 Editing gconf entries with the graphical gconf-editor

5.2.3 Restricting the desktop

You can use the same method for the GNOME desktop icons. The only difference is the GNOME desktop directory structure. When a user is created, her desktop structure is not created until she first logs on. Again by manually creating this desktop structure, the default is not created.

```
mkdir /home/fiona/.gnome2
mkdir /home/fiona/.gnome2/Business Applications
```

The current directory structure for user Fiona is:

- /home/fiona/.gnome2
- /home/fiona/.gnome2/Business Applicationsx3270
- /home/fiona/.gnome2/Business Applications/mozilla

We want Fiona to have access to these applications so she can run them. However, we do not want her to have the ability to make permanent changes to the desktop or to alter the applications that we set up. To do this we could change ownership and permission settings as follows:

```
chown -R <adminID> /home/<username>/.gnome2
chmod -R 755 /home/<username>/.gnome2
```

This allows the user to execute the icons and open the folders, but not to make any changes to them. But this is not foolproof, since Fiona can still rename the /home/fiona/.gnome2 directory and build another one on her own.

5.2.4 Visual configuration

This section explains how to configure the GNOME desktop environment from the GUI using the GNOME control center.

> **Note:** When using the GUI to personalize the GNOME environment, a subset of the Gconf database is created within the user’s home directory.

To access the GNOME preferences dialog, open the Nautilus URL:

```
preferences://
```

Or select the **Preferences** menu entry. Using `gnome-control-center` from the command line does exactly the same. You can now select all of the preferences from within the displayed overview (see Figure 5-11 on page 110 for a screenshot of Fedora Core II with Crystal Icons and Plastik theme clone for GNOME).

The GNOME control center presents a myriad of different configuration options, for example:

- Acessibility
- Desktop background
- Mouse
- Keyboard and keyboard shortcuts
- Menus and toolbars
- Fonts
- Sound
- Themes
- Window behavior
- Screen resolution
- Screensaver
- Session management
- Login photo for gdm
- Network proxy

And a lot more.

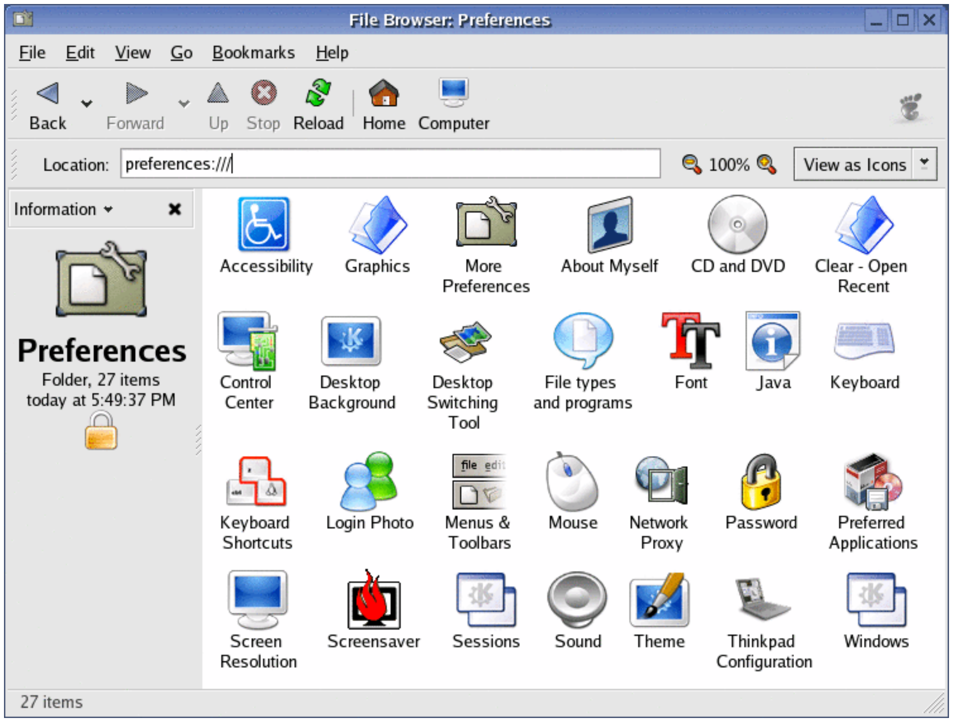
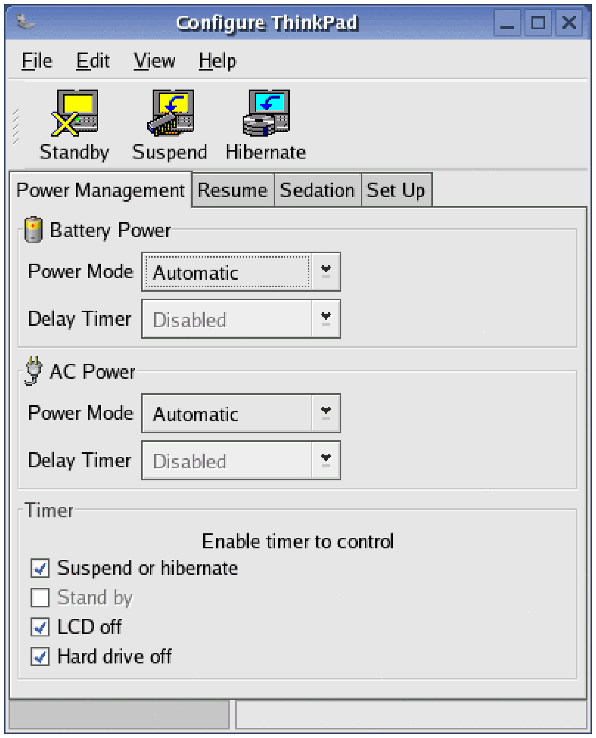


Figure 5-11 GNOME preferences

Other programs can stick into that extensible framework too. In Figure 5-12 on page 111 you see a Thinkpad configuration entry that has been added by the configure-thinkpad package you can download at the following URL:

http://tpctl.sourceforge.net/


Configure ThinkPad

Battery Power

Power Mode: Automatic

Delay Timer: Disabled

AC Power

Power Mode: Automatic

Delay Timer: Disabled

Timer

Enable timer to control

- [✓] Suspend or hibernate
- [ ] Stand by
- [✓] LCD off
- [✓] Hard drive off

Figure 5-12 GNOME Thinkpad configuration tool

5.3 The self-managing Linux client

The goal of each Linux client migration should be to build a more secure, easier to manage, and more productive user environment. In this section we concentrate on the managability aspect and present some general ideas that have been realized in several customer projects and are available as open source, commercial procucts or will even be integrated in future versions of the standard Linux distributions.

There are many ways to set up a Linux client environment depending on where certain resources lie and how they are used and managed. The resources we are speaking about are:

- ► Operating system
- ► Configuration data
- ► Application data
- ► User data
- ► Hard disks

- Audio devices
- Printers
- USB devices (memory sticks, WLAN cards, digital cameras, etc.)
- Bluetooth devices (cell phones and many others)
- PCMCIA cards (GPRS, ISDN, WLAN, and other cards)
- Smartcard readers
- Firewire devices (hard disks, video cameras, etc.)

Except for the hardware connected to the clients locally (like audio devices, printers, USB sticks, bluetotth devices, GPRS cards, smartcard readers, and firewire devices), all other resources can be either on the client or server side, depending on what kind of storage or memory devices are built into the system (like hard drives, RAM, Flash RAM, etc.)

A classic diskless X terminal can boot its operating system (hopefully Linux) from Flash RAM or over the network (that is, with PXE boot), and use all other resources from the server, which means all applications except the ones available in the Flash ROM (that is, Mozilla, Java JVM, etc.) are started on the server and use the memory there. A diskless client can also mount certain or all parts of the file system (for example, the home directory, application directory) from a server and run the application using the local memory (some call that a Linux Net PC).

Since modern thin client hardware is quite powerful, the second approach seems appropriate today and you do not waste local computing ressources. If you can migrate your Linux applications on the fly from one machine to the other (with virtual machines (JVM, Mono[2]), openMosix[3], or other single system image technologies) you could start your applications on some server, migrate them to you local client, and park them back to maybe another server when you disconnect. This allows you to continue your work later at exactly the same place where you left. Saving a complete application (state) to hard disk and restarting it would be needed here, so applications not being used would not consume any resources, but this is not possible right now with Linux.

You could use VMWare images loaded on demand on the server instead and use them remotely with Nomachine, VNC, Citrix, or Tarantella so you would not park single applications but the whole operating system. There are some innovative companies that are beginning to implement these kinds of scenarios today.

Let us get back to Linux clients having local hard disk drives. For this kind of situation you can install Linux and use your preferred systems management tools (from Novell, Red Hat, IBM, or open source) to manage them. The installation

[2] `http://www.mono-project.com`
[3] `http://www.openmosix.org`

can be fully automated with AutoYast, Kickstart, Debian FAI[4] (Fully Automated Installation), etc.

Automated installation can also be achieved by designing a PXE boot installation method (maybe using the Linux Terminal Server project as a starting point) in combination with specialized *initrd* images. This innovative method has been used within the OpenLDAP controlled SuSE Smart Client framework at a large insurance company in Germany. Their architecture relies on synchronization (using rsync) of a master server with 230 branch servers that are being used as local boot and installation servers for their branch clients.

A similar method using a highly specialized initrd adaptation has been implemented at anohter large insurance company in Germany. They use about 4,500 laptop computers and another 2,000 stationary clients. Those clients are booted over the network (PXELinux[5]) and configured while booting the systems according to:

- What hardware is detected (using Kudzu[6] autodetection), but overriding; for example, screen resolution, since 1024x768 is a must for certain applications, even if the hardware could do better
- How they are connected to the network (Olympic Tokenring, ISDN, GPRS Option Card, Bluetooth Mobile phone)
- System default login rights: Who you are and to which user group you belong to (meaning you have access to different applications, etc.)

All configuration data (/etc, configuration files in $HOME (for example, for Netscape, etc.), desktop profile data, etc.) is assembled during the boot process and written to memory, so that even if you get access to a hard drive on a system that was booted in this way you cannot tell which network this machine was connected to or who has been working with this Linux client. Stationary users have access to all applications via NFS, while mobile users (having a preinstalled Linux system on their Thinkpads) get their applications rsynced during the boot process, so everybody has an up-to-date system at any time. Certain parts of the system are rsynced only when the network connectivity is big enough, for example, for a X11 server update. After booting you have 30 secods to log into the system with a Omnikey USB smart card reader and your password; otherwise the system is rebooted, or in certain cases even repartitioned and destroyed. This type of client management architecture could be decribed as supporting self-managing clients or stateless Linux. Red Hat is prototyping

[4] `http://www.informatik.uni-koeln.de/fai`
[5] `http://syslinux.zytor.com/pxe.php`
[6] Red Hat Linux hardware probling library: `http://rhlinux.redhat.com/kudzu`

methods for supporting a stateless Linux client. They define some of the ideal properties of a stateless Linux client, as follows[7]:

- “If you throw a computer out of the window, you should be able to recreate its software, configuration, and user data bit-for-bit identically on a new piece of hardware.”
- “In any managed deployment from school workstation lab to enterprise server room, single computers should never be modified. Instead, all computers that need the modification should be modified in a single step.”

5.3.1 Summary

We have shown in this section that it is an oversimplification to simply lump all Linux clients within a traditional thin-to-fat spectrum. There is a multitude of tools and methods (like very sophisticated hardware detection) for Linux that support intermediate and/or managed client approaches. And they are diverse and growing.

[7] From “*Introduction to Stateless Linux, Proposal for the Fedora Project*”, `http://people.redhat.com/~hp/stateless/StatelessLinux.pdf`

6

Client migration scenario

In this chapter we demonstrate an example client migration. We describe the client as it is used before the migration. The important applications on the client will be identified. We describe a migration plan for this client based on the information in Chapter 3, “Organizational and human factors planning” on page 27, and Chapter 4, “Technical planning” on page 37. The result of the actual migration is shown in the last part of this chapter.

The sections in this chapter are:

- 6.1, “Example client migration” on page 116

  Details on the pre-migration client environment are discussed.
- 6.2, “Migration plan details” on page 117

  Migration plan details and strategy are presented as a result of assessing the existing client environment.
- 6.3, “Performing the migration” on page 119

  The actual migration sequence of steps, with some key configuration details included.

6.1 Example client migration

We perform a migration of a single type of client to show how the methods described in the planning part of this book can be put into practice. The example client migration will be a simple one. However, the infrastructure integration is almost completely as described in 4.2, “Integrating with existing network services” on page 45.

6.1.1 Assess the client usage pattern

The client we use for the example migration is an ITSO desktop used to write this book. The migration client has the following properties:

- It is a registered workstation in an NT4 domain.
- Intranet and Internet access modes using MS Internet Explorer.
- Adobe FrameMaker is the primary content authoring application on the client.
- Paintshop Pro is used to create and work on screenshots.
- Use network printers to print.
- Outlook to MS Exchange for e-mail.

The client is running Windows 2000 with Service Pack 4 installed. We will migrate to Linux, using the Red Hat Desktop distribution.

6.1.2 Identify important applications and infrastructure integration points

Since the user role for this workstation is primarily for writing books using Adobe FrameMaker, that most important application and the most important infrastructure components are printing and access to network file shares.

The first thing to note about the Adobe FrameMaker application is that there is no Linux alternative. There is no Linux native version, and moving to another application on Linux is not acceptable from a “business” point of view. In other words, FrameMaker is a unmigratable application that has to be handled like described in 4.7.2, “Terminal Server, Citrix Metaframe, or NoMachine solutions” on page 82.

Printing using the network printers and accessing shares in the NT4 domain means that the methods from 4.2, “Integrating with existing network services” on page 45, have to be followed.

6.2 Migration plan details

In this section we want to discuss in which steps the migration will take place, which possible problems we could come across, and how we will establish the functional continuity.

After assessing a current ITSO desktop and its environment,as we have done in the section before, we have to build a plan that will include the necessary work to make the client functional after removing the Windows operating system.

6.2.1 Client approach

The ITSO resident workstation would fit into either the Basic Office or Advanced Office workstation logical segment, as defined in 4.4.3, “Functional segmentation - Fixed function to general office” on page 65. And since the users use these workstations primarily for authoring technical papers using FrameMaker, as well as doing research using a Web browser, they map most closely to the Fat client type, as discussed in 4.4.2, “Logical segmentation - Thin, slim, or fat” on page 64

6.2.2 Graphical environment

Another key decision that has to be made is what type of graphical windowing environment will be used on the Linux client. Alternatives are discussed in 4.3.2, “Linux desktop environments” on page 53.

In this special case, we are not going to discuss advantages and disadvantages of one or the other environment; both of them would provide enough functionality.

But as Gnome is the default session manager for Red Hat distributions, we decided to stay with this choice for our sample scenario. The version that is included in Red Hat Desktop 3 at the time of this writing was Gnome 2.4.

6.2.3 Hardware

Another topic to be considered before migrating is the hardware that is in use.

In this case, no peripherals are connected to the client, and all other resources (like printers, for example) are accessed via the network.

The migration PC platform for this example is listed in Table 6-1 on page 118.

Table 6-1 Hardware and descriptions

Hardware	**Description**
PC model	IBM Netvista Type 6759
Processor	Intel Pentium III 866 MHz
Chipset	i810 integrated
Graphics card	i810
Sound card	i810
Optical drive	CD-ROM 24x IDE
Floppy	3,5” floppy drive
Harddisk	IDE Harddisk 40 GB

As the graphics and sound controller are integrated with the chipset, we are expecting some additional work to identify and install appropriate Linux drivers for these devices. A check on the Intel Web site provided us with drivers for these two components. All other parts are known to be working, as they are standard components.

6.2.4 Application continuity

In Table 6-2, we show our application mapping to support this sample migration scenario.

Table 6-2 Application mapping to support sample migration scenario

Application on Windows	**Application on Linux**
Internet Explorer 6.0 SP1	Mozilla Firefox 0.9.1
Outlook 2000	Ximian Evolution 1.4 with Ximian Connector for Exchange 2000
Windows-Explorer	Nautilus
WinZip	FileRoller
ICQ	Gaim
Microsoft Word 2002	OpenOffice.org 1.1Writer
Microsoft Excel 2002	OpenOffice.org 1.1 Spreadsheet
Microsoft Powerpoint 2002	OpenOffice.org 1.1 Impress
Adobe Reader 6.0	Acrobat Reador 5.08 for Linux

Application on Windows	Application on Linux
Adobe FrameMaker	Adobe FrameMaker (on WTS)
Paintshop Pro 8 (Jasc Software)	GIMP 1.2

6.2.5 Windows networking

Windows networks can become quite complex, especially when using roaming profiles, Active Directory structures, or when combining several trusted domains. In our environment none of these features are in use. From a network services integration point of view this makes the switch to Linux relatively easy.

Our migration clients will need to mount existing Windows file server shares, and be able to use existing printers also shared by the Windows domain servers. Therefore we need the domain name and a valid user for authentication to the existing ITSO domain.

The distinct names in this domain are:

- Domain name = ITSOAUSNT
- Primary domain controller = ITSONT00
- Backup domain controllers = ITSONT01,ITSONT02,ITSONT03
- User name(s) for residents = ausresxx

We will be using methods described in detail in 4.2, “Integrating with existing network services” on page 45, and in Chapter 7, “Integration how-tos” on page 137, to complete network services integration of the Linux client.

6.3 Performing the migration

In this section we discuss performing the migration.

6.3.1 Basic installation tasks

The first thing to do is complete a base installation from the distribution CDs. In the case of Red Hat Desktop, the CD set contains four CDs. Special packages for the desktop are available from the Red Hat Network. These contain third-party applications like Acrobat Reader or Real Player and a set of high-quality truetype fonts from AGFA.

During installation we accepted the default selections for application packages as defined by the installation program. On our target desktop hardware the Red Hat *Anaconda* installation program completed the base installation without any problems. Also, the Red Hat hardware probing library called *kudzu* worked as expected. All hardware components were detected correctly. The correct modules for the integrated video and audio chipset were loaded successfully. The network card driver was loaded, and the network was activated successfully at the end of the installation. The system successfully received a DHCP lease from the DHCP server in the ITSO network. Name resolution worked as well, without any extra configuration steps.

Setting up remote administrative access using VNC

Once the installation completed, we chose to immediately modify our client setup to provide remote terminal access to it from other workstations on the network using VNC. This way we could complete any additional administration tasks to complete the install remotely.

To begin this process we first check that the ssh daemon is running and that connections via VNC will get a complete desktop. One way to enable this for the root user is to modify the file xstartup in $HOME/.vnc , as shown in Example 6-1.

Example 6-1 Enabling VNC connections

```
#!/bin/sh

# Uncomment the follwing two lines for normal desktop:
unset SESSION_MANAGER
exec /etc/X11/xinit/xinitrc
.....
```

Also, you have to enable Xdmcp in /etc/X11/gdm/gdm.conf, and set up a session in /etc/sysconfig/vncservers.

After starting by executing the command **’service vncserver start’**, you can connect with any vnc client and work on the complete graphical interface.

Important: This can be dangerous, as it enables connecting to the local X server from any place. If it is needed, editing /etc/hosts.allow or /etc/hosts.deny can allow you to limit who can access the workstation.

Preparing to integrate with Windows network services

The next step in our migration is to connect the client to the existing networking services. In our existing environment we will be integrating with Windows NT/2000 servers, so we will follow the instructions in Chapter 7, “Integration how-tos” on page 137.

Our first goal is the ability to use the services of the existing Windows domain, primarily user authentication. In order to be able to log onto the Linux client using Windows user names and passwords, it is necessary to set up winbind.

This is done by editing /etc/samba/smb.conf for a proper setup in the network environment:

Example 6-2 Changes to smb.conf for Linux client domain authentication

```
[global]
   workgroup = ITSOAUSNT
   security = domain
   password server = ITSONT00,ITSONT01,ITSONT02,ITSONT03
...
...
   winbind separator = +
   idmap uid = 10000-20000
   idmap gid = 10000-20000
   winbind enum users = yes
   winbind enum groups = yes
   template homedir = /home/%D+%U
   template shell = /bin/bash
```

6.3.2 Integrating existing network services

After the winbind daemon has started, the command **`wbinfo -u`** should deliver the current user list of the Windows domain. But before this works, it is necessary to join the domain with the Linux client. As long there is no machine account in the domain for the client, it will not be possible to fetch the list of users via winbind. Therefore we have to join the Domain with the following command:

```
net join -a ITSOAUSNT -U administrator
```

> **Important:** Do not forget to join the domain with the command **`net join -a <domainname>`**. This creates a valid machine account on the domain controller and gives the Linux client a valid SID for authentication.

As we are now able to read the user database on Windows, only two more steps are needed to log in on the Linux client with an existing account managed by the existing Windows domain.

Next, nsswitch.conf is changed according to 7.2, “How to use winbind to make domain users known locally” on page 140 ,which delivers the mapping of Windows users and Groups to the Linux uids and gids.

Example 6-3 Changes to /etc/nsswitch.conf

```
passwd: files winbind
group: files winbind
```

By performing the command **getent passwd** for testing purposes, all users should be listed with their correct uids and gids.

Now it is time to enable the login process to use the mapped users. This is done by using two different PAM modules, which is described in detail in 7.3, “How to authenticate users against the domain using PAM” on page 143.

As we are using Red Hat Desktop for our pilot migration, all information in 7.3.1, “Winbind and PAM on Red Hat Desktop” on page 144, applies.

At first we have to use the the pam_winbind and the pam_smb_auth module, which allow system authentication with a user of the domain ITSOAUSNT. In order to enable this module, we have to edit the /etc/pam.d/system_auth file, as in Example 6-4.

Example 6-4 Edit the /etc/pam.d/system_auth file

```
#%PAM-1.0
# This file is auto-generated.
# User changes will be destroyed the next time authconfig is run.
auth        required      /lib/security/$ISA/pam_env.so
auth        sufficient    /lib/security/$ISA/pam_unix.so likeauth nullok
auth        sufficient    /lib/security/$ISA/pam_winbind.so use_first_pass
auth        sufficient    /lib/security/$ISA/pam_smb_auth.so use_first_pass nolocal
auth        required      /lib/security/$ISA/pam_deny.so

account     required      /lib/security/$ISA/pam_unix.so
account     sufficient    /lib/security/$ISA/pam_winbind.so
........................
```

It is now possible to authenticate both with a local or a domain user account. A first test on the console shows that it is working, but at this point logging in on the Gnome welcome screen will generate an error.

The reason for this problem is that at the time of the first login, there is still no home directory existing for the user ID. So Gnome has a file system path in which to create the personalization and settings files during first login. The solution arrives with another pam module that is available, pam_mkhomedir. By using this module, a home directory will be created at logon time if it does not exist yet.

It is necessary to add the following line in the system-auth file.

Example 6-5 Addition of pam_mkhomedir entry in /etc/pam.d/system-auth

```
#%PAM-1.0
# This file is auto-generated.
# User changes will be destroyed the next time authconfig is run.
```

```
auth        required       /lib/security/$ISA/pam_env.so
auth        sufficient     /lib/security/$ISA/pam_unix.so likeauth nullok
auth        sufficient     /lib/security/$ISA/pam_winbind.so use_first_pass
auth        sufficient     /lib/security/$ISA/pam_smb_auth.so use_first_pass nolocal
auth        required       /lib/security/$ISA/pam_deny.so

account     required       /lib/security/$ISA/pam_unix.so
account     sufficient     /lib/security/$ISA/pam_winbind.so
........................
session     optional       /lib/security/$ISA/pam_mkhomedir skel=/etc/skel umask=0022
```

Now we can enter the user name and password on the Gnome welcome screen, a home directory is created, and so the complete Gnome desktop is available.

We log on using the ID:

```
ITSOAUSNT+AUSRES06
```

The login user name has the format:

```
<domainname:winbind separator:username>.
```

So the Gnome desktop shows an icon with `ITSOAUSNT+AUSRES06's home`. Unfortunately, it is not possible to separate the domain and user name on a standard Gnome installation; the winbind separator will always be visible.

Mounting Windows file shares

After enabling domain user logon, it should also be possible to use the file shares on the Windows servers. Our example client needs to map three different file shares using the same credentials. We will have to store the user credentials in order to use them for interactively use of the **mount** command. The **smbmount** command allows the use of a credentials file. For security reasons we placed this in root's home directory and set the file permission in a way that only root has access to the file. The file /root/.credentials looks like Example 6-6.

Example 6-6 Credentials file for mounting shares

```
username = ausres06
password = password
domain = itsoausnt
```

The most logical place for such a mount execution is in /etc/fstab, where it is possible to define the file system to smbfs ,and the file shares would be mounted at the same time as the local file systems.

Example 6-7 Mounting file shares in /etc/fstab

```
/dev/hda1           /                    reiserfs   acl,user_xattr        1 1
```

```
/dev/hda5           /scr                auto      noauto,user         0 0
/dev/hda2           swap                swap      pri=42              0 0
......
/dev/fd0            /media/floppy       subfs     fs=floppyfss,procuid,nodev,nosuid,sync 0 0
//itsont05/data     /shares/data_g      smbfs     credentials=/root/.credentials 0 0
```

This works fine unless the Windows file share uses non-standard characters. In contrast to UNIX-like operating systems, Windows allows you to use space characters or even German umlauts in the names of file shares. You may run into problems in the case where the Windows file shares contain special characters. For example, the /etc/fstab file uses the space character as a delimiter, so if a share contains a space, it will take the following string as a mount point. So, while mounting the “project data” file share, the mount command complains about several errors in the fstab file, as it tries to use data as a mount point.

Tip: When mounting Windows file shares in a Linux file system, it is very reasonable to check the names of the shares for special characters. Renaming them UNIX-compliant can save a lot of time and annoyance.

Thus we had to use another way of mounting the file share, and we decided to extend the /etc/rc.local file with the corresponding **smbmount** commands.

Modifications to rc.local are shown in Example 6-8.

Example 6-8 Extended rc.local file

```
#!/bin/sh
#
# This script will be executed *after* all the other init scripts.
# You can put your own initialization stuff in here if you don't
# want to do the full Sys V style init stuff.

touch /var/lock/subsys/local

#mounting windows file shares
smbmount //itsont05/data /shares/data_g -o credentials=/root/.credentials,
gid=ITSOAUSNT+Domain\ Users,dmask=0775
smbmount //itsont02/"project data" /shares/projectdata_e -o credentials=/root/.credentials,
gid=ITSOAUSNT+Domain\ Users,dmask=0775
```

If using file shares with special characters here, it is necessary to put them in quotation marks, as shown in the last line of the example above. The gid and dmask options of the **smbmount** command are used to ensure that the permissions are set in the right way for the Linux directories in which the shares are mounted. If they are left out, users would not have write permissions because of the default umask in the root directory. The permissions that are set on the Windows file server are inspected after one user has full access to the mount point. It is evident that the smbmount does not override permissions on the server side.

Important: Be careful with permissions on the directories that will be used as mountpoints; make sure that they fit to the permissions given on the Windows server. Using a gid in the command prevents file access from users who are not member of the group.

Printers

Printer services are used via the smb protocol as well, but as Linux with CUPS has a nice printing system, we will use this to integrate the network printers on the client.

Tip: If drivers for your printer are not included in the standard installation of CUPS, it may be possible to find the right ppd (Postscript Printer Description) driver file for your printer on the Internet. The correct ppd driver file for your printer should be added to the directory /usr/share/cups/model. After restarting CUPS it is possible to use this driver while adding a printer. Methods for finding a ppd driver file are:

- Search your printer manufacturer's Web site for Linux support or ppd driver file downloads.
- Purchase printing software supporting Linux:

  `http://www.easysw.com/printpro`
- Try a Windows PPD file provided by Adobe:

  `http://www.adobe.com/products/printerdrivers/winppd.html`
- See the CUPS home page for more links:

  `http://www.cups.org`

Following the steps in "Create printer using CUPS Web interface" on page 153 it is very easy to add the printers when the right credentials are used.

6.3.3 Application configuration and installation

As pointed out earlier in this chapter, the main application on the resident PC is Adobe FrameMaker. Interestingly enough, Adobe offered a Linux version some time ago, but with Version 6.0 this product was cancelled. As the client needs FrameMaker Version 7.0 with some special plug-ins, it is not possible to install this application on Linux. Thus we have to find a way to handle this "unmigratable" application (see 4.7, "Un-migrateable applications" on page 81, for detailed discussion on strategies).

The solution we chose to support FrameMaker users on the migrated Linux client was to use a remote terminal services method using Windows Terminal Server (WTS). WTS makes it possible for a Linux client to connect using the RDP protocol to a windows application session running on the WTS host. Other terminal services solutions for this type of application hosting requirement exist, but for this small scenario the use of WTS was deemed to be satisfactory. Our next step was to install the application on the Terminal Server.

As it is not guaranteed that every application can be run on a Terminal Server, we ran some tests by starting FrameMaker several times in different sessions. We learned that our version of FrameMaker was not optimized for this scenario, since each FrameMaker application instance running on the WTS host consumed an equal amount of system memory.

Important: If an application is not optimized for terminal servers, it will take the same amount of memory for each instance. So it has to be considered that the server has enough main memory for handling the expected Terminal Server session load.

The connection to the Terminal Server is made by tsclient, a GTK gui for the rdesktop application. Using a launcher file on the desktop, it is possible to use a $HOME/*.tc configuration file while executing tsclient. Our command syntax for starting the tsclient session is as follows:

```
tsclient -x framemaker.tc
```

The file framemaker.tc contains all information for the connection, such as IP address, user name, etc.

An example screenshot of FrameMaker running inside a Terminal Server client on Linux is provided in Figure 6-1 on page 130.

Image manipulation

On the Windows-based client, Paint Shop Pro is used for manipulating images and taking screenshots. As Paint Shop Pro is not available for Linux, we decided to use the GIMP as a functionally equivalent Linux application instead. This open source application is very mature and provides a good functional equivalent to Paint Shop for the specific uses that the Windows client in this scenario needs. Further information can be found at:

```
http://www.gimp.org
```

Gimp is included as part of the standard installation of Red Hat Desktop, so you just have to start it. A screen shot showing the gimp running on the migrated desktop is shown on Figure 6-2 on page 131.

Browser

For our migration scenario, we decided to use the Mozilla Firefox Browser. Firefox is the browser-only project from Mozilla.org. As we do not need mail capabilities (these are provided by Ximian Evolution) using the stand-alone browser offering from Mozilla seemed reasonable.

We recommend that you check important Web sites or especially Web clients for compatibility with the browser before the migration. Some Web sites use ActiveX-controls or special JavaScript functions, which are only supported by Internet Explorer. As this is not the case for our migration client, it is easy to make the switch and use Mozilla Firefox instead of Internet Explorer.

E-mail client

In this section we talk about an e-mail client.

Novell Evolution with Novell Connector for Exchange 2000

To make our client migration scenario more realistic, we added the requirement that existing clients use Microsoft Outlook and Exchange 2000 messaging. So our migration task includes migrating from the Windows-based Outlook client to a Linux-based messaging client that provides similar functionality, and integrates with the existing Exchange 2000 messaging system.

Our migration solution consists of using the Novell Evolution e-mail client in conjunction with the Novell Connector for Microsoft Exchange Server 2000/2003. The Evolution e-mail client has an architecture that supports the building of special connectors for different groupware applications. Thus a connector for Microsoft Exchange 2000 and 2003 was developed, and was released under the terms of the GPL for the first time in May 2004. A key factor in enabling the development of the Exchange connectors was Microsoft's inclusion of a Web interface to Exchange called Outlook Web Access (OWA). OWA uses uses webDAV for communication to the exchange server message store. So it became possible to develop Exchange connectors by implementing a webDAV interface, instead of having to reverse engineer the proprietary MAPI protocol of Exchange.

Restriction: Only the Exchange versions 2000 and 2003 support webDAV connections. As of the writing of this book, a connector supporting native Linux client access to full Exchange 5.5 server functionality was still not available.

We installed the Red Hat RPMs that were delivered with Red Hat Desktop and found it overall to be quite easy to set up a connection to the Exchange Server. You still have to keep in mind some requirements to get this working.

Attention: Before using the Exchange connector, some prerequisites should be checked. It is necessary to start the Exchange virtual server in the http section of the Exchange system manager. You also need to enable HTTP protocol for the users. This is done in the user settings via the Active Directory Users console.

More information on the Novell Exchange Connector can be found at:

```
http://www.novell.com/products/connector/
```

Customization

You have a very high degree of flexiblity in modifying the look and feel of a Linux client. For our client migration scenario, we chose to modify just a few things as a way of demonstrating some possibilities.

The first thing we did was to adjust the default background color and insert the ITSO Redbooks image on the background. This can easily be done by using the background settings application for the Gnome Desktop. Another possibility, especially to set this as a default value, is the use of the gconftool-2 with a command syntax as in Example 6-9.

Example 6-9 Using gconftool to replace the screen background image

```
#  gconftool-2 --direct --config-source \
     xml:readwrite:/etc/gconf/gconf.xml.mandatory --type string --set \
     /desktop/gnome/background/picture_filename itso-logo.png
```

For further information on how to change values in the Gnome configuration or make changes that can be enforced for groups of users, see 5.2, “GNOME customization” on page 104. Specific details about using gconftool-2 can be found in 5.2.2, “gconftool-2” on page 105.

Gnome also offers a graphical editor for the settings, although the welcome screen of this tool says that using the editor is not the preferred way, so using it is at your own risk.

Another nice effect can be created by inserting another logo for the menu button. By default, Red Hat has replaced the Gnome logo with its own custom image. We replaced this by editing the following file:

```
/usr/share/pixmaps/redhat-main-menu.png
```

As the name of this file is set in the gconf.xml file, we decided to take the easy way and just rename our ITSO logo file to redhat-main-menu.png. Of course, it has to be 48x48 pixels large, and for a better look, the logo had to be made transparent in GIMP.

