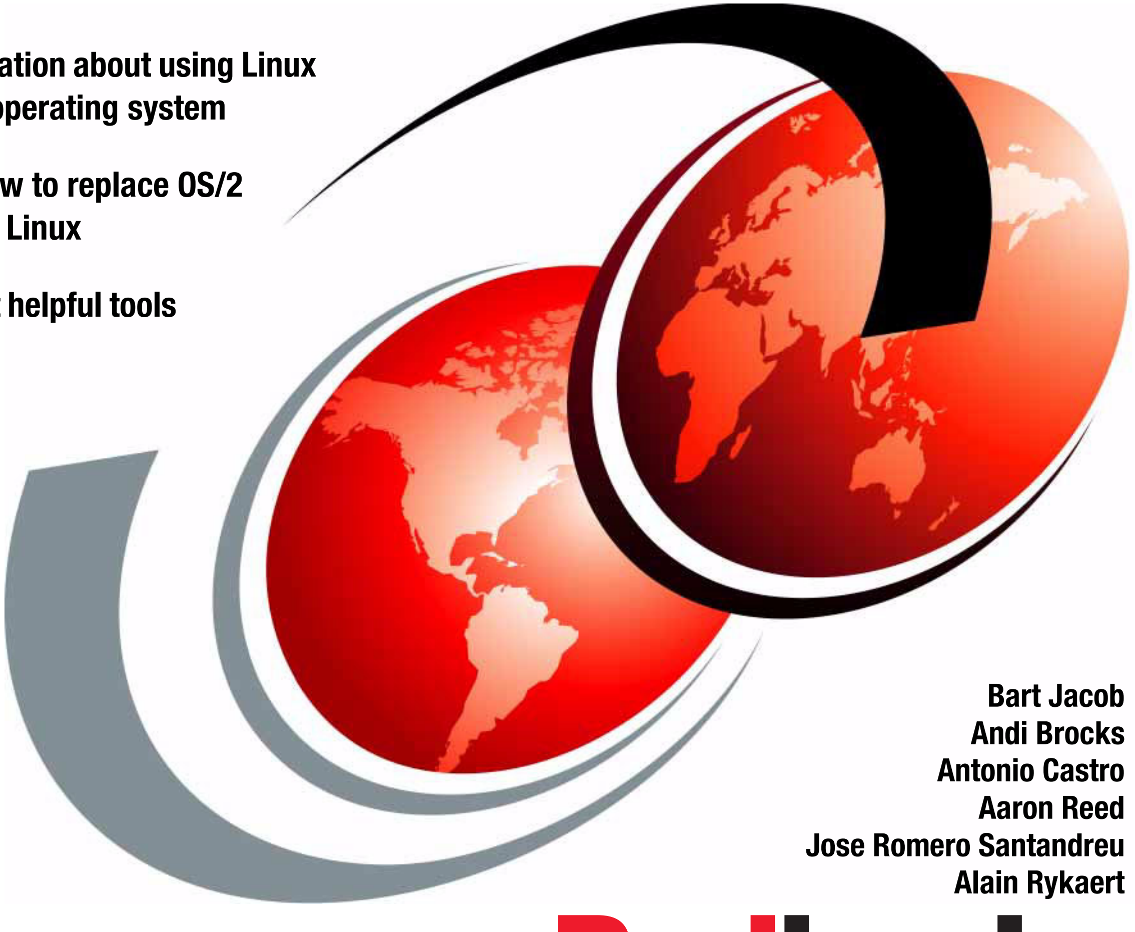
OS/2 to Linux Client Transition

Gain information about using Linux as a client operating system

Discover how to replace OS/2 clients with Linux

Learn about helpful tools and tips



Bart Jacob
Andi Brocks
Antonio Castro
Aaron Reed
Jose Romero Santandreu
Alain Rykaert

Redbooks

ibm.com/redbooks

International Technical Support Organization

OS/2 to Linux Client Transition

March 2004

SG24-6621-00

Note: Before using this information and the product it supports, read the information in “Notices” on page xi.

First Edition (March 2004)

Contents

Figures

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:
This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

eserver®
ibm.com®
iNotes™
iSeries™
pSeries®
xSeries®
z/OS®
zSeries®
AIX®
CICS®
Domino®
DB2®
Footprint®
Hummingbird®
Infoprint®
IBM®
Lotus Notes®
Lotus®
NetVista™
Notes®
OS/2®
Redbooks (logo) ™
Redbooks™
S/390®
Sametime®
Tivoli®
WebSphere®
Workplace Messaging™

The following terms are trademarks of other companies:

Intel, Intel Inside (logos), MMX, and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

SET, SET Secure Electronic Transaction, and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

Other company, product, and service names may be trademarks or service marks of others.

Preface

This IBM® Redbook provides information related to the viability of Linux as a client platform. It targets technical personnel who are involved in evaluating Linux as a possible client platform. It also targets administrators and support personnel that are responsible for supporting client systems.

This redbook can be helpful to anyone who is evaluating the potential of using Linux for enterprise client systems. However, the key focus is on environments where OS/2® is currently used.

Many enterprises have been using OS/2 as a stable platform for critical enterprise client applications. However, as those enterprises look to the future, they are looking for a platform on which they can build a strategy that is open, standards-based, secure, and provides a cost effective solution. Linux has become successful as a server platform in many of these same enterprises. It comes as no surprise that these enterprises also want to evaluate the possibility of including Linux for many of their client systems.

This redbook describes platform and functional considerations for choosing Linux as a client platform. It discusses the following topics of interest to system administrators:

- Techniques and facilities for administering Linux clients
- Coexistence of Linux clients with other platforms
- A technique to easily install Linux clients based on the well-known OS/2-based CID methodology

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 (ITSO), Austin Center.

Bart Jacob is a Senior Consulting IT Specialist for the IBM ITSO, Austin Center. He has 23 years of experience in providing technical support across a variety of IBM products and technologies, including communications, object-oriented software development, and systems management. He has over 12 years of experience at the ITSO, where he has been writing IBM Redbooks™ and creating and teaching workshops around the world on a variety of topics. He holds a masters degree in numerical analysis from Syracuse University.

Andi Brocks is a Senior Software Support Specialist working for IBM. He supports OS/2 for the major banks in the United Kingdom. He has been providing OS/2 support for many large enterprises for the past eight years. He also specializes in Linux support.

Antonio Castro is a PMI Certificated Project Manager. He joined IBM in 1996 in the Strategic Outsourcing organization. He then moved to the Sydney Olympics project to manage the testing of the Game Management Systems. He previously worked for another firm in the chemical analysis group developing new analytical methods and software and supported the Doping Control Labs of the Barcelona Olympic Games and Lillehammer Winter Olympics. His main areas of interest include application testing, Linux server and client systems, and project management.

Aaron Reed is an Advisory Software Programmer at IBM Austin. He has 11 years of experience in OS/2 programming, working on the Workplace Shell, Netscape for OS/2, and Mozilla development teams with a wide range of customer support experience. He holds a degree in mathematics from Iowa State University.

Jose Romero Santandreu is an IT Specialist in IBM Spain. He has six years of experience in UNIX® environments, especially AIX®, working in Adabas, DB2®, MQSeries®, and IBM Tivoli® Storage Manager projects over this platform. He spent two years providing UNIX infrastructure support for the e-business Integration Center in Madrid and two Linux projects. He is now working in the Madrid CAS center performing DB2, AIX, and Linux support.

Alain Rykaert has worked for 25 years at IBM Belgium in various positions within technical support and marketing. He spent the past dozen years mostly with OS/2 and related products. As a member of the rapid IBM deployment team, he does large scale rollouts of OS/2 for European enterprises. He has participated in several residencies in Austin and has coauthored several redbooks, mostly related to IBM LAN Server. He is also an active member of the OS/2 community, including the Belgian OS/2 User Group.

Thanks to the following people for their contributions to this project:

Dave Fritz
Paul Griffiths
Jason Kersten
Walter Lee
IBM Austin

Alejandro Gonzalez Madueno
IBM Spain

Oliver Mark
IBM Germany

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:

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:

  ibm.com/redbooks

- 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

1

Introduction to client systems and Linux

OS/2-based client systems have been used in large enterprises for the last 15 years. They have provided a rich and stable platform for the client portions of many sophisticated client-server applications.

As enterprises move to more flexible information technology (IT) infrastructures to support on demand business environments, many are looking to replace their current OS/2 clients with new systems. Enterprises are looking for cost-effective solutions that provide the capabilities and flexibility to meet new and changing requirements.

Linux has matured to the point where many enterprises are already using it on their servers. Its stability, security, and cost effectiveness in an on demand environment has prompted many organizations to start investigating its use as a client system.

This chapter describes the capabilities of Linux that make it a suitable client solution for many enterprise users and particularly for those environments where OS/2 is currently deployed.

1.1 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 (SOA) 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 to which many enterprises are moving. However, it won’t 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.

1.1.1 Workstation classification

There are many ways to segment client workstation types and requirements. When doing so for various user groups and looking for an alternative to current OS/2-based clients, consider Linux as a possible alternative. This redbook describes the various and growing number of capabilities available today on Linux.

Fixed function

Users of these client machines typically run only one designated application or application type. Applications are customized for specific usage. Examples, may include kiosk, point-of-sale terminal, or such legacy environments as IBM 3270-based applications. More recently deployed applications can simply be designed to run in a portal-based environment where the user's interface is through a Web browser.

Technical workstation

Users of these client machines work on industry-specific applications. They may require specific software packages, tailored to a business sector or problem domain. Examples include engineering applications (such as CAD/CAM applications) or entertainment applications (such as movie animation).

Transactional workstation

Users of these client machines run applications that range from simple query and update, to complex conversational, forms-based transactions. Users of these workstations may also have a requirement to browse the intranet and simple Internet sites, and process simple e-mail. Applications are often customized for specific usage, for example, e-mail that does not include attachments. Examples of such workstations are travel agency workstations, bank teller workstations, and front-office workstations in insurance companies.

Basic office workstation

Users run business applications necessary to drive the company business processes, such as Enterprise Resource Planning (ERP), Supply Chain Management (SCM). They browse the intranet and simple Internet sites. They also collaborate through instant messaging or e-mail.

They may also run applications to create and view simple documents (for example, memos, letters, spreadsheets) for use within the company. These applications create files in portable formats, such as Portable Document Format (PDF), rich text format (RTF), Hypertext Markup Language (HTML). Examples may include bank side-counter or loan officers.

Advanced office workstation

Users perform all the functions of the basic office workstation. In addition, they run applications to create and modify complex (compound) documents for use both within and outside their company. These applications involve advanced office productivity features such as charting, formatting, or embedding. Retaining the data format while exchanging files is important. Plus, users of these client machines may also download executable binary files from the Internet. Examples of users of these workstations include software developers, retail back-office workers, bank side-counter, or loan officers.

1.2 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.

Because of its core design and open nature, Linux can be easily customized. Linux is available under the GNU General Public License (GPL) agreement and can 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.

From the beginning, Linux was developed to 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.

1.3 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 could 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 Tovalds 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 was needing.

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*. The most well-known distributions include Red Hat, SuSE, and Debian. UnitedLinux is a consortium that includes several companies including SuSE, Connectiva, Turbolinux, and SCO Group.

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- 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.

1.3.1 Licensing

Linux is distributed under the GNU GPL agreement. This section summarizes briefly the terms of this license.

The GNU GPL allows and limits the licensee to the following terms:

- Free redistribution of the software is allowed. There are no restrictions in regard to selling or giving away the software.
- The program must include the source code and allow distribution source and compiled form.
- 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 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.

1.4 Summary

This short introduction has provided a brief overview of Linux and some of its characteristics that make it a candidate to be used in an enterprise environment. The next chapter looks at platform-related topics and the facilities that a Linux platform provides that make it a viable option for enterprise client systems.

2

Platform considerations

Before we address application-specific considerations in later chapters, we first discuss platform capabilities.

When choosing a client platform, one of the primary considerations is its usability, or more specifically, how the user interacts with the system. This includes the look and feel of the user interface. In an enterprise environment, a key consideration is also how easily the user interface can be customized and maintained by support personnel. Aside from the look and feel of the desktop, other aspects such as support for printing and multimedia capabilities also fall into the category of platform considerations.

2.1 Graphical user interface

Linux provides great flexibility when it comes to its graphical user interface (GUI). OS/2 included the Presentation Manager, and Windows® has its own proprietary graphical interface. But, Linux systems allow for much more flexibility in the choice of the windowing environment.

In both the OS/2 and Windows environment, the subsystem that provides the basic graphical capabilities and the subsystem that drives the windows and user experience are combined. Linux uses well-known standards that allow for the various components of the overall graphical interface to be chosen independently. This provides much more flexibility and allows for the user interface to be optimized for the specific needs of the users and ultimately the business.

The components of the graphical subsystem can be broken down into several layers. For the purposes of this discussion, we describe three layers:

- Windowing system
- Video support
- Desktop environments

2.1.1 X11 window systems

For many years, the X11 window standard has been the primary windowing for most UNIX distributions. It is well known and well understood. Also extensive expertise is available for developing X11 applications.

Several X11 implementations are available for Linux, of which we describe three of them. Linux distributions come with one or more X11 window implementations.

XFree86

XFree86 is an open-source (MIT license) X Window System implementation that is available for Linux and other platforms. XFree86 is by far the most popular X11 implementation and is included with most distributions.

Recent releases of XFree86 have added numerous performance and feature enhancements (for example, 3D support for games and other applications). The availability of many of these advanced features depends on the specific video card in use.

To learn more about XFree86, see:

http://www.xfree86.org

Metrolink

Metrolink offers several commercial X servers. These servers range from low-footprint versions for embedded applications to high performance versions with 3-D, touch-screen, and multi-head support.

You may consider commercial X servers, such as Metro-X and Xi Graphics, for special purpose applications (for example, Linux used as a platform for a touch-screen kiosk or asynchronous transfer mode (ATM)). You may also consider them for cases where the customer's hardware is unsupported by XFree86.

For more information about Metrolink, see:

`http://www.metrolink.com`

Xi Graphics

Xi Graphics offers a line of commercial X servers targeted to special requirements such as multi-head display and environments where high performance is critical. Xi Graphics also offers custom and original equipment manufacturer (OEM) versions for embedded solutions.

To learn more about Xi Graphics, go to the following Web site:

`http://www.xig.com`

2.1.2 Video support

In general, most video adapters work well with XFree86, although XFree86 drivers may take a while to appear when a new chipset is introduced. A generic unaccelerated VESA driver is included that may be useful when dealing with unsupported chipsets.

In enterprise environments where a large client rollout is being developed with multiple client types containing different chipsets, SciTech Software Inc. provides a multi-OS graphics driver solution. It includes support for OS/2, virtually all versions of Linux (based on XFree86 4.x and later), and embedded versions of Linux which use the Qt embedded graphics library.

IBM currently uses SciTech SNAP Graphics as the OEM graphics driver solution for OS/2, with support for nearly 200 graphic chipsets. SciTech SNAP Graphics allows for the same IBM-certified device driver binaries that are currently used by OS/2 for use under Linux. These drivers can be considered as a complete drop-in replacement for the more generic XFree86 drivers with the advantage of being a completely tested, certified, and source verified solution.

2.1.3 Desktop environments

Unlike OS/2 Presentation Manager, there is no one “standard” desktop environment for Linux. The graphics subsystem (X Window System), window manager, and desktop environment (if any) are all for the most part interchangeable components.

The situation is further compounded by the vast number of programming toolkits and widget libraries. For example, GTK2, motif, Athena, QT, TCL/Tk, and so on, are available for X-based applications.

The net result of all of this is that there is less consistency across various Linux deployments, distributions, and applications. 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 use and discuss KDE and GNOME in this redbook.

KDE provides a full-featured “Windows replacement” style desktop and application environment for Linux and other UNIX-like operating systems. KDE is a mature desktop environment for Linux. It is the default desktop for many distributions. KDE is based on the cross-platform Qt programming library from Trolltech. Significant KDE-based applications include Kmail, the Koffice suite, and the Konqueror Web browser.

You can learn more about KDE on the Web at:

http://www.kde.org

GNOME is an open source desktop environment. It is based on the GTK+, and more recently GTK2, programming libraries. GNOME is the default desktop for Red Hat and Ximian. It is also used by some traditional UNIX vendors including Sun. Significant GNOME applications include the AbiWord word processor, the Evolution mail solution, and the Nautilus file manager.

For more information about GNOME, see:

http://www.gnome.org

Desktop customization

Regardless of the desktop chosen for a particular environment, a feature that most clients want to use is the ability to customize the desktop using scripts. This allows administrators to provide a corporate desktop for end users and to easily add or remove available applications. The following sections deal with customizing and locking down the KDE and GNOME environments.

2.2 KDE desktop

The KDE desktop is much like the OS/2 desktop in that it has a desktop that can have icons and folders placed in a chosen arrangement. It also has a *kicker bar*. This is similar to the OS/2 WarpCenter in the sense that you can add icons to it. However, its functionality has more resemblance to the Windows Start bar.

Figure 2-1 shows a default Red Hat Workstation 3 KDE desktop.