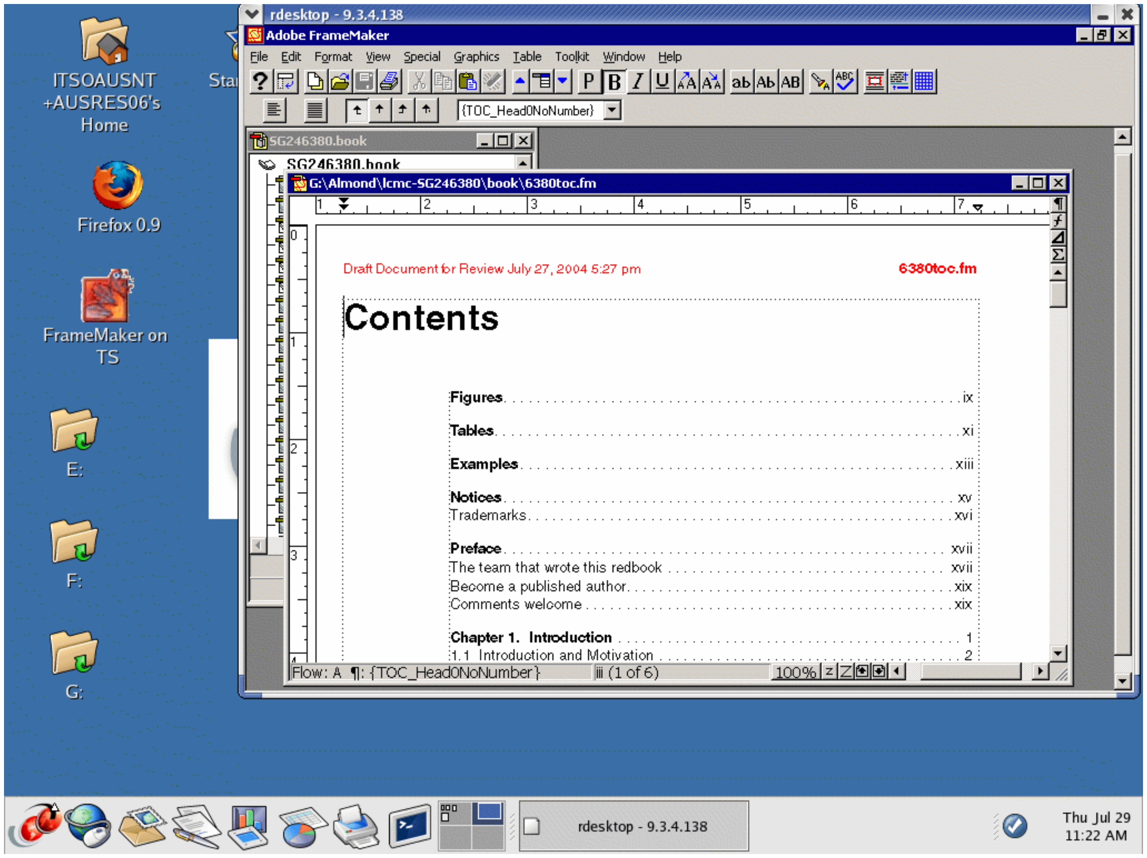
The last thing was the creation of links to the mapped Windows file shares. For covenience, we renamed them to the equivalent drive letters.

The output of these customizations can be viewed in Figure 6-5 on page 134.

6.3.4 Screenshots: Client migrated to Linux

On the following pages we provide screenshots of the most important applications now running on the Linux client:

- Adobe FrameMaker on Windows Terminal Server - Figure 6-1 on page 130
- The GIMP on Linux Client - Figure 6-2 on page 131
- Ximian Evolution on Linux Client connected to MS Exchange 2000 server - Figure 6-3 on page 132
- Mozilla Firefox on Linux client - Figure 6-4 on page 133
- Showing the new desktop, customized for ITSO - Figure 6-5 on page 134
- CUPS showing Windows network printer - Figure 6-6 on page 135


ITSOAUSNT +AUSRES06's Home
Firefox 0.9
FrameMaker on TS
E:
F:
G:
rdesktop - 9.3.4.138
Adobe FrameMaker
File Edit Format View Special Graphics Table Toolkit Window Help
{TOC_Head0NoNumber}
SG246380.book
G:\Almond\lcmc-SG246380\book\6380toc.fm
Flow: A ¶: {TOC_Head0NoNumber} iii (1 of 6) 100%
Thu Jul 29 11:22 AM

Draft Document for Review July 27, 2004 5:27 pm 6380toc.fm

Contents

Figure 6-1 FrameMaker on Windows Terminal Server session using rdesktop

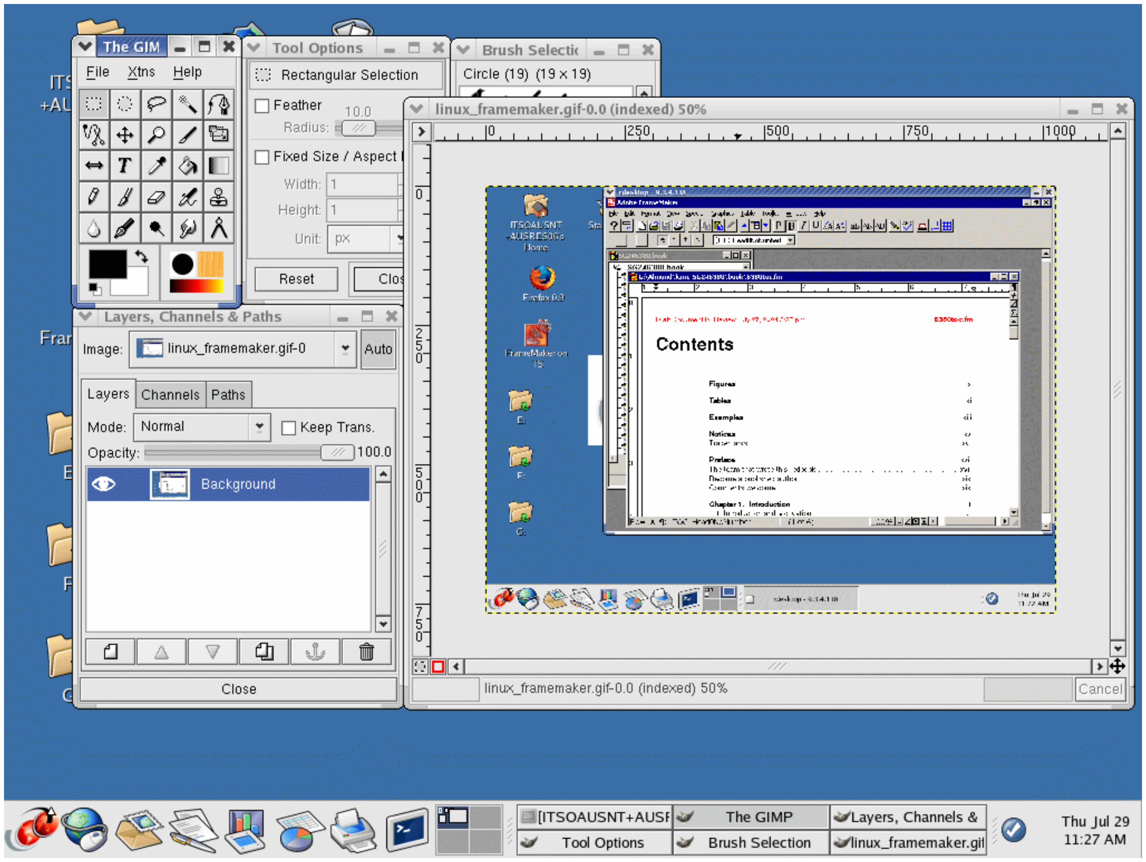
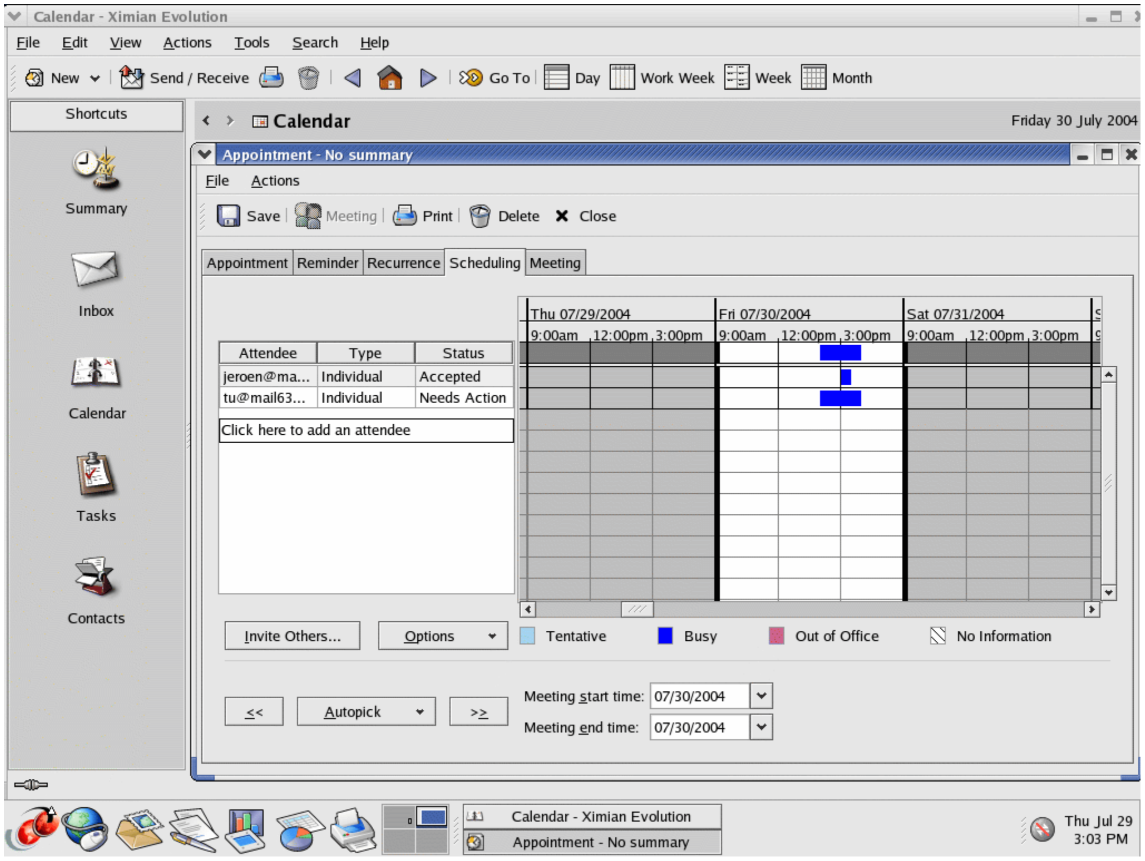


Figure 6-2 Running Gimp and taking a screenshot of FrameMaker


Calendar - Ximian Evolution
File Edit View Actions Tools Search Help
New Send / Receive Go To Day Work Week Week Month
Shortcuts
Summary
Inbox
Calendar
Tasks
Contacts
Calendar
Friday 30 July 2004
Appointment - No summary
File Actions
Save Meeting Print Delete Close
Appointment Reminder Recurrence Scheduling Meeting
Thu 07/29/2004
Fri 07/30/2004
Sat 07/31/2004
9:00am 12:00pm 3:00pm
Invite Others... Options
Tentative Busy Out of Office No Information
<< Autopick >>
Calendar - Ximian Evolution
Appointment - No summary
Thu Jul 29 3:03 PM

Attendee	Type	Status
jeroen@ma...	Individual	Accepted
tu@mail63...	Individual	Needs Action
Click here to add an attendee		

Meeting start time: 07/30/2004

Meeting end time: 07/30/2004

Figure 6-3 Ximian Evolution connected to MS Exchange 2000

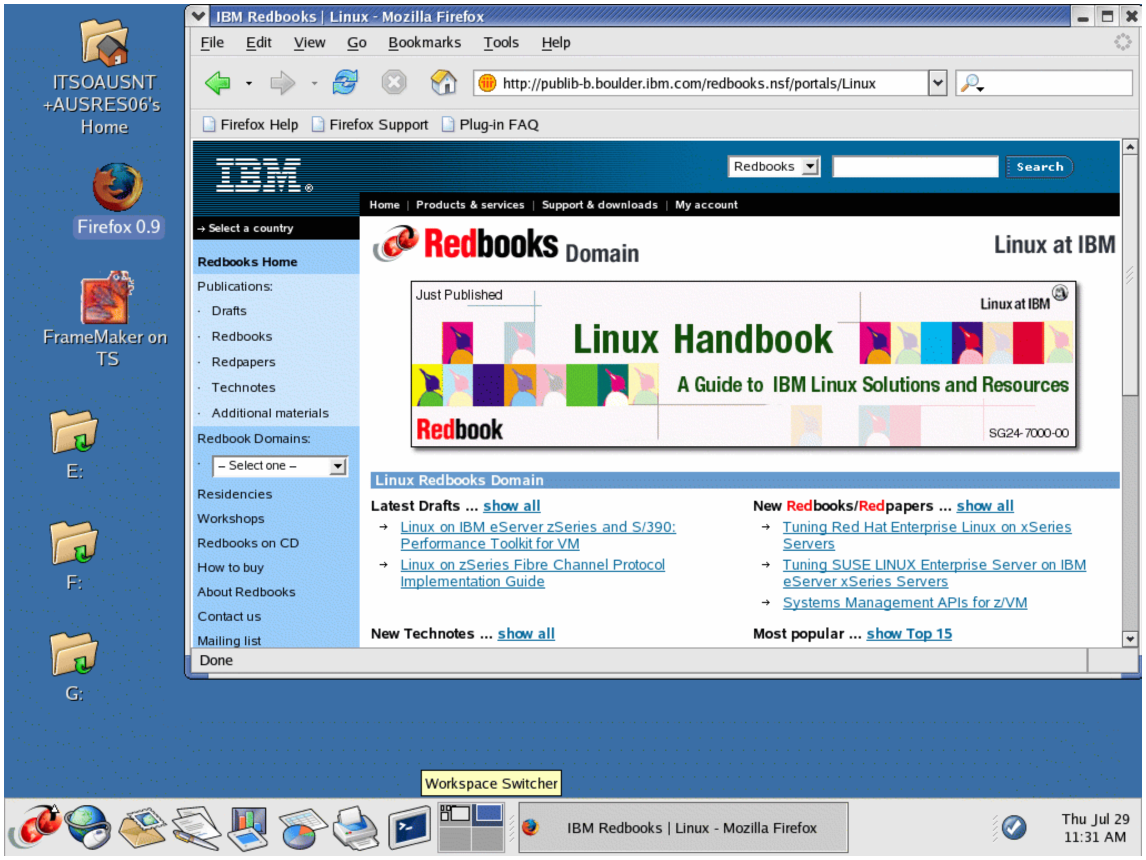

ITSOAUSNT +AUSRES06's Home
Firefox 0.9
FrameMaker on TS
E:
F:
G:
IBM Redbooks | Linux - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://publib-b.boulder.ibm.com/redbooks.nsf/portals/Linux
Firefox Help
Firefox Support
Plug-in FAQ
Redbooks
Search
Home | Products & services | Support & downloads | My account
→ Select a country
Redbooks Home
Publications:
Drafts
Redbooks
Redpapers
Technotes
Additional materials
Redbook Domains:
– Select one –
Residencies
Workshops
Redbooks on CD
How to buy
About Redbooks
Contact us
Mailing list
Redbooks Domain
Linux at IBM
Just Published
Linux at IBM
Linux Handbook
A Guide to IBM Linux Solutions and Resources
Redbook
SG24-7000-00
Linux Redbooks Domain
Latest Drafts ... show all
Linux on IBM eServer zSeries and S/390: Performance Toolkit for VM
Linux on zSeries Fibre Channel Protocol Implementation Guide
New Redbooks/Redpapers ... show all
Tuning Red Hat Enterprise Linux on xSeries Servers
Tuning SUSE LINUX Enterprise Server on IBM eServer xSeries Servers
Systems Management APIs for z/VM
New Technotes ... show all
Most popular ... show Top 15
Done
Workspace Switcher
IBM Redbooks | Linux - Mozilla Firefox
Thu Jul 29
11:31 AM

Figure 6-4 Mozilla Firefox on the Linux client

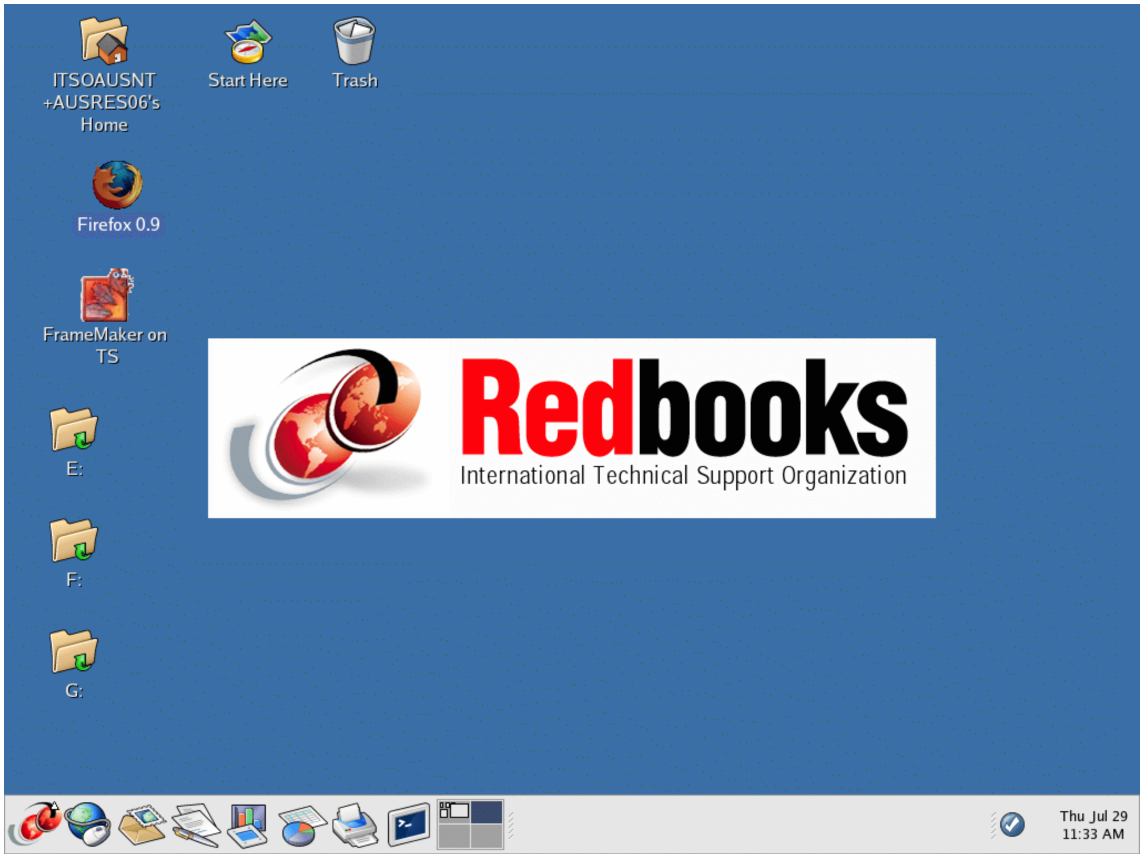
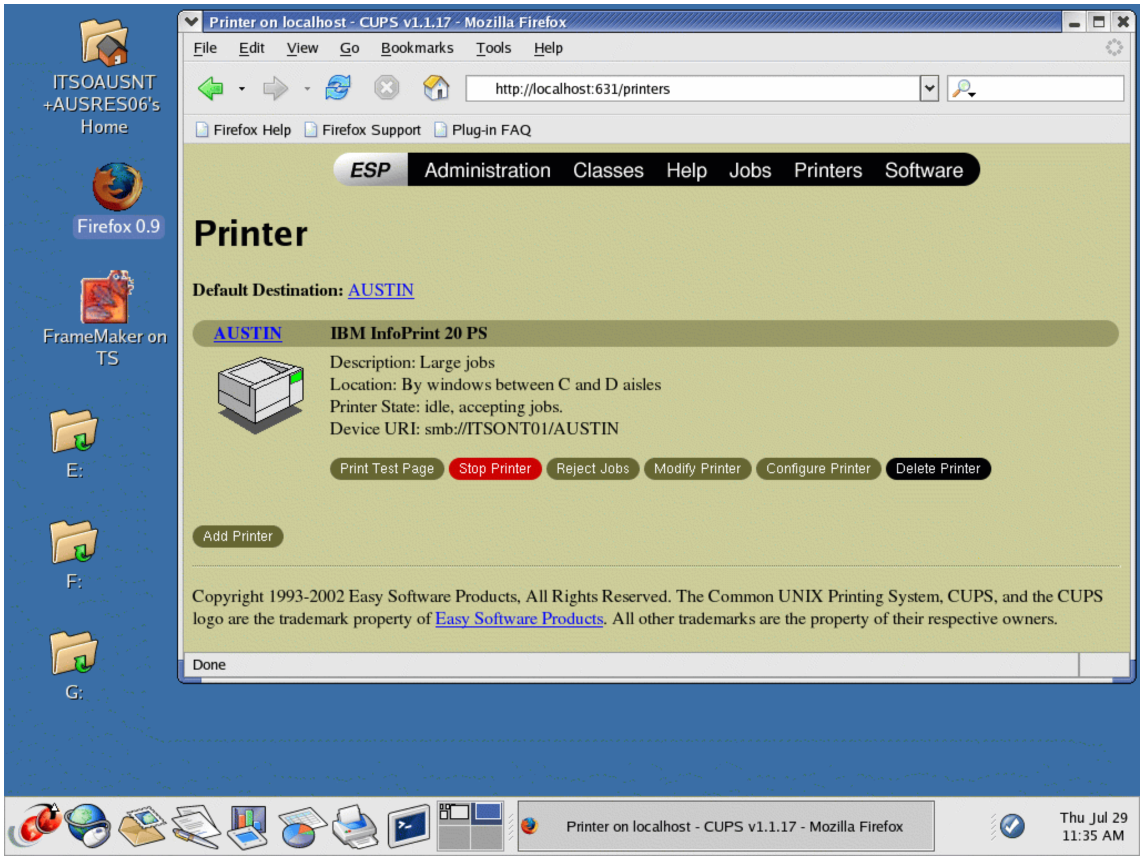


Figure 6-5 New Linux desktop showing organizational customizations for ITSO



Printer

Default Destination: AUSTIN

AUSTIN **IBM InfoPrint 20 PS**

Description: Large jobs
Location: By windows between C and D aisles
Printer State: idle, accepting jobs.
Device URI: smb://ITSONT01/AUSTIN

Copyright 1993-2002 Easy Software Products, All Rights Reserved. The Common UNIX Printing System, CUPS, and the CUPS logo are the trademark property of Easy Software Products. All other trademarks are the property of their respective owners.

Figure 6-6 Printer status page of CUPS in Mozilla Firefox

7

Integration how-tos

This chapter describes how to integrate a Linux client into an existing Windows domain (NT4 or Active Directory). Many integration issues are covered, including mounting home directories from SMB shares at logon.

The sections in this chapter are:

- 7.1, “How to join a Windows domain” on page 138
- 7.2, “How to use winbind to make domain users known locally” on page 140
- 7.3, “How to authenticate users against the domain using PAM” on page 143
- 7.4, “How to automatically mount home-directories at logon” on page 147
- 7.5, “How to use network printers in the domain” on page 151

All examples and how-tos in this section assume at least Samba Version 3.0. Some of the functionality is either not present or not mature in earlier versions.

7.1 How to join a Windows domain

Most how-tos discuss in detail how to add Linux *servers* to a domain. This how-to describes adding a Linux *client* to a domain. Adding a client to an NT4 domain is different from adding a client to an Active Directory domain. Both are discussed in detail.

In all examples in this section we use a domain AD6380 with Primary Domain Controller SMB3LAB26 and Backup Domain Controller SMB3LAB27.

7.1.1 Joining an NT4 domain

Samba is used to connect to the domain. The minimum smb.conf looks like Example 7-1.

Example 7-1 smb.conf for joining NT4 domain

```
[global]
   workgroup = AD6380
   security = domain
   password server = SMB3LAB26,SMB3LAB27
```

Replace the example domain and password servers with your own domain name and the correct names or addresses for the primary (and backup) domain controllers.

You can then join the domain using:

```
   net join -S SMB3LAB26 -U administrator
```

Replace SMB3LAB26 with the name (or IP) of your own primary domain controller and use any domain account that has the right to add machines to the domain. The command will prompt for the password of the domain account "administrator".

More details on joining an Active Directory domain can be found in the current Samba-3 release version of the Samba-HOWTO-Collection. The collection can be found at the following location:

http://samba.org/samba/docs/

7.1.2 Joining an Active Directory domain

In this case we need both Samba and Kerberos to connect. We need Kerberos to authenticate against a Windows 200x KDC.

In the example we use domain AD6380.LOCAL with AD server SMB3LAB26.

The minimum smb.conf contains the lines given in Example 7-2.

Example 7-2 smb.conf for joining Active Directory domain

```
[global]
   realm = AD6380.LOCAL
   security = ads
   password server = SMB3LAB26
```

For the realm take care to use the correct case, since Kerberos is case sensitive.

The minimum krb5.conf looks like Example 7-3.

Example 7-3 krb5.conf for joining Windows 200x Kerberos realm

```
[libdefaults]
   default_realm = AD6380.LOCAL

[realms]
   AD6380.LOCAL = {
      kdc = SMB3LAB26:88
      admin_server = SMB3LAB26
   }

[domain_realm]
   .kerberos.server = AD6380.LOCAL
```

Make sure the name of the Kerberos server is in the DNS in such a way that a reverse lookup on the IP address returns the NetBIOS name of the KDC or the NetBIOS name followed by the realm. It should not return the host name with a domain attached. The easiest way to ensure this is by putting it in the /etc/hosts entry.

Since Kerberos tickets are heavily time dependent, it is important to make sure that the AD server and clients have the same time. As Windows clients get their time from the domain controller the Linux client can use Samba tools to get the time from the server as well. You do this using the **`net time set`** command. This fetches the time from the AD server and sets the local clock.

Important: Make sure clients and the Active Directory (or Kerberos) server have the same time within a defined allowed skew.

You can test the Kerberos configuration by doing a **`kinit USERNAME@REALM`** to make sure the password is accepted by the Windows 200x KDC.

Attention: Only newly created accounts in ADS or accounts that have had their passwords changed once since migration will work. If an account stems from before the migration (or installation in the case of Administrator) the **kinit** command returns a message about a wrong encryption. Changing the password of the account resolves this problem.

To actually join the AD domain you execute the following:

```
net ads join -U administrator
```

This will prompt for the administrator password, for example, joining client machine client1 to the domain AD6380 using administrative user idsadmin (see Example 7-4).

Example 7-4 Example of joining client1 to domain AD6380

```
[root@client1 root]# net ads join -U idsadmin
idsadmin password:*******
Using short domain name -- AD6380
Joined 'CLIENT1' to realm 'AD6380.LOCAL'
```

Joining in a particular organizational unit can be done by first getting the correct credentials and then joining the unit. For example, if you want to join the domain (that is, create a computer account) in a container called Clients under the organizational directory Computers/ITSO, you execute:

```
kinit Administrator@AD6380.LOCAL
net ads join “Computers\ITSO\Clients”
```

Attention: Since Windows 2003 uses SMB signing you have to put the following line in the smb.conf file when trying to join a Windows 2003 ADS.

```
client use spnego = yes
```

More details on joining an Active Directory domain can be found in the *Samba HOWTO collection section 6.4:*

http://samba.org/samba/docs/Samba-HOWTO-Collection.pdf/

7.2 How to use winbind to make domain users known locally

After joining a domain as described in 7.1, “How to join a Windows domain” on page 138, it will be necessary to add all domain accounts to the Linux client if the domain accounts are going to log on to the client. In smaller domains this will not be a problem, but it is generally not good administrative practice. This would mean that adding an account to the domain would mean adding an account to all clients. That sort of takes the advantage out of using a domain.

In this section we describe how to use winbind to avoid creating domain accounts locally. The winbind daemon will take care of translating domain accounts to uids and gids to the client OS.

The winbind daemon will read its settings from the smb.conf file. The lines in Example 7-5 have to be added to the smb.conf to enable winbind to function.

Example 7-5 Lines added to smb.conf for winbind

```
[global]
   winbind separator = +
   idmap uid = 10000-20000
   idmap gid = 10000-20000
   winbind enum users = yes
   winbind enum groups = yes
   template homedir = /home/%D+%U
   template shell = /bin/bash
```

The separator means that accounts will be written as “AD6380+Administrator” for the Administrator account in the domain AD6380. The other entries give the uid and gid range to use for domain accounts and what to use as shell and home directory in case of actual logon. We have found that a plus sign (+) as a winbind separator will work most successfully within the Linux environment.

Important: If you have a very large domain, you may not be able to incorporate all users in your idmap uid range. This is only problematic if you use winbind on a domain client, which all users use via the network.

After these changes the winbind daemon has to be started. Make sure the winbind daemon starts on system boot. On most distributions (certainly on Red Hat and Novell/SuSE) this is done using the following command:

```
   chkconfig winbind on
```

The winbind functionality can be tested by using the **`wbinfo`** command. Executing with the -u option will show all domain accounts translated and the -g option will show all domain groups.

Example 7-6 Example output of wbinfo command

```
[root@client1 root]# wbinfo -u
AD6380+Administrator
AD6380+Guest
```

```
AD6380+TsInternetUser
AD6380+idsadmin
AD6380+krbtgt
AD6380+HOST/client1
AD6380+SMB3LAB26$
```

Important: The service nscd (name service caching daemon) interferes with proper functioning of winbind. *Never* run nscd on systems where winbindd runs.

With these settings the winbind daemon will be able to translate domain accounts to local users. We tell the system to actually use winbind by putting a reference in the Name Service Switch (NSS) configuration.

The configuration file for NSS is /etc/nsswitch.conf. This file contains entries like:

```
passwd: files example
```

This means that when a request for a password comes in, first a routine in /lib/libnss_files.so will be called to resolve it and then a routine in /lib/libnss_example.so. Once the request is resolved, the result will be returned to the application. To add winbind to the resolve path we change the following lines in /etc/nsswitch.conf in the following manner:

```
passwd: files winbind
group: files winbind
```

To test this execute the command:

```
getent passwd
```

This will return not only the entries in the file /etc/passwd but also the domain accounts that were translated by winbind.

Example 7-7 Partial example output of getent passwd using winbind in NSS

```
[root@client1 root]# getent passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
.......................
AD6380+Administrator:x:10000:10000:Administrator:/home/AD6380+Administrator:/bi
n/bash
AD6380+Guest:x:10001:10000:Guest:/home/AD6380+Guest:/bin/bash
AD6380+TsInternetUser:x:10002:10000:TsInternetUser:/home/AD6380+TsInternetUser:
/bin/bash
AD6380+idsadmin:x:10003:10000:idsadmin:/home/AD6380+idsadmin:/bin/bash
AD6380+krbtgt:x:10004:10000:krbtgt:/home/AD6380+krbtgt:/bin/bash
AD6380+HOST/client1:x:10005:10001:client1:/home/AD6380+HOST/client1:/bin/bash
```

```
AD6380+SMB3LAB26$:x:10006:10002:SMB3LAB26:/home/AD6380+SMB3LAB26_:/bin/bash
```

Because of the way that winbind works (namely handing out the first uid in the range to the first domain user it has to authenticate and then storing the mapping), the mapping between domain user and uid will not be the same on all clients. This leads to problems if the clients are using shared file system bases on NFS.

Important: When winbind fails because the Kerberos connection is lost because of time skew, the daemon has to be restarted after fixing the time skew.

The winbind daemon will translate users and generate home directories and shells. These home directories are not created by winbind. How to do this is shown in 7.3.3, "Winbind and home directories" on page 145.

Using the winbind daemon will create the connection to the domain for translating users. To enable domain users to log onto the Linux client we need a change in the PAM configuration, as shown in the next section.

7.3 How to authenticate users against the domain using PAM

Once the Linux client is part of a domain (as described in 7.1, "How to join a Windows domain" on page 138) and knows about the domain accounts (as described in 7.2, "How to use winbind to make domain users known locally" on page 140) we can configure the system to allow domain accounts to log on.

Pluggable Authentication Modules (PAM) is a system for abstracting authentication and authorization technologies. Using PAM modules it is possible to change the way applications authenticate or authorize accounts without having to recompile the application.

In most distributions the configuration of PAM is governed by files in /etc/pam.d/. Usually there is one file per application. So changing authentication modes for an application is as simple as adding a corresponding module to the config file.

Since the PAM implementation of Linux distribution differs we look at our test distributions in detail.

7.3.1 Winbind and PAM on Red Hat Desktop

Red Hat has implemented the pam_stack module. This means that configuration files in /etc/pam.d/ can use settings from other files, essentially stacking settings. For this purpose Red Hat has implemented most of the settings in the /etc/pam.d/system-auth configuration file. This means that we only have to add winbind PAM modules to this file and all applications using PAM will be winbind aware.

Since the file /etc/pam.d/system-auth is generated by **authconfig**, care has to be taken in running this command after making changes by hand.

A sample system-auth file contains the lines in Example 7-8.

Example 7-8 Example of part of /etc/pam.d/system-auth file

```
#%PAM-1.0
# This file is auto-generated.
# User changes will be destroyed the next time authconfig is run.
auth        required      /lib/security/$ISA/pam_env.so
auth        sufficient    /lib/security/$ISA/pam_unix.so likeauth nullok
auth        sufficient    /lib/security/$ISA/pam_winbind.so use_first_pass
auth        sufficient    /lib/security/$ISA/pam_krb5.so use_first_pass
auth        sufficient    /lib/security/$ISA/pam_smb_auth.so use_first_pass nolocal
auth        required      /lib/security/$ISA/pam_deny.so

account     required      /lib/security/$ISA/pam_unix.so
account     sufficient    /lib/security/$ISA/pam_winbind.so
........................
```

Once pam_winbind.so has been incorporated into the system-auth file all applications using the file through pam_stack.so are now winbind aware. This means that we can log onto the Linux client using a domain account, remembering to use:

```
<domainname><winbind-seperator><accountname>
```

So an example in our test domain would be AD6380+Administrator.

If it is problematic that winbind-enabled users are available for all applications using the pam_stack module, then pam_winbind.so calls could be placed only in the configuration files for those applications that need it.

7.3.2 Winbind and PAM on Novell Linux Desktop

The Novell Linux Desktop (NLD) does not use a pam_stack module to get settings from a central file. This means that for each application that needs to use winbind authentication the module has to be added to the configuration file in /etc/pam.d.

We take the configuration file for sshd as an example. After incorporating the pam_winbind.so module the file looks like Example 7-9.

Example 7-9 Configuration file /etc/pam.d/sshd after incorporating winbind

```
#%PAM-1.0
auth     sufficient     pam_winbind.so
auth     required       pam_unix2.so use_first_pass  # set_secrpc
auth     required       pam_nologin.so
auth     required       pam_env.so
account  required       pam_unix2.so
account  sufficient     pam_winbind.so
account  required       pam_nologin.so
password required       pam_pwcheck.so
password required       pam_unix2.souse_first_pass use_authtok
session  required       pam_unix2.sonone # trace or debug
session  required       pam_limits.so
```

For every PAM-enabled application that you want to enable for domain users, both the auth and account line for pam_winbind.so have to be added (before the nologin lines). Also to prevent a double password prompt the parameter use_first_pass should be added to any pam module needing a password in the configuration file apart from the pam_winbind.so module.

7.3.3 Winbind and home directories

Winbind-enabled users do not exist on the client locally. The winbind configuration in /etc/samba/smb.conf tells the system which shell to use and where the home directory of the user is located (this function is performed by the /etc/passwd file for local users). However the home directory is not created by winbind.

This problem can be solved in a number of ways:

- Create all possible home directories (empty) on all clients.
- Create all home directories on a server file system that is mounted on all clients (either through SMB or NFS).
- Use the pam_mkhomedir.so module to create a home directory at first logon.

The first two options seem straightforward to implement but lead to management issues every time a user is added to the domain. The last option is seen as a best practice and consists of adding a line to the PAM configuration files that will create the home directory if it does not exist. This change of the PAM configuration file has to be done once when the Linux client “master” is created.

Red Hat Desktop

On Red Hat systems the pam_mkhomedir module is added to /etc/pam.d/system-auth. This way the home directory will be created when a user logs in through a login (for example, a terminal prompt), a secure shell session, or a graphical logon.

The line that is to be added to the file is shown in Example 7-10.