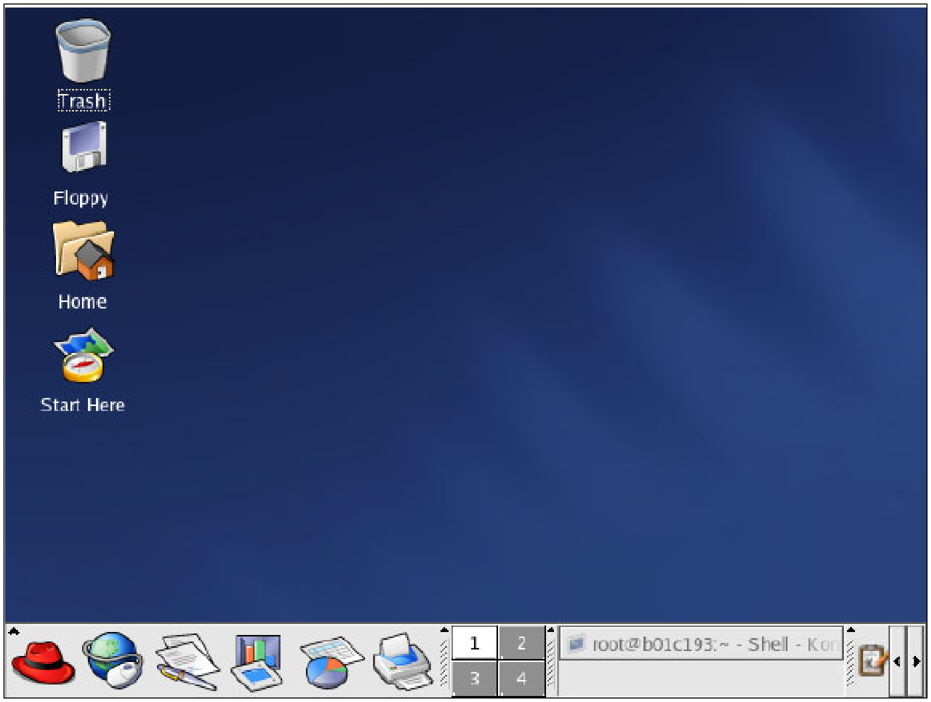


Figure 2-1 KDE desktop

2.2.1 KDE customization and lockdown

Figure 2-2 shows an example of a desktop that we created to be used by a light office user. The applications that we provide are:

- x3270: A 3270 host emulator
- Mozilla: A Web browsing suite
- OpenOffice: An office productivity suite
- LinNeighborhood: A tool to allow easy mounting of Server Messaging Block (SMB) shares
- A kicker bar configured with the same icons as the desktop for easy accessibility

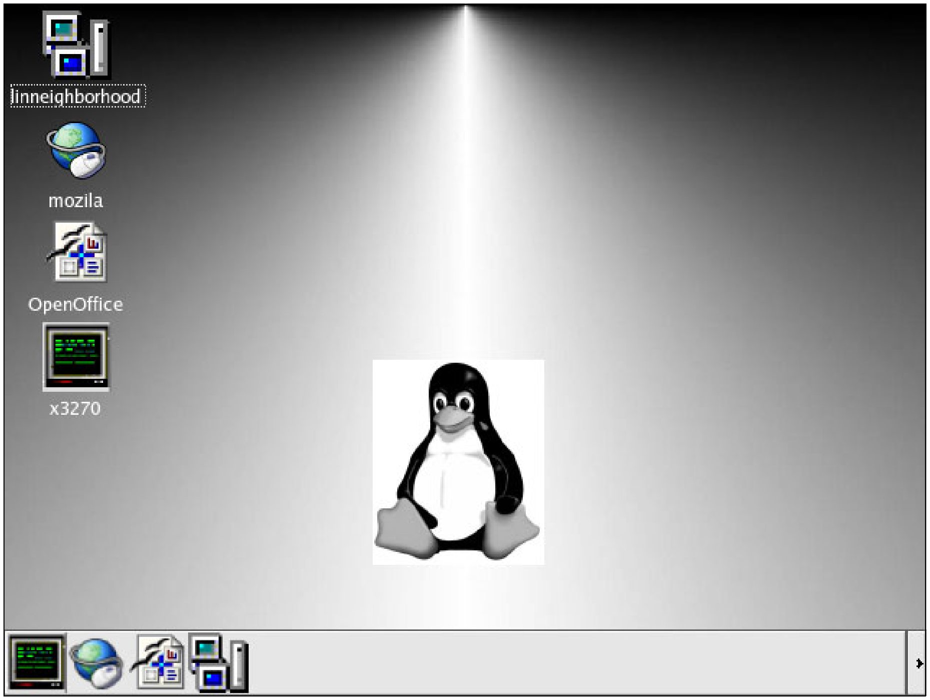


Figure 2-2 A customized KDE desktop

The following sections explain how you can manipulate the KDE environment to create a desktop like the one presented previously. You can achieve this by directly editing the files that control the environment or by using the graphical menus.

KDE environment

As with the OS/2 Presentation Manager desktop, folders are created as a directory structure. Unlike OS/2, these are not centralized in one place since Linux is a true multi-user operating system. The desktop structure is created in the user's home directory. This is shown in the following example for user *fiona*.

```
/home/fiona/Desktop
```

If you want to create a folder for all of Fiona's applications, you can do this from a command line by entering:

```
mkdir /home/fiona/Desktop/"Applications on Demand"
```

Icons are handled differently than with the OS/2 Presentation Manager, since the details are stored in text files. Example 2-1 shows a file that describes a basic icon. This icon file creates a clickable icon that starts the host terminal emulator x3270. The name of the file is *x3270*. This is the name that is displayed on the desktop.

Example 2-1 Icon file

```
[Desktop Entry]
Comment=
Comment[en_US]=3270 Host Emulator
Encoding=UTF-8
Exec=x3270 "myhost.mycompany.com"
GenericName=
GenericName[en_US]=
Icon=/usr/share/icons/Bluecurve/32x32/apps/alevt.png
MimeType=
Name=
Name[en_US]=x3270
Path=
ServiceTypes=
SwallowExec=
SwallowTitle=
Terminal=false
TerminalOptions=
Type=Application
X-KDE-SubstituteUID=false
X-KDE-Username=
```

There are significant entries in an icon configuration file. For example, the following flag is for the desktop environment to understand that this is an icon:

```
[Desktop Entry]
```

The comment [en_US] field is used to display a description in the fly-over bubble help:

```
Comment[en_US]=3270 Host Emulator
```

The Exec field is probably the most important section of an icon file since it is the the field where you specify the executable that the icon is to run. You can specify the full path to the executable. If the executable is found in a directory in the PATH environment variable, then you can specify the file name.

```
Exec=x3270 "myhost.mycompany.com"
```

x3270 is the name of the executable. The text encased in quotation marks is a parameter to be passed to the executable. In this case, it is the name of a Telnet 3270 server that will be connected to when the emulator is started.

To specify an icon (picture) file for a desktop icon, the icon field is used. Linux accepts an icon in any format that is recognized by the system, for example JPG, PNG, or XDM. PNG is the format that is used most widely.

```
Icon=/usr/share/icons/Bluecurve/32x32/apps/alevt.png
```

By saving icon files in the Desktop directory, they appear directly on the desktop. If an uncluttered desktop is preferred, the icons can be saved in a folder on the desktop by creating a subdirectory inside the desktop directory.

Configuring the desktop environment via file manipulation

KDE uses ASCII-based text files for all its configuration options. This makes it easy to configure KDE from scripts and editors. Three main files are used to configure the desktop:

- /usr/share/config/kdeglobals
- /usr/share/config/kdesktoprc
- /usr/share/config/kickerrc

Red Hat 9.0 places these files in the /usr/share/config directory. Other distributions may place these files in other locations, but the editing and customization are exactly the same. By editing these main files, any new user that is created inherits and uses the default settings from these files.

kdeglobals

The default kdeglobals file is used to configure such features as how the icons behave (for example, their size), how they are displayed, and whether a single or double-click is required. This file includes the theme that is to be used on the desktop. Example 2-2 shows a default kdeglobals file.

Example 2-2 Default kdeglobals file

```
[General]
widgetStyle=Bluecurve
alternateBackground=240,240,240
background=230,230,230
buttonBackground=230,230,230
buttonForeground=0,0,0
foreground=0,0,0
linkColor=0,0,192
selectBackground=76,89,166
selectForeground=255,255,255
visitedLinkColor=128,0,128
widgetStyle=Bluecurve
windowBackground=255,255,255
windowForeground=0,0,0
```

```
fixed=Monospace,10,-1,5,50,0,0,0,1,0
font=Sans,10,-1,5,50,0,0,0,0,0
menuFont=Sans,10,-1,5,50,0,0,0,0,0
taskbarFont=Sans,10,-1,5,50,0,0,0,0,0
toolBarFont=Sans,10,-1,5,50,0,0,0,0,0

[KDE]
SingleClick=false
ShowIconsOnPushButtons=true
AntiAliasing=true
ChangeCursor=false
DoubleClickInterval=400
colorScheme=Bluecurve.kcsrc
contrast=1
macStyle=false

[PanelIcons]
Size=48

[DesktopIcons]
Size=48

[WM]
activeBackground=70,79,134
activeBlend=115,127,203
activeForeground=255,255,255
activeTitleBtnBg=207,207,207
inactiveBackground=197,197,197
inactiveBlend=215,215,215
inactiveForeground=127,127,127
inactiveTitleBtnBg=238,238,238

[KSpell]
KSpell_Client=1

[Icons]
Theme=Bluecurve

[WM]
activeFont=Sans,10,-1,5,74,0,0,0,0,0

[KDE Action Restrictions]
action/help_about_kde=false

[KDE Action Restrictions]
action/help_report_bug=false
```

The file is broken into separate sections using square brackets. The KDE section shown in the following example specifies how the desktop interacts with its icons, the mouse, and the color scheme in use. Most values can be changed using the variables true or false.

```
[KDE]
SingleClick=false
ShowIconsOnPushButtons=true
AntiAliasing=true
ChangeCursor=false
DoubleClickInterval=400
colorScheme=Bluecurve.kcsrc
contrast=1
macStyle=false
```

The General section shown in the following example is used to configure such items as the fonts in use on the desktop and the colors used in windows and folders. Items in the General section that deal with colors are configured based on the RGB color format which is also used in OS/2.

```
[General]
widgetStyle=Bluecurve
alternateBackground=240,240,240
background=230,230,230
buttonBackground=230,230,230
buttonForeground=0,0,0
foreground=0,0,0
linkColor=0,0,192
selectBackground=76,89,166
selectForeground=255,255,255
visitedLinkColor=128,0,128
widgetStyle=Bluecurve
windowBackground=255,255,255
windowForeground=0,0,0
fixed=Monospace,10,-1,5,50,0,0,0,1,0
font=Sans,10,-1,5,50,0,0,0,0,0
menuFont=Sans,10,-1,5,50,0,0,0,0,0
taskbarFont=Sans,10,-1,5,50,0,0,0,0,0
toolBarFont=Sans,10,-1,5,50,0,0,0,0,0
```

A second kdeglobals file is created the first time a user logs in to the KDE environment. This is placed in the user's home directory and is created the first time they log on the KDE desktop. For example, user Fiona's second kdeglobals file is in the location /home/fiona/.kde/share/config/kdeglobals.

This second kdeglobals file allows for the configuration of extra variables that are not included in the first kdeglobals file. Example 2-3 shows the default file created when our user first logs on.

Example 2-3 Default user-specific kdeglobals file

```
[$Version]
update_info=kded.upd:kde3.0,kaccel.upd:kde3.1/r3,klippershortcuts.upd:04112002,
socks.upd:kde3.0/r1

[Desktops]
Name_1=
Name_2=
Name_3=
Name_4=
Number=4

[Global Shortcuts]
Desktop Screenshot=default(Ctrl+Print)
Enable/Disable Clipboard Actions=default(Alt+Ctrl+X)
Halt Computer=default(Alt+Ctrl+PageDown)
Halt without Confirmation=none
Kill Window=default(Alt+Ctrl+Escape)
Lock Screen=default(Alt+Ctrl+L)
Logout=default(Alt+Ctrl+Delete)
Logout without Confirmation=default(Alt+Ctrl+Shift+Delete)
Manually Invoke Action on Current Clipboard=default(Alt+Ctrl+R)
Mouse Emulation=default(Alt+F12)
Popup Launch Menu=default(Alt+F1)
Reboot Computer=default(Alt+Ctrl+PageUp)
Reboot without Confirmation=none
Run Command=default(Alt+F2)
Show Klipper Popup-Menu=default(Alt+Ctrl+V)
Show Taskmanager=default(Ctrl+Escape)
Show Window List=default(Alt+F5)
Switch One Desktop Down=none
Switch One Desktop Up=none
Switch One Desktop to the Left=none
Switch One Desktop to the Right=none
Switch to Desktop 1=default(Ctrl+F1)
Switch to Desktop 10=default(Ctrl+F10)
Switch to Desktop 11=default(Ctrl+F11)
Switch to Desktop 12=default(Ctrl+F12)
Switch to Desktop 13=default(Ctrl+Shift+F1)
Switch to Desktop 14=default(Ctrl+Shift+F2)
Switch to Desktop 15=default(Ctrl+Shift+F3)
Switch to Desktop 16=default(Ctrl+Shift+F4)
Switch to Desktop 2=default(Ctrl+F2)
Switch to Desktop 3=default(Ctrl+F3)
Switch to Desktop 4=default(Ctrl+F4)
Switch to Desktop 5=default(Ctrl+F5)
Switch to Desktop 6=default(Ctrl+F6)
Switch to Desktop 7=default(Ctrl+F7)
Switch to Desktop 8=default(Ctrl+F8)
Switch to Desktop 9=default(Ctrl+F9)
Switch to Next Desktop=none
Switch to Next Keyboard Layout=default(Alt+Ctrl+K)
Switch to Previous Desktop=none
Toggle Showing Desktop=default(Alt+Ctrl+D)
Toggle Window Raise/Lower=none
Walk Through Desktop List=default(Ctrl+Tab)
Walk Through Desktop List (Reverse)=default(Ctrl+Shift+Tab)
Walk Through Desktops=none
Walk Through Desktops (Reverse)=none
Walk Through Windows=default(Alt+Tab)
Walk Through Windows (Reverse)=default(Alt+Shift+Tab)
Window Close=default(Alt+F4)
Window Iconify=none
Window Lower=none
Window Maximize=none
Window Maximize Horizontal=none
Window Maximize Vertical=none
Window Move=none
Window Operations Menu=default(Alt+F3)
Window Raise=none
Window Resize=none
Window Screenshot=default(Alt+Print)
Window Shade=none
Window to Desktop 1=none
Window to Desktop 10=none
Window to Desktop 11=none
Window to Desktop 12=none
Window to Desktop 13=none
Window to Desktop 14=none
Window to Desktop 15=none
Window to Desktop 16=none
Window to Desktop 2=none
Window to Desktop 3=none
Window to Desktop 4=none
Window to Desktop 5=none
Window to Desktop 6=none
Window to Desktop 7=none
Window to Desktop 8=none
Window to Desktop 9=none
Window to Next Desktop=none
Window to Previous Desktop=none

[Paths]
Trash=$HOME/Desktop/Trash/
one
```

```
Switch to Next Keyboard Layout=default(Alt+Ctrl+K)
Switch to Previous Desktop=none
Toggle Showing Desktop=default(Alt+Ctrl+D)
Toggle Window Raise/Lower=none
Walk Through Desktop List=default(Ctrl+Tab)
Walk Through Desktop List (Reverse)=default(Ctrl+Shift+Tab)
Walk Through Desktops=none
Walk Through Desktops (Reverse)=none
Walk Through Windows=default(Alt+Tab)
Walk Through Windows (Reverse)=default(Alt+Shift+Tab)
Window Close=default(Alt+F4)
Window Iconify=none
Window Lower=none
Window Maximize=none
Window Maximize Horizontal=none
Window Maximize Vertical=none
Window Move=none
Window Operations Menu=default(Alt+F3)
Window Raise=none
Window Resize=none
Window Screenshot=default(Alt+Print)
Window Shade=none
Window to Desktop 1=none
Window to Desktop 10=none
Window to Desktop 11=none
Window to Desktop 12=none
Window to Desktop 13=none
Window to Desktop 14=none
Window to Desktop 15=none
Window to Desktop 16=none
Window to Desktop 2=none
Window to Desktop 3=none
Window to Desktop 4=none
Window to Desktop 5=none
Window to Desktop 6=none
Window to Desktop 7=none
Window to Desktop 8=none
Window to Desktop 9=none
Window to Next Desktop=none
Window to Previous Desktop=none

[Paths][$i]
Trash=$HOME/Desktop/Trash/
```

This second kdeglobals file allows the user or an administrator to configure virtual desktops and keyboard shortcuts used within KDE. Some examples are shown in the following discussion.

You can add names to virtual desktops and increase or decrease their number. In the following example, we reduced the number of available virtual desktops and gave them specific names:

```
[Desktops]
Name_1=Mozilla
Name_2=x3270
Name_3=
Name_4=
Number=2
```

The Global shortcuts section enables the definition of the keys that are used to manipulate the desktop. The following line is from Example 2-3 on page 17. The Run Command key sequence creates a window from which applications or commands may be issued.

```
Run Command=default(Alt+F2)
```

In the following example, we changed the key stroke from its default to using ALT and R and C:

```
Run Command=(Alt+R+C)
```

An administrator can remove the users' ability to use certain shortcuts by specifying *none* as the shortcut:

```
Run Command=none
```

Merging the two kdeglobals files

When a new user logs on for the first time, their personal KDE environment configuration files are created from the original files that were created by the KDE installation program. These are based on the files located in the /usr/share/config directory. When this first logon is performed and the second kdeglobals file is created, this file is created based on default values as defined to the KDE desktop.

An administrator may want all new users that are created on the workstation to have specific restrictions or, for instance, the same shortcuts. By merging this second file with the first means that all new users who are created on the Linux workstation have the same restrictions or shortcuts.

For example, earlier we discussed changing the key strokes used to execute the Run dialog. Consider the example where the following changes are made to the second kdeglobals file:

```
Run Command=(Alt+R+C)
```

The second kdeglobals file /home/fiona/.kde/share/config/kdeglobals is merged with the original kdeglobal file /usr/share/config/kdeglobals. All new users who are added to the Linux workstation have the key stroke ALT and R and C to execute the Run Dialog.

kdesktoprc

The kdesktoprc file allows such features as the screen saver, background wallpaper, and the individual configurations of the extra virtual desktops. For example, Desktop 1 can have a different configuration than Desktop 2. The file in Example 2-4 is the default installation file found in /usr/share/config.

Example 2-4   Default kdesktoprc file

```
[Background Common]
CacheSize=2048
CommonDesktop=true
Dock=true
Export=false
LimitCache=true

[Desktop0]
BackgroundMode=VerticalGradient
BlendBalance=100
BlendMode=NoBlending
ChangeInterval=60
Color1=138,148,198
Color2=104,112,150
CurrentWallpaper=0
LastChange=0
MinOptimizationDepth=1
MultiWallpaperMode=NoMulti
ReverseBlending=false
UseSHM=false
Wallpaper=/usr/share/backgrounds/images/default.png
WallpaperMode=Scaled

[FMSettings]
NormalTextColor=255,255,255
UnderlineLinks=false
StandardFont=Sans,10,-1,0,50,0,0,0,0,0

[ScreenSaver]
Enabled=true
Lock=false
Priority=19
Saver=KRandom.desktop
Timeout=300
```

In kdesktoprc, you can make such changes as the background wall paper. For example, consider the following line:

```
Wallpaper=/usr/share/backgrounds/images/default.png
```

To specify a specific image to be used as the wallpaper, you can change the previous line to the following line:

```
Wallpaper=/usr/share/backgrounds/images/corporate_logo.jpg
```

The file corporate_logo.jpg is a JPG file not a PNG file. Desktop backgrounds are like the icon files and can be any one of multiple formats.

kickerrc

The kickerrc file is the configuration file that manages and controls the KDE kicker bar. By editing this file, you can choose the icons that are available to the user. You can also add or remove standard KDE applets such as the clock and the workspace selector.

Example 2-5 shows the default kickerrc file as found in /usr/share/config.

Example 2-5 Default kickerrc file

```
[Applet_1]
ConfigFile=kminipagerappletrc
DesktopFile=minipagerapplet.desktop
FreeSpace=0
WidthForHeightHint=77

[Applet_2]
ConfigFile=taskbar_panelappletrc
DesktopFile=taskbarapplet.desktop
FreeSpace=0
WidthForHeightHint=208

[Applet_3]
ConfigFile=klipper_panelappletrc
DesktopFile=klipper.desktop
FreeSpace=1
WidthForHeightHint=28

[Applet_4]
ConfigFile=systemtray_panelappletrc
DesktopFile=systemtrayapplet.desktop
FreeSpace=1
WidthForHeightHint=36
```

```
[Applet_5]
ConfigFile=clockappletrc
DesktopFile=clockapplet.desktop
FreeSpace=1
WidthForHeightHint=75

[DesktopButton_1]
FreeSpace=0

[General]
Applets=KMenuButton_1,ServiceButton_1,ServiceButton_2,ServiceButton_3,ServiceBu
tton_4,ServiceButton_5,ServiceButton_6,Applet_1,Applet_2,Applet_3,Applet_4,Appl
et_5
Size=54
SizePercentage=100
ShowLeftHideButton=false
ShowRightHideButton=false

[KMenuButton_1]
FreeSpace=0

[ServiceButton_1]
DesktopFile=Internet/redhat-web.desktop
FreeSpace=0

[ServiceButton_2]
DesktopFile=Internet/redhat-email.desktop
FreeSpace=0

[ServiceButton_3]
DesktopFile=Office/redhat-word-processor.desktop
FreeSpace=0

[ServiceButton_4]
DesktopFile=Office/redhat-presentations.desktop
FreeSpace=0

[ServiceButton_5]
DesktopFile=Office/redhat-spreadsheet.desktop
FreeSpace=0

[ServiceButton_6]
DesktopFile=Accessories/redhat-print.desktop
FreeSpace=0

[buttons]
EnableIconZoom=false

[KMenu]
```

```
UseSidePixmap=false

[menus]
DetailedMenuEntries=false
UseBookmarks=false
UseBrowser=false
UseRecent=false
```

Within the file in Example 2-5, the main entries define the applications that are used on the kicker bar. For example, the following snippet defines the clock that runs on the kicker bar. The ConfigFile and DesktopFile that configure this applet are again ASCII text files. These files can be complex but this is outside the scope of this redbook.

```
[Applet_5]
ConfigFile=clockappletrc
DesktopFile=clockapplet.desktop
FreeSpace=1
WidthForHeightHint=75
```

The following entries define the look, feel, and behavior of the kicker bar:

```
[buttons]
EnableIconZoom=false

[KMenu]
UseSidePixmap=false

[menus]
DetailedMenuEntries=false
UseBookmarks=false
UseBrowser=false
UseRecent=false
```

Locking down the desktop

With all of the file and potential customization we have described, the administrator may choose to lock these changes. This prevents the user from modifying their desktop (or portions of their desktop). This locking capability helps to ensure that the desktop environments remain standard. It also simplifies the job of the help desk and support staff.

Locking any section or item in the files

Within the files, the [$i] flag is used to mark sections or subsections to lock them. This prevents all users from making modifications to their desktops and saving such modifications.

In this example from the kdedsktoprc file, we added the [$i] flag to the screen saver section heading, [ScreenSaver]. By adding this value, we locked the entire ScreenSaver section from any changes. The user is now unable to apply any changes from their control center.

```
[ScreenSaver][$i]
Enabled=true
Lock=false
Priority=19
Saver=KBanner.desktop
Timeout=60
```

Using the [$i] flag, we can lock individual items within the files. In the following example, we locked the screen saver to Banner, locked the timeout value before the screen locks due to inactivity to one minute, and forced the screen saver to be enabled. This leaves the user the ability to change the remaining values. In this example, the user can change only Priority and Lock.

```
[ScreenSaver]
Enabled[$i]=true
Lock=false
Priority=19
Saver[$i}=KBanner.desktop
Timeout[$i]=60
```

Examples of configured files

The following sections provide a few practical examples of how you may customize the files previously described to meet the requirements of a specific department or enterprise.

kdeglobals

In Example 2-6, we made only one configuration change to the kdeglobals file. However we locked down each section of the file (except one), so the user cannot alter any of the other settings. This example only shows the first two sections of the file.

Example 2-6 Custom kdeglobals file

```
[General]
widgetStyle[$i]=Bluecurve
alternateBackground[$i]=240,240,240
background[$i]=230,230,230
buttonBackground[$i]=230,230,230
buttonForeground[$i]=0,0,0
foreground[$i]=0,0,0
linkColor[$i]=0,0,192
```

```
selectBackground[$i]=76,89,166
selectForeground[$i]=255,255,255
visitedLinkColor[$i]=128,0,128
widgetStyle[$i]=Bluecurve
windowBackground[$i]=255,255,255
windowForeground[$i]=0,0,0
fixed[$i]=Monospace,10,-1,5,50,0,0,0,1,0
font[$i]=Sans,10,-1,5,50,0,0,0,0,0
menuFont[$i]=Sans,10,-1,5,50,0,0,0,0,0
taskbarFont[$i]=Sans,10,-1,5,50,0,0,0,0,0
toolBarFont[$i]=Sans,10,-1,5,50,0,0,0,0,0

[KDE][$i]
SingleClick=true
ShowIconsOnPushButtons=true
AntiAliasing=true
ChangeCursor=false
DoubleClickInterval=400
colorScheme=Bluecurve.kcsrc
contrast=1
macStyle=false
.
.
.
```

In Example 2-6, notice that we separately locked each item in the [General] section of the kdeglobals. In testing, we found that if we lock the entire section [General][$i], then any changes that we made to the kickerrc file to customize the kicker bar are ignored.

kdesktoprc

Example 2-7 is an excerpt from a merged kdesktoprc file that contains an excerpt of the [Global Shortcuts] section. In this example, we edited many of the defaults to remove a lot of features to which we did not want users to have access. By adding the [$i] flag that was discussed earlier, the user cannot re-enable the shortcuts that were disabled. You can define shortcut keystrokes to give users an experience similar to what is provided by the OS/2 Work Place Shell.

Example 2-7 Custom kdesktoprc file

```
[Icons][$i]
Theme=Bluecurve

[WM][$i]
activeFont=Sans,10,-1,5,74,0,0,0,0,0
```

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

[KDE Action Restrictions][$i]
action/help_report_bug=false
[Global Shortcuts][$i]
Desktop Screenshot=none
Enable/Disable Clipboard Actions=none
Halt Computer=none
Halt without Confirmation=none
Kill Window=default(Alt+Ctrl+Escape)
Lock Screen=default(Alt+Ctrl+L)
Logout=default(Alt+Ctrl+Delete)
Logout without Confirmation=none
Manually Invoke Action on Current Clipboard=default(Alt+Ctrl+R)
Mouse Emulation=default(Alt+F12)
Popup Launch Menu=none
Reboot Computer=none
Reboot without Confirmation=none
Run Command=none
Show Klipper Popup-Menu=none
Show Taskmanager=none
Switch to Desktop 1=default(Ctrl+F1)
Switch to Desktop 2=default(Ctrl+F2)
```

Example kdesktoprc

Example 2-8 shows a kdesktoprc file with the [$i] flag added to all sections. Also in this file, we manually added the [Mouse Buttons] section and configured the right mouse button to have no action. This disables the users' ability to right-click the desktop and access the menu. This does not affect right-clicking in applications or on the kicker bar. We also changed the default background to a corporate background which is common for all users and is unchangeable.

Example 2-8 Custom kdestoprc with the right mouse button disabled

```
[Background Common][$i]
CacheSize=2048
CommonDesktop=true
Dock=true
Export=false
LimitCache=true

[Desktop0][$i]
BackgroundMode=VerticalGradient
BlendBalance=100
BlendMode=NoBlending
```

```
ChangeInterval=60
Color1=138,148,198
Color2=104,112,150
CurrentWallpaper=0
LastChange=0
MinOptimizationDepth=1
MultiWallpaperMode=NoMulti
ReverseBlending=false
UseSHM=false
Wallpaper=/usr/share/backgrounds/images/corporate_logo.jpg
WallpaperMode=Scaled

[FMSettings][$i]
NormalTextColor=255,255,255
UnderlineLinks=false
StandardFont=Sans,10,-1,0,50,0,0,0,0,0

[Mouse Buttons]
Right[$i]=none

[ScreenSaver][$i]
Enabled=true
Lock=true
Priority=19
Saver=KMatrix.desktop
Timeout=150
```

Examples of configured kicker bars

Example 2-9 shows the kicker bar locked down without any icons or extra desktops available. The file that controls the kicker bar is called *kickerrc.* In this example, again you can see the use of the [$i] flag. If the user does make any local changes to the task bar, these are disregarded at the next logon.

Example 2-9 Custom kickerrc file

```
[Applet_1][$i]
ConfigFile[$d]
DesktopFile[$d]
FreeSpace[$d]
WidthForHeightHint[$d]

[Applet_2][$i]
ConfigFile[$d]
DesktopFile[$d]
FreeSpace[$d]
WidthForHeightHint[$d]
```

```
[Applet_3][$i]
ConfigFile[$d]
DesktopFile[$d]
FreeSpace[$d]
WidthForHeightHint[$d]

[Applet_4][$i]
ConfigFile[$d]
DesktopFile[$d]
FreeSpace[$d]
WidthForHeightHint[$d]

[Applet_5][$i]
ConfigFile[$d]
DesktopFile[$d]
FreeSpace[$d]
WidthForHeightHint[$d]

[General][$i]
Applets=

[KMenuButton_1][$i]
FreeSpace[$d]

[ServiceButton_1][$i]
DesktopFile[$d]
FreeSpace[$d]

[ServiceButton_6][$i]
DesktopFile[$d]
FreeSpace[$d]
```

Example 2-10 shows a kicker bar that is fully locked down but is preconfigured with the x3270 application icon. This user only needs access to mainframe applications via a 3270 emulator.

Example 2-10 Custom kickerrc with x3270 icon

```
[Applet_1][$i]
ConfigFile[$d]
DesktopFile[$d]
FreeSpace[$d]
WidthForHeightHint[$d]

[Applet_2][$i]
ConfigFile[$d]
```

```
DesktopFile[$d]
FreeSpace[$d]
WidthForHeightHint[$d]

[Applet_3][$i]
ConfigFile[$d]
DesktopFile[$d]
FreeSpace[$d]
WidthForHeightHint[$d]

[Applet_4][$i]
ConfigFile[$d]
DesktopFile[$d]
FreeSpace[$d]
WidthForHeightHint[$d]

[Applet_5][$i]
ConfigFile[$d]
DesktopFile[$d]
FreeSpace[$d]
WidthForHeightHint[$d]

[General][$i]
Applets=URLButton_1

[KMenuButton_1][$i]
FreeSpace[$d]

[ServiceButton_1][$i]
DesktopFile[$d]
FreeSpace[$d]

[URLButton_1][$i]
FreeSpace=0
URL=file:/usr/share/apps/kicker/x3270

[ServiceButton_6][$i]
DesktopFile[$d]
FreeSpace[$d]
```

The changes in Example 2-10 are in the sections [General][$i] and [URLButton_1][$i]. In the general section, we added the following string:

```
Applets-URLButton_1
```

This is required to allow the definition of [URLButton_1][$i] to function. As explained earlier, icons are based on text files. This is the same for icons that are placed on the kicker bar. To share these icons between all users, a shared directory is required. We use the /usr/share/apps/kicker directory to store kicker bar icon files.

Within this directory, we can place all the shared kicker applications that we want our users to access. Only the applications that are defined in the kickerrc appear on the kicker bar so all application definitions can be placed in the one directory.

The definition that is used to create the button on the desktop is:

```
[URLButton_1][$i]
FreeSpace=0
URL=file:/usr/share/apps/kicker/x3270
```

The entry `URL=file` defines the location of the icon file, which is in the directory that was created earlier.

Restricting the desktop

When a user is added and the home directory structure is created, the user's desktop is not created by KDE until their first logon. Before the user logs on, the administrator can create the user's desktop icons and folders manually using the command line (for example):

```
mkdir /home/fiona/Desktop
mkdir /home/fiona/Desktop/"Applications on demand"
```

By doing this, the default KDE environment's Home, Trash, Start Here, and Floppy icons are not created.

Restricting the desktop from having permanent changes made to it by a user is an easy process. We performed this from the command line using a simple, common Linux command.

The current directory structure for the user Fiona is:

- /home/fiona/Desktop
- /home/fiona/Desktop/Applications on demand/x3270
- /home/fiona/Desktop/Applications on demand/mozilla

We want Fiona to have access to these applications to run them but 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, the administrator can limit Fiona's ability to write to these directories by using the following command,

```
chmod -R 755 /home/<username>/Desktop
```

This allows the user to execute the icons and open the folders, but not make any changes to them.

Note: Using **chmod** on the desktop directory structure does not affect the kicker bar.

Resulting desktop

Figure 2-3 shows the desktop that we created after performing the actions in the previous sections. As discussed in Chapter 7, “Linux client installation” on page 145, we can create this desktop during the installation process. This enables more efficient deployment and administration by the desktop system administrator.

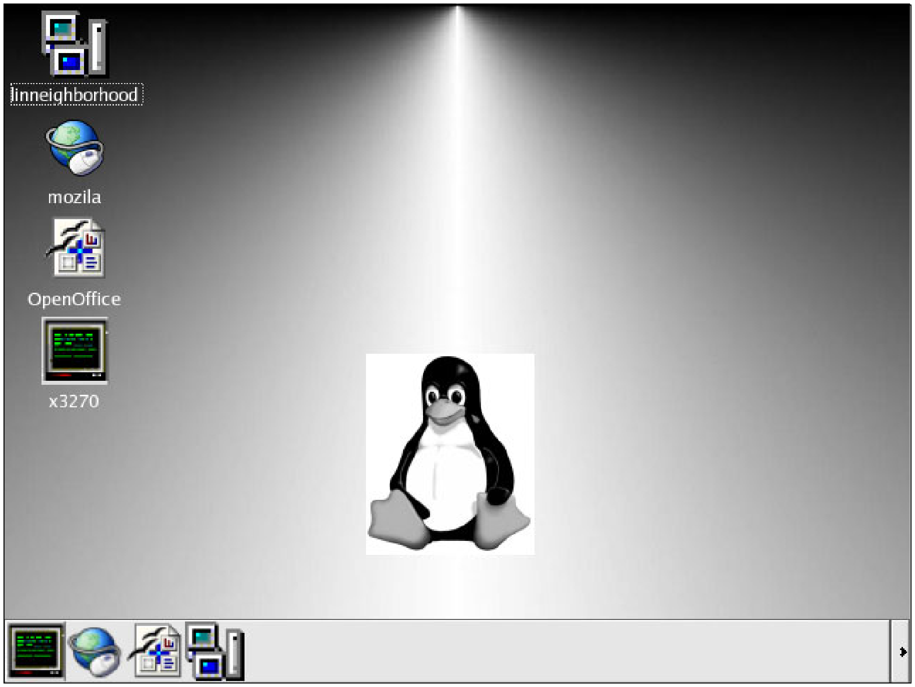


Figure 2-3 Customized desktop

2.2.2 Configuring the KDE desktop environment using the GUI

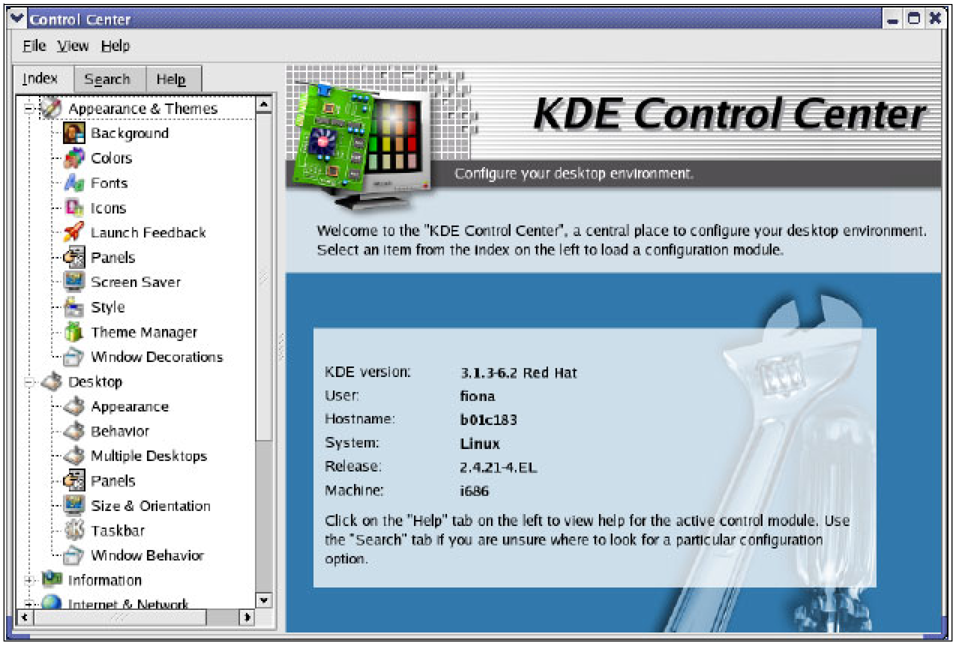
This section explains how to configure the KDE desktop environment from the GUI using the KDE Control Center.

Note: When you make changes to the desktop using the GUI, changes are saved in files in the /home/<user>/.kde/share/config directory. The files that contain these changes are kdeglobals, kdesktoprc, and kickerrc as discussed earlier. These files only contain the personalization made for the current logged on user.

All configuration for the KDE desktop environment is performed using the KDE Control Center. You invoke this program by using the kicker bar:

```
“Red Hat Menu” > Select Control Center
```

Figure 2-4 shows the Control Center with the Appearance & Themes and Desktop options expanded.



Welcome to the "KDE Control Center", a central place to configure your desktop environment. Select an item from the index on the left to load a configuration module.

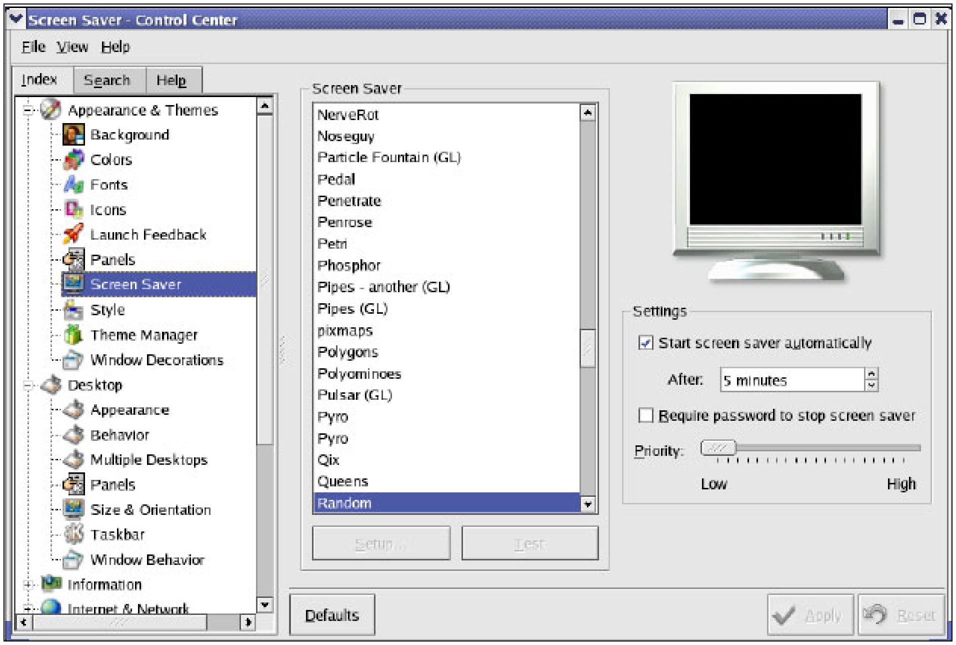
KDE version: 3.1.3-6.2 Red Hat
User: fiona
Hostname: b01c183
System: Linux
Release: 2.4.21-4.EL
Machine: i686

Click on the "Help" tab on the left to view help for the active control module. Use the "Search" tab if you are unsure where to look for a particular configuration option.

Figure 2-4 KDE Control Center

Appearances and Themes

The menu on the left allows the user to configure all options that pertain to the appearance and themes used within KDE. To configure the various options, use the mouse pointer to select the item to personalize on the left side. For example, if the Screen Saver icon on the left is selected, the user sees the window shown in Figure 2-5.



Settings

☑ Start screen saver automatically

After: 5 minutes

□ Require password to stop screen saver

Priority: Low High

Figure 2-5 Screen Saver configuration

After you personalize the screen saver and its options, click **Apply** and exit the Control Center.

> **Tip:** Viewing the personal files that are created in the.kde/share/config directory is a good way to learn how to configure the main KDE configuration files using file manipulation.

The following options are configurable within the Appearance & Themes option:

- Background
- Colors
- Fonts
- Icons
- Launch feedback
- Panels (some kicker bar options)
- Screen Saver
- Style
- Theme manager
- Window decorations

After you make a change to the user’s screen saver, the /.kde/share/config/kdesktoprc file is created in the user’s KDE directory.

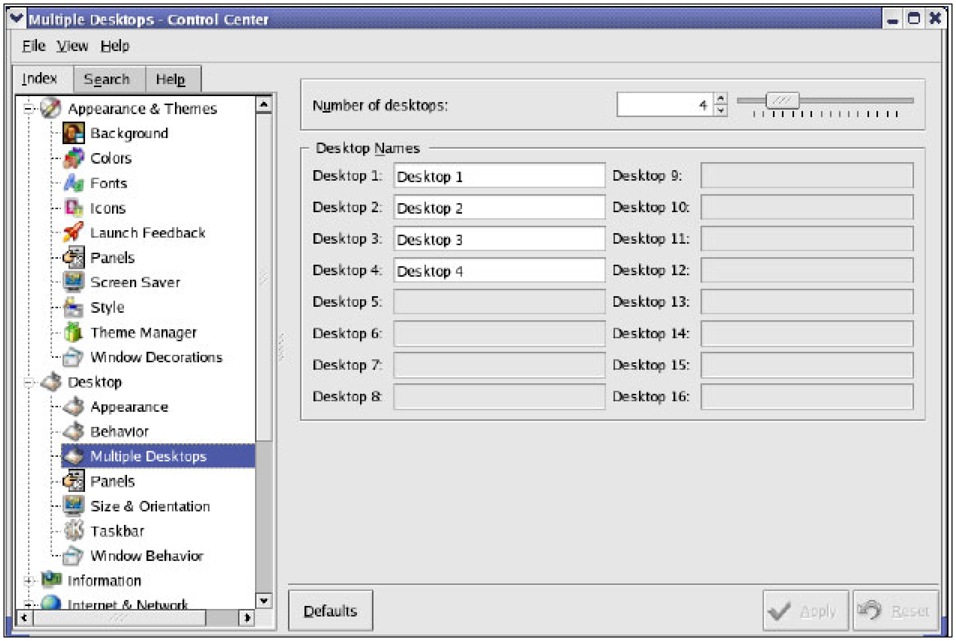
The contents of the file are:

```
[ScreenSaver]
Saver=xmatrix.desktop

[Version]
KDEVersionMajor=3
KDEVersionMinor=1
KDEVersionRelease=3
```

As mentioned earlier, this file is a subset of the main kdesktoprc. It contains only configuration information that was changed by the user.

Desktop

The menu on the left also allows the user to configure all options that pertain to the Desktop used within KDE. To configure the various options, again use the mouse pointer to select the item to personalize. For example, select the Multiple Desktops icon on the left. Then you see the window shown in Figure 2-6.



Number of desktops: 4

Desktop Names

Desktop 1: Desktop 1
Desktop 2: Desktop 2
Desktop 3: Desktop 3
Desktop 4: Desktop 4
Desktop 5:
Desktop 6:
Desktop 7:
Desktop 8:
Desktop 9:
Desktop 10:
Desktop 11:
Desktop 12:
Desktop 13:
Desktop 14:
Desktop 15:
Desktop 16:

Figure 2-6 Desktops configuration

After you personalize the virtual desktops, you click the Apply button and exit the Control Center.

The following options are configurable within the Desktop option:

- Appearance
- Behavior
- Multiple desktops
- Panel (more kicker options)
- Size and orientation
- Taskbar (more kicker options)
- Window behavior

While within the control center, you can edit and customize the kicker bar. It has many characteristics in common with the OS/2 WarpCenter and Launchpad. If the icon to be added to the kicker bar already exists on the desktop, select it with the left mouse button and drag and drop it onto the kicker bar.

2.3 GNOME desktop

The GNOME default desktop looks similar to the default KDE desktop. This is due to the Red Hat distribution using the BlueCurve Scheme throughout its default desktops.

Much the same as KDE, the GNOME desktop has a desktop upon which icons and folders can be placed in a chosen arrangement. As with KDE, it also has a launch panel called the *GNOME-Panel*. It can include icons to launch programs and select one of several desktops.

Figure 2-7 shows a default Red Hat GNOME desktop.

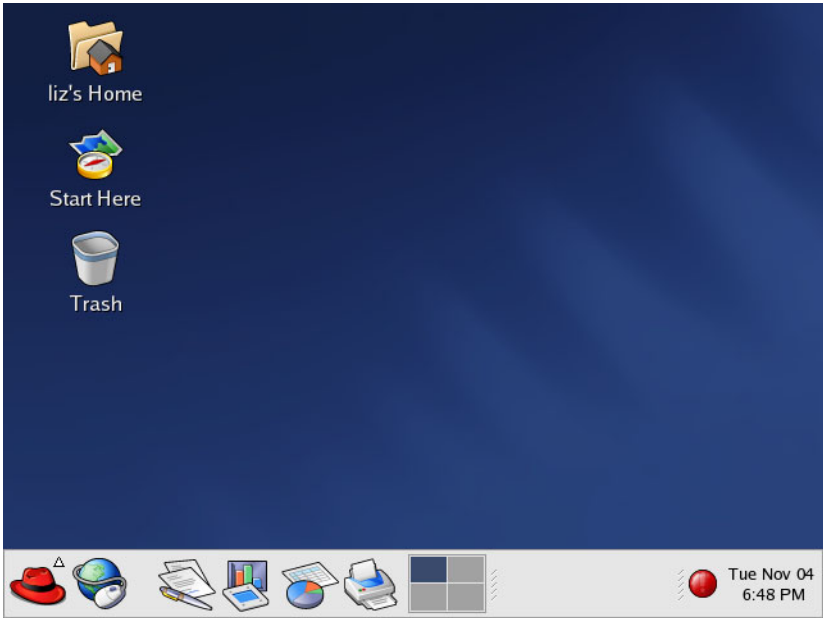


Figure 2-7 GNOME desktop

2.3.1 GNOME customization and lockdown

We use a desktop that we created to be used by a light office user. The applications we provided are the same as for a light office KDE user. They are:

- x3270
- Mozilla
- OpenOffice
- LinNeighborhood

A brief explanation of the GNOME desktop

As with the KDE desktop, folders are created as a directory structure. They are not centralized in one place since Linux is a true multi-user operating system. The desktop structure within GNOME is slightly different than that of KDE. In GNOME, the desktop directory is hidden and renamed as /home/fiona/.gnome-desktop.

As we did before, we manipulated the desktop structure in the same way as in KDE as shown here to reflect the difference in the GNOME desktop structure:

```
mkdir /home/fiona/.gnome-desktop/Applications\ on\ Demand
```

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/fiona/.gnome-desktop directory to populate the desktop.

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/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*, which is shown in Example 2-11.

Example 2-11 %gconf.xml file

```
/etc/gconf/gconf.xml.defaults/desktop/gnome/background

<?xml version="1.0"?>
<gconf><entry name="draw_background"
mtime="1067344442"schema="/schemas/desktop/gnome/background/draw_background"/>
<entry name="picture_options" mtime="1067344442"
schema="/schemas/desktop/gnome/background/picture_options"/>
<entry name="picture_filename" mtime="1067344442"
schema="/schemas/desktop/gnome/background/picture_filename"/>
<entry name="picture_opacity" mtime="1067344442"
schema="/schemas/desktop/gnome/background/picture_opacity"/>
<entry name="primary_color" mtime="1067344442"
schema="/schemas/desktop/gnome/background/primary_color"/>
<entry name="secondary_color"mtime="1067344442"
schema="/schemas/desktop/gnome/background/secondary_color"/>
<entry name="color_shading_type" mtime="1067344442"
```

```
schema="/schemas/desktop/gnome/background/color_shading_type"/>
</gconf>
```

Configuring the desktop from the command line

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 relate to GNOME 2.2 and its tools as shipped with Red Hat Workstation 3. Some of the switches used in **gconf** have changed in later versions.

The following example resets the default background and make 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 work spaces available to the user. In this instance, we set two work spaces, since this user only has access to and uses two applications. With one in each work space, they can run each application 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 customized our desktop and explain each of the commands that are used.

Configuring the GNOME panel

To configure the GNOME panel, you use the **gconftool-2**. The configuration files are based on XML as discussed earlier. 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
```

Restricting the desktop

As shown in 2.2, “KDE desktop” on page 11, 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, their desktop structure is not created until they first logon. Again by manually creating this desktop structure, the default is not created.

```
mkdir /home/fiona/.gnome-desktop
mkdir /home/fiona/.gnome-desktop/Applications\ on\ demand
```

Restricting the desktop from having permanent changes made to it by a user is the same process as before with KDE. The current directory structure for user Fiona is:

- /home/fiona/.gnome-desktop
- /home/fiona/.gnome-desktop/Applications on demand/x3270
- /home/fiona/.gnome-desktop/Applications on demand/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 use the following command:

```
chmod -R 755 /home/<username>/.gnome-desktop
```

This allows the user to execute the icons and open the folders, but not to make any changes to them.

Configuring the GNOME desktop environment using the GUI

This section explains how to configure the GNOME desktop environment from the GUI using the GNOME preferences folder.

> **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, select **Red Hat Menu -> Preferences**.

You can select all of the preferences from within the displayed menu. If you select Red Hat Menu -> Preferences and select Control Center, a folder reminiscent of the Windows Control Panel opens. Figure 2-8 shows the GNOME Control Center folder.

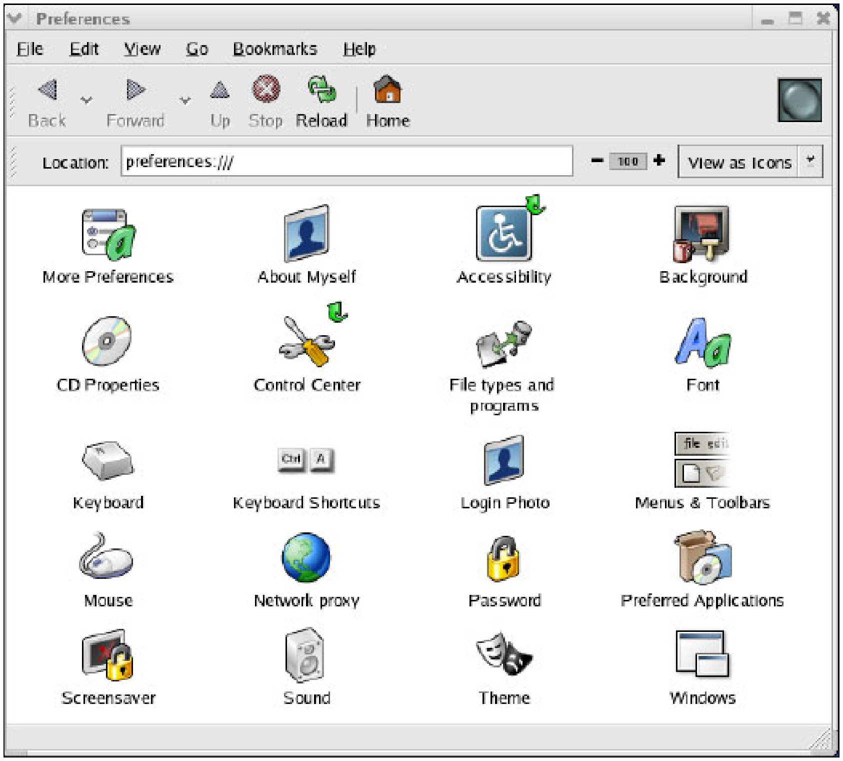
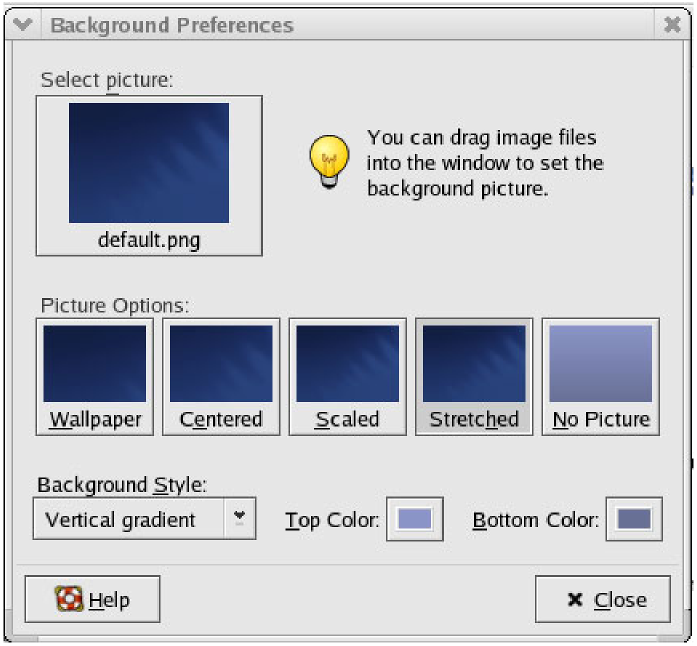


Figure 2-8 GNOME Control Center

Unlike the KDE Control Center that provides a single interface to configure various aspects of the desktop, the GNOME Control Center presents a myriad of different programs to configure the GNOME environment.

To change the desktop background image, select the **Background** icon. The Background Preferences window (Figure 2-9) opens.

To browse the directory structure to select a new image, under Select picture, select the blue icon. You can select from the available options how you want to display the image, for example, scaled or centered.



Select picture: default.png

You can drag image files into the window to set the background picture.

Picture Options: Stretched

Background Style: Vertical gradient

Figure 2-9 GNOME background preferences

2.3.2 Roaming users

For information about using the preconfigured desktop solutions for KDE and GNOME to allow users to roam, see Chapter 6, “Migration considerations” on page 137.

2.4 Printing

Although local printing support for Linux is fairly robust, network printing historically is more difficult. However, that is changing. Much of the difficulty associated with network printing is due to the weaknesses of the classic line printer requester (LPR)/line printer daemon (LPD) system used by most UNIX-based operating systems, including Linux. In recent years, two major open source projects have emerged that attempt to address this situation.

2.4.1 CUPS

Common UNIX Printing System (CUPS) is a new printing system based on the Internet Printing Protocol (IPP). In addition to IPP, CUPS includes limited compatibility with the classic UNIX LPR/LPD system and other proprietary protocols. Most recent distributions (including Red Hat and UnitedLinux-based distributions) have adopted CUPS as their default printing subsystems.

Windows XP supports IPP and can detect and use CUPS-based queues without special configuration. IPP support can be added to Windows 2000 through an optional download. For this reason, CUPS may be the preferred printing subsystem when access to Linux-based printer queues is required in a mixed Linux and Windows client environment.

If you are using Samba to control access to network resources, you can share and manage access to CUPS-based printer queues from Samba. Depending on the distribution, this may require a recompilation of Samba to include CUPS support. See the Samba Web site for more information:

http://www.samba.org

For more information about IPP, see:

http://www.pwg.org/ipp/

2.4.2 LPRng

LPRng is a new printing system. Its primary aim is to address many of the deficiencies of the classic UNIX LPR printing system (RFC1179) while maintaining maximum compatibility with the old system. Improvements in LPRng include dynamic queue redirection, printer pooling (one queue sharing several printers), and a more robust security model.

Compared to CUPS, LPRng is generally thought of as the more mature alternative. For this reason, it was the default printing system for many distributions until recently. LPRng may still be preferred when integrating Linux clients into environments where the classic LPR/LPD system is firmly entrenched.