Example 7-10 Example of part of the /etc/pam.d/system-auth file, including pam_mkhomedir

```
#%PAM-1.0
# This file is auto-generated.
# User changes will be destroyed the next time authconfig is run.
auth        required        /lib/security/$ISA/pam_env.so
auth        sufficient      /lib/security/$ISA/pam_unix.so likeauth nullok
auth        sufficient      /lib/security/$ISA/pam_winbind.so use_first_pass
auth        sufficient      /lib/security/$ISA/pam_krb5.so use_first_pass
auth        sufficient      /lib/security/$ISA/pam_smb_auth.so use_first_pass nolocal
auth        required        /lib/security/$ISA/pam_deny.so

account     required        /lib/security/$ISA/pam_unix.so
account     sufficient      /lib/security/$ISA/pam_winbind.so
........................
session     optional        /lib/security/$ISA/pam_mkhomedir skel=/etc/skel umask=0022
```

The skel option tells the module where to get the skeleton files to copy to the newly created home directory. The umask governs the creation and subsequent permission settings on the directory.

Novell Linux Desktop

Since NLD does not use system-auth and pam_stack.so, a pam_mkhomedir.so has to be added to every configuration file for applications that let users log on. Examples of these applications are ssh, telnet, gdm, xdm, and login. The first login of every domain user throught any of these applications will create the home directory.

The line that is added to the file is of the form:

```
session   optional  pam_mkhomedir skel=/etc/skell umask=0022
```

An example for the ssh application is given in Example 7-11 on page 147.

Example 7-11 Example of /etc/pam.d/sshd file including pam_mkhomedir.so

```
#%PAM-1.0
auth     sufficient   pam_winbind.so
auth     required     pam_unix2.so use_first_pass
auth     required     pam_nologin.so
auth     required     pam_env.so
account  required     pam_unix2.so
account  sufficient   pam_winbind.so
account  required     pam_nologin.so
password required     pam_pwcheck.so
password required     pam_unix2.so use_first_pass use_authtok
session  optional     pam_mkhomedir.so  skel=/etc/skel umask=0022
session  required     pam_unix2.so none        # trace or debug
session  required     pam_limits.so
```

The skel option tells the module where to get the skeleton files to copy to the newly created home directory. The umask governs the creation and subsequent permission settings on the directory.

7.4 How to automatically mount home-directories at logon

One of the great functionalities of Windows is the Single Sign On (SSO) function. Once you log onto a Windows OS, it takes your password to try and mount shares that you have installed as remountable at next logon.

A similar functionality can be created on a Linux client using the pam_mount module. This PAM module is not completely mature yet. It is not included in all enterprise distributions. But with a little extra work it can be made to function.

Restriction: Take care when mounting SMB shares as users' home directories when using graphical logon. Some graphical desktop environments will not work in a SMB-mounted file system, most importantly, those that depend on symbolic links or sockets, since SMB file systems will not work with symbolic links and sockets.

Tip: Mount the user's domain share in a subdirectory of the home directory, thus avoiding all issues with desktop environments.

Restriction: The pam_mount module does not follow the rules of PAM implementation, since the actual mount is done in the auth part instead of the session part may mean that the mount will occur before a pam_mkhomedir has actually created a mount point.

The pam_mount module not only mounts SMB file systems, but also NCP, loop-mounted encrypted file systems, and basically any file system handled by the **mount** command.

7.4.1 pam_mount on Red Hat Desktop

The module is added to the /etc/pam.d/system-auth file to enable automatic mounting for all login modes. The module consists of both an "auth" part, which acquires the password through PAM; and a "session" part, which does the actual mounting. Since this is a session module this enables an unmount of the file systems when the session closes.

Tip: Check if you need automatic mounting for all modes to login. For example, a **su** from root to a user will not work because a password is not provided.

The lines to add look like this:

```
auth      required     pam_mount.so
```

And:

```
session   optional     pam_mount.so
```

The auth line should go before pam_unix and pam_winbind lines. The session line should go in the session section.

The pam_mount module has its own configuration file /etc/security/pam_mount.conf. This file contains settings like where the **mount** and **umount** commands are found, debug settings, whether mount points should be created, and which file systems should be mounted for which users.

A minimum pam_mount.conf file for just mounting SMB shares looks like Example 7-12.

Example 7-12 Minimum pam_mount.conf

```
debug 1
mkmountpoint 1
options_require    nosuid,nodev
```

```
lsof /usr/sbin/lsof
fsck /sbin/fsck
losetup /sbin/losetup
unlosetup /sbin/losetup -d
smbmount /usr/bin/smbmount
umount   /usr/bin/smbumount
volume * smb smb3lab26 & /home/&/domainshare uid=&,dmask=0750 - -
```

This enables debugging on pam_mount. Mount points are created when non-existent and the share that will be mounted is indicated in the last line.

The astrick)*) in second place in the last line indicates that this volume is mounted for every user. The ampersand (&) in the definition is expanded to the user name of the user logging in. So the volume line in Example 7-12 on page 148 tells pam_mount to mount a share named after the user name from server smb3lab26 onto mount point /home/<username>/domainshare. The options indicate that the mount should be owned by the indicated uid and have permissions as indicated by dmask.

Attention: In some cases the pam_mount module will fail to setuid root, which means that normal **mount** commands will fail with and `only root can do that` message. This is the case for Red Hat Desktop.

Possible workaround for pam_mount failing to setuid root

In this case you will find errors in the logs (/var/log/messages, most likely) that look like:

```
pam_mount: error setting uid to 0
```

Or:

```
pam_mount: mount: only root can do that
```

When these messages are present, this means that pam_mount is not working using the normal mount tools. A workaround can be created where the normal **mount** command is not used, but a script that will call **smbmount**. Since the **smbmount** and **smbumount** are allowed to be used by normal users, this will work. We give an example of a possible script in Example 7-13 on page 150.

Another possible problem when mounting shares from a domain is that the ampersand (&) in the pam_mount.conf file expands to the entire user name, including the domainvname and the winbind separator. As we stated in 7.2, “How to use winbind to make domain users known locally” on page 140, the best winbind separator in a Linux environment is a plus sign (+). This, however, is an illegal character in a share name under Windows. This means that if we want to use pam_mount to mount a share that has the user name in it, we have to get rid of the <domainname><winbind-separator> part. We can tackle this in the same script that handles the mount problem. We devised such a script to get pam_mount working on Red Hat Desktop. The script for the mount command we give here (as is) as an example in Example 7-13.

Example 7-13 Script domount to handle problems with pam_mount

```
#!/bin/sh  -x
#
# Script to fix pam_mount.so not getting uid=0 rights
#
# create mountpoint if it does not exist
if [ ! -d $4 ]; then
  mkdir $4
  chmod 777 $4
fi
#
# if mount is smbfs and user is domain-user substract 'DOM+' from the sharename
# this assumes that the winbind-seperator is '+'
#
if [ $2 = smbfs ]; then
  IFS="/"
  host=`echo $3 |awk '{print $1}' `
  duser=`echo $3 |awk '{print $2}' `
  IFS="+"
  nuser=`echo $duser |awk '{print $2}'`
  if [ "$nuser" = "" ]; then
# if not domain-user share than contains no + and nuser will be empty
# restore sharename from original argument 3
    sharename=$3
  else
# construct a sharename consisting of only username
# shares do not handly +
    sharename="//$host/$nuser"
  fi
  IFS=" "
  /usr/bin/smbmount $sharename $4 $5 $6 $7 $8 $9
else
# if not smb mount try to perform normal mount
  /usr/bin/mount $1 $2 $3 $4 $5 $6 $7 $8 $9
fi
```

This script works with pam_mount Version 0.9.18 on Red Hat Desktop 3. The script has to be put somewhere in the file system and should be executable and setuid root. If the script is called /usr/local/bin/domount ,we do this by executing the following commands:

```
chown root.root /usr/local/bin/domount
chown 4755 /usr/local/bin/domount
```

The script is now executable for all users and will be run as root. A corresponding umount script is very simple, as shown in Example 7-14.

Example 7-14 Script doumount to handle problems with pam_mount

```
#!/bin/sh -x
#
# umount script to fix pam_mount.so functionality
#
# smbumount is allowed for user
/usr/bin/smbumount $1
```

This file has to have the same properties as the domount script.

Once the scripts have been installed with the proper owner and permissions, the pam_mount.conf file has to be changed to use these scripts. To enable this we make the following changes to the /etc/security/pam_mount.conf file:

```
smbmount /usr/local/bin/domount
umount   /usr/local/bin/doumount
```

7.5 How to use network printers in the domain

Being able to use existing printers is one of the most important pre-conditions for a new type of client in any environment. We do not want to keep separate printing environments for every type of client.

This means that the Linux client will have to use the (network) printers available in the domain. We can achieve this in two ways:

- Print directly to the printer using its network interface.
- Print to the printer using the Windows print server and its SMB interface.

Both options are available using CUPS from the Linux client. We suggest using the second option, however, since all prints in the domain (and user's environment) will be on equal footing. This means there are just prints and not Windows prints and Linux prints.

You need to be running CUPS on the Linux client. This can be checked by:

```
/etc/init.d/cups status
```

This should return that CUPS is running and give its PID. If not it can be started using:

```
/etc/init.d/cups start
```

Make sure it is started at next reboot using:

```
chkconfig cups on
```

You also need to make sure that there is a link to smbspool in the CUPS backend directory:

```
# ls -al /usr/lib/cups/backend/smb
lrwxrwxrwx 1 root root 21 Mar 1 12:34 /usr/lib/cups/backend/smb ->
../../../bin/smbspool
```

You can verify smb support in CUPS using this command:

```
# lpinfo -v | grep smb
network smb
```

You can create a new printer in two ways:

- Using the **`lpadmin`** command
- Using the CUPS Web interface

We describe both methods in detail.

Create printer using lpadmin command

You can create a printer from the command line. This enables scripted creation of many printers. This means that creation of domain printers can be incorporated into a login profile, such that users do not have to create printers by hand on their client.

To create the printer on the command line we use the following command:

```
lpadmin -p printer1 -E -v smb://Administrator:*****@SMB3LAB26/printer1 -D
"printer1 in AD6380 example domain"
```

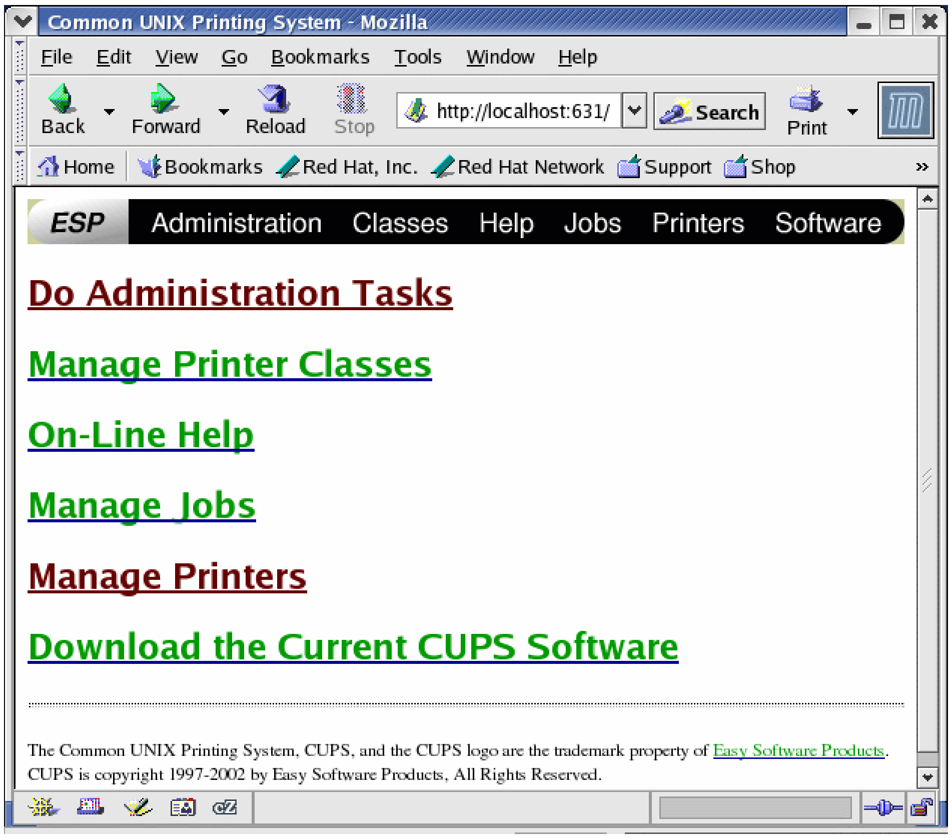
You need to change the fields to the names in your own domain. The first printer1 is the local printer name and the second printer1 is the name of the printer share. The server SMB3LAB26 is the print server. If the printer share is available for everyone, you can leave out the user name and password in the smb: URL. Since using a user-protected printer share will expose passwords in scripts and login profiles, the best practice is to make printer shares available for everyone.

You can also add the -m option to set the model. Or use the -l option to set a location.

Create printer using CUPS Web interface

You could accomplish the same thing as above by using the CUPS Web interface. We do this following these steps:

1. In a browser, load the CUPS admin interface using this URL: http://localhost:631/.


Common UNIX Printing System - Mozilla
File Edit View Go Bookmarks Tools Window Help
Back Forward Reload Stop
http://localhost:631/
Search
Print
Home Bookmarks Red Hat, Inc. Red Hat Network Support Shop
ESP Administration Classes Help Jobs Printers Software

Do Administration Tasks

Manage Printer Classes

On-Line Help

Manage Jobs

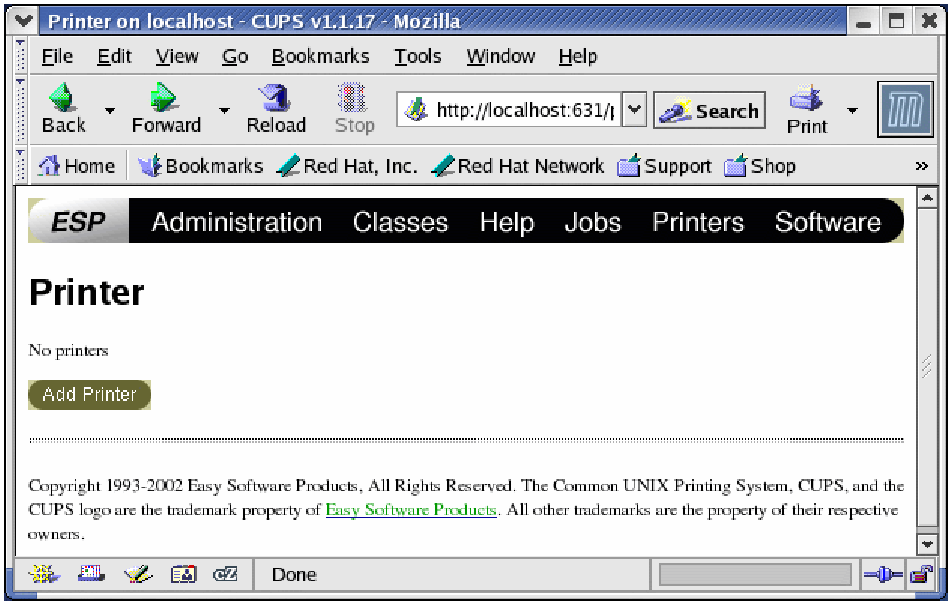
Manage Printers

Download the Current CUPS Software

The Common UNIX Printing System, CUPS, and the CUPS logo are the trademark property of Easy Software Products. CUPS is copyright 1997-2002 by Easy Software Products, All Rights Reserved.

Figure 7-1 CUPS Web interface initial page

2. Then click **Manage Printers**.


Printer on localhost - CUPS v1.1.17 - Mozilla
File Edit View Go Bookmarks Tools Window Help
Back Forward Reload Stop
http://localhost:631/
Search
Print
Home Bookmarks Red Hat, Inc. Red Hat Network Support Shop
ESP Administration Classes Help Jobs Printers Software
Add Printer
Done

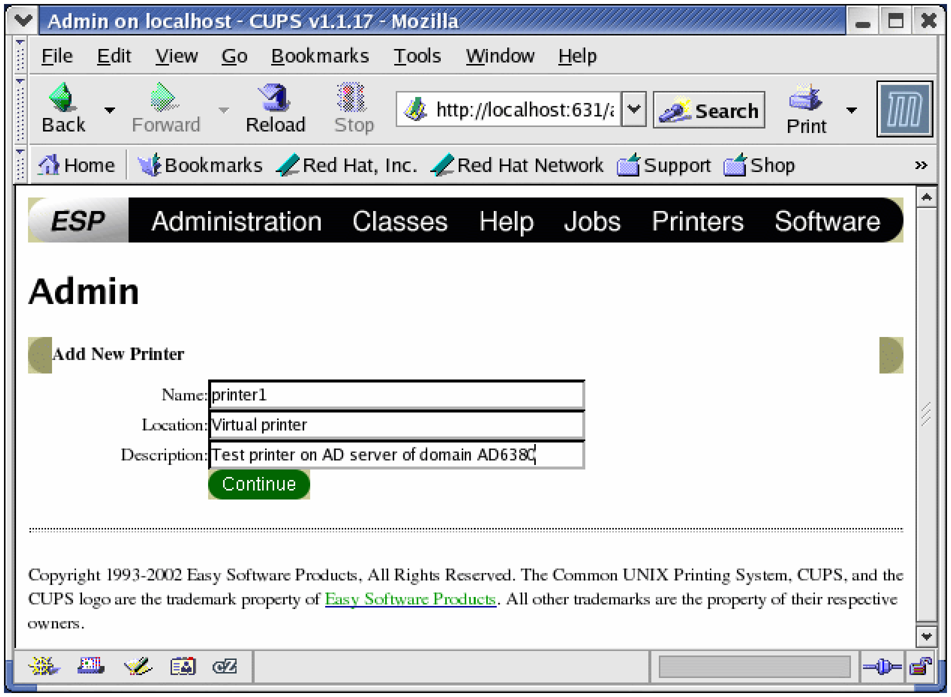
Printer

No printers

Copyright 1993-2002 Easy Software Products, All Rights Reserved. The Common UNIX Printing System, CUPS, and the CUPS logo are the trademark property of Easy Software Products. All other trademarks are the property of their respective owners.

Figure 7-2 CUPS Web interface Manage Printers page

3. Click **Add Printer**.


Admin on localhost - CUPS v1.1.17 - Mozilla
File Edit View Go Bookmarks Tools Window Help
Back Forward Reload Stop
http://localhost:631/a
Search
Print
Home Bookmarks Red Hat, Inc. Red Hat Network Support Shop
ESP Administration Classes Help Jobs Printers Software

Admin

Add New Printer

Name: printer1

Location: Virtual printer

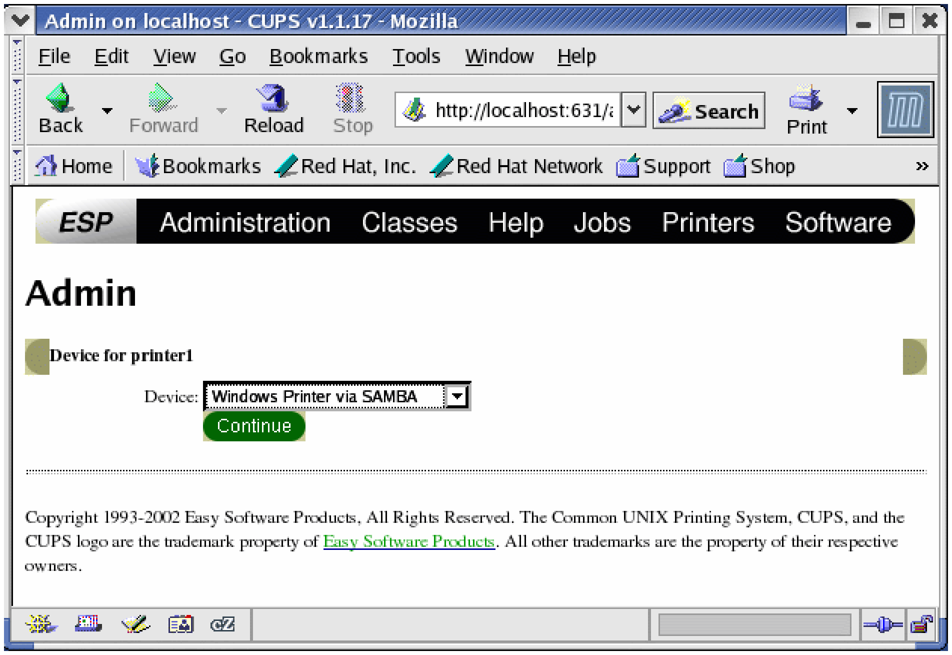
Description: Test printer on AD server of domain AD6380

Continue

Copyright 1993-2002 Easy Software Products, All Rights Reserved. The Common UNIX Printing System, CUPS, and the CUPS logo are the trademark property of Easy Software Products. All other trademarks are the property of their respective owners.

Figure 7-3 CUPS Web interface Add Printer page

4. You get a page where you enter the Name, Location, and Description of the printer. Then click **Continue**.


Admin on localhost - CUPS v1.1.17 - Mozilla
File Edit View Go Bookmarks Tools Window Help
Back Forward Reload Stop
http://localhost:631/a
Search
Print
Home Bookmarks Red Hat, Inc. Red Hat Network Support Shop
ESP Administration Classes Help Jobs Printers Software

Admin

Device for printer1

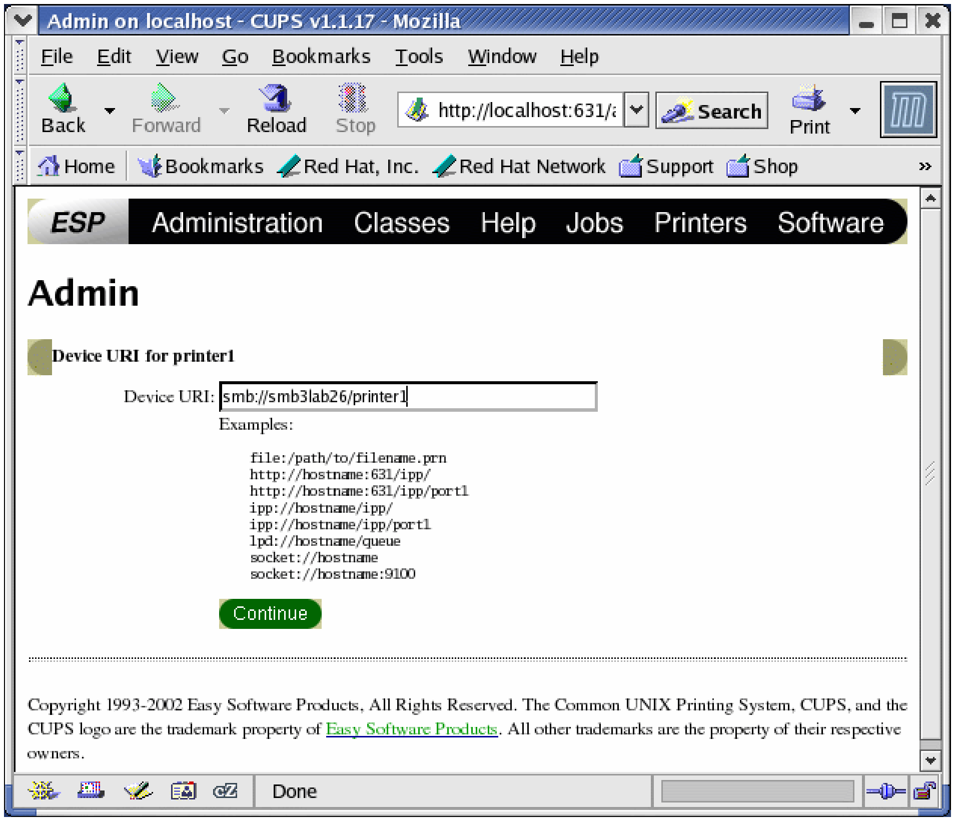
Device: Windows Printer via SAMBA

Continue

Copyright 1993-2002 Easy Software Products, All Rights Reserved. The Common UNIX Printing System, CUPS, and the CUPS logo are the trademark property of Easy Software Products. All other trademarks are the property of their respective owners.

Figure 7-4 CUPS Web interface Device for page

5. On the next page you choose the device for the printer. Choose **Windows Printer via SAMBA** and click **Continue**.


Admin on localhost - CUPS v1.1.17 - Mozilla
File Edit View Go Bookmarks Tools Window Help
Back Forward Reload Stop
http://localhost:631/a
Search
Print
Home Bookmarks Red Hat, Inc. Red Hat Network Support Shop
ESP Administration Classes Help Jobs Printers Software
Continue
Done

Admin

Device URI for printer1

Device URI: smb://smb3lab26/printer1

Examples:

```
file:/path/to/filename.prn
http://hostname:631/ipp/
http://hostname:631/ipp/port1
ipp://hostname/ipp/
ipp://hostname/ipp/port1
lpd://hostname/queue
socket://hostname
socket://hostname:9100
```

Copyright 1993-2002 Easy Software Products, All Rights Reserved. The Common UNIX Printing System, CUPS, and the CUPS logo are the trademark property of Easy Software Products. All other trademarks are the property of their respective owners.

Figure 7-5 CUPS Web interface Device URI for page

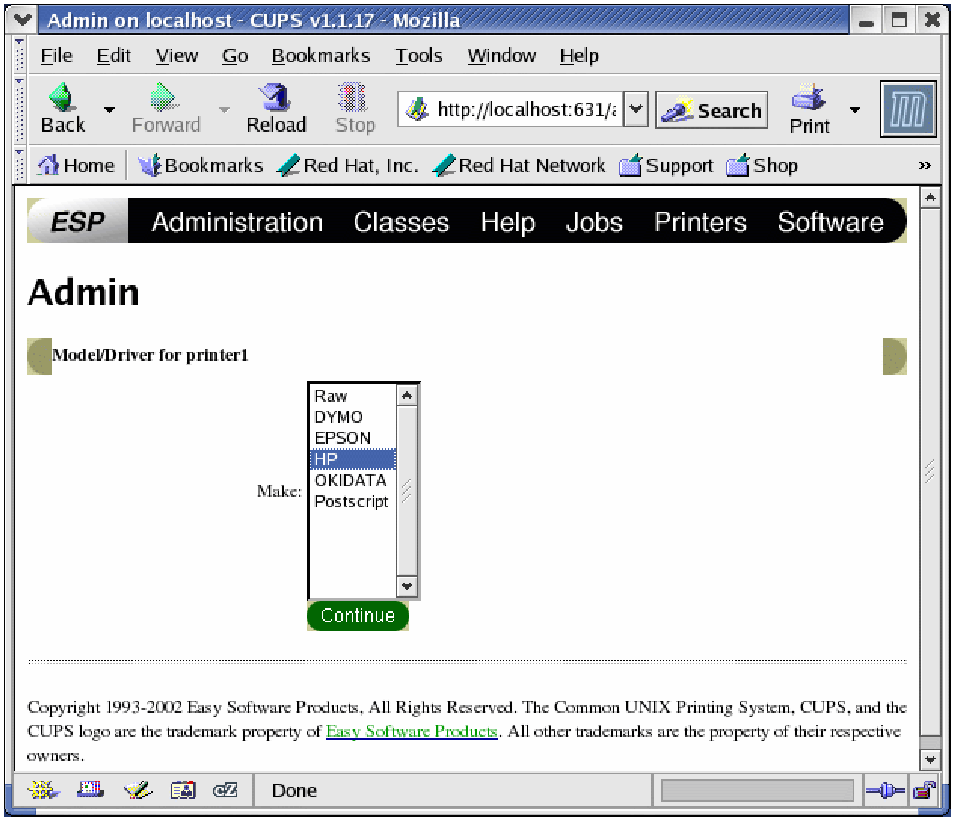
6. The next page asks for a device URI. Input:

   `smb://<domainname>:<password>@<printserver>/<printershare>`

   Or:

   `smb://<printserver>/<printershare>`

   Then click **Continue**.


Admin on localhost - CUPS v1.1.17 - Mozilla
File Edit View Go Bookmarks Tools Window Help
Back Forward Reload Stop
http://localhost:631/a
Search
Print
Home Bookmarks Red Hat, Inc. Red Hat Network Support Shop
ESP Administration Classes Help Jobs Printers Software
Done

Admin

Model/Driver for printer1

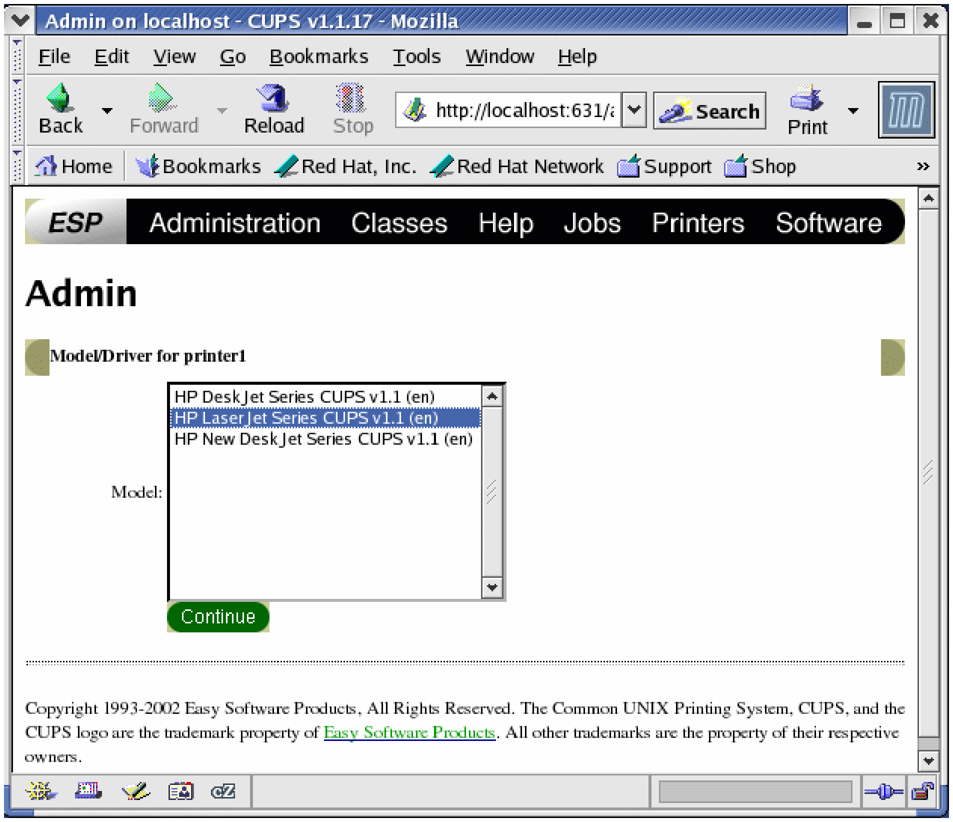
Make: Raw, DYMO, EPSON, HP, OKIDATA, Postscript

Continue

Copyright 1993-2002 Easy Software Products, All Rights Reserved. The Common UNIX Printing System, CUPS, and the CUPS logo are the trademark property of Easy Software Products. All other trademarks are the property of their respective owners.

Figure 7-6 CUPS Web interface Model/Driver for page

7. On the next page choose a model/driver for the printer and click **Continue**.


Admin on localhost - CUPS v1.1.17 - Mozilla
File Edit View Go Bookmarks Tools Window Help
Back Forward Reload Stop
http://localhost:631/a
Search
Print
Home Bookmarks Red Hat, Inc. Red Hat Network Support Shop
ESP Administration Classes Help Jobs Printers Software

Admin

Model/Driver for printer1

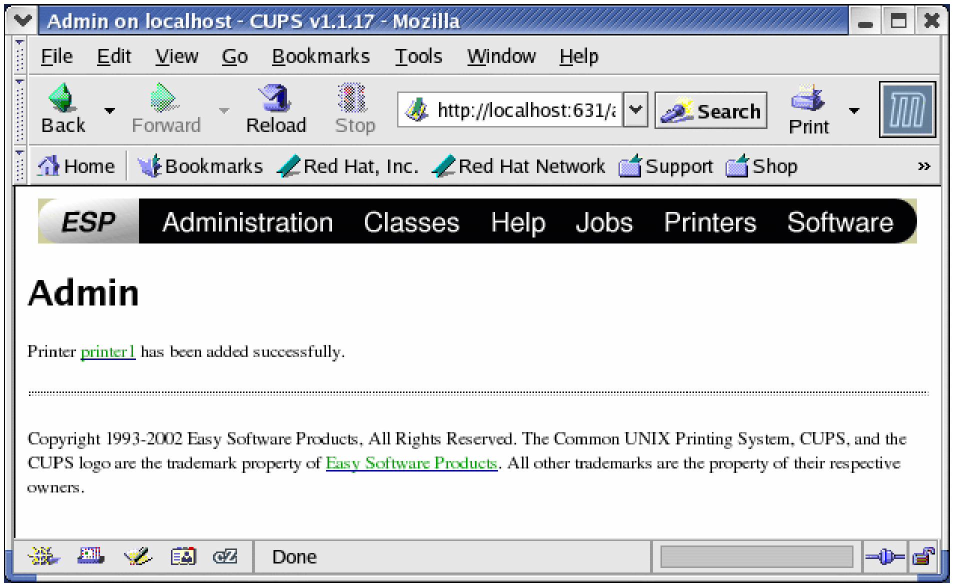
Model:
HP DeskJet Series CUPS v1.1 (en)
HP LaserJet Series CUPS v1.1 (en)
HP New DeskJet Series CUPS v1.1 (en)

Continue

Copyright 1993-2002 Easy Software Products, All Rights Reserved. The Common UNIX Printing System, CUPS, and the CUPS logo are the trademark property of Easy Software Products. All other trademarks are the property of their respective owners.

Figure 7-7 CUPS Web interface Model/Driver for second page

8. The next screen gives a more detailed selection based on the model chosen on the previous screen. Choose the correct one and click **Continue**.


Admin on localhost - CUPS v1.1.17 - Mozilla
File Edit View Go Bookmarks Tools Window Help
Back Forward Reload Stop
http://localhost:631/
Search
Print
Home Bookmarks Red Hat, Inc. Red Hat Network Support Shop
ESP Administration Classes Help Jobs Printers Software
Done

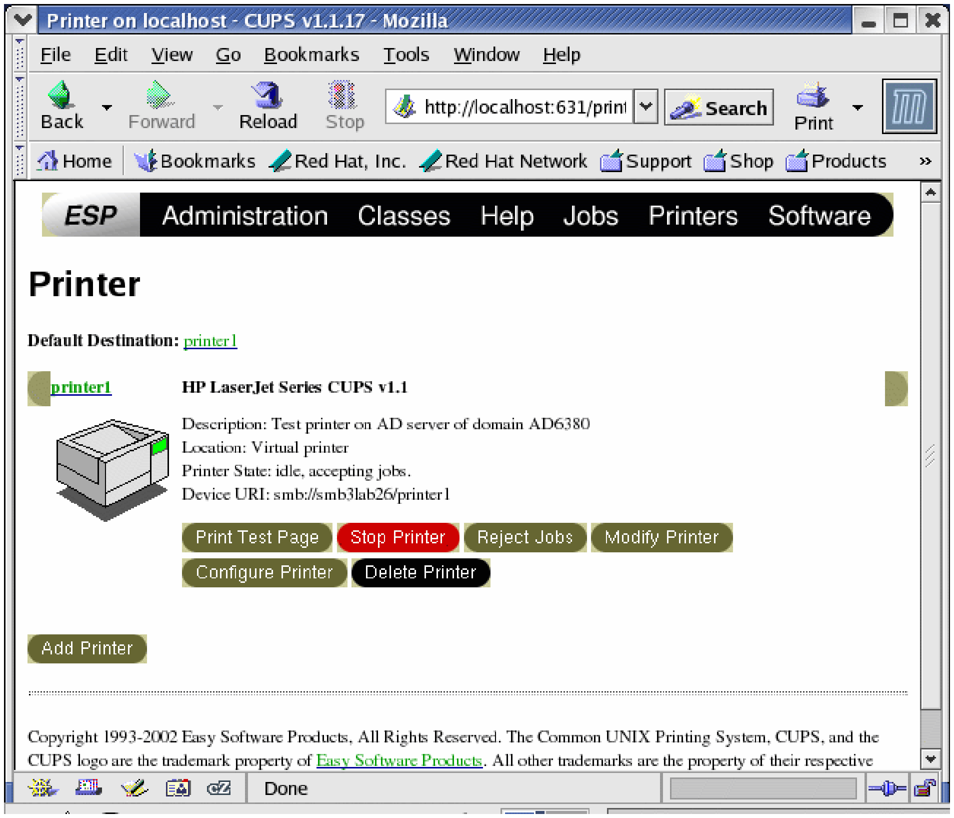
Admin

Printer printer1 has been added successfully.

Copyright 1993-2002 Easy Software Products, All Rights Reserved. The Common UNIX Printing System, CUPS, and the CUPS logo are the trademark property of Easy Software Products. All other trademarks are the property of their respective owners.

Figure 7-8 CUPS Web interface Printer added page

9. You have added the printer to CUPS.


Printer on localhost - CUPS v1.1.17 - Mozilla
File Edit View Go Bookmarks Tools Window Help
Back Forward Reload Stop
http://localhost:631/print
Search
Print
Home Bookmarks Red Hat, Inc. Red Hat Network Support Shop Products
ESP Administration Classes Help Jobs Printers Software
Print Test Page
Stop Printer
Reject Jobs
Modify Printer
Configure Printer
Delete Printer
Add Printer
Done

Printer

Default Destination: printer1

printer1 **HP LaserJet Series CUPS v1.1**

Description: Test printer on AD server of domain AD6380
Location: Virtual printer
Printer State: idle, accepting jobs.
Device URI: smb://smb3lab26/printer1

Copyright 1993-2002 Easy Software Products, All Rights Reserved. The Common UNIX Printing System, CUPS, and the CUPS logo are the trademark property of Easy Software Products. All other trademarks are the property of their respective

Figure 7-9 CUPS Web interface Manage Printers'page with printer1 added

The screenshots in this section are created using a Red Hat Desktop client connected to an Active Directory domain AD6380 with domain controller smb3lab26.

To test the printer you added using **`lpadmin`** or the Web interface just print a file to it, or use the Print Test Page button on the Web interface.

Attention: Using a user name and password in the printer creation process will expose your password in the CUPS definition files. It might be best to use a special printer user or make printer shares available for everyone.

Part 4

Appendixes

A

Using enterprise management tools

Here we provide analysis of the management tools provided by two major Linux enterprise distribution vendors

- Red Hat Satellite Server 3.4.0 - Management tool for Red Hat distributions
- ZENworks 6.5 Linux Management - Preferred management tool for Novell/SuSE distributions

In this chapter, the following topics are discussed:

- Why Enterprise management tools?
- Satellite server and Red Hat Network (RHN)
  - Architecture
  - Concepts
  - First impressions
- ZENworks Linux management
  - Architecture
  - Concepts
  - First impressions

Why enterprise management tools

In this chapter we aim to show how useful an enterprise management tool for Linux can be if the number of client systems to manage becomes large. Just like you need to automate certain tasks when using other operating systems, it is also advantageous to do this when using Linux. The two major enterprise distributions, Novell/SuSE and Red Hat, both provide options for automating client management tasks. We describe their enterprise management tools in this appendix.

The tools are designed to manage thousands of clients and hundreds of servers. They can be called enterprise tools because all the Linux systems in an enterprise can be managed by these systems.

Since both tools have a similar but different approach to the problem of managing many systems, we start each "impressions" section on the tool with a description of the architecture behind the tool. We also introduce the terms and concepts used in each tool. Since both tools are themselves based on complex application architectures, we do not go into full detail. This would be a topic for another book. Therefore we only give a first impression glance at the tools that can be used to manage the Linux client in the enterprise. This should give you a good idea of the power of each of the tools.

Although both tools have a Web interface and a command-line interface, we concentrate on the Web-based processes.

Satellite server and Red Hat Network (RHN)

This section describes the Satellite server and its relation to the Red Hat Network. This is the preferred architecture for enterprise customers running a large number of Red Hat Enterprise installations within their organization.

In this section we look at:

- Architecture - Of RHN in general and the Satellite model in particular
- Concepts - The terms used in RHN and Satellite like channel, entitlement, activation key, and others
- First impressions - A first look at the satellite server and how it can be used to manage Linux clients within the organization

Architecture

The architecture of the Red Had Network is one where there is a central server from which all systems are managed. In general this is the RHN server located at Red Hat on the Internet. For small numbers of systems and with low security constraints this is a feasible solution. When the number of systems in an enterprise grows or when security constraints prevent all systems from reaching the Internet, this simple architecture breaks down.

Red Hat uses three models for their solution:

- Hosted model - All systems connect to the RHN server at Red Hat through the organization's firewall.

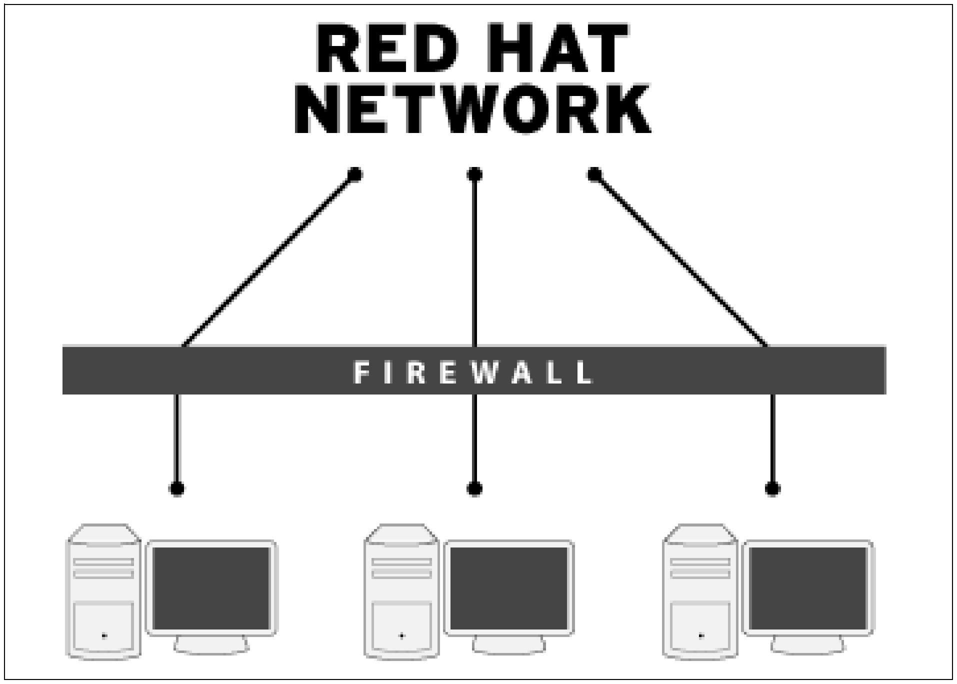


Figure A-1 [1]Red Hat Network - Hosted model

- Proxy model - Systems connect to a proxy server running in the local network, which aggregates data and can perform some functions locally. The proxy server connects with the central RHN servers at Red Hat through the firewall.

[1] Copyright 2004 Red Hat, Inc. All rights reserved. Reprinted by permision.

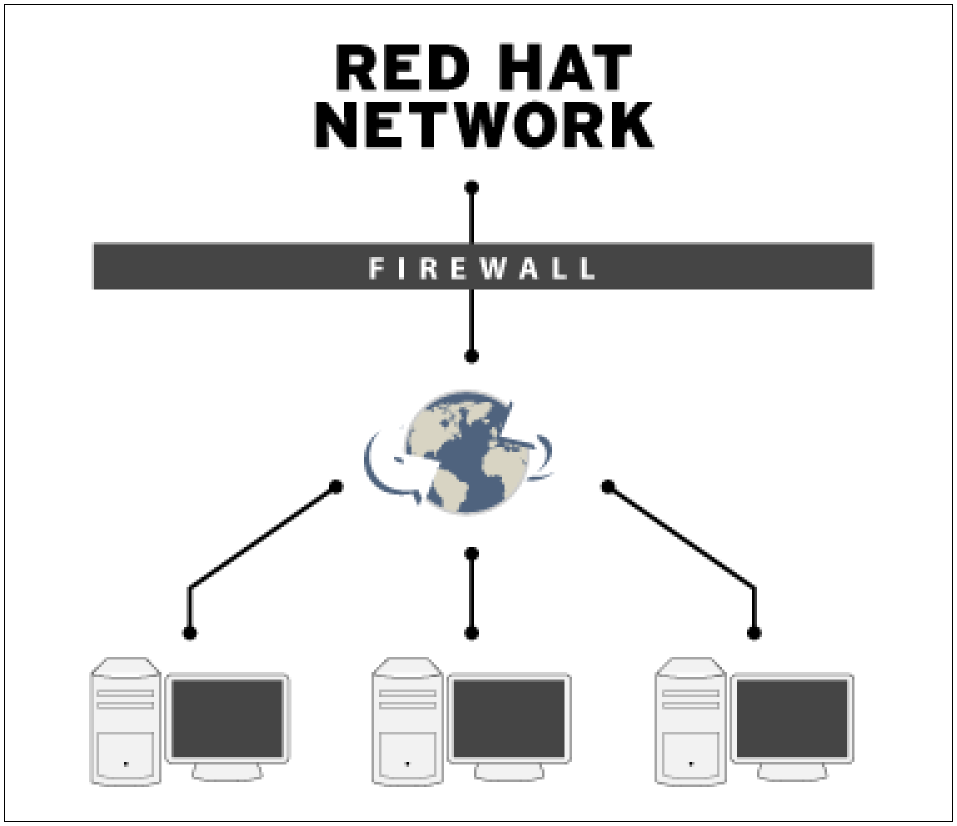


Figure A-2 [2]Red Hat Network - Proxy model

- Satellite model - Systems connect to the satellite server running locally. The satellite server functions as a locally connected RHN update server. It can receive updates for distribution to your network via Internet connection to RHN servers. It can also be loaded with updates manually, using physical media, thus eliminating the need for an external Internet connection for the satellite server. The satellite server performs all functions of the remote RHN server hosted by Red Hat in the Hosted and Proxy models above.

[2] Copyright 2004 Red Hat, Inc. All rights reserved. Reprinted by permision.

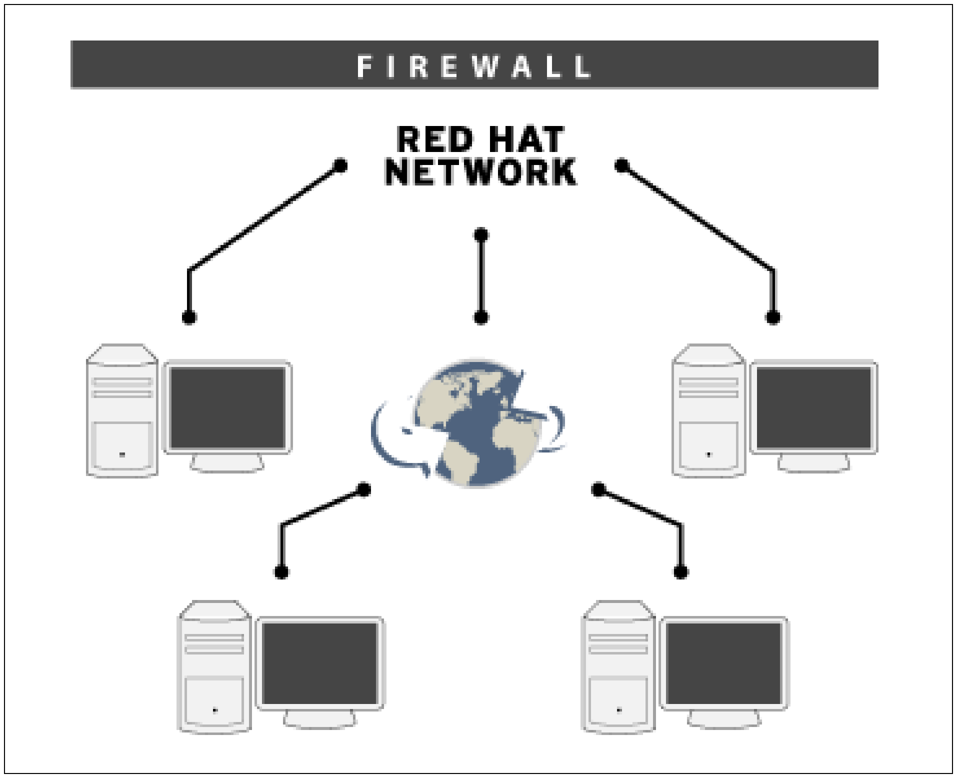


Figure A-3 [3]Red Hat Network - Satellite model

The pictures in figures Figure A-1 on page 167, Figure A-2 on page 168, and Figure A-3 are taken from the Red Hat Web page “Red Hat Network Architecture” at:

http://www.redhat.com/software/rhn/architecture/

In the Proxy and Satellite models it is still possible for systems to connect directly to Red Hat Network at Red Hat, although that is not necessary.

In this section we concentrate on the Satellite model, which is the preferred solution for large numbers of systems.

Technical architecture

In short, the technical architecture of the satellite server consists of a server application on top of a database to contain all the data about system, groups, entitlement, channels, packages, and errata. This application has a management

[3] Copyright 2004 Red Hat, Inc. All rights reserved. Reprinted by permision.

Web interface. The communication with the clients is performed through the `up2date` application and several daemons running on the client.

Concepts

In this section we describe some of the concepts used in the RHN/satellite server. These concepts include:

- System
- System group
- System set
- Entitlement
- Channel
- Errata
- Action
- Activation key

We describe each of these concepts in some detail in the satellite server context. Some of these same terms exist as a concept in ZENworks as well, but might have a slightly different meaning.

System

A system can be any Red Hat Linux installed machine with `up2date` configured and connected to the satellite server. The connection to the satellite server is done using a user ID. This user ID is created in the satellite server and is the initial administrator for the new system.

System group

A system group is an entity in the satellite server that can contain systems. When created, the container is empty. Systems can be added to the system group. The system group is used in the satellite server to perform actions for multiple systems (the ones in the group) simultaneously.

System set

Where the system group is more or less fixed, the system set can be used to construct a selection of systems to perform an action on. These systems can be picked individually or on a group level or based on some property of the system.

Entitlement

The entitlement that a system gets tells the satellite server what can be done for the system, what it is entitled to. In the current Red Hat Network and satellite server there are three levels of entitlement:

- Update - The entry-level entitlement, complimentary with Red Hat Network subscription, most appropriate for single or few systems
- Management - Includes Update entitlement features plus additional features that enable more efficient system grouping, management, and channel definition to support larger numbers of systems
- Provisioning - Includes Management entitlement features plus additional features that provide for full Linux infrastucture life-cycle management

> **Note:** For the most current information about RHN entitlement levels, please refer to the Red Hat Web site:
>
> `http://www.redhat.com/rhn`

With each higher entitlement more functions of the satellite server/RHN become available. And you can change entitlement levels whenever necessary.

Channel

A channel is a collection of updates. There are two sorts of channels: Base channels and child channels.

A base channel consists of a list of packages based on a specific architecture and Red Hat Enterprise Linux release. For example, all the packages in Red Hat Enterprise Linux 2.1 for the x86 architecture are a base channel.

A child channel is a channel associated with a base channel but contains extra packages. This way it is possible for an organization to use a base channel to create a child channel that will contain the base packages and extra packages specifically needed by the organization.

Action

Actions can be performed on clients from the satellite server. Since the rhnd daemon is running as root, even a reboot can be scheduled from the satellite server. Some actions that can be scheduled are:

- Boot
- Package update
- Hardware properties update

Errata

Errata is the name for the collection of all updates and patches. Red Hat distinguishes between three types of errata:

- Security updates
- Bugfix updates
- Enhancement updates

These three sorts of errata each have their own priority for an application. Security updates should be applied immediately, Bugfix updates if necessary, and Enhancement updates when needed.

Activation key

An activation key is used to register the system, entitle the system to a service level, and subscribe the system to specific channels and system groups through the command line utility **`rhnreg_ks`**. These keys can only be created for Management and Provisioning entitlements. The keys are generated in the satellite server.

First impressions

In this section we describe some standard actions in a satellite server. This gives a good first impression of how the tool works and what it can do.

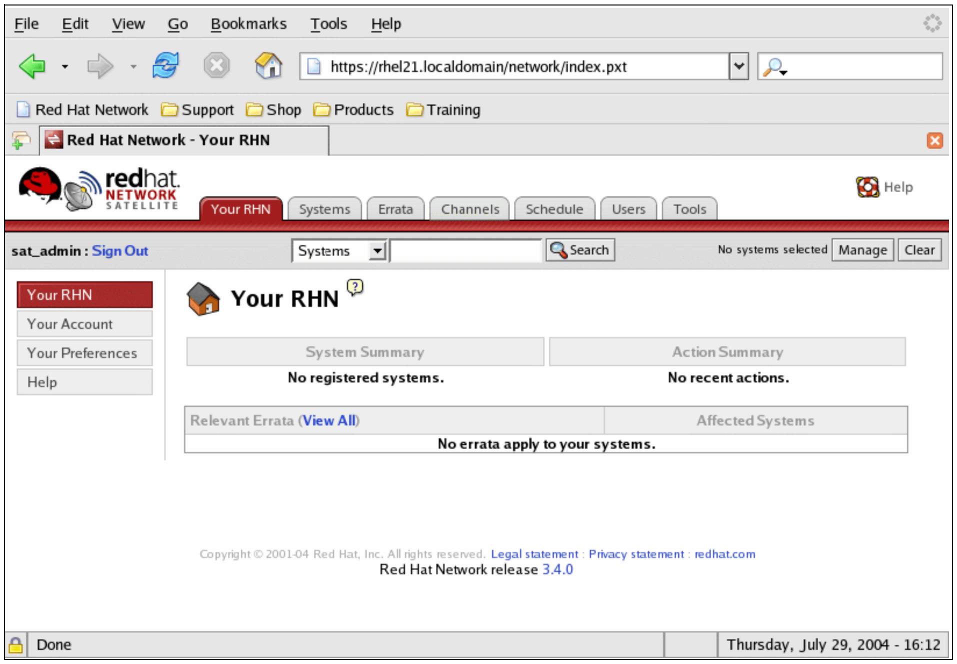
The standard actions we describe are:

- Adding a user in the satellite server
- Connecting a client to the satellite server and configuring the system
- Creating a system group
- Adding a system to a group
- Adding a channel to the server
- Performing a scheduled action on the client system

When logging onto the satellite server using the administrative user, one gets the page shown in Figure A-4 on page 173. This shows tabs at the top that take you to the different areas in the server. These areas are:

- Systems - Here all modifications with respect to systems and system groups are performed.
- Errata - Here packages that are relevant to the user’s systems are listed and can be managed.
- Channels - Here all channels are managed.
- Schedule - All scheduled actions are available here.
- Users - Satellite user management.

- Tools - RHN/satellite internal tools.



Your RHN

System Summary	Action Summary
No registered systems.	No recent actions.

Relevant Errata (View All)	Affected Systems
No errata apply to your systems.	

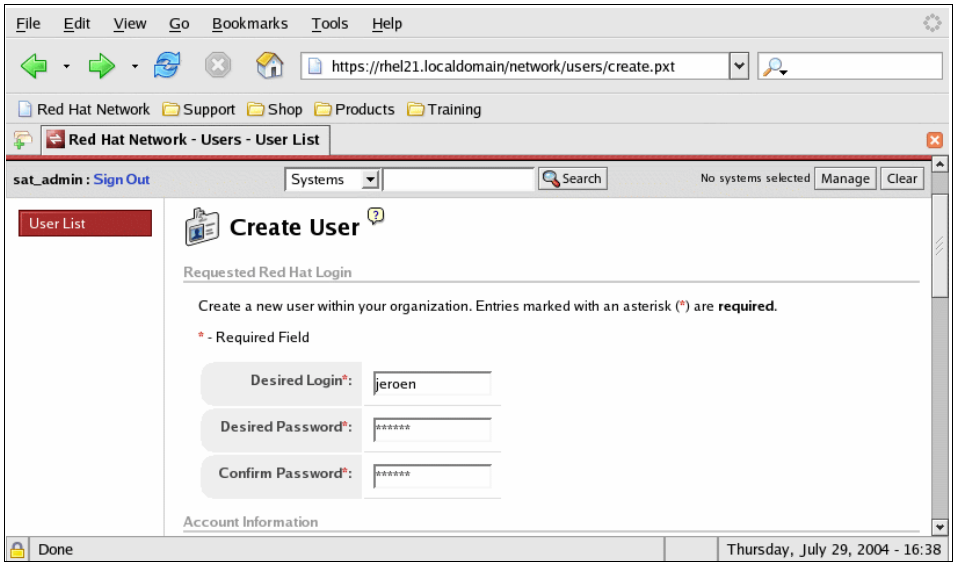
Copyright © 2001-04 Red Hat, Inc. All rights reserved. Legal statement : Privacy statement : redhat.com
Red Hat Network release 3.4.0

Done   Thursday, July 29, 2004 - 16:12

Figure A-4 Initial screen satellite server after logon as admin

Add a user

First go to the Users tab. Initially this page lists the users present in the satellite server. Select the **Create user** link on this page. This gives the page shown in Figure A-5 on page 174. Enter entries as needed and click the **Create login** button. This adds the normal user, as shown in Figure A-6 on page 174.


File Edit View Go Bookmarks Tools Help
https://rhel21.localdomain/network/users/create.pxt
Red Hat Network Support Shop Products Training
Red Hat Network - Users - User List
sat_admin : Sign Out
Systems
Search
No systems selected Manage Clear
User List
Done
Thursday, July 29, 2004 - 16:38

Create User

Requested Red Hat Login

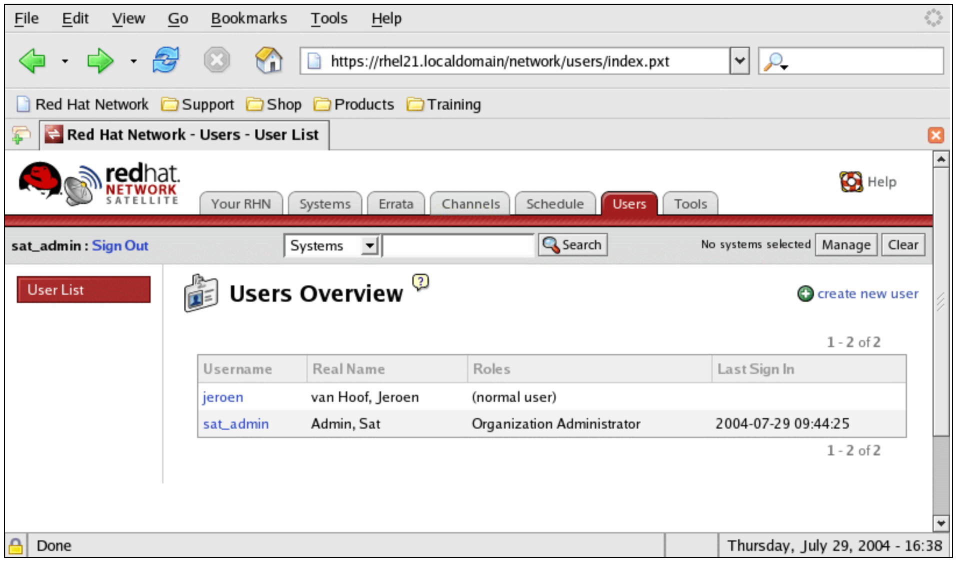
Create a new user within your organization. Entries marked with an asterisk (*) are **required**.

* - Required Field

Desired Login*:	jeroen
Desired Password*:	******
Confirm Password*:	******

Account Information

Figure A-5 Add user page for satellite server


File Edit View Go Bookmarks Tools Help
https://rhel21.localdomain/network/users/index.pxt
Red Hat Network Support Shop Products Training
Red Hat Network - Users - User List
redhat NETWORK SATELLITE
Your RHN Systems Errata Channels Schedule Users Tools
Help
sat_admin : Sign Out
Systems
Search
No systems selected Manage Clear
User List
create new user
Done
Thursday, July 29, 2004 - 16:38

Users Overview

1 - 2 of 2

Username	Real Name	Roles	Last Sign In
jeroen	van Hoof, Jeroen	(normal user)	
sat_admin	Admin, Sat	Organization Administrator	2004-07-29 09:44:25

1 - 2 of 2

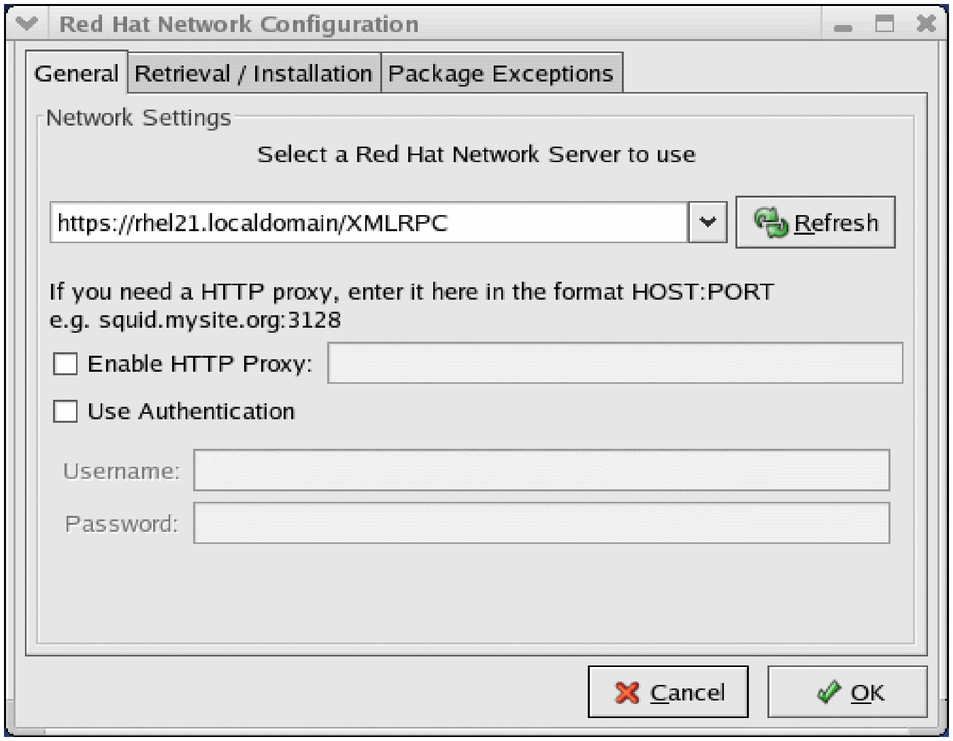
Figure A-6 Screen after user has been added

Connecting a client

To connect the client we start the Red Hat Network Update agent on the client. On first execution this starts the RHN Configuration application. Here we replace the address of the central RHN server at Red Hat with our own satellite server, as shown in Figure A-7. After we OK this application the Red Hat Network Update agent (**up2date**) is started. After the welcome screen we can log onto the satellite server. In Figure A-8 on page 176 we do this using the user we added in “Add a user” on page 173.

In the next screen the profile name for the system is input and indicates how much information up2date can send to the server; see Figure A-9 on page 177. In the next screen, shown in Figure A-10 on page 178, we indicate which installed packages to send to the server. The server needs this information to determine which errata are relevant for this system.

Checking the server, we see in Figure A-11 on page 179 that the system *client1* has been added to the server. This process of connecting a system to the satellite server can be automated using the commands **rhn_register** and **up2date**.



Select a Red Hat Network Server to use

https://rhel21.localdomain/XMLRPC

If you need a HTTP proxy, enter it here in the format HOST:PORT
e.g. squid.mysite.org:3128

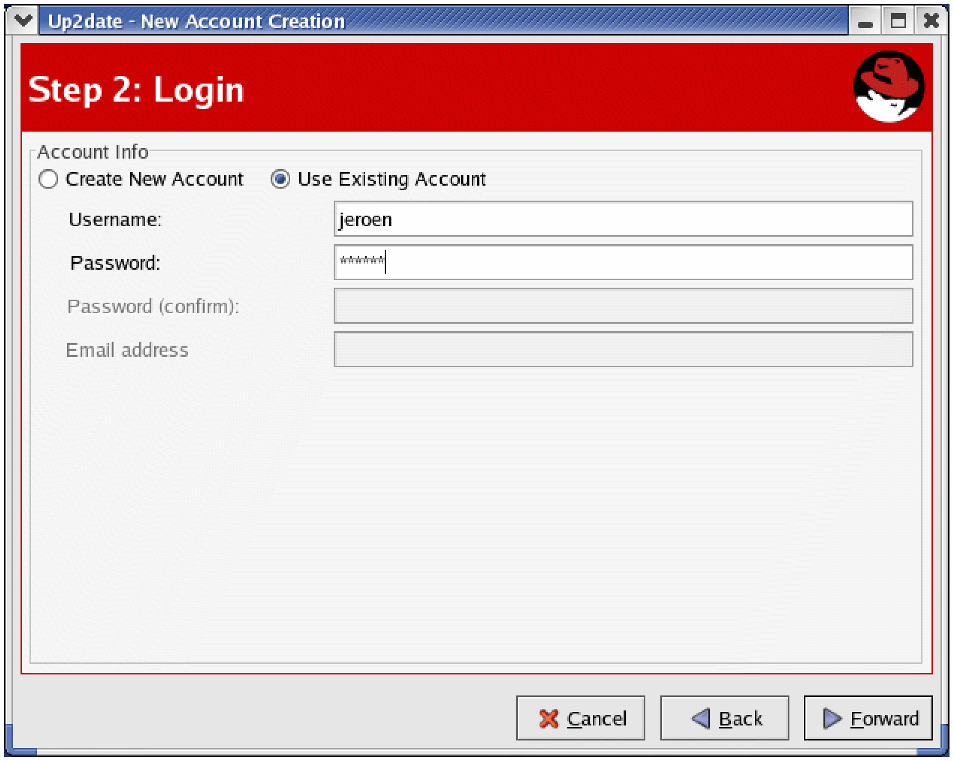
☐ Enable HTTP Proxy:

☐ Use Authentication

Username:

Password:

Figure A-7 Configuration of rhnsd daemon

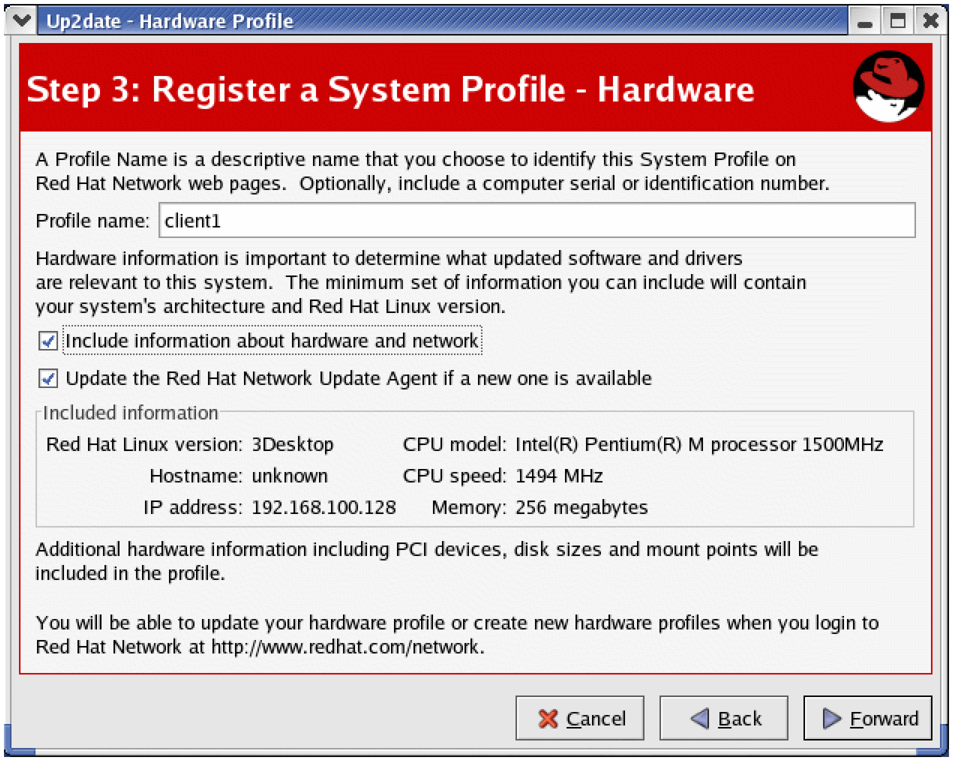


Account Info

( ) Create New Account [✓ Use Existing Account]

Username:	jeroen
Password:	******
Password (confirm):	
Email address	

Figure A-8 Setting the user to connect to satellite server



Step 3: Register a System Profile - Hardware

A Profile Name is a descriptive name that you choose to identify this System Profile on Red Hat Network web pages. Optionally, include a computer serial or identification number.

Profile name: client1

Hardware information is important to determine what updated software and drivers are relevant to this system. The minimum set of information you can include will contain your system's architecture and Red Hat Linux version.