If a Linux server is used to manage printer queues in a mixed Windows and Linux environment, you can use Samba to provide and manage access to LPRng-based queues from Windows clients. For more information, see:

http://www.lprng.com/LPRng-HOWTO-Multipart/smb.htm

You can also find general information about LPRng on the Web at:

http://www.lprng.org/

2.4.3 Print driver solutions

The drivers for most modern PostScript printers are implemented using a platform-independent PostScript Printer Description (PPD) file. If you are attempting to use a PostScript printer that is not supported “out of the box” by a particular distribution, you can often use the PPD file supplied with the Windows driver on Linux. Consult the documentation for the Linux distribution or printing subsystem (CUPS or LPRng) for instructions about using Windows PPD files.

IBM has contributed an OMNI driver framework to open source. It improves support for non-PostScript printers on Linux. The OMNI driver provides a universal core that implements the common set of functions supported by most printers. This simplifies the development of printer drivers, since the driver developer only needs to be concerned with implementing support for the unique features of a particular device. Most recent distributions ship the OMNI drivers. You can obtain the latest IBM version from the Web at:

`http://www.ibm.com/developer/opensource/linux/projects/omni/`

2.4.4 Creating printer definitions using lpadmin

The Red Hat and SuSE distributions both have a common method for setting up printers from the command line. The command can be used within a script during the installation or at a later date if the environment changes and new printers need to be added.

As mentioned earlier, each printer to be set up needs a driver. To print, Linux only needs one file from a Windows driver package. The file in question is the PPD file. Follow these steps:

1. Download the required printer device driver.
2. Unpack the package using **`unzip`** or **`tar`**, for example.
3. Locate the required PPD file, for example, Ibm27701.ppd.

You can use **`lpadmin`** to set up:

- Locally attached printers
- Networked printers using LPR
- Networked printers using IPP
- SMB printers (Samba and Windows)

The following list shows examples of using **`lpadmin`** to define an IBM Infoprint® 70 in various contexts. The PPD file is called Ibm27701.ppd. We saved this file in the /usr/share directory. For more complete information, access the man pages for **`lpadmin`**.

- For a locally attached printer:

```
lpadmin -p IBMInfoprint70Local -E -v parallel:/dev/lp0 -P
/usr/share/Ibm27701.ppd
```

- For a network printer and to connect to it via LPR:

```
lpadmin -p IBMInfoprint70LPR -E -v lpd://printer_ip_address/queuename -P
/usr/share/Ibm27701.ppd
```

- To install a network printer using IPP:

```
lpadmin -p IBMInfoprint70IPP -E -v
ipp://server_ip_address/printers/queuename -P /usr/share/Ibm27701.ppd
```

- To install a network printer using SMB:

```
lpadmin -p IBMInfoprint70SMB -E -v smb://uid:pwd@server_name//share -P
/usr/share/Ibm27701.ppd
```

2.4.5 Creating printers using printconf-tui (Red Hat)

You can use the **printconf-tui** tool, from a command shell, to install printers to Red Hat. There are advantages to using this tool to create printers instead of using **lpadmin**. For example, after you create a reference printer and export the configuration, you can edit the configuration file easily to make such changes as changing the default paper, which is set to U.S. letter by default.

Manually installing a printer using printconf-tui

For first-time users, the following example acts as a guide for using **printconf-tui**. In this example, we set up a locally attached printer.

1. From the Red Hat Printer Config window (Figure 2-10), select **New**.

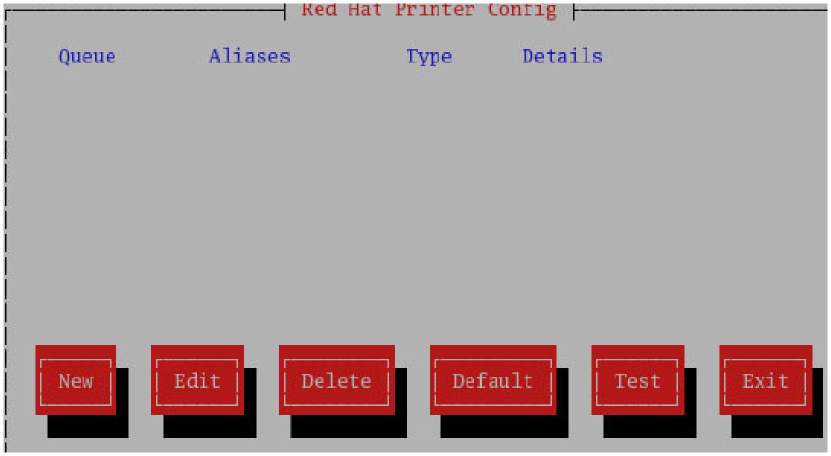


Figure 2-10 Initial set up window

2. You see a list where you can select the type of printer to set up (Figure 2-11). In this example, we select a locally attached printer. Click **Next**.

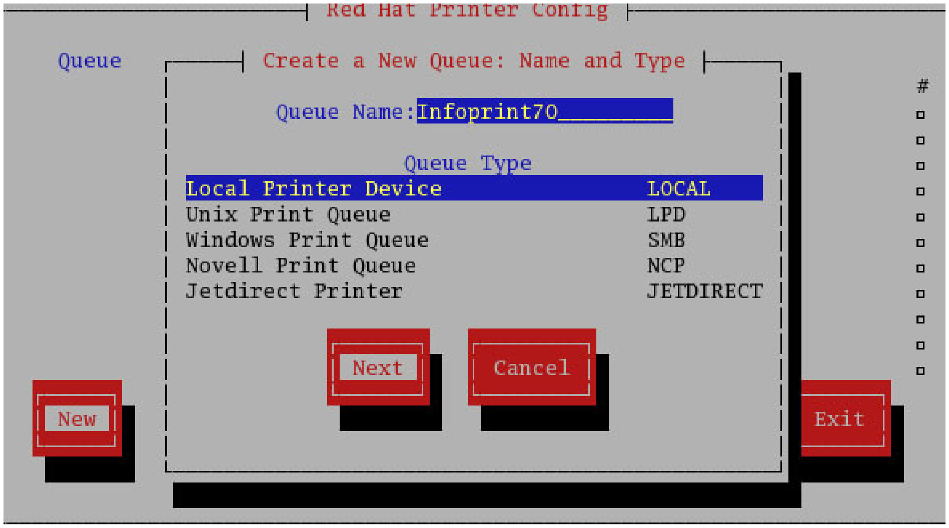

```
Red Hat Printer Config
Create a New Queue: Name and Type

Queue Name:Infoprint70

Queue Type
Local Printer Device                LOCAL
Unix Print Queue                    LPD
Windows Print Queue                 SMB
Novell Print Queue                  NCP
Jetdirect Printer                   JETDIRECT
```

Figure 2-11 Select queue type

3. On the next window (Figure 2-12), you see the device to which the local printer is connected. In our example, it is /dev/lp0. Click **Next**.

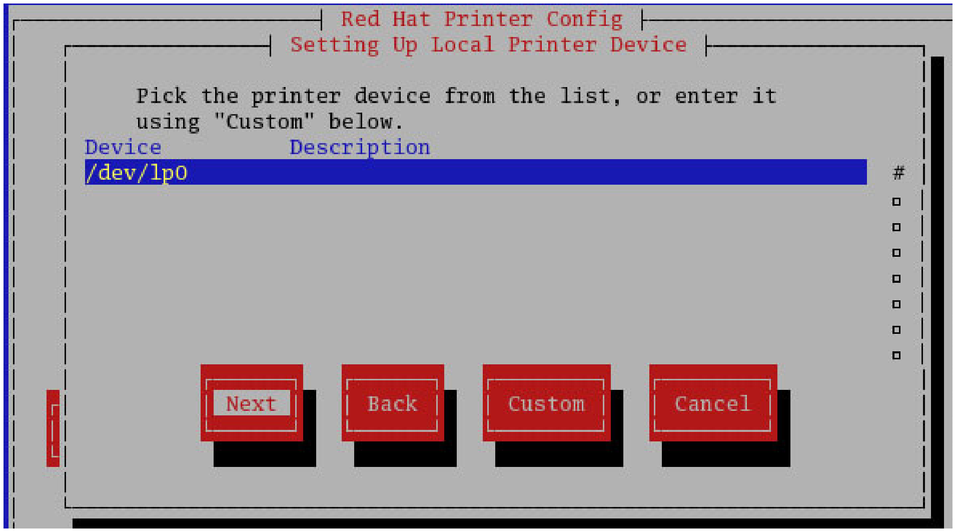
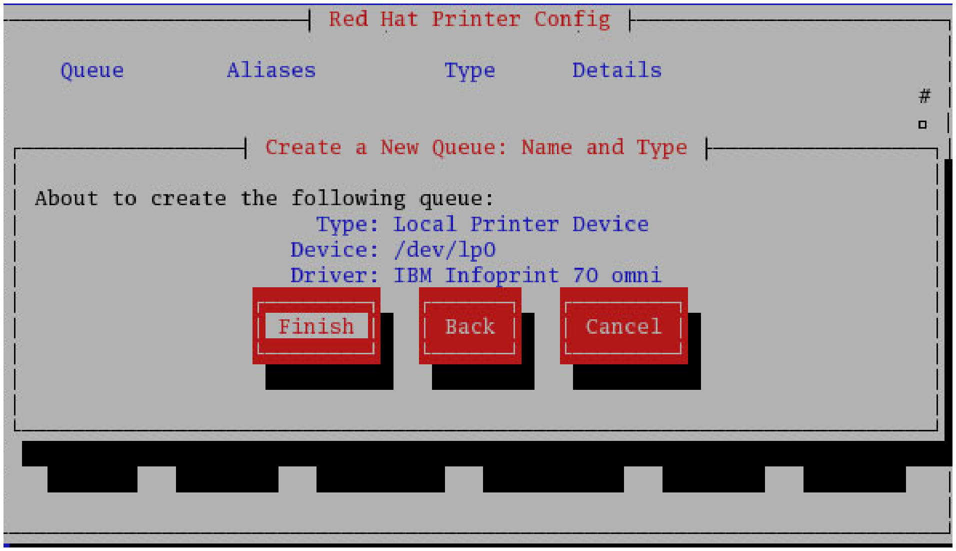

```
Red Hat Printer Config
Setting Up Local Printer Device

     Pick the printer device from the list, or enter it
     using "Custom" below.
Device          Description
/dev/lp0
```

Figure 2-12 Choosing the device

4. You now see the printer device driver list. In this example, we use IBM Infoprint 70. Select this driver and click **Next**.
5. On the final window (Figure 2-13), review the settings that you chose. If all is correct, click **Finish** and then exit the program.



```
About to create the following queue:
             Type: Local Printer Device
           Device: /dev/lp0
           Driver: IBM Infoprint 70 omni
```

Figure 2-13 Defined settings for the new printer

Setting up local printers

To set up printers in an unattended state, for example during installation or remotely when a user requires a new installation, we use the **`printconf-tui`** program. Run this program on an installed reference system to create a printer configuration that is replicated to other systems.

Using the configuration program in the previous section, we created a local printer called Infoprint70. This is setup to print to /dev/lp0 (the parallel port). After you create this printer, enter the following command on a terminal session:

```
printconf-tui --Xexport > infoprint70.xml
```

By using the **`export`** command, we create a text file in XML format. It contains all of the printer definition information. The redirection symbol (>) is the same as in OS/2. It pipes the data to a filename of our choice (infoprint70 in our example).

Use the following command to display the XML printer configuration data:

```
printconf-tui --Xexport
```

Example 2-12 shows the configuration data.

Example 2-12 Configuration data using printconf-tui

```
<?xml version="1.0"?>
<adm_context VERSION="0">
  <id NAME="local" SERIAL="1067523338">
    <null/>
    <null/>
    </id>
  <datatree>
    <printconf TYPE="LIST">
      <print_queues TYPE="LIST">
        <Infoprint70 ATOMIC="TRUE" TYPE="LIST">
          <alias_list ANONYMOUS="TRUE" TYPE="LIST">
            </alias_list>
          <lpoptions TYPE="LIST">
            <page-bottom TYPE="STRING" VALUE="36"/>
            <cpi TYPE="STRING" VALUE="12"/>
            <lpi TYPE="STRING" VALUE="7"/>
            <scaling TYPE="STRING" VALUE="100"/>
            <page-right TYPE="STRING" VALUE="36"/>
            <page-left TYPE="STRING" VALUE="36"/>
            <wrap TYPE="STRING" VALUE="true"/>
            <page-top TYPE="STRING" VALUE="36"/>
            </lpoptions>
          <queue_type TYPE="STRING" VALUE="LOCAL"/>
          <queue_data TYPE="LIST">
            <local_printer_device TYPE="STRING" VALUE="/dev/lp0"/>
            </queue_data>
          <filter_data TYPE="LIST">
            <mf_type TYPE="STRING" VALUE="MFOMATIC"/>
            <flags TYPE="LIST">
              <assume_data_is_text TYPE="BOOL" VALUE="FALSE"/>
              <rerender_Postscript TYPE="BOOL" VALUE="FALSE"/>
              <convert_text_to_Postscript TYPE="BOOL" VALUE="TRUE"/>
              </flags>
            <printer_id TYPE="STRING" VALUE="IBM-Infoprint_70"/>
            <gs_driver TYPE="STRING" VALUE="omni"/>
            <foomatic_defaults ANONYMOUS="TRUE" TYPE="LIST">
              </foomatic_defaults>
            </filter_data>
          <filter_type TYPE="STRING" VALUE="MAGICFILTER"/>
          </Infoprint70>
        </print_queues>
      </printconf>
    </datatree>
  </adm_context>
```

After you save this file, you can import it into any other system and set it up without any input from the user. A system administrator can execute its installation from a Secure Shell (SSH) session or make it a part of a remote installation process as discussed in Chapter 7, “Linux client installation” on page 145.

Enter the following command to install the chosen printer using the configuration file:

```
printconf-tui --Ximport < infoprint70.xml
```

The locally attached InfoPrint 70 is now ready for use.

Connecting to SMB print servers

To connect a client to an SMB-based printer, use **printconf-tui** and select the option to install an SMB printer instead of a local printer as describe earlier. After you create the printer definition, export its configuration as you did earlier:

```
printconf-tui --Xexport >infoprint70smb.xml
```

Another XML file is created and has the values as shown in Example 2-13.

Example 2-13 XML file created

```
<?xml version="1.0"?>
<adm_context VERSION="0">
  <id NAME="local" SERIAL="1067526465">
    <null/>
    <null/>
    </id>
  <datatree>
    <printconf TYPE="LIST">
      <print_queues TYPE="LIST">
        <SMBIBM70 ATOMIC="TRUE" TYPE="LIST">
          <alias_list ANONYMOUS="TRUE" TYPE="LIST">
            </alias_list>
          <lpoptions TYPE="LIST">
            <page-bottom TYPE="STRING" VALUE="36"/>
            <cpi TYPE="STRING" VALUE="12"/>
            <lpi TYPE="STRING" VALUE="7"/>
            <scaling TYPE="STRING" VALUE="100"/>
            <page-right TYPE="STRING" VALUE="36"/>
            <page-left TYPE="STRING" VALUE="36"/>
            <wrap TYPE="STRING" VALUE="true"/>
            <page-top TYPE="STRING" VALUE="36"/>
            </lpoptions>
          <queue_type TYPE="STRING" VALUE="SMB"/>
          <queue_data TYPE="LIST">
            <smb_share TYPE="STRING" VALUE="info70"/>
            <smb_ip TYPE="STRING" VALUE="9.3.4.21"/>
            <smb_workgroup TYPE="STRING" VALUE="lyoung"/>
            <smb_user TYPE="STRING" VALUE="crispin"/>
            <smb_password TYPE="STRING" VALUE="guest"/>
            <smb_translate TYPE="BOOL" VALUE="FALSE"/>
            </queue_data>
          <filter_data TYPE="LIST">
            <mf_type TYPE="STRING" VALUE="MFOMATIC"/>
            <flags TYPE="LIST">
              <assume_data_is_text TYPE="BOOL" VALUE="FALSE"/>
              <rerender_Postscript TYPE="BOOL" VALUE="FALSE"/>
              <convert_text_to_Postscript TYPE="BOOL" VALUE="TRUE"/>
              </flags>
            <printer_id TYPE="STRING" VALUE="IBM-Infoprint_70"/>
            <gs_driver TYPE="STRING" VALUE="omni"/>
            <foomatic_defaults ANONYMOUS="TRUE" TYPE="LIST">
              </foomatic_defaults>
            </filter_data>
          <filter_type TYPE="STRING" VALUE="MAGICFILTER"/>
          </SMBIBM70>
        </print_queues>
      </printconf>
    </datatree>
  </adm_context>
```

The file in Example 2-13, after it is created, allows an administrator to reconfigure areas of the file. In this example, we printed to an SMB server which had the address of 9.3.4.21 using the user ID *crispin* and password *guest* as shown in the following excerpts:

```
<smb_share TYPE="STRING" VALUE="info70"/>
<smb_ip TYPE="STRING" VALUE="9.3.4.21"/>
<smb_workgroup TYPE="STRING" VALUE="lyoung"/>
<smb_user TYPE="STRING" VALUE="guest"/>
<smb_password TYPE="STRING" VALUE="guest"/>
```

You can easily reconfigure the previous entries for a user other than *crispin* and connect to a different share, server, or workgroup.

As with the local printer, you can import the modified file during installation or when a user requests access to a new printer.

2.5 Protocols

Linux mostly uses TCP/IP as its networking protocol. This is the standard protocol for most UNIX-like operating systems. Linux has the advantage over

UNIX systems by having a completely new kernel developed from new ideas. One is to support several non-TCP/IP protocols.

Linux has created a completely new TCP/IP implementation that has gained a reputation as one of the best for its reliability and speed. Linux can use TCP/IP as the basis to support several other protocols such as Systems Network Architecture (SNA) and NetBIOS. SNA is often used to connect to host systems, NetBIOS to share printers, and files with OS/2 and Windows machines.

2.5.1 NetBIOS over IP

NetBIOS over IP is implemented in Linux by use of a Samba server, Samba client, and programs that search over the network as LinNeighborhood. LinNeighborhood is an Xwindow graphical application, similar to Windows Network Neighborhood, and calls Samba client commands in the background.