[✓] Include information about hardware and network

[✓] Update the Red Hat Network Update Agent if a new one is available

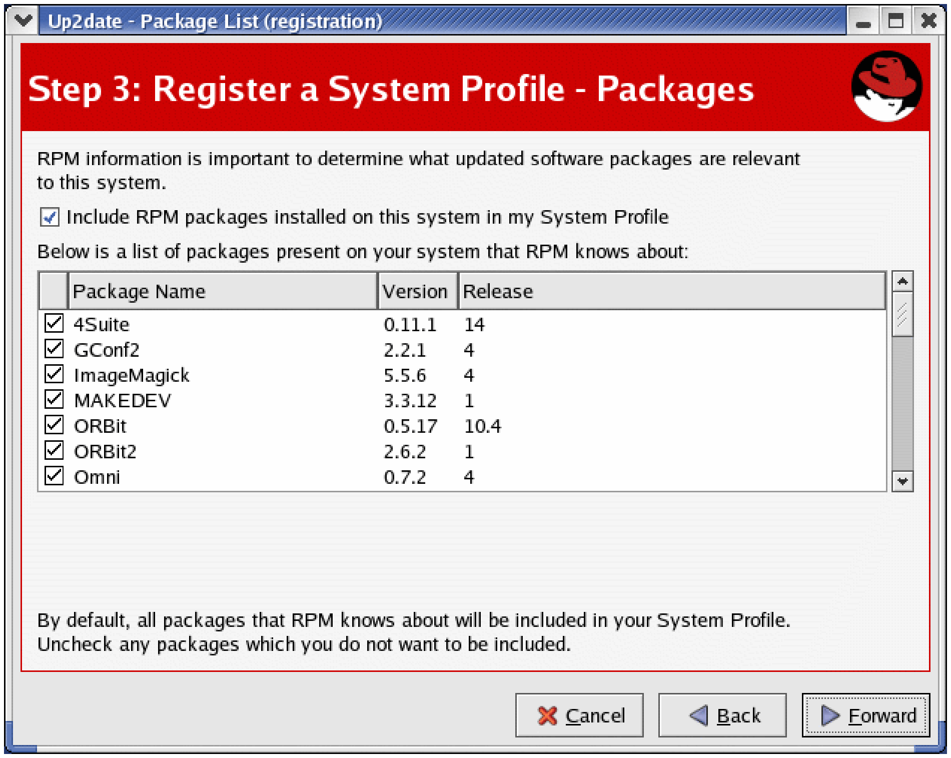
Included information

```
Red Hat Linux version: 3Desktop        CPU model: Intel(R) Pentium(R) M processor 1500MHz
             Hostname: unknown         CPU speed: 1494 MHz
           IP address: 192.168.100.128    Memory: 256 megabytes
```

Additional hardware information including PCI devices, disk sizes and mount points will be included in the profile.

You will be able to update your hardware profile or create new hardware profiles when you login to Red Hat Network at http://www.redhat.com/network.

Figure A-9 Creating a profile to use for the system in the satellite server


Up2date - Package List (registration)
Step 3: Register a System Profile - Packages

RPM information is important to determine what updated software packages are relevant to this system.

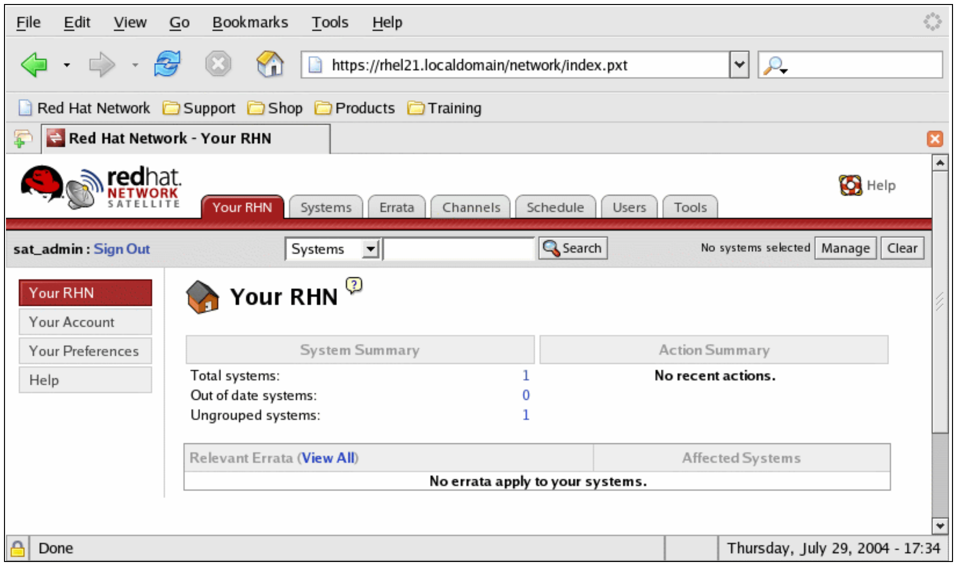
☑ Include RPM packages installed on this system in my System Profile

Below is a list of packages present on your system that RPM knows about:

	Package Name	Version	Release
☑	4Suite	0.11.1	14
☑	GConf2	2.2.1	4
☑	ImageMagick	5.5.6	4
☑	MAKEDEV	3.3.12	1
☑	ORBit	0.5.17	10.4
☑	ORBit2	2.6.2	1
☑	Omni	0.7.2	4

By default, all packages that RPM knows about will be included in your System Profile. Uncheck any packages which you do not want to be included.

Figure A-10 Installed packages to send to the satellite server


File Edit View Go Bookmarks Tools Help
https://rhel21.localdomain/network/index.pxt
Red Hat Network Support Shop Products Training
Red Hat Network - Your RHN
redhat NETWORK SATELLITE
Help
Your RHN Systems Errata Channels Schedule Users Tools
sat_admin : Sign Out
Systems Search No systems selected Manage Clear
Your RHN
Your Account
Your Preferences
Help
Done
Thursday, July 29, 2004 - 17:34

Your RHN

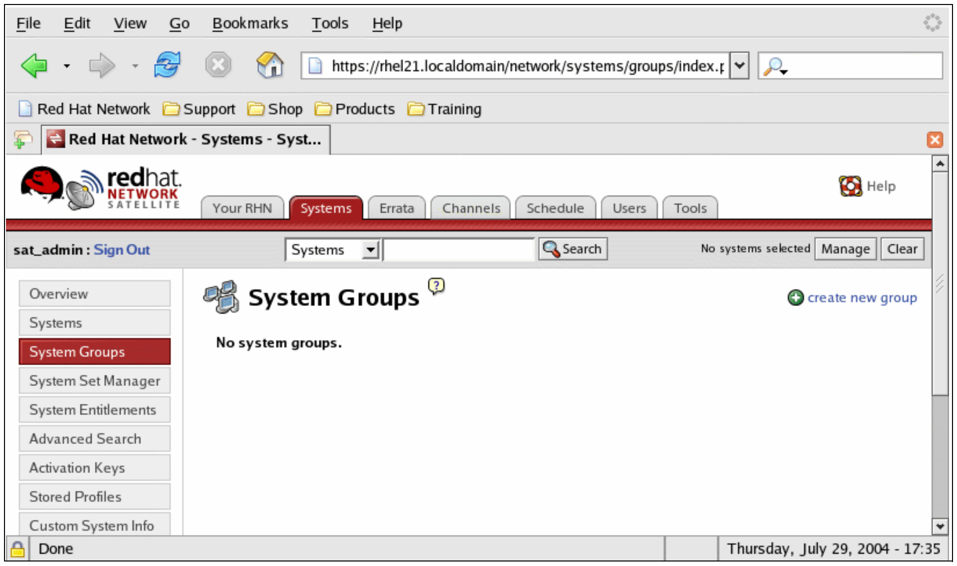
System Summary		Action Summary
Total systems:	1	No recent actions.
Out of date systems:	0	
Ungrouped systems:	1	

Relevant Errata (View All)	Affected Systems
No errata apply to your systems.	

Figure A-11   Client system in satellite server

Create system group

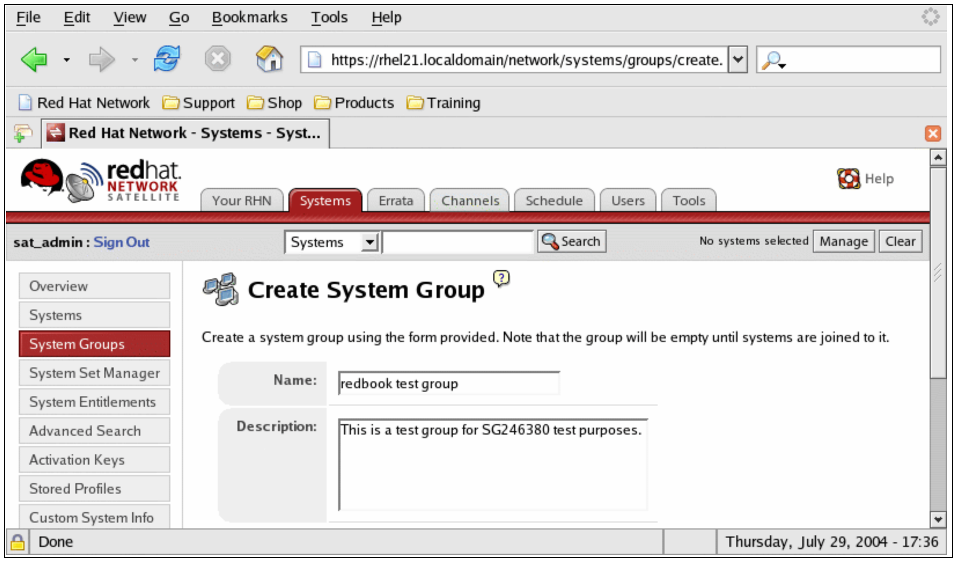
A system group is created using the System groups page shown in Figure A-12. Follow the link **Create new system** and enter the needed information, as shown in Figure A-13 on page 180.


File Edit View Go Bookmarks Tools Help
https://rhel21.localdomain/network/systems/groups/index.p
Red Hat Network Support Shop Products Training
Red Hat Network - Systems - Syst...
redhat NETWORK SATELLITE
Help
Your RHN Systems Errata Channels Schedule Users Tools
sat_admin : Sign Out
Systems Search No systems selected Manage Clear
Overview
Systems
System Groups
System Set Manager
System Entitlements
Advanced Search
Activation Keys
Stored Profiles
Custom System Info
create new group
Done
Thursday, July 29, 2004 - 17:35

System Groups

No system groups.

Figure A-12   System group primary page


File Edit View Go Bookmarks Tools Help
https://rhel21.localdomain/network/systems/groups/create.
Red Hat Network Support Shop Products Training
Red Hat Network - Systems - Syst...
Your RHN Systems Errata Channels Schedule Users Tools Help
sat_admin : Sign Out
Systems Search
No systems selected Manage Clear
Overview
Systems
System Groups
System Set Manager
System Entitlements
Advanced Search
Activation Keys
Stored Profiles
Custom System Info
Done
Thursday, July 29, 2004 - 17:36

Create System Group

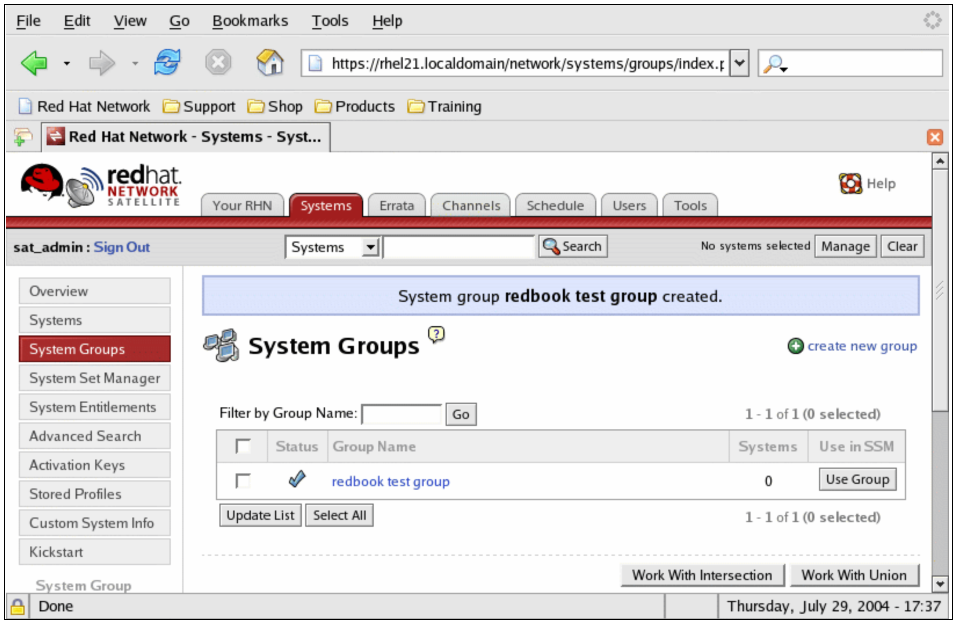
Create a system group using the form provided. Note that the group will be empty until systems are joined to it.

Name: redbook test group

Description: This is a test group for SG246380 test purposes.

Figure A-13 Create a system group page

Figure A-14 shows the new system group added to the list. It shows that currently there are no systems in the group.


File Edit View Go Bookmarks Tools Help
https://rhel21.localdomain/network/systems/groups/index.p
Red Hat Network Support Shop Products Training
Red Hat Network - Systems - Syst...
Your RHN Systems Errata Channels Schedule Users Tools Help
sat_admin : Sign Out
Systems Search
No systems selected Manage Clear
Overview
Systems
System Groups
System Set Manager
System Entitlements
Advanced Search
Activation Keys
Stored Profiles
Custom System Info
Kickstart
System Group
create new group
Go
Update List
Select All
Work With Intersection
Work With Union
Done
Thursday, July 29, 2004 - 17:37

System group **redbook test group** created.

System Groups

Filter by Group Name:

1 - 1 of 1 (0 selected)

	Status	Group Name	Systems	Use in SSM
☐	✓	redbook test group	0	Use Group

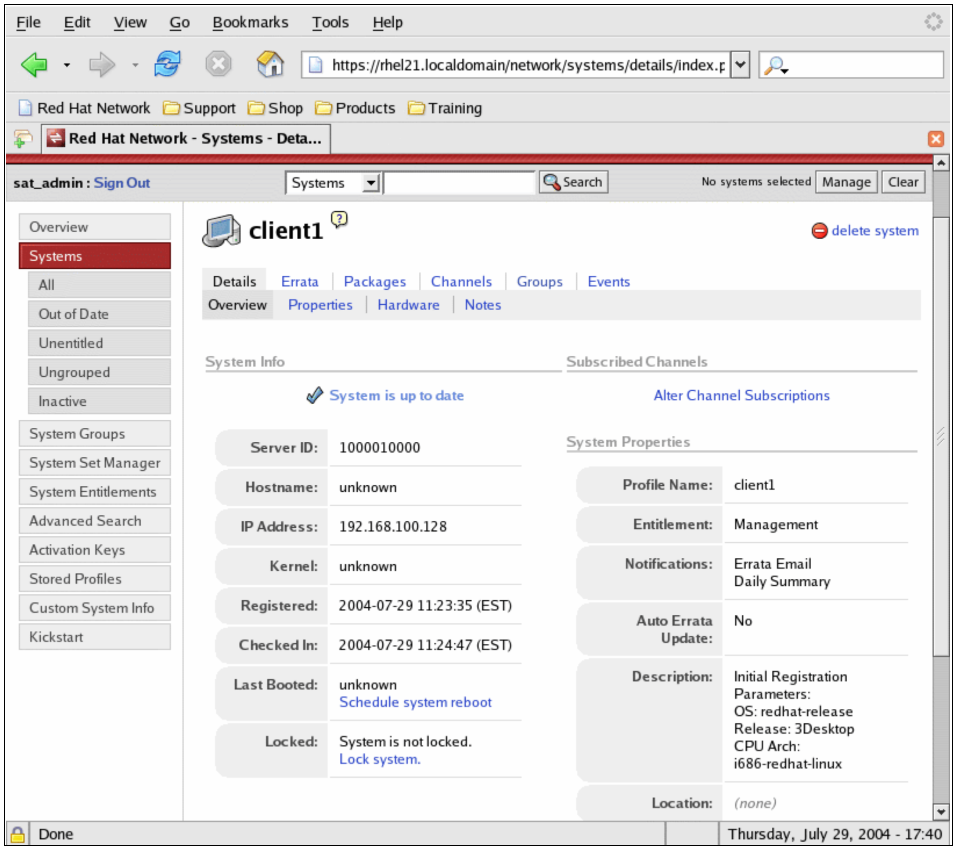
1 - 1 of 1 (0 selected)

Figure A-14 System group primary page with new group added

Add system to group

To add a system to a system group we first go to the Details page for the system like the one shown in Figure A-15 for system client1. Notice the details that are in the server through the registration process, like IP address, OS release, and architecture. On this page are tabs that take you to all data relevant to this system. We choose the Groups tab and get a page like in Figure A-16 on page 182.

By putting the check in the box before the wanted group and clicking the Update Membership button, the system is now part of the group redbook testgroup, as shown in Figure A-17 on page 183.

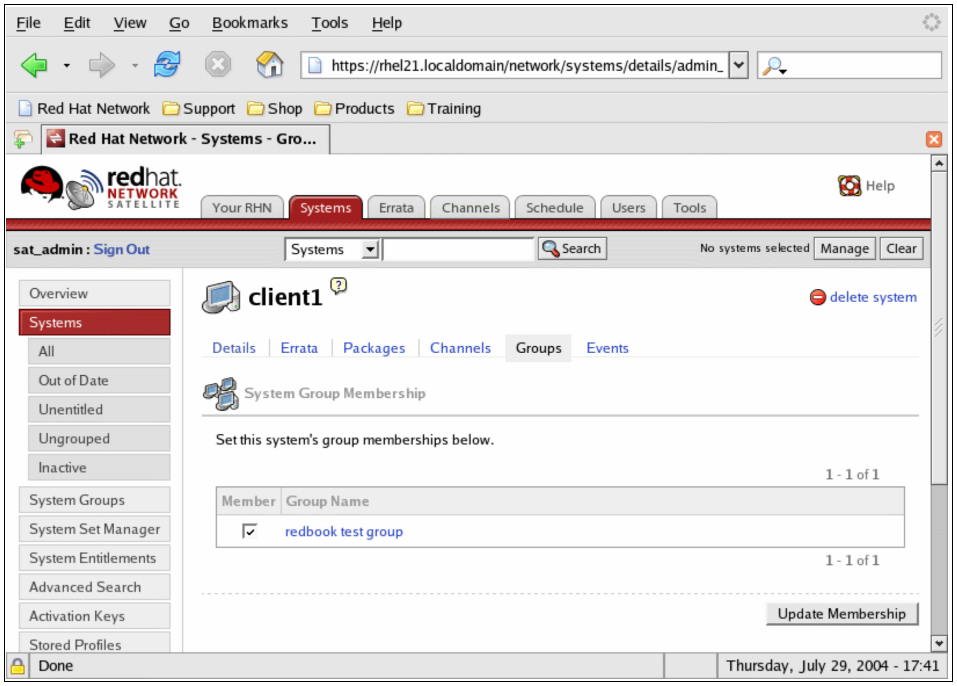


File Edit View Go Bookmarks Tools Help
https://rhel21.localdomain/network/systems/details/index.p
Red Hat Network Support Shop Products Training
Red Hat Network - Systems - Deta...
sat_admin : Sign Out
Systems
Search
No systems selected Manage Clear
Overview
Systems
All
Out of Date
Unentitled
Ungrouped
Inactive
System Groups
System Set Manager
System Entitlements
Advanced Search
Activation Keys
Stored Profiles
Custom System Info
Kickstart
client1
delete system
Details Errata Packages Channels Groups Events
Overview Properties Hardware Notes
System Info
Subscribed Channels
System is up to date
Alter Channel Subscriptions
System Properties
Done
Thursday, July 29, 2004 - 17:40

Server ID:	1000010000
Hostname:	unknown
IP Address:	192.168.100.128
Kernel:	unknown
Registered:	2004-07-29 11:23:35 (EST)
Checked In:	2004-07-29 11:24:47 (EST)
Last Booted:	unknown Schedule system reboot
Locked:	System is not locked. Lock system.

Profile Name:	client1
Entitlement:	Management
Notifications:	Errata Email Daily Summary
Auto Errata Update:	No
Description:	Initial Registration Parameters: OS: redhat-release Release: 3Desktop CPU Arch: i686-redhat-linux
Location:	*(none)*

Figure A-15 Client system primary page


File Edit View Go Bookmarks Tools Help
https://rhel21.localdomain/network/systems/details/admin_
Red Hat Network Support Shop Products Training
Red Hat Network - Systems - Gro...
Your RHN Systems Errata Channels Schedule Users Tools
Help
sat_admin : Sign Out
Systems Search
No systems selected Manage Clear
Overview
Systems
All
Out of Date
Unentitled
Ungrouped
Inactive
System Groups
System Set Manager
System Entitlements
Advanced Search
Activation Keys
Stored Profiles
client1
delete system
Details Errata Packages Channels Groups Events
System Group Membership
Update Membership
Done
Thursday, July 29, 2004 - 17:41

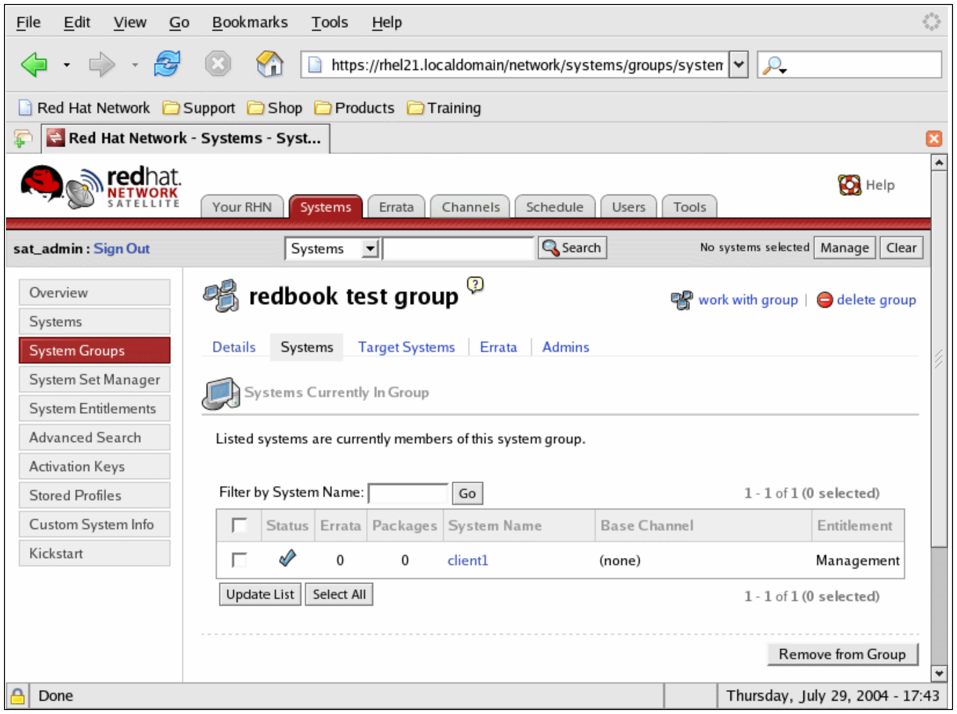
Set this system's group memberships below.

1 - 1 of 1

Member	Group Name
✓	redbook test group

1 - 1 of 1

Figure A-16 Client system group page



File Edit View Go Bookmarks Tools Help
https://rhel21.localdomain/network/systems/groups/systen
Red Hat Network Support Shop Products Training
Red Hat Network - Systems - Syst...
redhat. NETWORK SATELLITE
Your RHN Systems Errata Channels Schedule Users Tools Help
sat_admin : Sign Out
Systems Search
No systems selected Manage Clear
Overview
Systems
System Groups
System Set Manager
System Entitlements
Advanced Search
Activation Keys
Stored Profiles
Custom System Info
Kickstart
redbook test group
work with group | delete group
Details Systems Target Systems Errata Admins
Systems Currently In Group
Filter by System Name: Go
1 - 1 of 1 (0 selected)
Update List Select All
Remove from Group
Done
Thursday, July 29, 2004 - 17:43

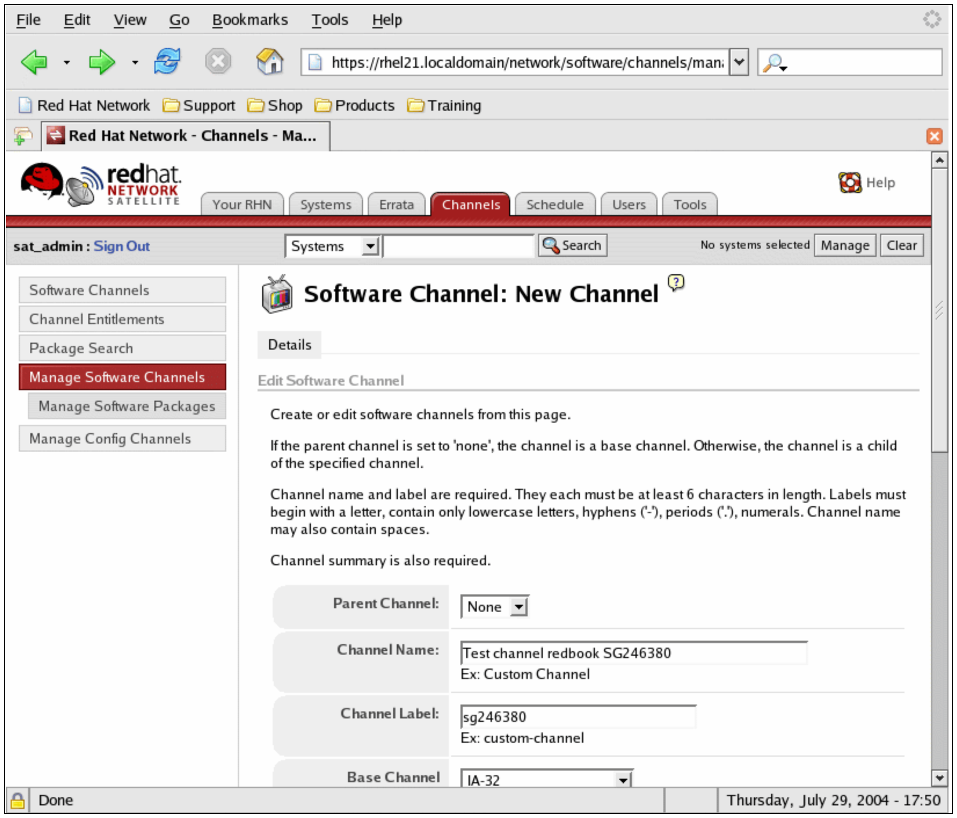
Listed systems are currently members of this system group.

	Status	Errata	Packages	System Name	Base Channel	Entitlement
☐	✓	0	0	client1	(none)	Management

Figure A-17 System group system page with client1 system added

Add a channel

To create a new channel we first go to the Channels area through the tab at the top of the page and choose **Manage Software Channels**. We fill in the relevant information, as shown in Figure A-18 on page 184. Once we create a channel we can use this channel for a system. To do this go back to the system page for the client, use the tab Channels, and set the Channel as shown in Figure A-19 on page 185.



Software Channel: New Channel

Edit Software Channel

Create or edit software channels from this page.

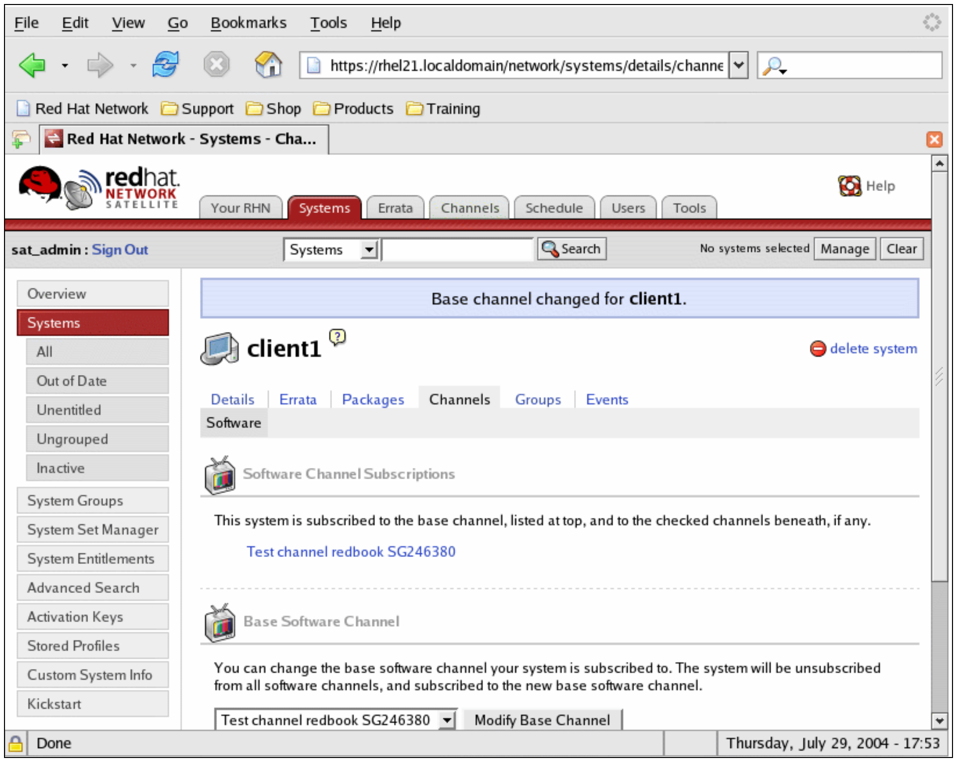
If the parent channel is set to 'none', the channel is a base channel. Otherwise, the channel is a child of the specified channel.

Channel name and label are required. They each must be at least 6 characters in length. Labels must begin with a letter, contain only lowercase letters, hyphens ('-'), periods ('.'), numerals. Channel name may also contain spaces.

Channel summary is also required.

Parent Channel:	None
Channel Name:	Test channel redbook SG246380 Ex: Custom Channel
Channel Label:	sg246380 Ex: custom-channel
Base Channel	IA-32

Figure A-18 New channel input



Base channel changed for **client1**.

Software Channel Subscriptions

This system is subscribed to the base channel, listed at top, and to the checked channels beneath, if any.

Test channel redbook SG246380

Base Software Channel

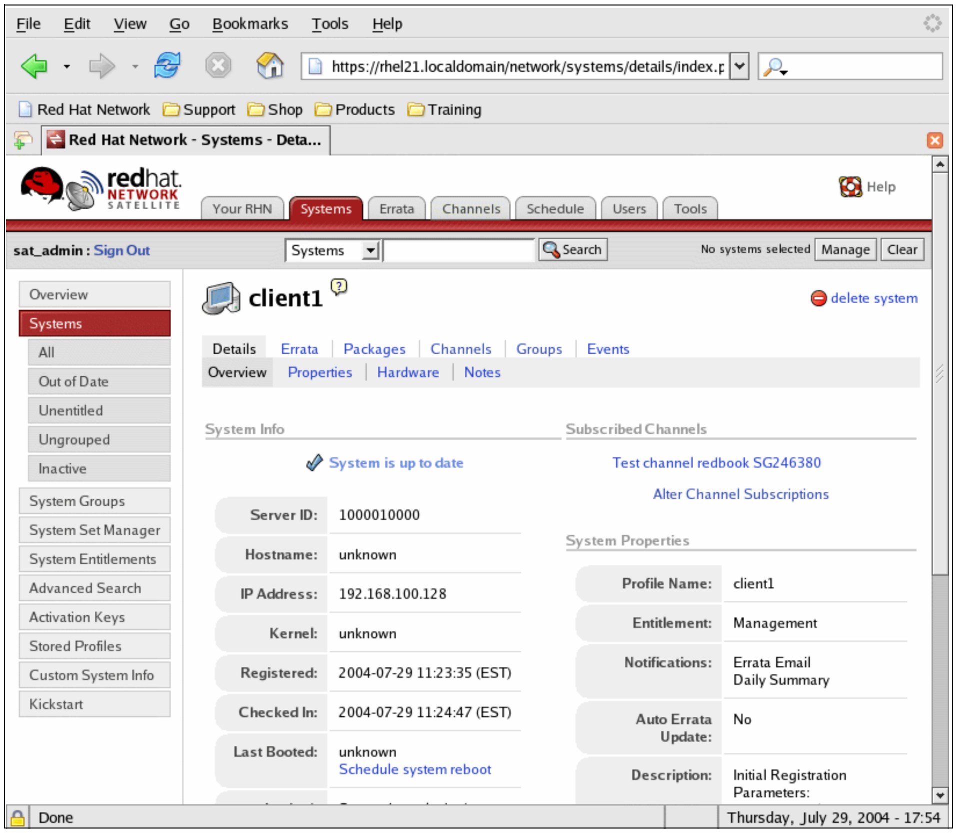
You can change the base software channel your system is subscribed to. The system will be unsubscribed from all software channels, and subscribed to the new base software channel.

Figure A-19 Client1 system channels page with new base channel

Schedule an action

As an example we schedule a reboot of system client1. To do this go to the system page for client1, as shown in Figure A-20 on page 186. Notice that the channel “Test channel redbook SG246380” is now the base channel for client1. To reboot the system we can use the “Schedule system reboot” link at the bottom of the page. This link takes us to the page where we can set the time for the reboot. This page is shown in Figure A-21 on page 187. Once the action is scheduled, it shows up in the pending scheduled actions page, as shown in Figure A-22 on page 188.

The client gets the command from the server the next time the **`rhnsd`** daemon connects. This process can be triggered by hand by the **`rhn_check`** command. After performing this action on client1, we see the client rebooting, as shown in Figure A-23 on page 189. Afterwards the action is visible in the Completed actions list on the server, Figure A-24 on page 190.



System Info

System is up to date

Server ID:	1000010000
Hostname:	unknown
IP Address:	192.168.100.128
Kernel:	unknown
Registered:	2004-07-29 11:23:35 (EST)
Checked In:	2004-07-29 11:24:47 (EST)
Last Booted:	unknown Schedule system reboot

Subscribed Channels

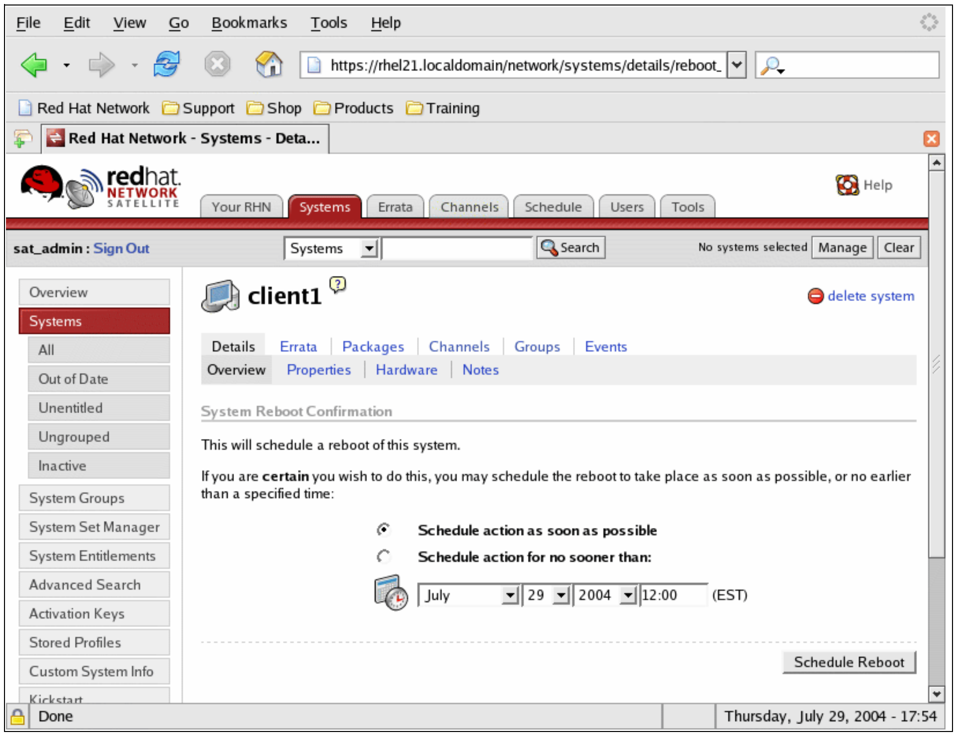
Test channel redbook SG246380

Alter Channel Subscriptions

System Properties

Profile Name:	client1
Entitlement:	Management
Notifications:	Errata Email Daily Summary
Auto Errata Update:	No
Description:	Initial Registration Parameters:

Figure A-20 Client1 primary page



File Edit View Go Bookmarks Tools Help
https://rhel21.localdomain/network/systems/details/reboot_
Red Hat Network Support Shop Products Training
Red Hat Network - Systems - Deta...
redhat. NETWORK SATELLITE
Your RHN Systems Errata Channels Schedule Users Tools Help
sat_admin : Sign Out
Systems Search No systems selected Manage Clear
Overview
Systems
All
Out of Date
Unentitled
Ungrouped
Inactive
System Groups
System Set Manager
System Entitlements
Advanced Search
Activation Keys
Stored Profiles
Custom System Info
Kickstart
client1
delete system
Details Errata Packages Channels Groups Events
Overview Properties Hardware Notes
Schedule Reboot
Done
Thursday, July 29, 2004 - 17:54

System Reboot Confirmation

This will schedule a reboot of this system.

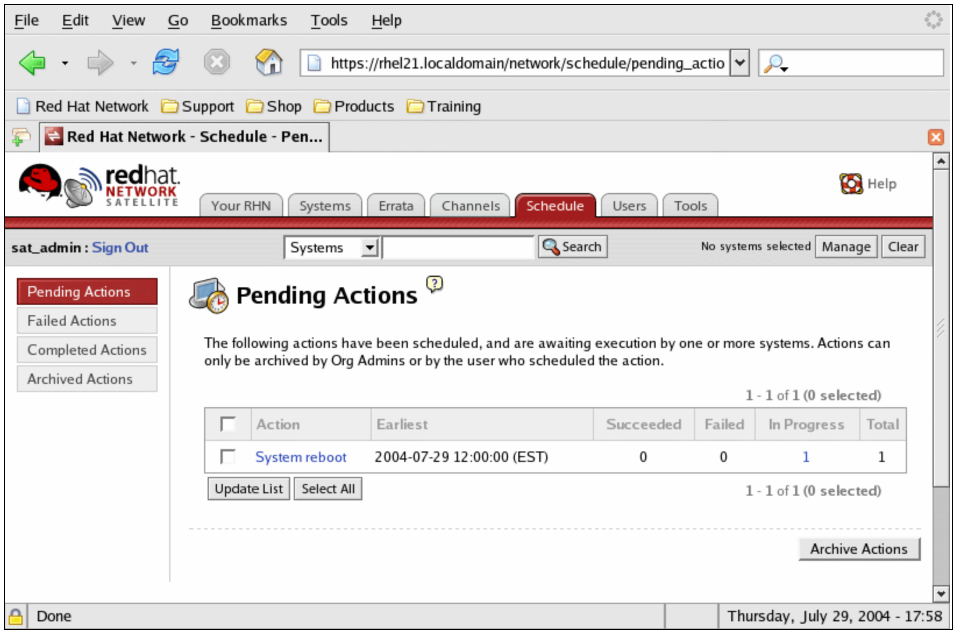
If you are **certain** you wish to do this, you may schedule the reboot to take place as soon as possible, or no earlier than a specified time:

[✓ **Schedule action as soon as possible**]

Schedule action for no sooner than:

July 29 2004 12:00 (EST)

Figure A-21 Client1 system reboot schedule page


File Edit View Go Bookmarks Tools Help
https://rhel21.localdomain/network/schedule/pending_actio
Red Hat Network Support Shop Products Training
Red Hat Network - Schedule - Pen...
Your RHN Systems Errata Channels Schedule Users Tools
Help
sat_admin : Sign Out
Systems Search
No systems selected Manage Clear
Pending Actions
Failed Actions
Completed Actions
Archived Actions
Update List Select All
Archive Actions
Done
Thursday, July 29, 2004 - 17:58

Pending Actions

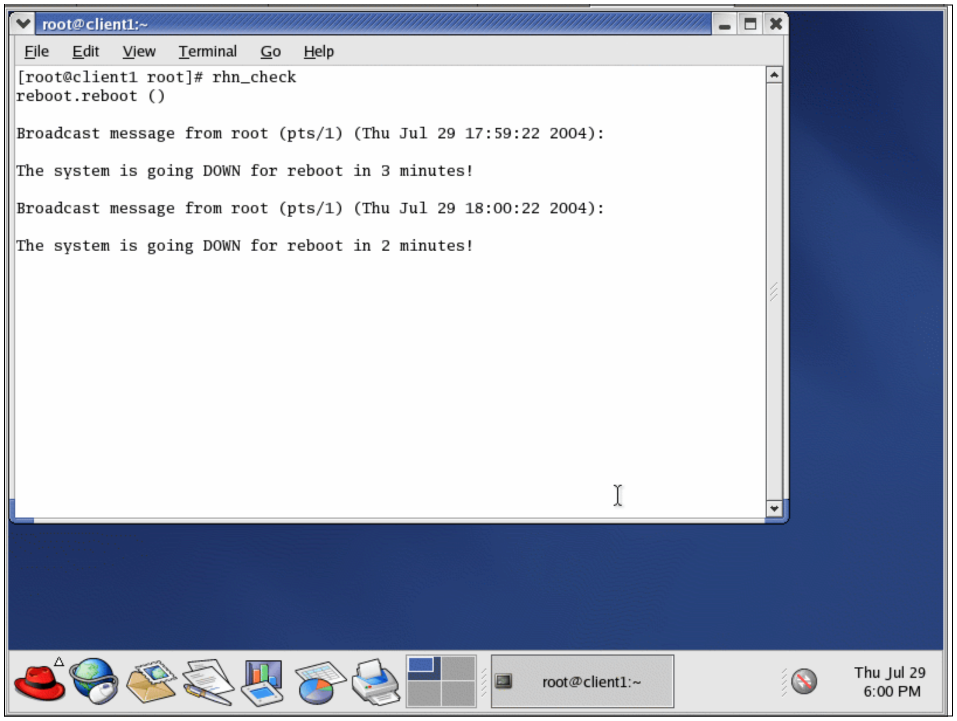
The following actions have been scheduled, and are awaiting execution by one or more systems. Actions can only be archived by Org Admins or by the user who scheduled the action.

1 - 1 of 1 (0 selected)

	Action	Earliest	Succeeded	Failed	In Progress	Total
☐	System reboot	2004-07-29 12:00:00 (EST)	0	0	1	1

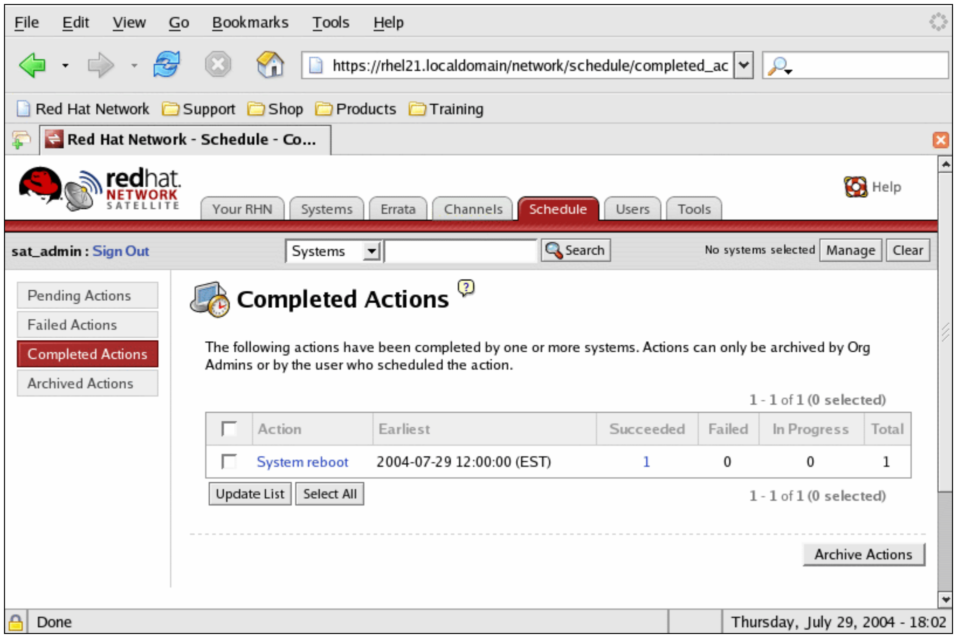
1 - 1 of 1 (0 selected)

Figure A-22 Schedule page pending actions



```
[root@client1 root]# rhn_check
reboot.reboot ()

Broadcast message from root (pts/1) (Thu Jul 29 17:59:22 2004):

The system is going DOWN for reboot in 3 minutes!

Broadcast message from root (pts/1) (Thu Jul 29 18:00:22 2004):

The system is going DOWN for reboot in 2 minutes!
```

Figure A-23 Client1 system gets reboot signal through rhnsd daemon



Completed Actions

The following actions have been completed by one or more systems. Actions can only be archived by Org Admins or by the user who scheduled the action.

1 - 1 of 1 (0 selected)

	Action	Earliest	Succeeded	Failed	In Progress	Total
	System reboot	2004-07-29 12:00:00 (EST)	1	0	0	1

Update List | Select All

1 - 1 of 1 (0 selected)

Archive Actions

Figure A-24   Completed actions page

ZENworks Linux management

This section describes ZENworks Linux management. In Version 6.5 of this application Novell has managed to start to integrate the Red Carpet Enterprise application (obtained through Ximian aquisition) into the Novell ZENworks framework. Novell positions this new addition to ZENworks as the preferred application for enterprise customers running a large number of Linux installations within their organization.

ZENworks Linux management is the preferred tool for Novell/SuSE distribution, but can handle any Linux distribution that can run the **`rcd`** daemon and is rpm base.

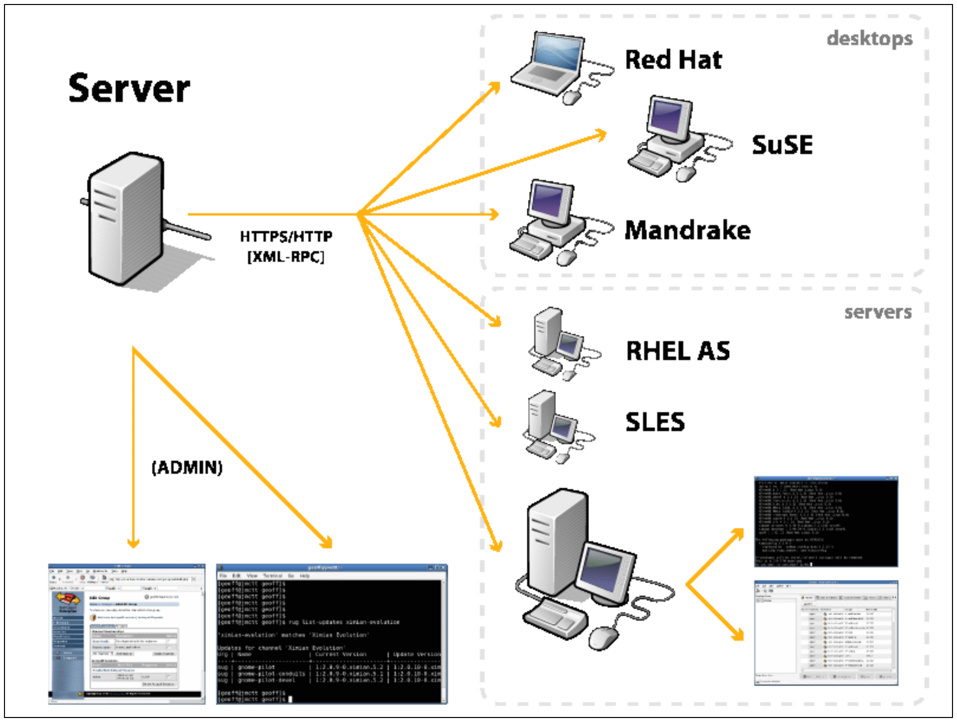
In this section we look at:

- Architecture - Essentially still the Red Carpet Enterprise architecture
- Concepts - The terms used in ZENworks Linux management like channel, machine, group, machine set, activation key, and others
- First impressions - A first look at ZENworks Linux management and how it can be used to manage Linux clients within the organization

Architecture

The system architecture of the ZENworks Linux management server is shown in Figure A-25. The server consists of an application running on top of a database and using a http/https interface to communicate. The http/https interface runs the Web interface for management on one side and is used by the client to communicate with the server. The server can also be managed through the **`rcman`** command-line interface.

The client runs the **`rcd`** daemon. This daemon can be controlled by either the **`rug`** command-line interface or the **`red-carpet`** GUI application.



```
[geoff@jmcit geoff]$
[geoff@jmcit geoff]$
[geoff@jmcit geoff]$
[geoff@jmcit geoff]$
[geoff@jmcit geoff]$
[geoff@jmcit geoff]$
[geoff@jmcit geoff]$ rug list-updates ximian-evolution

'ximian-evolution' matches 'Ximian Evolution'

Updates for channel 'Ximian Evolution'
Urg | Name                   | Current Version        | Update Version
----+------------------------+------------------------+----------------
sup | gnome-pilot            | 1:2.0.9-0.ximian.5.2 | 1:2.0.10-0.xim
sup | gnome-pilot-conduits   | 1:2.0.9-0.ximian.5.1 | 1:2.0.10-0.xim
sup | gnome-pilot-devel      | 1:2.0.9-0.ximian.5.2 | 1:2.0.10-0.xim

[geoff@jmcit geoff]$
[geoff@jmcit geoff]$
```

Figure A-25 ZENworks Linux management architecture[4]

Concepts

In this section we describe some of the concepts used in the RHN/satellite server. These concepts include:

- Machine

[4] Reproduced with permission from "*ZENworks Linux Management 6.5 Administrator's Guide*" (http://www.novell.com/products/zenworks/linuxmanagement/)

- Group
- Machine set
- Channel
- Transaction
- Activation key

We describe each of these concepts in some detail in the ZENworks Linux management context. Some of these same terms exist as a concept in the satellite server as well (see “Concepts” on page 170), but might have a slightly different meaning.

Machine
Machine is the term used for a client system. The ZENworks server collects a lot of information about the system through the rcd daemon. This information can be used to search machines, but also to create sets to perform transactions on.

Group
A group of machines is a collection of machines that can share certain properties, like administrators or channels. The group is used to segment an enterprise into smaller entities, for example, to limit the scope of administrators.

Machine set
The machine set is like a group. Only one set exists and it is populated from all machines by hand or by a search on properties of the machines. The machine set can contain all machines with a certain graphics card, or all machines needing a specific update and not having installed it yet. The set is a very powerfull mechanism to perform a change accross the enterprise.

Channel
A channel is a collection of RPM packages or package sets for a specific distribution and/or architecture. The package set is a collection of packages that is handled by ZENworks as a single package. It can have a version and dependencies.

Transaction
Transactions are generally updates that can be performed on machines, groups, or the machine set. The transactions can be scheduled. A new feature in ZENworks 6.5, for example, is the ability to do a “dry-run” of a transaction on the server to decrease the load on client machines.

Activation key

The activation key is the primary mechanism to register machines in ZENworks. The key is generated and certain properties are given to the key, like groups, channels, and administrators. When a machine is activated using this key it gets all these properties automatically in ZENworks. So the activation key used to activate the client more or less sets its role in the organization.

First impressions

In this section we describe some standard actions in the ZENworks Linux management server. This gives a good first impression of how the tool works and what it can do.

The standard actions we describe are:

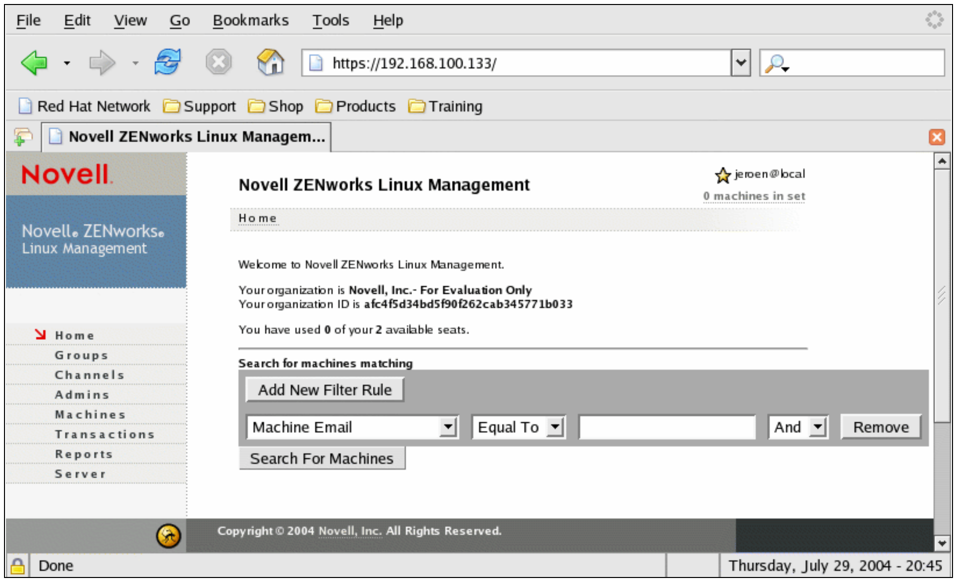
- Creating a group
- Adding a channel with packages
- Creating an administrator
- Adding an activation key
- Adding a client to ZENworks using Red Carpet

On first login using the administrator account, the Web page of ZENworks Linux management looks like Figure A-26 on page 194. These tests were performed using an evalution copy of ZENworks 6.5 Linux management, obtained from:

```
http://www.novell.com/products/zenworks/eval.html
```

This is a limited version but all functionality is present.

The different areas of the server are reached through navigation links on the left side of the page. These links are present on all pages to enable quick navigation through the Web-enabled administration of the ZENworks Linux management server.



Novell ZENworks Linux Management

Home

Welcome to Novell ZENworks Linux Management.

Your organization is **Novell, Inc.- For Evaluation Only**
Your organization ID is **afc4f5d34bd5f90f262cab345771b033**

You have used **0** of your **2** available seats.

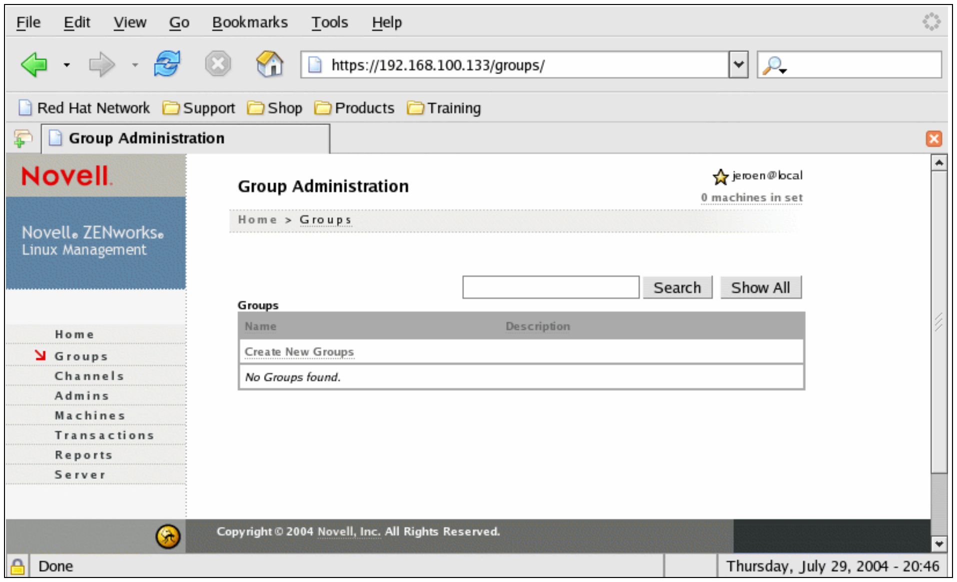
Search for machines matching

Copyright © 2004 Novell, Inc. All Rights Reserved.

Figure A-26 ZENworks Linux management initial screen

Creating a group

A group is created by going to the Group administration page, as shown in Figure A-27 on page 195. After selecting **Create New Groups**, the necessary information is entered on the next page, as shown in Figure A-28 on page 195. When saving this page the Edit page for the new group is reached, shown in Figure A-29 on page 196. Through the tabs on this page it is possible to edit all properties of the group, for example, administrators or channel permissions, as shown in Figure A-30 on page 197.

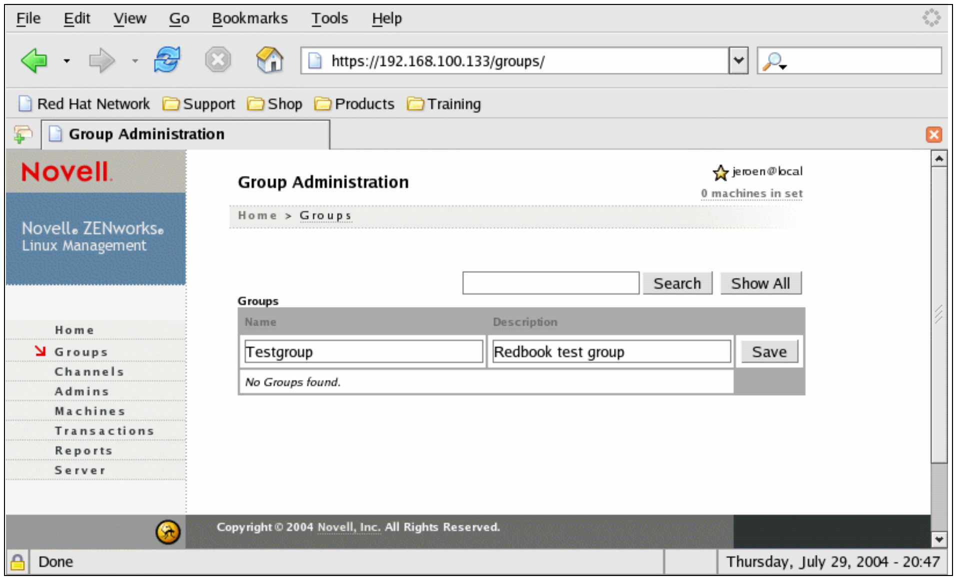


Group Administration

Groups

Name	Description
Create New Groups	
No Groups found.	

Figure A-27 Group administration inital page

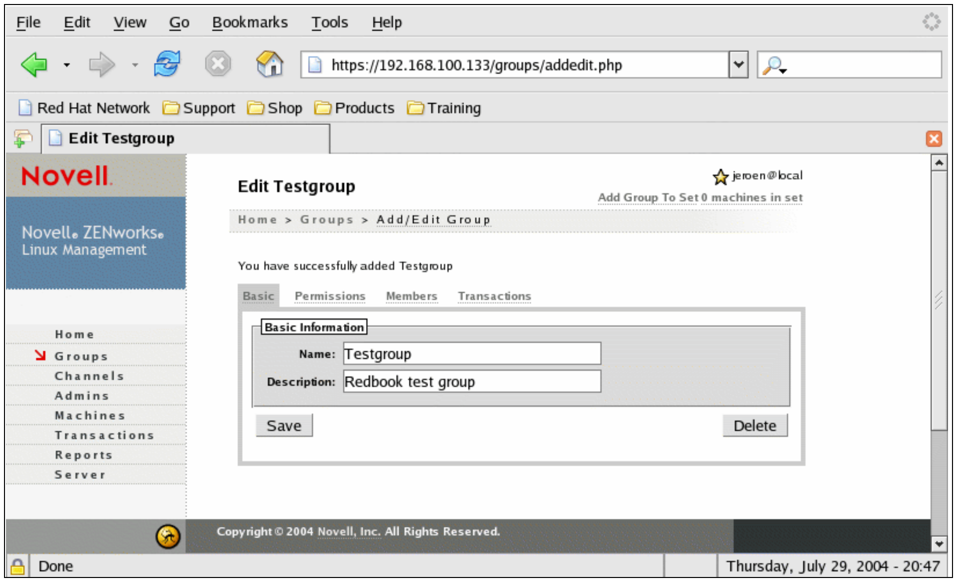


Group Administration

Groups

Name	Description
Testgroup	Redbook test group
No Groups found.	

Figure A-28 Group administration add a group page


File Edit View Go Bookmarks Tools Help
https://192.168.100.133/groups/addedit.php
Red Hat Network
Support
Shop
Products
Training
Edit Testgroup
Novell
Novell ZENworks Linux Management
Home
Groups
Channels
Admins
Machines
Transactions
Reports
Server
jeroen@local
Add Group To Set 0 machines in set
Home > Groups > Add/Edit Group
Basic
Permissions
Members
Transactions
Save
Delete
Copyright © 2004 Novell, Inc. All Rights Reserved.
Done
Thursday, July 29, 2004 - 20:47

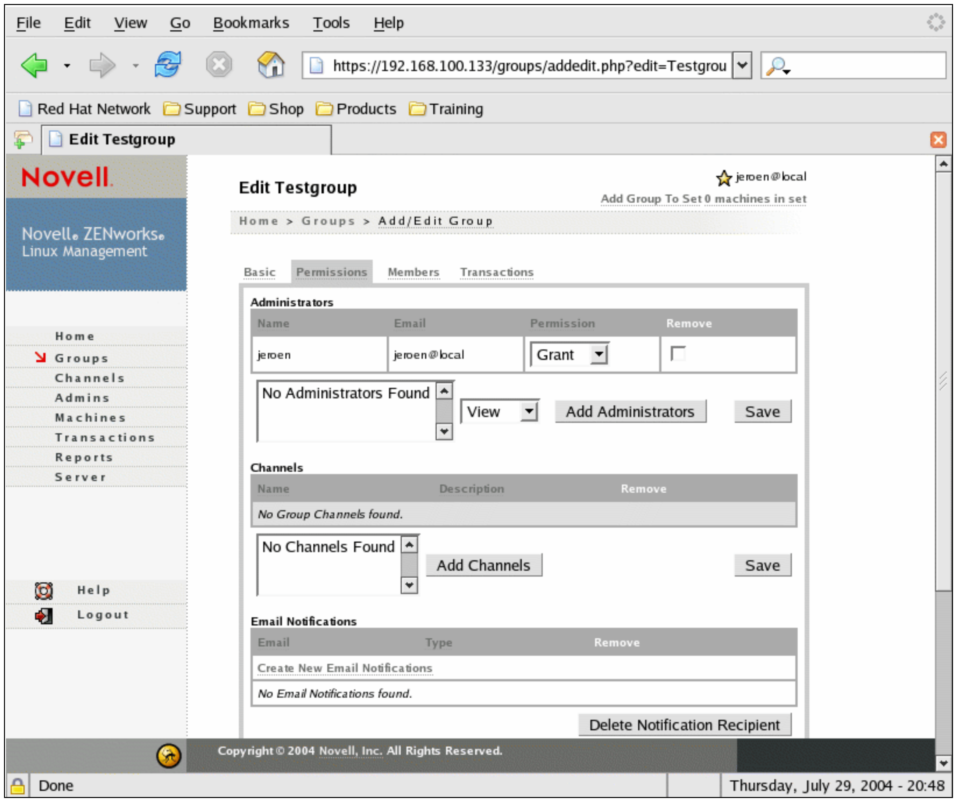
Edit Testgroup

You have successfully added Testgroup

Basic Information

Name:	Testgroup
Description:	Redbook test group

Figure A-29 Group administration edit group page


File Edit View Go Bookmarks Tools Help
https://192.168.100.133/groups/addedit.php?edit=Testgrou
Red Hat Network Support Shop Products Training
Edit Testgroup
Novell.
Novell. ZENworks. Linux Management
Home
Groups
Channels
Admins
Machines
Transactions
Reports
Server
Help
Logout
Edit Testgroup
jeroen@local
Add Group To Set 0 machines in set
Home > Groups > Add/Edit Group
Basic Permissions Members Transactions
Administrators

Name	Email	Permission	Remove
jeroen	jeroen@local	Grant	☐

No Administrators Found
View
Add Administrators
Save
Channels

Name	Description	Remove
No Group Channels found.		

No Channels Found
Add Channels
Save
Email Notifications

Email	Type	Remove
Create New Email Notifications		
No Email Notifications found.		

Delete Notification Recipient
Copyright © 2004 Novell, Inc. All Rights Reserved.
Done
Thursday, July 29, 2004 - 20:48

Figure A-30 Group administration edit group permissions page

Adding a channel with packages

To work with channels we go to the Channel administration page, Figure A-31 on page 198. Here we click the **Create New Channels** link and on the next page enter the Name, Description, and Alias for the new channel; see Figure A-32 on page 198. When saving the new channel we come to the Edit page for this channel, as shown in Figure A-33 on page 199.

To add a package to the channel we click the **Add Package** link. Saving this page will add the package for the selected platforms to the channel. See Figure A-34 on page 200 and Figure A-35 on page 201.

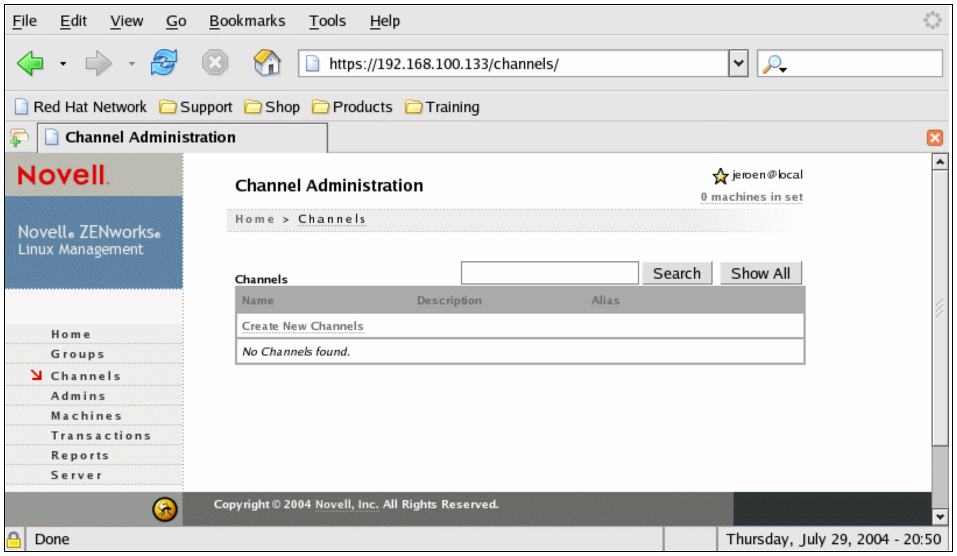
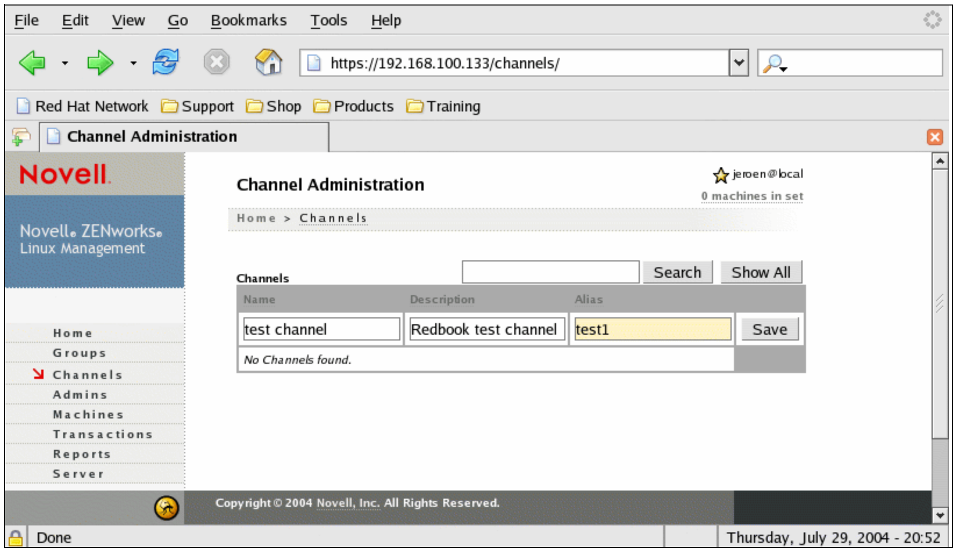
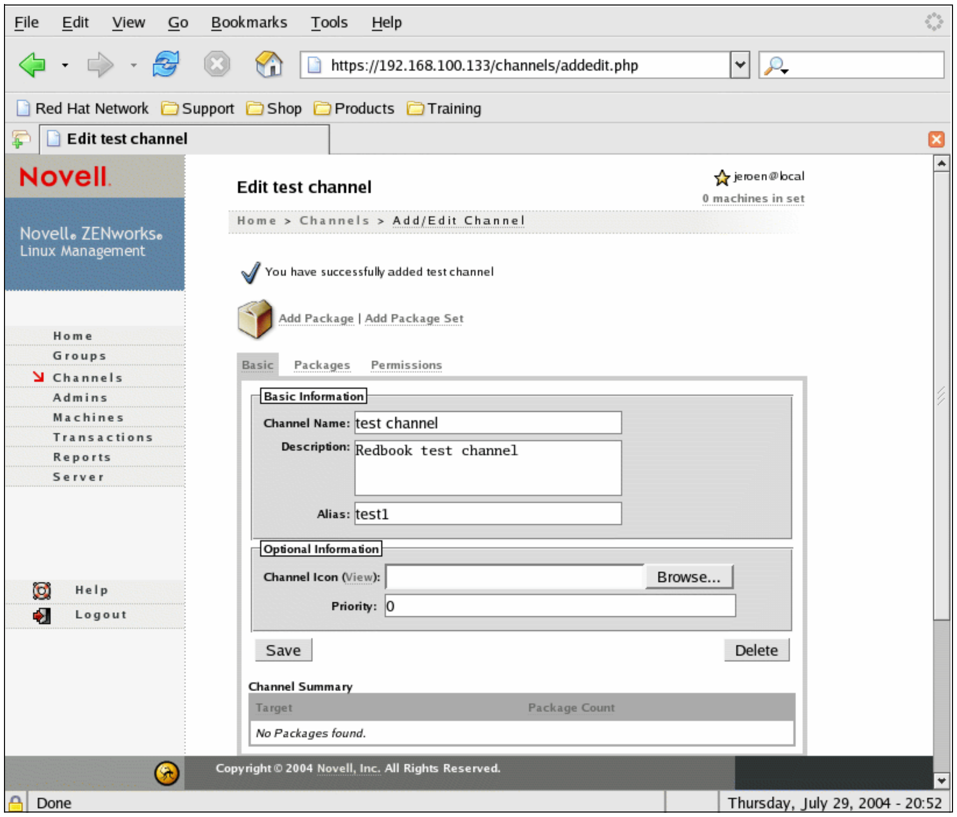


Figure A-31 Channel administration initial page



Name	Description	Alias
test channel	Redbook test channel	test1

Figure A-32 Channel administration add channel page


File Edit View Go Bookmarks Tools Help
https://192.168.100.133/channels/addedit.php
Red Hat Network
Support
Shop
Products
Training
Edit test channel
Novell
Novell ZENworks Linux Management
Home
Groups
Channels
Admins
Machines
Transactions
Reports
Server
Help
Logout
jeroen@local
0 machines in set
Home > Channels > Add/Edit Channel
Add Package | Add Package Set
Basic
Packages
Permissions
Browse...
Save
Delete
Done
Thursday, July 29, 2004 - 20:52

Edit test channel

You have successfully added test channel

Basic Information

Channel Name: test channel

Description: Redbook test channel

Alias: test1

Optional Information

Channel Icon (View):

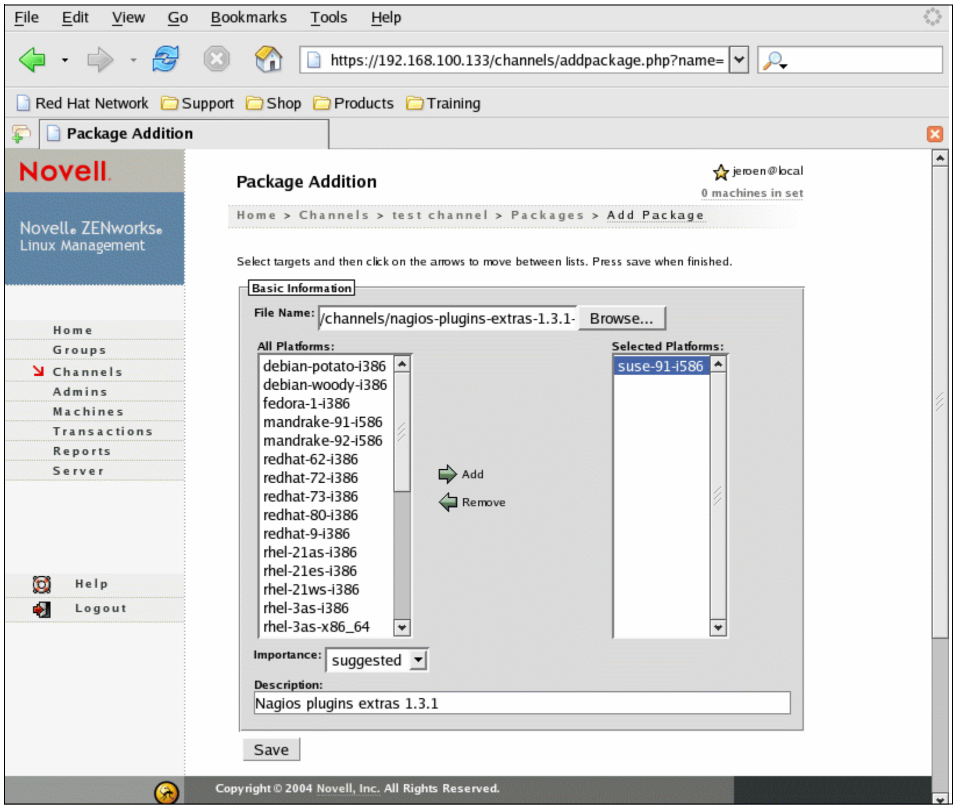
Priority: 0

Channel Summary

Target	Package Count
No Packages found.	