2.5.2 SNA

Linux supports the SNA protocol to access host systems. Several client software tools are available to open a terminal emulation session to a host from Linux. Some of them include:

- IBM WebSphere Host On-Demand to access host systems
- x3270 to access IBM @server xSeries® servers
- PR3287 to allow printer output from an x3270 session to be directed to a Linux printer
- TN5250 to access iSeries servers

These terminal emulation sessions need an intermediary machine and a Linux (or OS/2) network node running the software needed to act as an *SNA Gateway*. IBM Communications Server for Linux provides this capability.

Communications Server must be installed on a Linux system that provides access to the host or hosts for all the end nodes that are connected to it, using SNA Gateway.

To serve terminal emulation sessions, another part of Communications Server is configured—TN Redirector.

For example, suppose that a host is divided into two logical partitions (LPARs), LPAR1 and LPAR2. This host communicates using SNA as its only protocol. If it also uses TCP/IP, it can be accessed directly without any SNA Gateway.

It is necessary to configure the Communications Server on Linux, called for example *cslinux*, to connect to both LPARs to define a link station for each LPAR.

Now, it is necessary to configure TN Redirector to assign a port number to each LPAR of the host. For example, LPAR1 can be 50007, and LPAR2 can be 50008. For an x3270 to connect, it is given this address to connect:

- `cslinux:50007` to connect to LPAR1
- `cslinux:50008` to connect to LPAR2

The Communications Server features include:

- Telnet Redirector
- SNA Gateway
- Peer to peer networking
- Application programming interfaces (APIs)

IBM Communications Server for Linux is installed in two main steps:

- Prerequisites installation: Install Open Motif and LiS first. LiS (streams) comes in source format. Therefore, you must compile it since it depends strongly on the kernel. To do this, you need a kernel-source package to compile LiS.
- Communication Server installation: You install Communication Server by running a shell script called **`installcslinux`**. If the installation fails indicating that LiS is not installed, and it is, you may need to start LiS with the **`streams start`** command.

Communications Server is installed into the /opt/sna directory. It is started using the system start script /etc/init.d/init.d/snastart.

2.5.3 IPX and SPX

The IPX protocol is supported by the Linux kernel. To enable it, the kernel must be recompiled with IPX support. Or, if this support is enabled as a loadable module, this module must be present with the name *ipx.o*. Look for it in /lib/modules/2.4.xx-xx/kernel/net/ipx.

2.5.4 PPP

Serial connections via modem or null cable modem are easily configured using **`kinternet`** or **`qinternet`**. Point to Point Protocol (PPP) can allow different alternative connections to different Internet providers.

Attention: **`kinternet`** is installed by default in SuSE and is an individual package. Red Hat uses **`kppp`**, included in the kdenetwork package.

kinternet is part of KDE and **qinternet** is similar but can be used from outside KDE. In the background, they both use **wvdial** to make the connections.

The connection configuration is stored in /etc/syscofig/network/providers in the form of a single file per configuration made.

PPP connections need **smpppd** to run. You start **smpppd** with the script:

```
/etc/init.d/smpppd start
```

To stop **smpppd**, enter:

```
etc/init.d/smpppd stop
```

And to check whether it is running, enter:

```
etc/init.d/smpppd status
```

After you start it, you can start **kinternet** for its first configuration from the user that started KDE, from a terminal, or by pressing Alt-F2 to open a command launcher. Then, run **kinternet**. An icon appears at the right side of the toolbar.

> **Note:** If GNOME is the desktop manager that is used, don't use **kinternet**, since it is intended for KDE. Instead, install and run **qinternet**, since it is virtually the same as **kinternet**. Don't start **kinternet** as *user root* since the configurations created work only with the user that creates them.

It is quite simple to configure a PPP connection from **kinternet** (or **qinternet**).

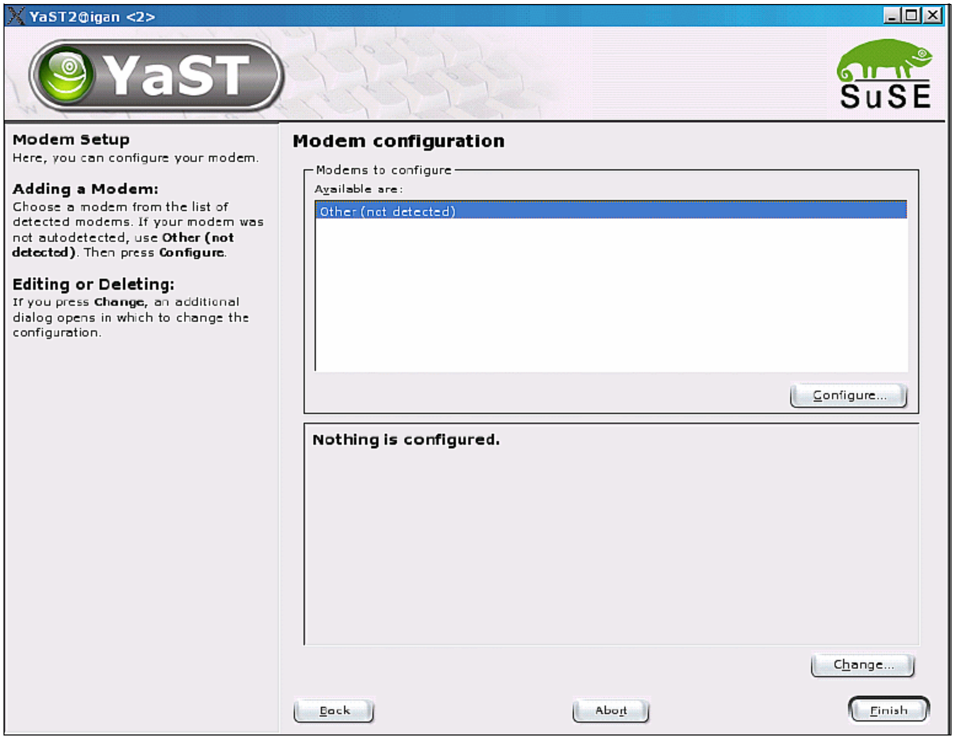
The following scenario is based on SuSE 8.2.

1. Start YaST2. Type the following command to configure modem connections:

   ```
   yast2 modem
   ```

   After you configure the connections, they are used by **kinternet**.

2. The Modem configuration panel (Figure 2-14) opens. Click **Configure**.



Modem Setup

Here, you can configure your modem.

Adding a Modem:

Choose a modem from the list of detected modems. If your modem was not autodetected, use **Other (not detected)**. Then press **Configure**.

Editing or Deleting:

If you press **Change**, an additional dialog opens in which to change the configuration.

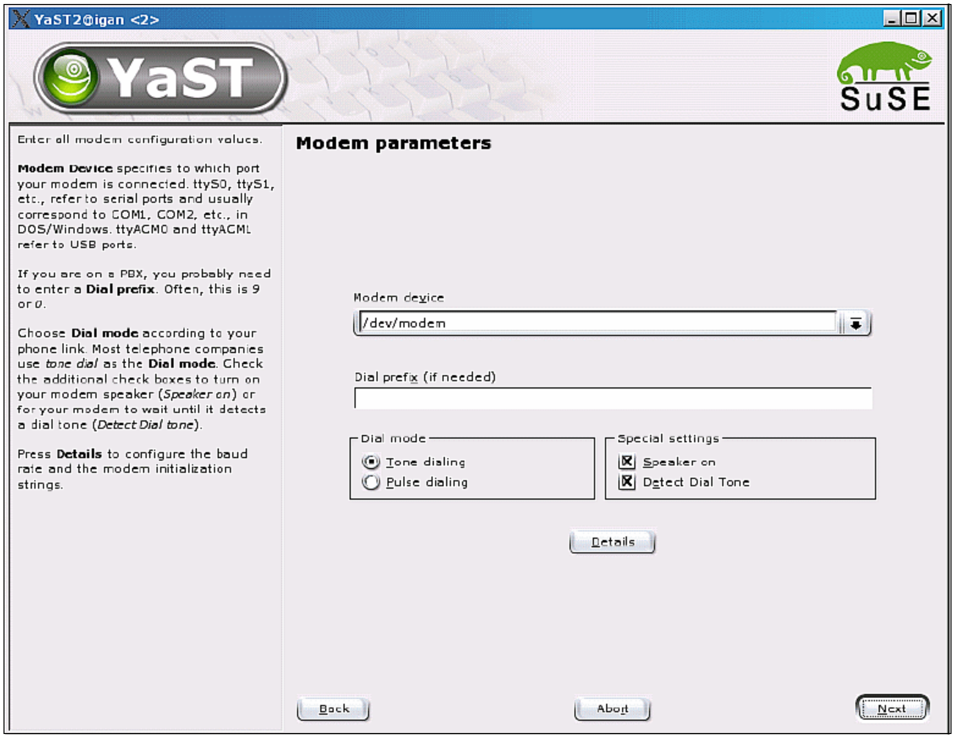
Modem configuration

Other (not detected)

Nothing is configured.

Figure 2-14 Using YaST to configure a modem

3. On the Modem Parameters panel (Figure 2-15), click **Next**.



YaST2@igan <2>

Enter all modem configuration values.

Modem Device specifies to which port your modem is connected. ttyS0, ttyS1, etc., refer to serial ports and usually correspond to COM1, COM2, etc., in DOS/Windows. ttyACM0 and ttyACML refer to USB ports.

If you are on a PBX, you probably need to enter a **Dial prefix**. Often, this is *9* or *0*.

Choose **Dial mode** according to your phone link. Most telephone companies use *tone dial* as the **Dial mode**. Check the additional check boxes to turn on your modem speaker (*Speaker on*) or for your modem to wait until it detects a dial tone (*Detect Dial tone*).

Press **Details** to configure the baud rate and the modem initialization strings.

Modem parameters

Modem device: /dev/modem

Dial prefix (if needed):

Dial mode
● Tone dialing
○ Pulse dialing

Special settings
☒ Speaker on
☒ Detect Dial Tone

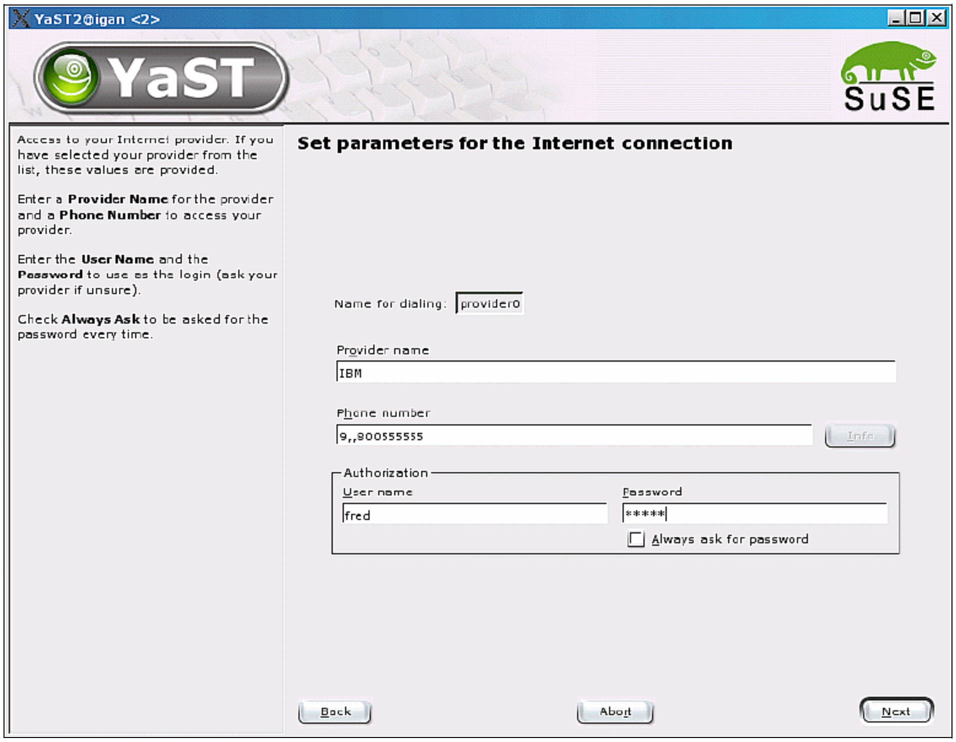
Figure 2-15 Setting modem parameters

4. On the next panel (not shown), you see a long list of popular Internet Service Providers (ISPs). Select the one that is suitable for you. Otherwise, click **Custom Providers-> New**.

5. On the Set parameters for the Internet connection panel (Figure 2-16), enter the required data.

   This provider is identified as provider0, which cannot be changed. provider0 is also the name of the configuration file and is used to identify the provider.

   The phone number to be called can include commas—one per second of delay necessary to call from some telephone switchboards. If you cannot type the commas, you can add them later to the configuration file. Click **Next**.



Access to your Internet provider. If you have selected your provider from the list, these values are provided.

Enter a **Provider Name** for the provider and a **Phone Number** to access your provider.

Enter the **User Name** and the **Password** to use as the login (ask your provider if unsure).

Check **Always Ask** to be asked for the password every time.

Set parameters for the Internet connection

Name for dialing: provider0

Provider name: IBM

Phone number: 9,,800555555

Authorization

User name: fred

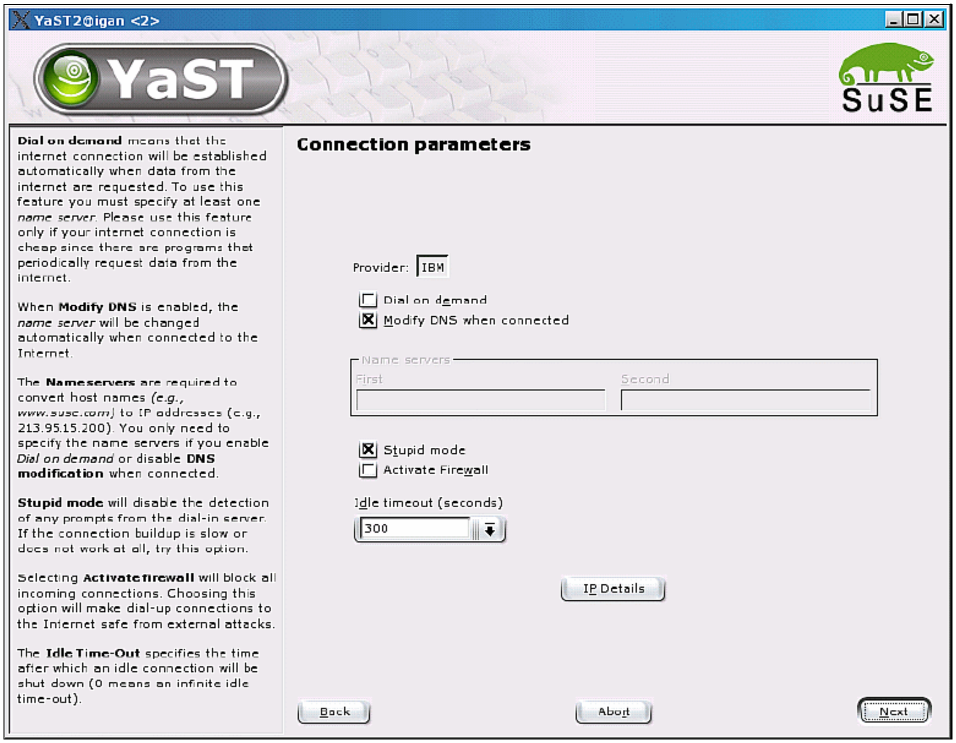
Password: *****

□ Always ask for password

Figure 2-16 Configuring an Internet connection

6. The program advises the presence of commas. Click **Yes** to continue.

7. On the Connection parameters panel (Figure 2-16), select **Stupid mode** (recommended) to let the program negotiate the connection. Then, click **Next**.



Dial on demand means that the internet connection will be established automatically when data from the internet are requested. To use this feature you must specify at least one *name server*. Please use this feature only if your internet connection is cheap since there are programs that periodically request data from the internet.

When **Modify DNS** is enabled, the *name server* will be changed automatically when connected to the Internet.

The **Nameservers** are required to convert host names *(e.g., www.suse.com)* to IP addresses (e.g., 213.95.15.200). You only need to specify the name servers if you enable *Dial on demand* or disable **DNS modification** when connected.

Stupid mode will disable the detection of any prompts from the dial-in server. If the connection buildup is slow or does not work at all, try this option.

Selecting **Activate firewall** will block all incoming connections. Choosing this option will make dial-up connections to the Internet safe from external attacks.

The **Idle Time-Out** specifies the time after which an idle connection will be shut down (0 means an infinite idle time-out).

Connection parameters

Provider: IBM

☐ Dial on demand
☒ Modify DNS when connected

Name servers
First:
Second:

☒ Stupid mode
☐ Activate Firewall

Idle timeout (seconds): 300

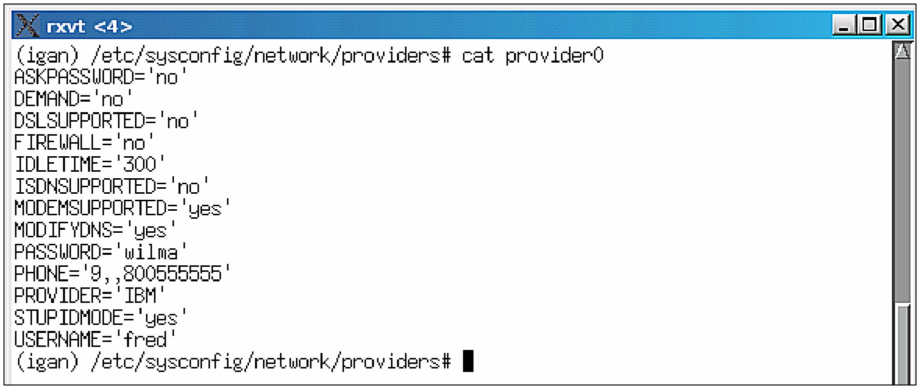
Figure 2-17 Connection parameters

8. On the last panel, click **Finish**.
9. Click **No** if you are asked to configure the mail. Then the configuration is complete.

To connect, follow these steps:

1. Right-click over the **kinternet** icon on the toolbar.
2. Select **Interface -> ppp0**.
3. On the next panel, select **Provider -> IBM**.
4. On the last panel, click **Dial-up** to start the call.

The configuration is stored in the form of several files, one per provider, in the /etc/sysconfig/network/providers directory. Figure 2-18 shows the file that corresponds to the provider that was just created.



```
(igan) /etc/sysconfig/network/providers# cat provider0
ASKPASSWORD='no'
DEMAND='no'
DSLSUPPORTED='no'
FIREWALL='no'
IDLETIME='300'
ISDNSUPPORTED='no'
MODEMSUPPORTED='yes'
MODIFYDNS='yes'
PASSWORD='wilma'
PHONE='9,,800555555'
PROVIDER='IBM'
STUPIDMODE='yes'
USERNAME='fred'
(igan) /etc/sysconfig/network/providers# 
```

Figure 2-18 Sample configuration file for an Internet connection

Only root can edit or view this file.

Linux works well with external modems that are usually Hayes compatible and with Peripheral Component Interconnect (PCI) modems that are not created especially to work with Windows, called *Winmodems*. These modems require a special driver, written for Windows, only with a proprietary set of instructions that are not available for Linux developers. Driver developers for Linux create generic drivers for these modems on a trial-and-error basis.

To see which modem is installed on a system, type the command:

```
lspci
```

Then look for a modem. Search on the Internet for the driver that was created for that modem since these drivers may not come with the distribution and must be downloaded.

2.6 Multimedia solutions

More and more, clients systems have a use for multimedia content. Linux multimedia capabilities, which were once thought to be somewhat limited, are improving almost daily. The following sections discuss some of the solutions that are available for handling multimedia content.

2.6.1 Generic audio support

The Advanced Linux Sound Architecture (ALSA) provides audio and MIDI functionality to the Linux operating system. ALSA has the following significant features:

- Efficient support for all types of audio interfaces, from consumer sound cards to professional multichannel audio interfaces
- Fully modularized sound drivers
- Symmetric multiprocessor (SMP) and thread-safe design
- User space library (alsa-lib) to simplify application programming and provide higher level functionality
- Support for the older OSS API, providing binary compatibility for most OSS programs

ALSA is released under the GPL and the GNU Lesser General Public License (LGPL). For more information, see:

http://www.alsa-project.org

2.6.2 Configuring audio

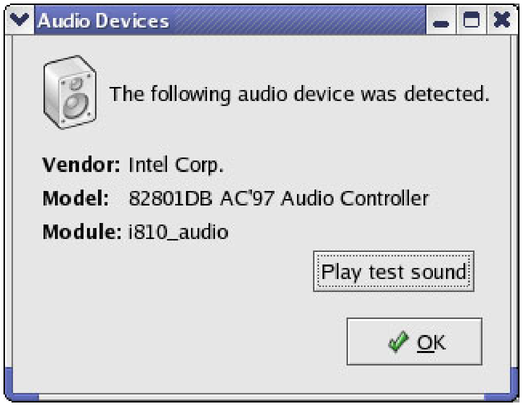
As distributions progress and mature, they standardize their own methods or front ends for configuring hardware. The following sections show how to configure audio support on Red Hat and SuSE.

Configuring audio for Red Hat

To start the Red Hat audio configuration program at a terminal shell, enter:

```
redhat-config-soundcard
```

If a sound card is detected as indicated in Figure 2-19, click **OK** to configure the sound modules.

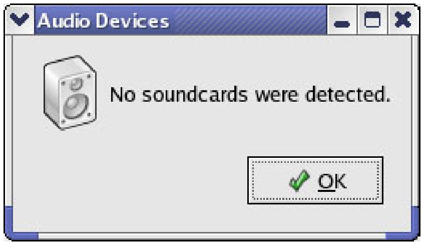


The following audio device was detected.

Vendor: Intel Corp.
Model: 82801DB AC'97 Audio Controller
Module: i810_audio

Figure 2-19 Red Hat audio configuration

If a sound card is not detected, then you see a window like the example in Figure 2-20.



No soundcards were detected.

Figure 2-20   No audio card detected

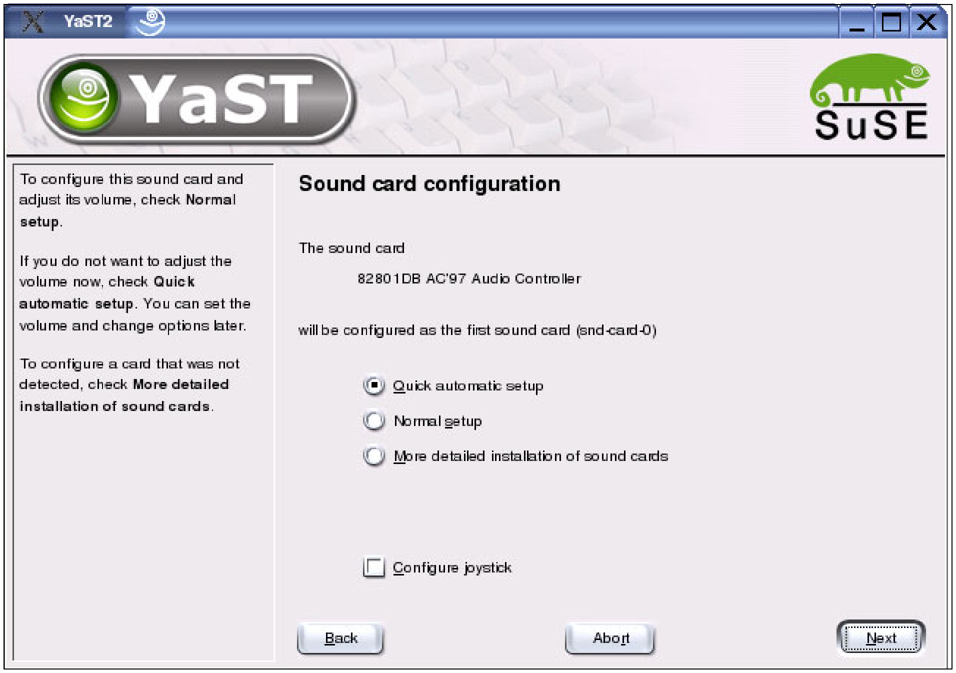
If the sound card is not detected, download a newer RPM that contains a newer version of *redhat-config-soundcard*. If the card is not detected, there are no parameters to use to force it to detect or use a specific driver. This may sound limited, but Red Hat has invested a lot of time and effort into the sound configuration program and can detect most cards.

Configuring audio for SuSE

To start the audio configuration program at a terminal shell, enter:

```
yast2 sound
```

When the program starts, it attempts to auto detect a sound card. On an IBM NetVista™ system, YaST detects the card and displays the panel shown in Figure 2-21. If a sound card is detected and the default values are acceptable, click **Next**. This configures the sound modules. Then the configuration program exits.



To configure this sound card and adjust its volume, check **Normal setup**.

If you do not want to adjust the volume now, check **Quick automatic setup**. You can set the volume and change options later.

To configure a card that was not detected, check **More detailed installation of sound cards**.

Sound card configuration

The sound card

82801DB AC'97 Audio Controller

will be configured as the first sound card (snd-card-0)

● Quick automatic setup

○ Normal setup

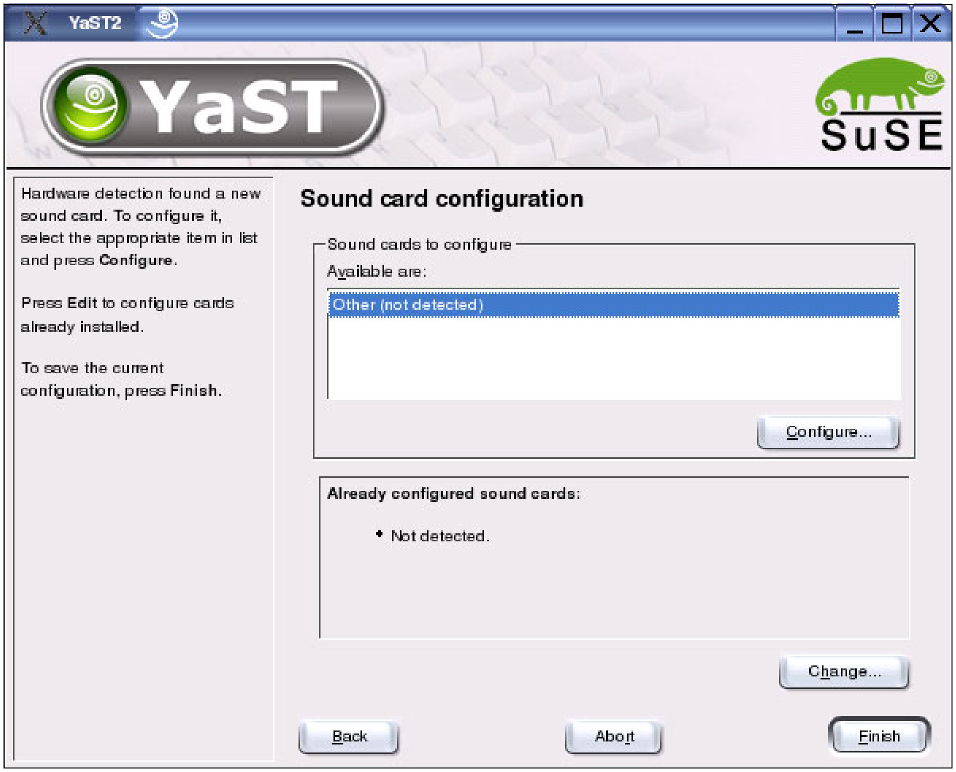
○ More detailed installation of sound cards

□ Configure joystick

Figure 2-21 YaST sound card configuration

If the sound card is not detected, select **More detailed installation of sound cards** and click **Next**.

Then you see the panel shown in Figure 2-22. In the Sounds cards to configure box, click the **Configure** button.



Hardware detection found a new sound card. To configure it, select the appropriate item in list and press **Configure**.

Press **Edit** to configure cards already installed.

To save the current configuration, press **Finish**.

Sound card configuration

Sound cards to configure

Available are:

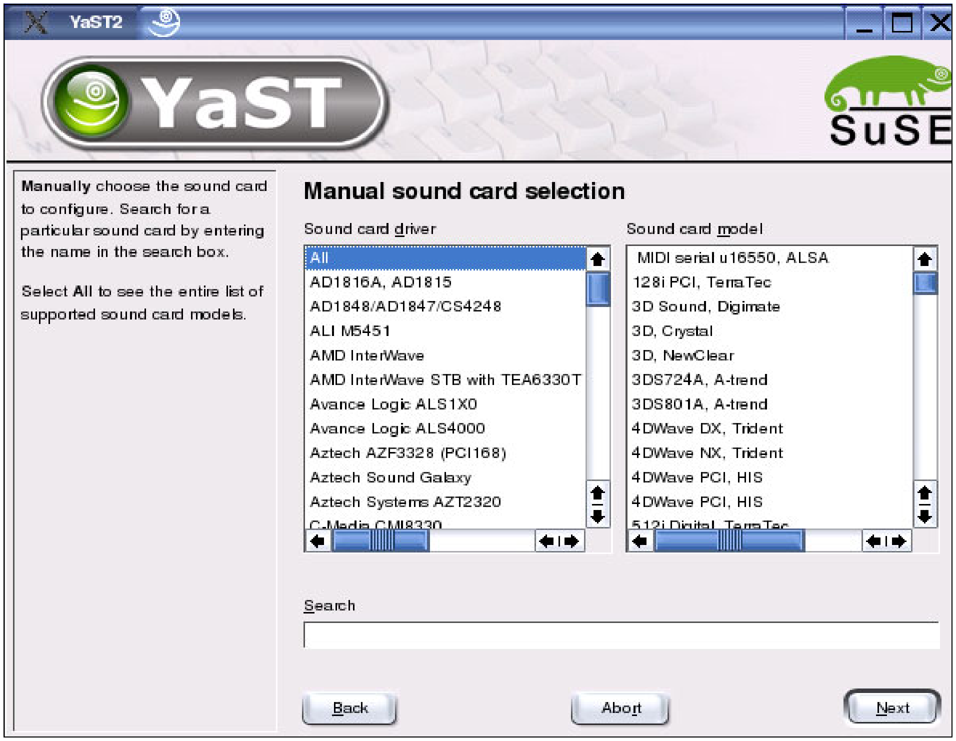
Other (not detected)

Already configured sound cards:

- Not detected.

Figure 2-22 Audio card not detected

On the Manual sound card selection panel (Figure 2-23), select the sound driver that relates to the card that is installed and click **Next**.



Manually choose the sound card to configure. Search for a particular sound card by entering the name in the search box.

Select All to see the entire list of supported sound card models.

Manual sound card selection

Sound card driver

All
AD1816A, AD1815
AD1848/AD1847/CS4248
ALI M5451
AMD InterWave
AMD InterWave STB with TEA6330T
Avance Logic ALS1X0
Avance Logic ALS4000
Aztech AZF3328 (PCI168)
Aztech Sound Galaxy
Aztech Systems AZT2320
C-Media CMI8330

Sound card model

MIDI serial u16550, ALSA
128i PCI, TerraTec
3D Sound, Digimate
3D, Crystal
3D, NewClear
3DS724A, A-trend
3DS801A, A-trend
4DWave DX, Trident
4DWave NX, Trident
4DWave PCI, HIS
4DWave PCI, HIS
512i Digital, TerraTec

Search

Figure 2-23 Manually selecting audio cards

The panel shown in Figure 2-21 on page 62 is displayed. Either select **Quick automatic setup** to accept the defaults or select **Normal setup** to configure the default volume level for the system.

2.6.3 Configuring video

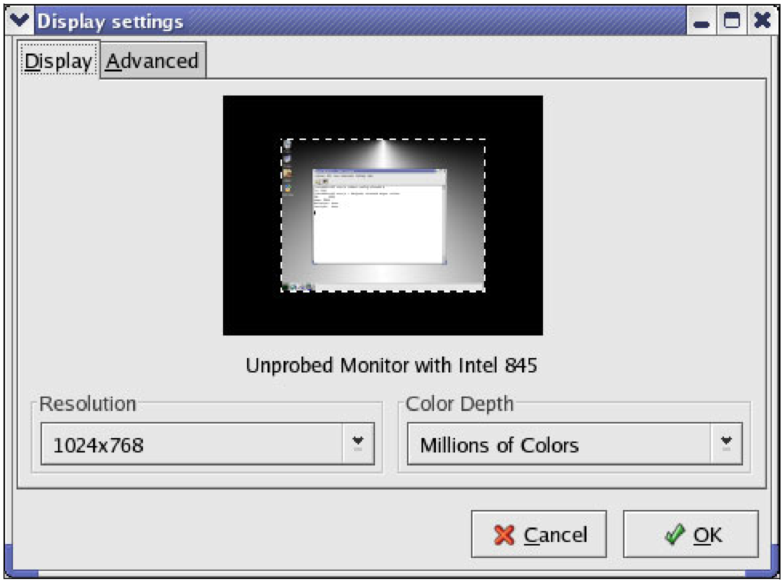
As with audio configuration, current distributions have their own methods of detecting and installing video drivers. The following sections explain how to configure video for Red Hat and SuSE.

Configuring video for Red Hat

To configure video settings in Red Hat, from a terminal shell, enter the following command:

```
redhat-config-xfree86
```

You run this command regardless of whether the video was configured before or if this is the first time that the configuration program is run. The Display settings window (Figure 2-24) opens.



Unprobed Monitor with Intel 845

Resolution: 1024x768

Color Depth: Millions of Colors

Figure 2-24 Red Hat video configuration

This window has two tabs

- Display
  - Configuration of resolution
  - Configuration of color depth
- Advanced
  - Configuration of monitor type
  - Selection of the VideoCard driver

When you make the required configuration changes, click **OK**. This saves the configuration to the /etc/X11/XF86Config file. This is an ASCII-based configuration file that is read by X upon the start of the X server to set the video settings.

Configuring video for SuSE

As with the configuration of the audio within SuSE, configure the video by entering the following command:

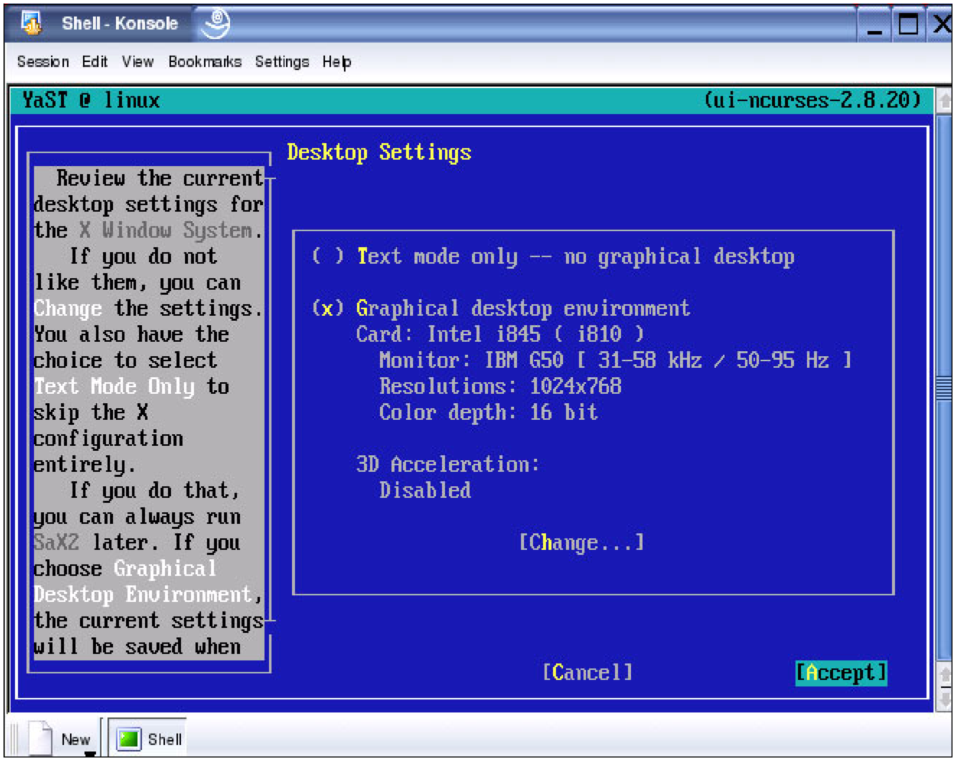
```
yast2 x11
```

YaST2 is a graphical program. It relies on the GUI that is being preconfigured and working. For this example configuration of the video, we use *yast*, which is a text-based version of the configuration program. You can use it within an existing configured GUI environment to make changes. Or you can use it from a terminal if no GUI has been configured or was configured incorrectly.

To execute the text-based version of the X server configuration program, enter the following command:

```
yast x11
```

When the utility starts, it auto detects the video driver and shows the YaST @ linux Desktop Settings panel (Figure 2-25). To change these settings, tab to the [Change] button.