Copyright © 2004 Novell, Inc. All Rights Reserved.

Figure A-33 Channel administration edit channel page



Package Addition

jeroen@local
0 machines in set

Home > Channels > test channel > Packages > Add Package

Select targets and then click on the arrows to move between lists. Press save when finished.

Basic Information

File Name: /channels/nagios-plugins-extras-1.3.1- Browse...

All Platforms:
debian-potato-i386
debian-woody-i386
fedora-1-i386
mandrake-91-i586
mandrake-92-i586
redhat-62-i386
redhat-72-i386
redhat-73-i386
redhat-80-i386
redhat-9-i386
rhel-21as-i386
rhel-21es-i386
rhel-21ws-i386
rhel-3as-i386
rhel-3as-x86_64

Add
Remove

Selected Platforms:
suse-91-i586

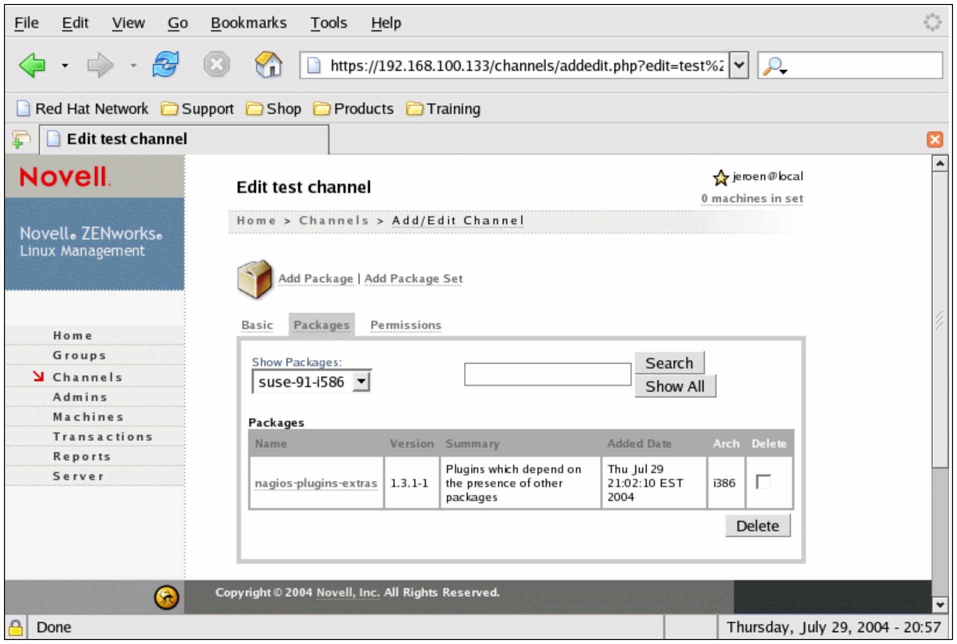
Importance: suggested

Description:
Nagios plugins extras 1.3.1

Save

Copyright © 2004 Novell, Inc. All Rights Reserved.

Figure A-34 Channel administration add package page


File Edit View Go Bookmarks Tools Help
https://192.168.100.133/channels/addedit.php?edit=test%2
Red Hat Network Support Shop Products Training
Edit test channel
Novell
Novell ZENworks Linux Management
Home
Groups
Channels
Admins
Machines
Transactions
Reports
Server
jeroen@local
0 machines in set
Home > Channels > Add/Edit Channel
Add Package | Add Package Set
Basic Packages Permissions
Show Packages:
suse-91-i586
Search
Show All
Delete
Copyright © 2004 Novell, Inc. All Rights Reserved.
Done
Thursday, July 29, 2004 - 20:57

Edit test channel

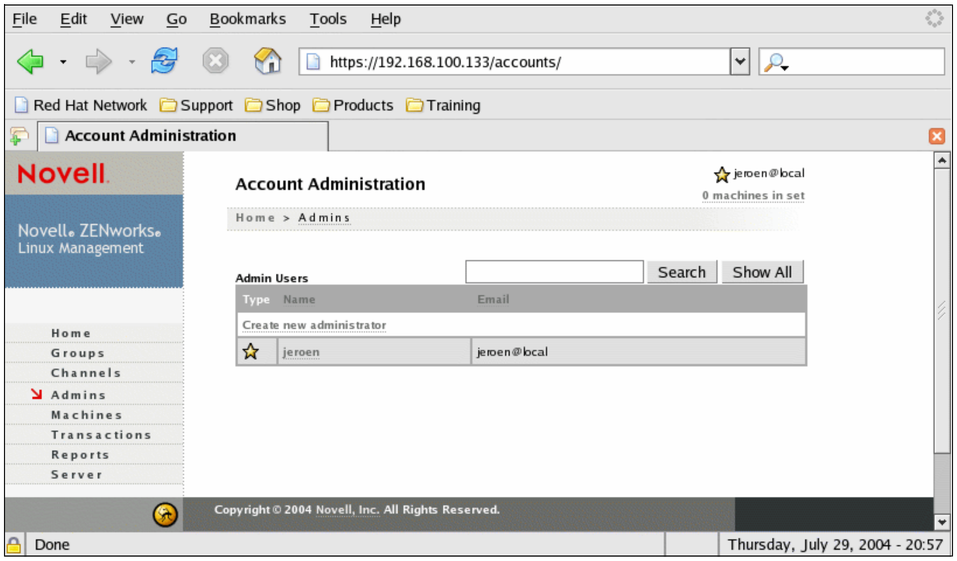
Packages

Name	Version	Summary	Added Date	Arch	Delete
nagios-plugins-extras	1.3.1-1	Plugins which depend on the presence of other packages	Thu Jul 29 21:02:10 EST 2004	i386	☐

Figure A-35 Channel administration edit channel packages page

Create an administrator

First go to the Account administration page shown in Figure A-36 on page 202. Here we follow the link **Create new administrator** and fill in the needed information, as in Figure A-37 on page 202. Saving this information shows us the edit page (shown in Figure A-38 on page 203), where groups and channels to administer can be added to the new administrator.



Account Administration

jeroen@local
0 machines in set

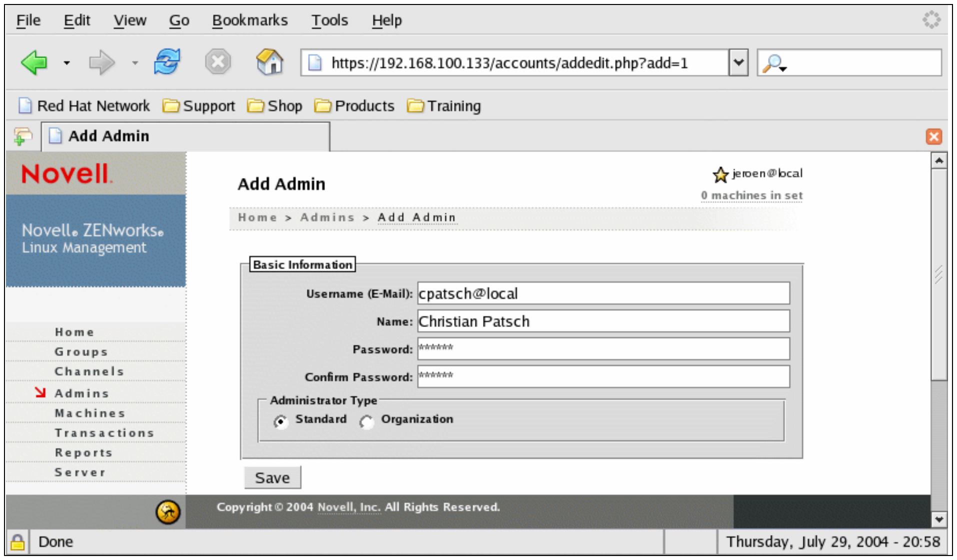
Home > Admins

Admin Users

Type	Name	Email
Create new administrator		
☆	jeroen	jeroen@local

Copyright © 2004 Novell, Inc. All Rights Reserved.

Figure A-36 Account administration initial page


File Edit View Go Bookmarks Tools Help
https://192.168.100.133/accounts/addedit.php?add=1
Red Hat Network
Support
Shop
Products
Training
Add Admin
Novell.
Novell. ZENworks. Linux Management
Home
Groups
Channels
Admins
Machines
Transactions
Reports
Server
Done
Thursday, July 29, 2004 - 20:58

Add Admin

jeroen@local
0 machines in set

Home > Admins > Add Admin

Basic Information

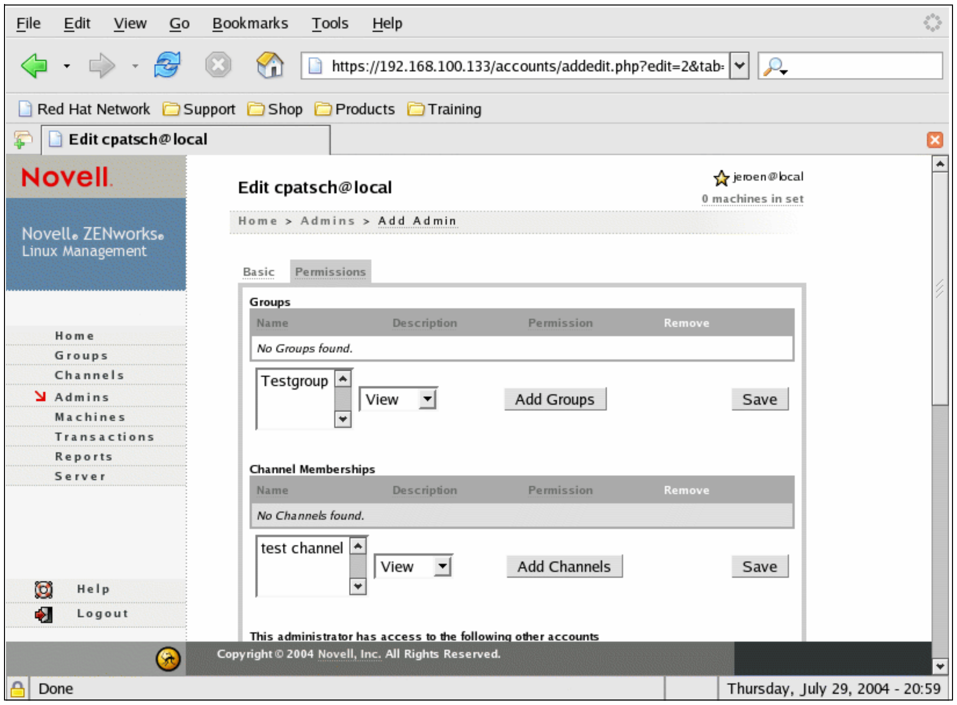
Username (E-Mail): cpatsch@local
Name: Christian Patsch
Password: ******
Confirm Password: ******

Administrator Type
[✓ Standard] Organization

Save

Copyright © 2004 Novell, Inc. All Rights Reserved.

Figure A-37 Account administration add admin page


File Edit View Go Bookmarks Tools Help
https://192.168.100.133/accounts/addedit.php?edit=2&tab=
Red Hat Network Support Shop Products Training
Edit cpatsch@local
Novell
Novell. ZENworks. Linux Management
Home
Groups
Channels
Admins
Machines
Transactions
Reports
Server
Help
Logout
jeroen@local
0 machines in set
Basic
Permissions
Add Groups
Save
Add Channels
Save
Copyright © 2004 Novell, Inc. All Rights Reserved.
Done
Thursday, July 29, 2004 - 20:59

Edit cpatsch@local

Home > Admins > Add Admin

Groups

Name	Description	Permission	Remove
No Groups found.			

Testgroup — View

Channel Memberships

Name	Description	Permission	Remove
No Channels found.			

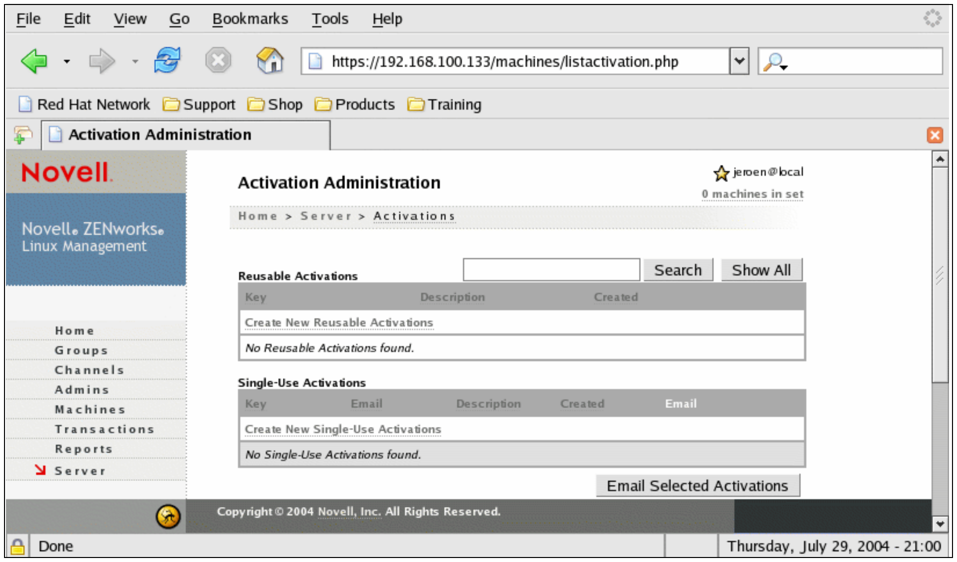
test channel — View

This administrator has access to the following other accounts

Figure A-38 Account administration edit admin permissions page

Add an activation key

The activation code pages are not mentioned in the navigation menu. To get there you go to Server, and on that page choose Activation codes. You then get the Activation administration page shown in Figure A-39 on page 204. Here you can create either a reusable or a single-use activation. The single-use activation becomes invalid after one machine has been activated using that code. The reusable activation key can be used many times and is the ideal way to activate a collection of systems with the same function. Following the Create New Reusable Activations link we come to a page like that shown in Figure A-40 on page 204. Entering the description and saving generates a value for the key. On the Edit page for the activation key we can set the groups, channels, and administrators that are going to be set for every machine using this key to activate, as shown in Figure A-41 on page 205.



https://192.168.100.133/machines/listactivation.php

Activation Administration

jeroen@local
0 machines in set

Home > Server > Activations

Reusable Activations

Key	Description	Created
Create New Reusable Activations		
No Reusable Activations found.		

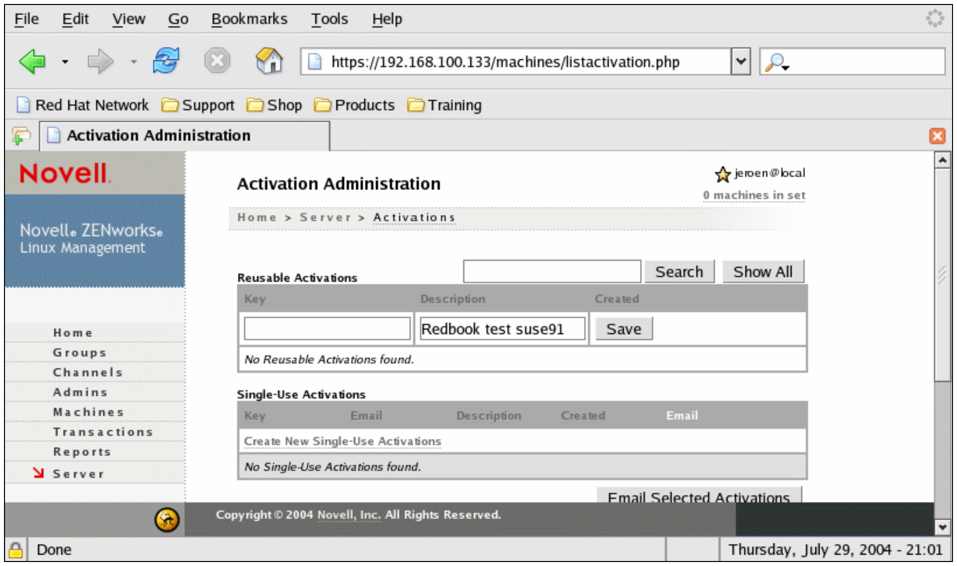
Single-Use Activations

Key	Email	Description	Created	Email
Create New Single-Use Activations				
No Single-Use Activations found.				

Copyright © 2004 Novell, Inc. All Rights Reserved.

Thursday, July 29, 2004 - 21:00

Figure A-39   Activation administration initial page



https://192.168.100.133/machines/listactivation.php

Activation Administration

jeroen@local
0 machines in set

Home > Server > Activations

Reusable Activations

Key	Description	Created
	Redbook test suse91	Save
No Reusable Activations found.		

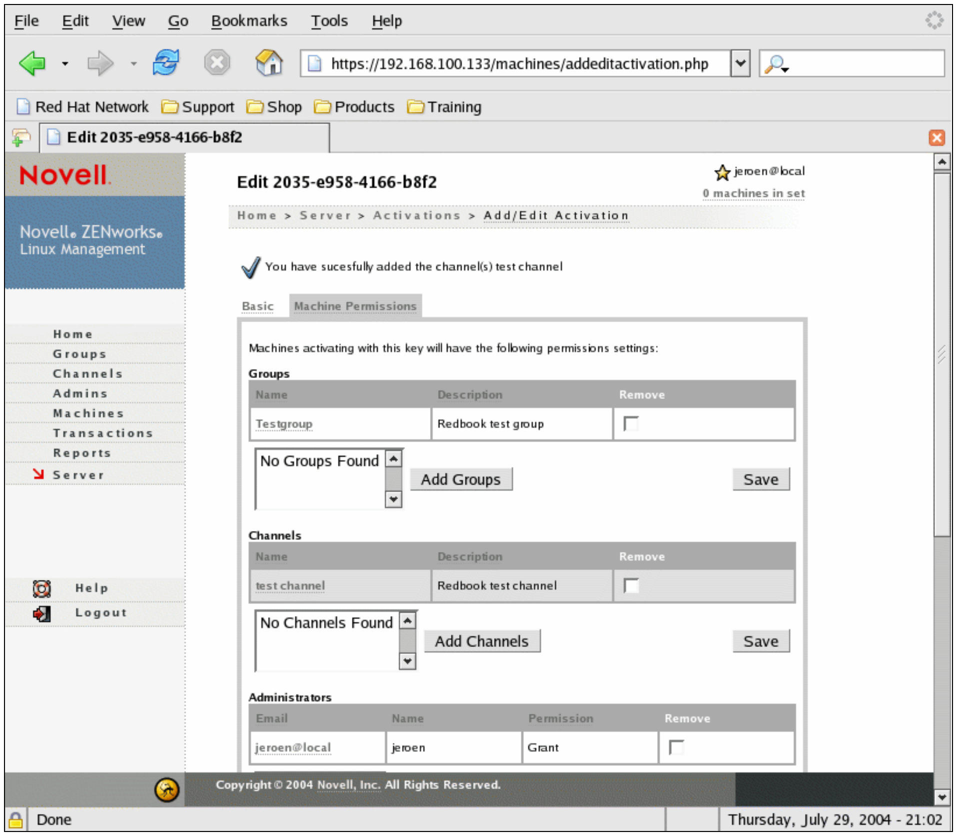
Single-Use Activations

Key	Email	Description	Created	Email
Create New Single-Use Activations				
No Single-Use Activations found.				

Copyright © 2004 Novell, Inc. All Rights Reserved.

Thursday, July 29, 2004 - 21:01

Figure A-40   Activation administration add activation page



File Edit View Go Bookmarks Tools Help
https://192.168.100.133/machines/addeditactivation.php
Red Hat Network
Support
Shop
Products
Training
Edit 2035-e958-4166-b8f2
Novell.
Novell. ZENworks. Linux Management
Home
Groups
Channels
Admins
Machines
Transactions
Reports
Server
Help
Logout
jeroen@local
0 machines in set
Home > Server > Activations > Add/Edit Activation
Basic
Machine Permissions
Done
Thursday, July 29, 2004 - 21:02

Edit 2035-e958-4166-b8f2

You have sucesfully added the channel(s) test channel

Machines activating with this key will have the following permissions settings:

Groups

Name	Description	Remove
Testgroup	Redbook test group	☐

No Groups Found — Add Groups — Save

Channels

Name	Description	Remove
test channel	Redbook test channel	☐

No Channels Found — Add Channels — Save

Administrators

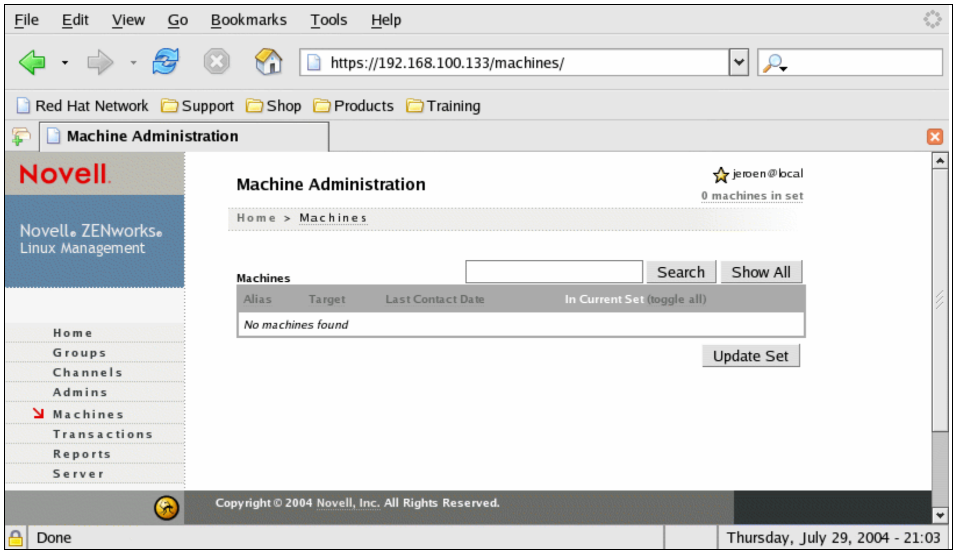
Email	Name	Permission	Remove
jeroen@local	jeroen	Grant	☐

Copyright © 2004 Novell, Inc. All Rights Reserved.

Figure A-41 Settings for groups, channels, and administrators

Add a client to ZENworks using Red Carpet

The client is added as a machine in ZENworks Linux management through either the **`red-carpet`** GUI application or the **`rug`** command-line tool. Both communicate with the server through the **`rcd`** daemon. Before we add the client we see that there are no machines present in ZENworks, as shown in Figure A-42.


File Edit View Go Bookmarks Tools Help
https://192.168.100.133/machines/
Red Hat Network Support Shop Products Training
Machine Administration
Novell
Novell ZENworks Linux Management
Home
Groups
Channels
Admins
Machines
Transactions
Reports
Server
Search
Show All
Update Set
Done
Thursday, July 29, 2004 - 21:03

Machine Administration

jeroen@local
0 machines in set

Home > Machines

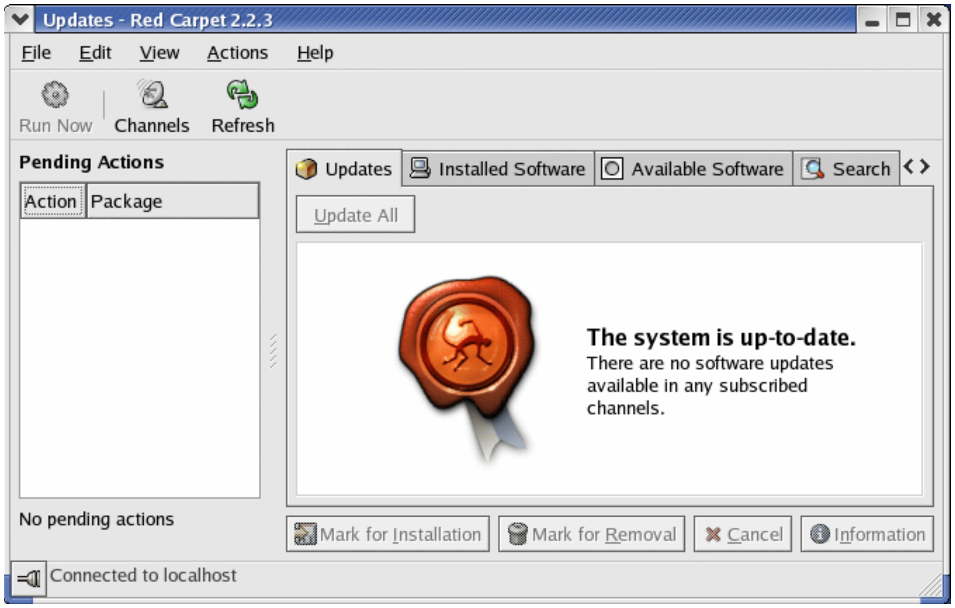
Machines

Alias	Target	Last Contact Date	In Current Set (toggle all)
No machines found			

Copyright © 2004 Novell, Inc. All Rights Reserved.

Figure A-42 Machine administration initial page

We start the Red Carpet application on the client (see Figure A-43). First we need to set the location of the ZENworks server to communicate with.

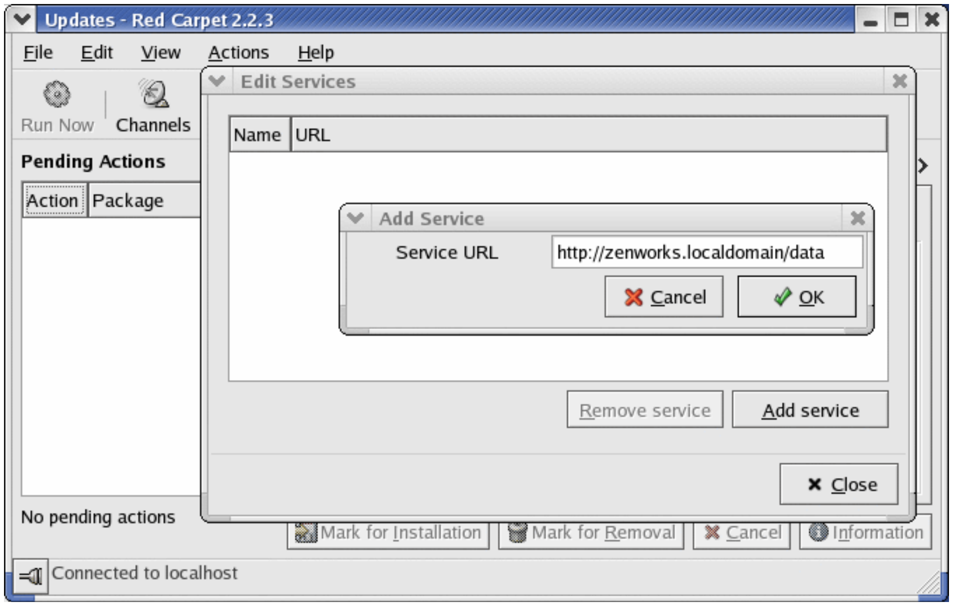


The system is up-to-date.
There are no software updates available in any subscribed channels.

Connected to localhost

Figure A-43 Red Carpet application

This is done using **Edit** → **Edit Services** and then **Add service**, and entering the service URL, as shown in Figure A-44 on page 208.

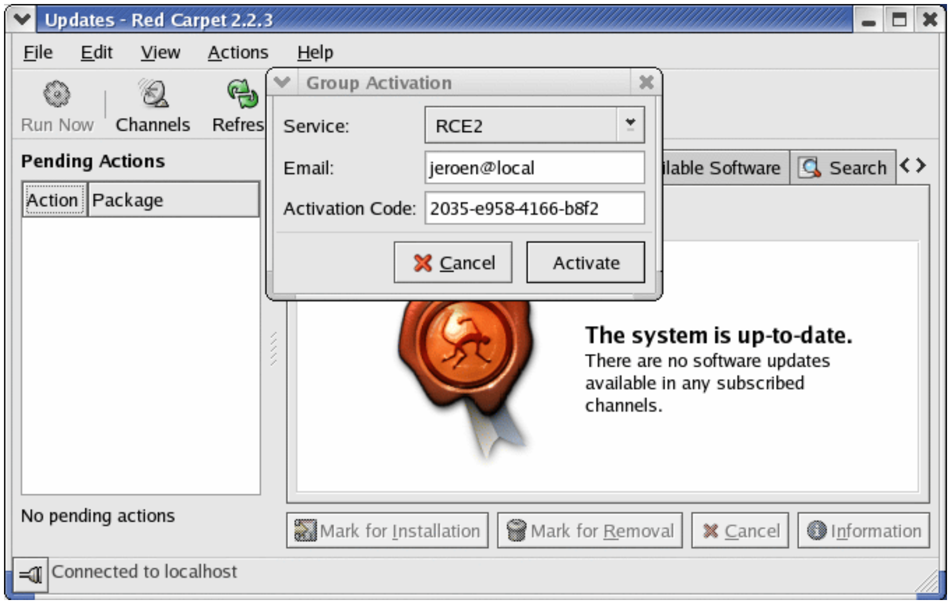


Service URL: http://zenworks.localdomain/data

Connected to localhost

Figure A-44   Red Carpet application edit services page

Next we activate using the activation key by going to **File** → **Activate** and entering the key, as shown in Figure A-45 on page 209.



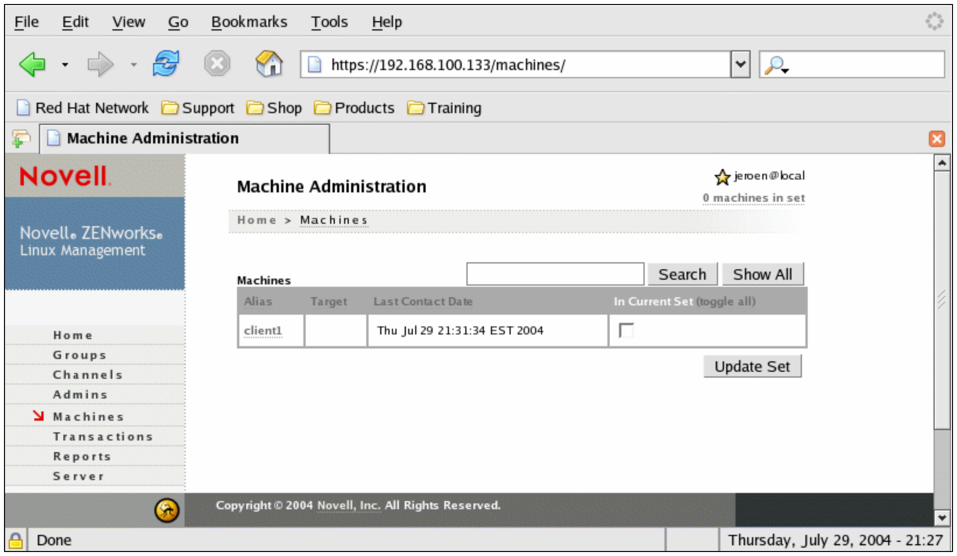
Service: RCE2
Email: jeroen@local
Activation Code: 2035-e958-4166-b8f2

Cancel | Activate

The system is up-to-date.
There are no software updates available in any subscribed channels.

Figure A-45 Red Carpet application activation

Going back to the ZENworks pages we see the machine client1 on the machine administration page, as shown in Figure A-46.



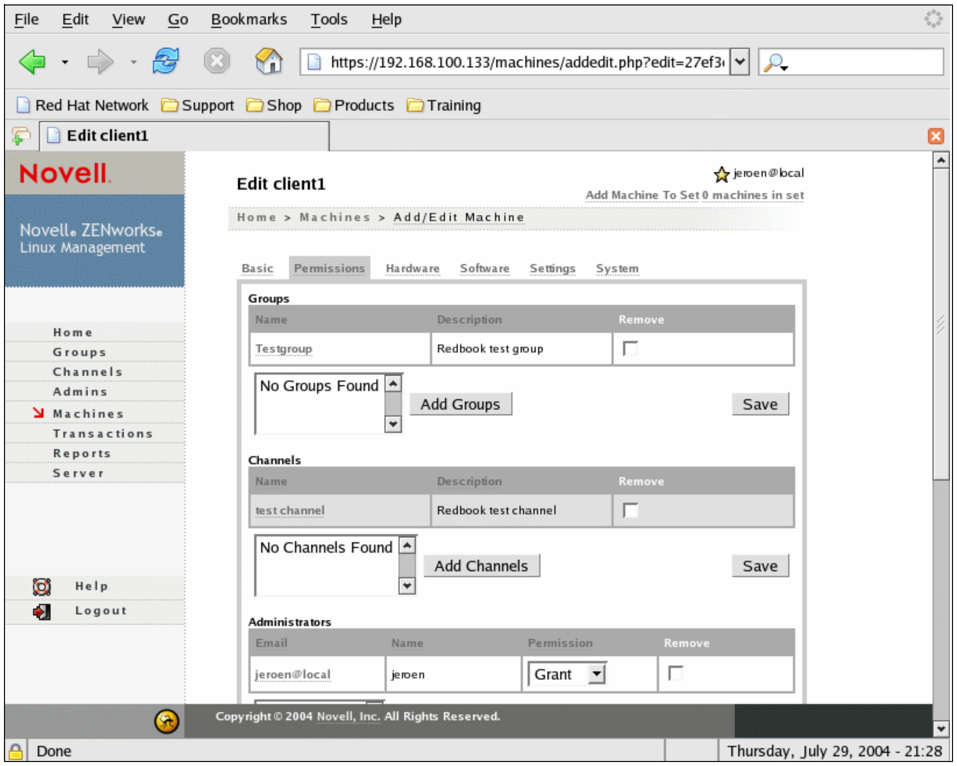
Machine Administration

Machines

Alias	Target	Last Contact Date	In Current Set (toggle all)
client1		Thu Jul 29 21:31:34 EST 2004	☐

Figure A-46 Client1 machine added to ZENworks

If we look at the edit page (Figure A-47) we see that the groups and channels we set for the activation key have been applied.


File Edit View Go Bookmarks Tools Help
https://192.168.100.133/machines/addedit.php?edit=27ef3
Red Hat Network Support Shop Products Training
Novell
Novell ZENworks Linux Management
Home
Groups
Channels
Admins
Machines
Transactions
Reports
Server
Help
Logout
Basic Permissions Hardware Software Settings System
Add Groups
Save
Add Channels
Save

Edit client1

jeroen@local

Add Machine To Set 0 machines in set

Home > Machines > Add/Edit Machine

Groups

Name	Description	Remove
Testgroup	Redbook test group	☐

No Groups Found

Channels

Name	Description	Remove
test channel	Redbook test channel	☐

No Channels Found

Administrators

Email	Name	Permission	Remove
jeroen@local	jeroen	Grant	☐

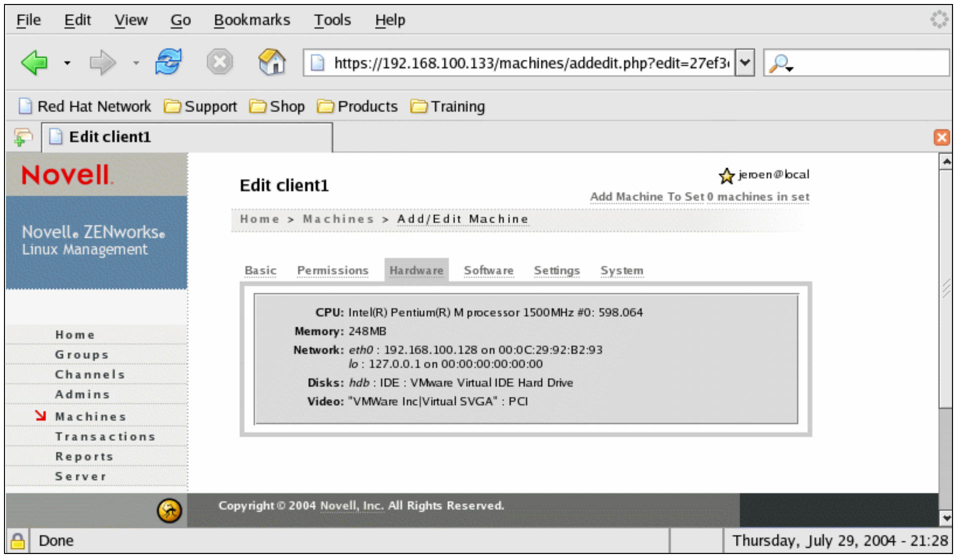
Copyright © 2004 Novell, Inc. All Rights Reserved.

Done

Thursday, July 29, 2004 - 21:28

Figure A-47 Edit machine page for client1

Figure A-48 on page 211 and Figure A-49 on page 211 show the information that has been passed to the ZENworks server by the `rcd` daemon. This information can be used in searches to create a machine set.

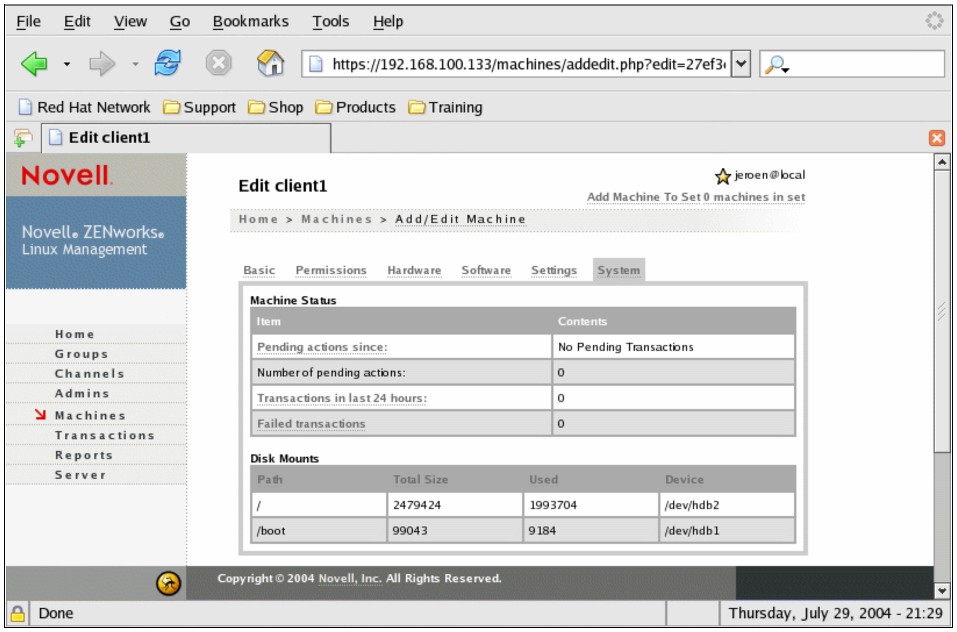

File Edit View Go Bookmarks Tools Help
https://192.168.100.133/machines/addedit.php?edit=27ef3
Red Hat Network Support Shop Products Training
Edit client1
Novell.
Novell. ZENworks. Linux Management
Home Groups Channels Admins Machines Transactions Reports Server
jeroen@local
Add Machine To Set 0 machines in set
Home > Machines > Add/Edit Machine
Basic Permissions Hardware Software Settings System
Done
Thursday, July 29, 2004 - 21:28

Edit client1

CPU: Intel(R) Pentium(R) M processor 1500MHz #0: 598.064
Memory: 248MB
Network: *eth0* : 192.168.100.128 on 00:0C:29:92:B2:93
lo : 127.0.0.1 on 00:00:00:00:00:00
Disks: *hdb* : IDE : VMware Virtual IDE Hard Drive
Video: "VMWare Inc|Virtual SVGA" : PCI

Copyright © 2004 Novell, Inc. All Rights Reserved.

Figure A-48 Edit machine page hardware for client1


File Edit View Go Bookmarks Tools Help
https://192.168.100.133/machines/addedit.php?edit=27ef3
Red Hat Network Support Shop Products Training
Edit client1
Novell.
Novell. ZENworks. Linux Management
Home Groups Channels Admins Machines Transactions Reports Server
jeroen@local
Add Machine To Set 0 machines in set
Home > Machines > Add/Edit Machine
Basic Permissions Hardware Software Settings System
Done
Thursday, July 29, 2004 - 21:29

Edit client1

Machine Status

Item	Contents
Pending actions since:	No Pending Transactions
Number of pending actions:	0
Transactions in last 24 hours:	0
Failed transactions	0

Disk Mounts

Path	Total Size	Used	Device
/	2479424	1993704	/dev/hdb2
/boot	99043	9184	/dev/hdb1

Copyright © 2004 Novell, Inc. All Rights Reserved.

Figure A-49 Edit machine page system for client1

B

Application porting

When planning for a Linux client migration you may discover early on in the assessment process that you have custom-developed applications that will not run natively on a Linux-based client (any Microsoft Visual Basic application would fit into this category).

This appendix provides some information and links for learning more about the application porting process. This topic alone could easily fill an entire redbook. We introduce the topic here for the sake of completeness (many Windows-to-Linux client migration projects will have application porting considerations).

REALBasic

REALbasic, provided by REAL Software, Inc., is a cross-platform software development environment that also provides a very effective path for porting Visual Basic applications so that they can run natively on a Linux client. A technical white paper is available that provides a description of the porting process using REALbasic. The white paper can be accessed at the following URL:

http://www.realbasic.com/vb

Here is a summary of that white paper:

> "This white paper explains how to port Windows applications developed in Visual Basic to Linux using REAL Software's REALbasic integrated development environment. It includes detailed code examples and provides a roadmap for how to port (and how not to port) Visual Basic applications to Linux. This white paper references REALbasic 5.5.3, a modern software development environment that is very similar to Microsoft Visual Basic® in terms of the GUI and syntax. It is not intended to provide a full comparison of Visual Basic and REALbasic, but rather to show where they are different in ways that will impact porting to Linux. It also explains how to make best use of the automated conversion tools provided by REAL Software."

wxWindows

wxWindows provides an open source C++ GUI framework for cross-platform programming. For more information, see the project Web site:

http://www.wxwindows.org

Some tutorials are available at the IBM Developerworks Web site that explain how to use this toolkit in detail:

- "*Looking through wxWindows: An introduction to the portable C++ and Python GUI toolkit*":

  http://www-106.ibm.com/developerworks/library/l-wxwin.html

- "*Porting MFC Applications to Linux: A step-by-step guide to using wxWindows*":

  http://www-106.ibm.com/developerworks/linux/library/l-mfc

C

Desktop automation and scripting

In this appendix we present details on some scripting and automation capabilities of Linux/Windows applications and desktops. These capabilities allow users and administrators to fully control their environment and automate tedious desktop tasks.

Scripting languages

Open source scripting languages for Linux like ECMAScript, Perl, Python, Ruby, Tcl, etc. have been ported to many different operating systems including Mac OS X and Windows. Typical Windows languages like Visual Basic, Visual Basic for Applications (VBA), or VB.NET are available for Linux to a certain degree as commercial offerings (for example, REALbasic) or as free implementations (like VB.NET in the Mono project from Novell):

http://www.thefreecountry.com/compilers/basic.shtml

All these Basic dialects are certainly relevant from a compatibility point of view (for example, for Excel macro migration), but do not play a central role in the foundation of the GNOME or KDE desktop environments since there are so many other natural choices there.

Python

Red Hat, for example, uses Python extensively in the installation process and at many other places. Tools like kudzu (which is also used for hardware recognition in the Debian-based Knoppix distribution) are available as Python modules too. While using **`kudzu --probe`** is straightforward, the Python wrapper allows system administrators to write their own harware recognition scripts in a much more elegant fashion, as you can see in Example C-1.

Example: C-1 Hardware recognition with kudzu and Python

```
#!/usr/bin/env python

import kudzu

print kudzu.probe(kudzu.CLASS_SOCKET, kudzu.BUS_PCI, kudzu.PROBE_ALL)
print kudzu.probe(kudzu.CLASS_NETWORK,kudzu.BUS_PCI, kudzu.PROBE_ALL)

devices = kudzu.probe(kudzu.CLASS_UNSPEC, kudzu.BUS_UNSPEC, kudzu.PROBE_ALL)
for dev in devices: print dev
```

QT Script for applications

Most Linux scripting languages have wrappers for the GTK+, QT, and WxWidgets libraries and even the whole KDE/GNOME APIs, so you can easily write simple desktop applications or panel applets for your own purposes. It is also possible to embed these languages in your programs to make them scriptable. Trolltech, for example, decided to give ECMAScript (the standardized version of JavaScript) a dominant role in QT by releasing the QT Script for Applications toolkit (QSA), which is tightly integrated into their framework:

http://www.trolltech.com/products/qsa/

QSA plays the same role as VBA for Windows applications and is very easy to use for QT/KDE programmers. QSA is released under a commercial license for Linux/Unix (X11), Windows, and Mac platforms. In addition to that, QSA is also licensed under the GNU GPL for free software development on the Linux/Unix (X11) and Mac OS X platforms.

KJSEmbed

A very similar approach is the KDE JavaScript Engine KJSEmbed with bindings for QT/KDE and the its interprocess communication mechanisms. You can find KSJEmbed (which is licenced unter the LGPL) and a lot of documentation at:

http://xmelegance.org/kjsembed
http://www.sourcextreme.com/presentations/KJSEmbed/

In Example C-2 you see how easy it is to write to a simple HTML browser with JavaScript using this toolkit.

Example: C-2 Basic HTML browser with KJSEmbedd

```
#!/usr/bin/env kjscmd

var url = 'http://www.kde.org/';
if ( application.args.length )
   url = application.args[0];

var mw = new KMainWindow();
var box = new QHBox( mw );
mw.setCentralWidget(box);

var view = Factory.createROPart( "text/html", box, "view" );
view.connect(view.child(0),
             'openURLRequest(const KURL&,const KParts::URLArgs&)',
             'openURL(const KURL&)' )
view.openURL( url );

mw.resize(650,500);
mw.show();

application.exec();
```

Many applications use other scripting languages of their choice (for example, Perl/Python in Gimp/Gnumeric, Python in Scribus, Star Basic in OpenOffice, JavaScript/XUL in Mozilla, etc.) so a universal language independent approach would make a lot of sense. This is where DCOP and D-BUS come into play, but in contrast to the embedded scripting language approach they face the problem with IPC/RPC..

Desktop interprocess communication

DCOP is the standard KDE Desktop Communication Protocol and can be used for all kinds of KDE desktop automation and scripting purposes. It is based on the Inter Client Exchange (ICE) protocol and uses UNIX sockets instead of remote calls. As with D-BUS (which is the corresponding GNOME approach and hosted by freedesktop.org), it has its roots in a CORBA implementation but was completely redesigned for desktop needs since the KDE developers where not happy with the CORBA performance at that time. The GNOME developers sticked to CORBA (with their optimized ORBit2 implementation) but added D-BUS to their toolkit (for example, for hardware event notfication).

On an even higher level ,you can use Web services to integrate applications, but the Linux desktop itself provides much more efficient tools like DCOP and D-BUS to communicate with your aplications. Let us take a closer look at DCOP since this is a proven KDE technology that can also be used for KDE applications running on the GNOME desktop.

DCOP

When you log into your KDE session a program called **kdeinit** is started (by **startkde**). It triggers other applications like **dcopserver** and the KDE daemon **kded**, which are responsible for system configuration management and caching (syscoca), inter-process communication, and a lot more. If you really want to understand how this process works and which environment variables and configuration files can influence it, you should read the *KDE for System Administrators Guide* available at:

http://www.kde.org/areas/sysadmin/

You can now explore your desktop environment and make DCOP calls with the command-line tool **dcop**, its graphical counterpart **kdcop**, or with the DCOP language bindings for Perl, Python, Ruby, Java, etc., which are part of the **kdebindings** package. The **dcop** command syntax is as follows:

```
dcop [ application [object [function [arg1] [arg2] [arg3] ... ] ] ]
```

Desktop automation

To begin with your desktop automation work just call **dcop**, check the output, and go further down the pipeline. In Example C-3 you can see how this works in practice. After listing all applications registered with the DCOP server (for example, **konqueror**'s process ID 21209), we open a new Konsole session and display the first session name afterwards. Next we list our e-mail accounts, check e-mail, compact all e-mail folders, and open a new KMail composer window with some predefined values.

Example: C-3 DCOP in action

```
% dcop
konsole-366
kmail
...
% dcop konsole-366 konsole newSession Test
% dcop konsole-366 session-1 sessionName
% dcop kmail KMailIface accounts

% dcop kmail default checkMail
% dcop kmail default compactAllFolders
% dcop kmail default openComposer info@kde.org "" "" "DCOP!" "Thanks” 0
```

In Example C-4 we open our favorite URL in Konqueror and generate another browser window with two preloaded tabs. The next line hides the **kicker** panel, which is useful for presentations or unattended Kiosk mode. The next line brings back the panel to make sure you do not panic. Finally we switch to desktop 2 by using the published interface of the KDE window manager **kwin**.

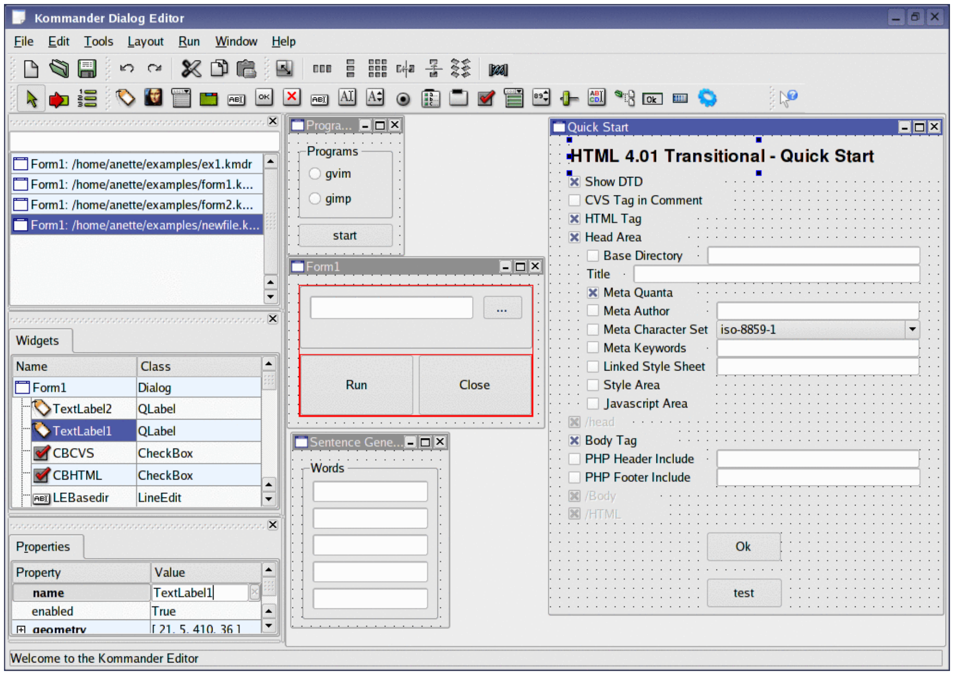
Example: C-4 More DCOP magic

```
% dcop
konqueror-21209
kicker
kwin
...
% dcop konqueror-21209 konqueror-mainwindow#1 openURL http://www.ibm.com/linux
% dcop konqueror-21209 default openBrowserWindow http://www.kde.org
% dcop konqueror-21692 konqueror-mainwindow#2 newTab http://www.linux.com

% dcop kicker qt/Panel hide
% dcop `dcop konqueror-21209 konqueror-mainwindow#1 action fullscreen` activate
% dcop kicker qt/Panel show

% dcop kwin default setCurrentDesktop 2
```

DCOP is a very powerful mechanism and can even be used with XML Remote Procedure Calls (through the XML-RPC to DCOP bridge), but one has to be aware of the fact that DCOP is an unencrypted protocol. By using standard UNIX security measures for the ICE commmunication sockets, users should not be able to attack DCOP sessions from other users.

Kommander

The developers of the Quanta+ Web development framework introduced a new toolkit called Kommander, which can be used to design KDE Dialogs and widgets by using **`kmdr-editor`** (a program derived from QT Designer) and executing them with **`kmdr-executor`**. Kommander uses XML-based GUI description files with embedded DCOP scripts (for KDE program automation, Signal-Slots, internal script execution, etc.), which can be written in any language able to speak the DCOP protocoll (Bash, Python, Perl, Java, JavaScript, etc.). Briefly, Kommander allows you to write KDE dialogs by using the powerful DCOP mechanisms itself. It is already used at many places inside Quanta+ (DCOP is also used for the inter-component communication in the KDE PIM framework Kontakt). Take a look at Figure C-1 to see how the integrated DCOP development with **`kmdr-editor`** looks and can be used for your own purposes.



Form1: /home/anette/examples/ex1.kmdr
Form1: /home/anette/examples/form1.k...
Form1: /home/anette/examples/form2.k...
Form1: /home/anette/examples/newfile.k...

Name	Class
Form1	Dialog
TextLabel2	QLabel
TextLabel1	QLabel
CBCVS	CheckBox
CBHTML	CheckBox
LEBasedir	LineEdit

Property	Value
name	TextLabel1
enabled	True
geometry	[21, 5, 410, 36]

HTML 4.01 Transitional - Quick Start

[✓] Show DTD
[ ] CVS Tag in Comment
[✓] HTML Tag
[✓] Head Area
[ ] Base Directory
Title
[✓] Meta Quanta
[ ] Meta Author
[ ] Meta Character Set: iso-8859-1
[ ] Meta Keywords
[ ] Linked Style Sheet
[ ] Style Area
[ ] Javascript Area
[✓] /head
[✓] Body Tag
[ ] PHP Header Include
[ ] PHP Footer Include
[✓] /Body
[✓] /HTML

Welcome to the Kommander Editor

Figure C-1 Integrated DCOP development with Kommander

D

Client personalization

This appendix compares the methods for storing client-side personalization data between Windows and Linux. As this book focuses on migrating clients from Windows to Linux, migration of client-side personalization data could also be an important topic for planning and implementation phases of a project.

Microsoft Windows client personalization

The main container for personalization data in Windows is called the user's *profile*. In Windows 2000 profile data is stored in:

```
c:\documents and settings\username
```

In Windows NT it is stored in:

```
c:\winnt\profiles\username
```

Windows uses these locations to store various kinds of personalization data, for example, user documents, application data and settings, temporary Internet files, "favorites", etc. This directory is named after the user name, but can also have a more detailed distinction, as sometimes the computer name or the name of the domain is appended. This happens if a user exists already and a new one is created with the same name but integrated in the domain. So one user name can exist several times on a machine, connected with the domain name or the local machine name or other combinations. The consequence is that Windows has to distinguish between user username.localcomputername and username.localdomainname.

The selection field in the login screen of Windows is where the user selects which profile to use. The data that defines personalization of the desktop and some other user-based settings resides in the NTUSER.dat file. At logon time these settings are loaded to show the customized desktop.

In order to provide profiles that are saved on a network share and that can be used from more than one client, Microsoft uses different types of profiles.

A local profile is stored localy to the client and therefore is only accessible on that computer. If the same user logs onto another machine, a new profile for that user will be created on that machine.

To make a single profile available to the same user from different client machines in the domain, a roaming profile must be created and stored in a shared location on the network.

The third type or profile is the mandatory type. It is intended for environments in which a change of the profile is not desired. Only an administrator can edit this type of profile, and all changes during a session will be deleted after logging out. Thus an identical profile is provided each time a user logs on.

Linux client personalization

Considering the modular structure of Linux, it becomes clear that profiles must differ a lot from the Windows profiles. As there is no registry in Linux, all settings must be saved in another way. Linux does this mostly via using readable text files, which have file name extensions such as .conf or .profile.

The file /etc/profile sets the system-wide environment and startup programs for logins. These system-wide defaults may be extended by profiles in the local or home directory of the user. The local .bash_profile (Bourne Again SHell) is one example on Red Hat; the .profile file on SuSE is another example.

Structurally, Linux stores user profile data in a very different way than Windows does. All individual user settings are stored inside the user's home directory, $HOME. For example:

```
/home/username
```

Linux does not implement a registry the way Windows does. Instead, user profile information is stored in configuration files within the individual user home directories. And these files are usually created by specific applications that need to store user personalization data required by that application.

One advantage of these files is that they are ASCII-based text formatted and thus human readable. Another is that the content can be automatically manipulated by scripts if desired. Also, by modifying permissions on these files it is possible to prevent undesired changes by the user. This could have the effect of "locking down" certain aspects of the user's login environment.

Most applications create a hidden folder, beginning with a dot or period (.), in which configuration files for those applications are created and maintained.

Further discussion of Linux methods for containment of personalization data will focus on the two currently most popular desktop session managers: KDE and Gnome. This topic strongly overlaps with methods for standardizing the desktop look and feel (in other words, how to design and enforce consistent personalization data). See 4.3, "Standardizing the desktop" on page 51, for more details.

Desktop personalization: KDE containment model

In KDE, desktop personlization is based on information found in a standard directory structure within the user's home directory in the following location:

```
/home/username/Desktop
```

The Desktop folder contains the icon files as well as all of the folders that are presented on the desktop.

KDE uses ASCII-based text files for all its configuration options. This makes it easy to configure KDE using scripts and editors. Three main files are used to configure the desktop:

- kdeglobals
- kdesktoprc
- kickerrc

Red Hat 9.0 places these files in the /usr/share/config directory. Other distributions may place these files in different locations. These files contain the default settings used to personalize the KDE desktop for new users. Thus any changes that are made to these files will be inherited by all new users created after making the changes. See 5.1, “KDE Kiosk framework” on page 90, for more details.

Desktop personalization: GNOME containment model

As with the KDE desktop, personalization data used by Gnome is stored in folders that are created as a directory structure within the user’s home directory. The desktop structure within GNOME is different than that of KDE. In GNOME, this information is contained within this directory:

```
/home/username/.gnome-desktop
```

Icons are handled in the same way as they are in KDE. This means that they are ASCII text files containing information that defines the application behavior when the icon is launched. You can save them in the /home/<user>/.gnome-desktop directory to populate the desktop.

The task of making general configuration changes to the GNOME desktop is very different than in KDE. KDE uses simple and easy-to-edit ASCII files. The GNOME desktop uses a database of Extensible Markup Language (XML) files to build and customize all of its settings. These files are numerous and can be found in this location:

```
/etc/gconf/gconf.xml.defaults/desktop/gnome/ directory.
```

This directory contains subdirectories for each portion of the GNOME desktop environment, as shown here:

```
/etc/gconf/gconf.xml.defaults/desktop/gnome/accessibility
/etc/gconf/gconf.xml.defaults/desktop/gnome/applications
/etc/gconf/gconf.xml.defaults/desktop/gnome/background
/etc/gconf/gconf.xml.defaults/desktop/gnome/file_views
```

```
/etc/gconf/gconf.xml.defaults/desktop/gnome/font_rendering
/etc/gconf/gconf.xml.defaults/desktop/gnome/interface
/etc/gconf/gconf.xml.defaults/desktop/gnome/peripherals
/etc/gconf/gconf.xml.defaults/desktop/gnome/sound
/etc/gconf/gconf.xml.defaults/desktop/gnome/thumbnailers
/etc/gconf/gconf.xml.defaults/desktop/gnome/url-handlers
```

Within each of these subdirectories is a file called %gconf.xml.

Gnome provides a command-line tool called gconftool-2, which can be used to change settings and preferences in these XML format files.

In order to make settings available from different applications and also from tools within the Gnome Desktop, there is a daemon running that notifies all involved applications when a value has changed and that manages all access between applications and the configuration sources. This daemon is called gconfd-2 and runs in one instance for each user.

For further information on how to change values in the Gnome configuration or make changes that can be enforced for groups of users, see 5.2, “GNOME customization” on page 104.

Related publications

IBM Redbooks

For information on ordering these publications, see "How to get IBM Redbooks" on page 232. Note that some of the documents referenced here may be available in softcopy only.

- *Linux Handbook: A guide to IBM Linux Solutions and Resources,* SG24-7000
- *Domino Web Access 6.5 on Linux*, SG24-7060
- *OS/2 to Linux Client Transition*, SG24-6621

Other publications

These publications are also relevant as further information sources:

- *Printing with Linux on zSeries Using CUPS and Samba*, REDP-3864-00
- *Migrate Exchange 5.5 to Domino on Linux,* REDP-3777-00
- *Linux: Why it Should Replace Your Windows Domains,* REDP-3779
- *Open Your Windows with Samba on Linux,* REDP-3780
- The IDA Open Source Migration Guidelines, netproject Ltd © European Communities 2003

Online resources

These Web sites and URLs are also relevant as further information sources:

- Samba-3 HOWTO Collection

  `http://samba.org/samba/docs/Samba-HOWTO-Collection.pdf/`
- Red Hat Network Architecture

  `http://www.redhat.com/software/rhn/architecture/`
- Webmin

  `http://www.sourceforge.net/projects/webadmin`
  `http://www.webmin.com`

- Big Brother/Big Sister

  `http://www.sourceforge.net/projects/big-brother;`
  `http://www.sourceforge.net/projects/bigsister`
  `http://www.bb4.org`

- Nagios

  `http://sourceforge.net/projects/nagios`
  `http://www.nagios.org/`

- Copies of the GNU GPL Licenses

  `http://ww.gnu.org/licences/licenses.html`

- Free Software Foundation (FSF)

  `http://www.gnu.org/fsf/fsf.html`

- OpenOffice.org Writer

  `http://www.openoffice.org`

- The GIMP

  `http://www.gimp.org`

- GAIM

  `http://gaim.sourceforge.net`

- rdesktop

  `http://sourceforge.net/projects/rdesktop/`

- NoMachine NX

  `http://www.nomachine.com`

- Knoppix:

  `http://www.knoppix.com`

- IBM Redpaper *Using Asset Depot for Inventory Management*

  `http://publib-b.boulder.ibm.com/Redbooks.nsf/RedbookAbstracts/redp3763.html`
  `?Open`

- Personal Computing Support: Linux for IBM Personal Systems

  `http://www-306.ibm.com/pc/support/site.wss/MIGR-48NT8D.html`

- Linux Standard Base

  `http://www.linuxbase.org`

- KDE

  `http://www.kde.org`

- SuperKaramba

  `http://netdragon.sourceforge.net/`

- GNOME

  http://www.gnome.org
- Significant GNOME desktop applications

  http://www.gnomefiles.org
- GDesklets

  http://gdesklets.gnomedesktop.org/
- SVG Clip Art Library

  http://www.openclipart.org
- W3 standard SVG

  http://www.w3.org/Graphics/SVG/
- Theme Web sites

  http://themes.freshmeat.net
  http://www.customize.org
  http://www.kde-look.org
  http://art.gnome.org
  http://www.crystalgnome.org
- KDE OpenOffice integration and Mozillux projects

  http://www.polinux.upv.es/mozilla
  http://kde.openoffice.org
- Success story of Linux usability on the desktop

  http://www.linux-usability.de/download/linux_usability_report_en.pdf
  http://www.userinstinct.com/viewpost.php?postid=gnome26review
- UniConf

  http://open.nit.ca/wiki/
- Linux Registry Project

  http://registry.sourceforge.net
- InterMezzo

  http://www.inter-mezzo.org
- Linux at IBM Solutions page

  http://www.ibm.com/linux/solutions
- RHN

  http://www.redhat.com/software/rhn
- Zenworks 6.5 Linux management

  http://www.novell.com/documentation/zenworks65/index.html

- ndiswrapper project

  http://ndiswrapper.sourceforge.net
- PC model types suitable for Linux

  http://www.ibm.com/pc/support/site.wss/MIGR-48NT8D.html
- Common Unix Printing System

  http://www.cups.org
- Scanner Access Now Easy (SANE)

  http://www.sane-project.org
- M.U.S.C.L.E. - Project for integrating SmartCards

  http://www.linuxnet.com
- Unix SmartCard Driver Project

  http://smartcard.sourceforge.net
- Project gphoto2 - Supporting digital cameras

  http://www.gphoto.org
- For winmodems, some Linux drivers are available for some models

  http://linmodems.org
- Mozilla Thunderbird

  http://www.mozilla.org
- iFolder

  http://www.novell.com/products/ifolder
- rsync

  http://samba.anu.edu/rsync
- Kiosk Admin Tool

  http://extragear.kde.org/apps/kiosktool.php
- Kiosk readme

  http://webcvs.kde.org/cgi-bin/cvsweb.cgi/kdelibs/kdecore/README.kiosk
- Mono

  http://www.mono-project.com
- openMosix

  http://www.openmosix.org
- Debian FAI (Fully Automated Installation)

  http://www.informatik.uni-koeln.de/fai

- PXELinux

  http://syslinux.zytor.com/pxe.php
- Red Hat Linux hardware probling library

  http://rhlinux.redhat.com/kudzu
- *Introduction to Stateless Linux, Proposal for the Fedora Project*

  http://people.redhat.com/~hp/stateless/StatelessLinux.pdf
- Purchase printing software supporting Linux

  http://www.easysw.com/printpro
- Try a Windows PPD file provided by Adobe

  http://www.adobe.com/products/printerdrivers/winppd.html
- CUPS home page

  http://www.cups.org
- GIMP as a functionally equivalent Linux application

  http://www.gimp.org
- Novell Exchange Connector

  http://www.novell.com/products/connector/
- *Samba HOWTO Collection* section 6.3

  http://samba.org/samba/docs/Samba-HOWTO-Collection.pdf/
- Red Hat Web page, Red Hat Network Architecture

  http://www.redhat.com/software/rhn/architecture/
- Red Hat Web site

  http://www.redhat.com/rhn
- *ZENworks Linux Management 6.5 Administrator's Guide*

  http://www.novell.com/products/zenworks/linuxmanagement/
- Evalution copy of ZENworks 6.5 Linux management

  http://www.novell.com/products/zenworks/eval.html
- REALbasic for porting Visual Basic applications so that they can run natively on a Linux client white paper

  http://www.realbasic.com/vb
- wxWindows provides an open source C++ GUI framework for cross-platform programming - project Web site

  http://www.wxwindows.org

- “Looking through wxWindows: An introduction to the portable C++ and Python GUI toolkit”

  http://www-106.ibm.com/developerworks/library/l-wxwin.html
- “Porting MFC Applications to Linux: A step-by-step guide to using wxWindows”

  http://www-106.ibm.com/developerworks/linux/library/l-mfc
- Free BASIC Compilers and Interpreters

  http://www.thefreecountry.com/compilers/basic.shtml
- QSA overview

  http://www.trolltech.com/products/qsa/
- *KDE for System Administrators Guide*

  http://www.kde.org/areas/sysadmin/

How to get IBM Redbooks

Help from IBM

Index

IBM Redbooks

Linux Client Migration Cookbook - A Practical Planning and Implementation Guide for Migrating to Desktop Linux

Linux Client Migration Cookbook

A Practical Planning and Implementation Guide for Migrating to Desktop Linux

For anyone who is exploring or planning for a Linux desktop migration

Provides in-depth detail on the technical and organizational challenges

Includes methods for planning and implementation

The goal of this IBM Redbook is to provide a technical planning reference for IT organizations large or small that are now considering a migration to Linux-based personal computers. For Linux, there is a tremendous amount of "how to" information available online that addresses specific and very technical operating system configuration issues, platform-specific installation methods, user interface customizations, etc. This book includes some technical "how to" as well, but the overall focus of the content in this book is to walk the reader through some of the important considerations and planning issues you could encounter during a migration project. Within the context of a pre-existing Microsoft Windows-based environment, we attempt to present a more holistic, end-to-end view of the technical challenges and methods necessary to complete a successful migration to Linux-based clients.

SG24-6380-00 ISBN 0738491527