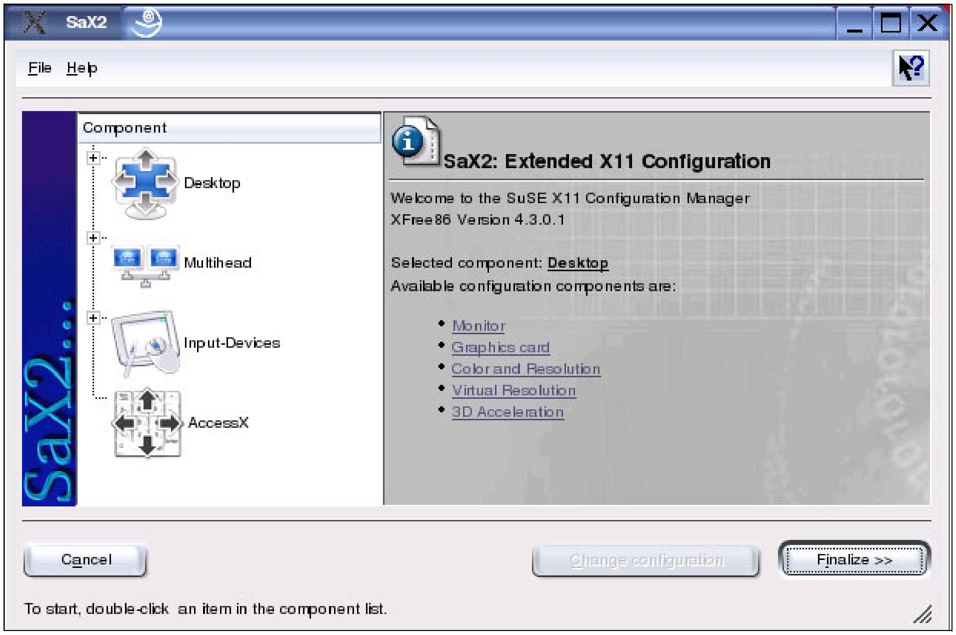
```
YaST @ linux                                              (ui-ncurses-2.8.20)

 Review the current   Desktop Settings
desktop settings for
the X Window System.   ( ) Text mode only -- no graphical desktop
  If you do not
like them, you can     (x) Graphical desktop environment
Change the settings.       Card: Intel i845 ( i810 )
You also have the            Monitor: IBM G50 [ 31-58 kHz / 50-95 Hz ]
choice to select             Resolutions: 1024x768
Text Mode Only to            Color depth: 16 bit
skip the X
configuration              3D Acceleration:
entirely.                    Disabled
  If you do that,
you can always run                         [Change...]
SaX2 later. If you
choose Graphical
Desktop Environment,
the current settings
will be saved when
                                        [Cancel]             [Accept]
```

Figure 2-25 Video configuration using YaST

At this point, the configuration program starts a GUI environment. In SuSE, it is called SaX2. You can start the program directly from a terminal shell by entering:

```
SaX2
```

From the SaX2: Extended X11 Configuration panel (Figure 2-26) that opens, configure the monitor, resolution, graphics adapter, virtual resolution, and 3-D accelerators. Then click **Finalize**. Now the settings are saved.



SaX2: Extended X11 Configuration

Welcome to the SuSE X11 Configuration Manager
XFree86 Version 4.3.0.1

Selected component: Desktop
Available configuration components are:

- Monitor
- Graphics card
- Color and Resolution
- Virtual Resolution
- 3D Acceleration

To start, double-click an item in the component list.

Figure 2-26 SaX2 configuration window

2.7 Summary

Linux systems have a robust environment for interaction with the user. The windowing system has many similarities to OS/2 or Windows from a user's perspective. However, from an administrator's perspective, the ability to customize and lock down the desktops of users can provide many benefits and reduce support requirements.

Aside from the GUI, the Linux platform offers rich support for printing and multi-media devices.

The next chapter looks at how Linux addresses functional requirements such as the support required for enterprise applications.

3

Functional considerations

This chapter describes the main areas of application functionality that are most prevalent in OS/2 client installations. It suggests some applications that you may install on Linux clients which address these tasks. Even though the migration from OS/2 to Linux is a non-trivial enterprise, we believe that this move can be accomplished without the user suffering any loss of usability, capability, or functionality.

3.1 Host connectivity

Throughout the evolution of network computing, there have been shifts in desktop strategy. These vary from character-based terminals connected to mainframes, local area network (LAN)-based clients interacting with Web-based application servers, and everything in between. It is often cost prohibitive to migrate all mission-critical applications to take advantage of the benefits of each choice.

Many products are available that address the need to run host applications from a Linux client. They range in capability from simple text-based emulators (useful when using Linux as a simple terminal) to full featured “screen scraping” applications that can provide a graphical user interface (GUI) or Web-based front end to host applications.

Most of the open source clients described in this chapter use TCP/IP to communicate with the host. If Systems Network Architecture (SNA) connectivity is required, you can use IBM Communications Server for Linux as a gateway to allow access to SNA-based applications from any of the TCP/IP clients that are described. To learn more about this product, see the following Web site:

http://www.ibm.com/software/network/commserver/about/cslinux.html

Many emulators are available. This list is not meant to be exhaustive, but rather offers a glimpse at possible solutions.

3.1.1 IBM WebSphere Host On-Demand

IBM WebSphere Host On-Demand is the IBM solution for delivering host terminal access to any client equipped with a supported browser and Java runtime. WebSphere Host On-Demand supports several terminal types including 3270, 5250, VT, and CICS®. The client-side WebSphere Host On-Demand application is delivered as a Java Archive (JAR) file that can be cached at the client for increased performance.

Using WebSphere Host On-Demand, you can create a user-friendly Web or portlet-based front end to host applications and deploy it across multiple client types and platforms from a centrally managed server environment. WebSphere Host On-Demand may be the best solution when rolling out host-based applications to users who are unfamiliar with traditional green-screen environments. It also has the advantage of being a Web application that is already deployed in many OS/2 environments.

Since WebSphere Host On-Demand is Java based, it is a great choice for a solution that involves clients running a variety of operating systems. For more information about WebSphere Host On-Demand, see:

http://www.ibm.com/software/webservers/hostondemand/

3.1.2 x3270

x3270 is a small, fast X Window System-based terminal emulator that is included in most Linux distributions. It may be the best choice for users who are accustomed to working with key-board driven, host-based applications (for example, transactional users). It may also be suitable for those who require responsive, direct (green screen) access to host applications. A curses (text-mode console) version is also available for use in text-only terminal environments.

Note: As of this writing, the current version of x3270 (3.2) does not support double-byte character sets (DBCS). A version was in alpha that adds DBCS support and support for connections over Secure Sockets Layer (SSL).

3.1.3 tn5250 and tn5250j

tn5250 and tn5250j are open source 5250 emulators. tn5250 is a basic emulator that runs on Linux, other UNIX-like operating systems, and Windows. tn5250j is Java-based and implements several additional features that are not present in tn5250. This includes intelligent handling of non-displayable characters and so on.

For more information about tn5250, see:

http://tn5250.sourceforge.net/

To learn more about tn5250j, see:

http://tn5250j.sourceforge.net/

3.1.4 PowerTerm InterConnect

PowerTerm InterConnect is developed by Ericom Software. It is a terminal emulation product that can emulate tn3270, tn5250, and other emulation types over a variety of security protocols, including Secure Shell (SSH), SSL, and TLS1. This product is also available on such other platforms as Windows, MAC OS X, and Sun Solaris, making it a viable solution for mixed client environments.

You can learn more about PowerTerm InterConnect (Linux Edition) on the Web at:

http://www.ericom.com/pti4linux.asp

3.2 Web browsers

Web browsers are now widely deployed on clients of all types. You can use them to scour the Internet and corporate intranets for information of all types. Among the many tasks that today's browsers perform, they also perform these functions:

- Serve as a way to deliver cross-platform applications to end users
- Initiate and manage electronic forms
- Display presentations
- Facilitate electronic meetings
- Collaborate with remote team members

As customers make the transition from OS/2 clients, they are in a position to re-evaluate their current desktop strategy. Each of the Web browsers mentioned in the following sections has its own strengths and weaknesses. Here are some points to consider when choosing a Web browser:

- **Standards compliance**

  Standards compliance is an important topic when referring to Web browsers. Internet standards, as compiled by the World Wide Web Consortium (W3C), are based on requirements and ideas from a wide variety of individuals, corporations, and browser developers who have a special interest and expertise in the area in question. These items of interest are debated, modified, tested, and finally agreed upon. When a Web browser declares itself in compliance with a certain standard, this browser can accurately handle the given technology in the manner described by the standard.

  As long as the Web page or application developers write their code to the specification, then any browser that complies with the standard processes the code with the same result as other such browsers. This eliminates the need for Web developers to test for and code to the peculiarities of an individual browser.

- **Footprint® and performance**

  The most important aspect of performance when referring to browsers is the amount of time it takes a browser to parse and render Web pages and Web applications. Browsers may be written to work specifically on a single platform and only offer the ability to browse as opposed to handling mail, news, and so on. Such browsers are most likely the fastest loading and performing Web browsers available. They do not have the overhead associated with trying to "be all things for all people". However, the ability to have the same user experience on a variety of clients may be important, as well as having an integrated mail, news, and Web browser solution.

Referring back to "standards compliance", the ability to completely and accurately load all necessary documents is required before performance of such a load can be an issue.

3.2.1 Mozilla

Mozilla is an open-source Internet suite. It is the most popular choice for IBM OS/2 and Linux client customer sets. The Mozilla suite provides a full-featured Web browser. It includes:

- A What You See Is What You Get (WYSIYG) Hypertext Markup Language (HTML) editor
- An IRC client application
- Java integration
- A fairly robust e-mail client (Post Office Protocol (POP3)/Internet Message Access Protocol (IMAP))
- A news reader (Network News Transfer Protocol (NNTP))

Mozilla is hosted on the Web at:

`http://www.mozilla.org`

Mozilla also includes a Java Script Console and Java Script debugger. The console logs any problems that the document parser encounters when formatting the HTML page and when the Java Script code is executed. The debugger allows for breakpoints, call stack and local variable evaluation, and much more. Mozilla also sports an Extensible Markup Language (XML)-based user interface (UI) referred to as XUL (pronounced "zool"). This UI lends itself to complete customization. With Mozilla, companies can now have total control over the look and feel of their browser and the end-user experience across all client platforms.

In general, you should deploy only Mozilla versions that come with a particular Linux distribution or recommended official "stable" releases (1.x or 1.x.x for patches) that are available on the Web at:

`http://www.mozilla.org`

We encourage you to focus your testing of in-house Web applications on a particular release of Mozilla. Then standardize on that release after you verify stability across all deployed platforms.

As you find bugs, you can work with your Linux distributors or directly with mozilla.org to create bug reports and fix the bugs. Mozilla's bug tracking system is also available online so users can monitor bugs and contribute comments and test cases as needed. For more information about Mozilla bug management, see the following Web site:

http://bugzilla.mozilla.org/

3.2.2 Konqueror

Konqueror is the Web browser that is built into the KDE desktop. It is based on the object-component technology within KDE and provides local and remote graphical file manager features. It supports most of the W3C standards.

Apple has chosen Konqueror technology (more specifically the KHTML rendering engine) as the basis for their newly-released Safari Web browser for Mac OS X. Konqueror is a viable choice for a Linux-only client rollout that has standardized on the KDE environment and controls the development of its own Web content.

You can learn more about Konqueror on the Web at:

http://www.konqueror.org

3.2.3 Galeon

Galeon was the Web browser that shipped with the GNOME desktop prior to 2003. GNOME is the other major desktop environment along with KDE. You can find it on the Web at:

http://galeon.sourceforge.net

Galeon is based on the Gecko component of Mozilla (layout and rendering engine). Therefore, it offers the same standards support as Mozilla. It also benefits from using completely native Linux frame controls to achieve GUI performance gains.

3.2.4 Epiphany

Epiphany is the Web browser that is currently bundled with the GNOME desktop. You can find it on the Web at:

http://www.gnome.org/projects/epiphany/

Epiphany is based on the Gecko component of Mozilla (layout and rendering engine) and was founded using the Galeon source code. This browser offers the same standards support as Mozilla and has the same advantage of the native frame controls performance as Galeon. In addition to the Web standards support and responsive user interface, Epiphany users the benefit of tight integration of Epiphany into the GNOME desktop environment. Epiphany is a viable choice for a Linux-only client rollout that has standardized on the GNOME environment.

You can find more information about Galeon and the distinctions between Galeon and Epiphany in *Galeon, A History*, which is on the Web at:

http://galeon.sourceforge.net/links/history.php

3.2.5 Opera

Opera is a fee-based, highly-rated, standards-based Web browser. Versions of Opera are available for several platforms including Windows, most major distributions of Linux, and non-PC devices, such as interactive television, personal digital assistants (PDAs), mobile phones, and automobiles. Opera has the advantage of completely native controls for UI responsiveness and quick rendering, along with a small footprint.

To learn more about Opera, go to:

http://www.opera.com

3.3 Web browser plug-ins

Plug-ins are native programs that a browser can launch automatically to handle specialized content. A Web page can contain data that is “typed” (referred to as a Multipurpose Internet Mail Extensions (MIME) type). Browsers maintain an internal list of MIME types that have been registered and a list of which plug-ins to launch to manage the content of the specified type. To learn about which plug-ins are registered for Netscape for OS/2, Mozilla for OS/2, or the IBM Web Browser for OS/2, in the Address line of the browser, enter:

```
about:plugins
```

The data types that are discussed in the following sections have native Linux plug-in support for Mozilla. We believe that Mozilla will be the browser chosen by the majority of clients that transition from OS/2 to Linux due to the fact that most OS/2 customers currently use the IBM Web Browser for OS/2. These data types are also supported on any browser that implements the Netscape Plug-in programming model.

3.3.1 PDF

The Portable Document Format (PDF) file format was designed by Adobe. It is generated by the Adobe Acrobat program and is viewable by the Adobe Acrobat Reader program. The PDF format is known for its ability to preserve document fidelity when printed and displayed. Many applications also provide the ability to write files in the PDF format.

Adobe provides Acrobat Reader in plug-in form for several platforms, including Linux. You can download Acrobat Reader (5.0 as of this writing) from the Web at:

http://www.adobe.com/

On Linux, the reader is a motif application. That is, it does not require KDE or GNOME. You should run it on any distribution with at least kernel 2.2 or higher.

3.3.2 Java plug-in for applets

Applets are delivered within a Web page and provide graphical interfaces for Java programs that execute in the context of a Web browser. The Java plug-in provides the link between the Web browser and the client's Java 2 Standard Edition (J2SE) Runtime Environment (JRE). JRE Version 1.4.1 is currently available for Linux and the installation package includes the plug-in. The most prevalent Java Virtual Machines (JVMs) that are available on Linux are produced by:

- Sun Microsystems, Inc.

  http://java.sun.com/linux/

- IBM

  http://www.ibm.com/java/

  Click the links **IBM developer kits -> Linux**.

- Blackdown.org with assistance from Sun Microsystems, Inc.

  http://www.blackdown.org/

3.3.3 Flash

Flash is a rich multimedia file format that is designed by Macromedia. Version 6 is currently available for Linux. To learn more about Flash, see:

http://www.macromedia.com/software/flash/

3.3.4 RealOne and Helix

Produced by Real Networks, RealOne is another popular multimedia plug-in that allows the user to experience rich video and sound. It is distributed and supported on Windows and Mac from the Real Networks Web site at:

http://www.real.com

Real Networks is working with the Helix Community to produce a product that is based on RealOne. It targets UNIX, Linux, and Solaris users. While Real Networks doesn't support the resulting product, it backs the project and the Helix community provides a customer forum to help resolve issues. You can find more information on the Web at:

http://www.helixcommunity.org/

3.4 Instant messaging

Instant messaging has become core to the collaboration strategy of many enterprises. The following sections discuss popular instant messaging products.

3.4.1 Lotus Instant Messaging and Web Conferencing 3.1

IBM provides support for Lotus Instant Messaging and Web Conferencing (also known as Sametime®), with Release 3.1, to Linux and UNIX desktops. These abilities can be integrated into portal environments or work as stand-alone applications.

Lotus Instant Messaging can be extended (securely) to interface with the instant messaging systems at customer and partner sites. Web conferencing features allow users to experience streaming audio, video, and shared documents, as well as presentations and application sharing. The cross-platform support for Lotus Instant Messaging and Web Conferencing makes this a valuable option for many enterprises.

3.4.2 Yahoo! Messenger

Yahoo! Messenger is one of the most popular instant messaging systems. It is freely available for a wide variety of client platforms, including Windows, Mac, UNIX, and Linux. It is a proven product that may already be familiar to most of the targeted end users. You can download it for free from the Web.

For more information, see:

http://messenger.yahoo.com/

3.4.3 Gaim

Gaim is an open source Linux instant messaging client that can run on Windows and MacOS X desktops. While it has not yet reached a 1.0 release, it is getting many positive reviews. You can use it to access a wide variety of instant messaging networks available on the Web, such as Yahoo, AOL Instant Messenger, MSN Messenger, IRC, ICQ, among others. A user can carry on multiple chats with different users on different messaging systems concurrently.

For more information, see:

http://gaim.sourceforge.net/

3.5 Office suite

Several good office products are available on Linux today that are already widely used.

The first consideration in a transition is document conversion. The problem is that almost all proprietary formats are undocumented. Converters, if available, are typically limited in both the completeness and the fidelity of the conversion.

The second consideration is *round tripping*. Round tripping is the act of converting a file from one format, to a second, and back to the original format. As discussed earlier, conversions are not guaranteed to be 100% accurate. In round tripping, there are two conversions. This increases the chance that there is more content and format loss.

If round tripping with proprietary formats is a major requirement for a customer's enterprise, then Linux is not the best answer. However, many users do not need their documents to be handled in a proprietary format, only a consistent one. If you choose Linux for the client platform, then we recommend that you migrate everything once into a new format that the majority of end users can handle. This minimizes the inaccuracies that conversion may introduce.

3.5.1 IBM Lightweight Productivity Editors

IBM includes Lightweight Productivity Editors in the IBM WebSphere® Portal Server. Lightweight Productivity Editors are portlets, which allow the user to create and edit rich text, spreadsheet, or presentation documents. They are not meant to replace a fully functional office suite. However a relatively few end users need the full capabilities of the office products that they currently have at their disposal.

As a result, IBM developed these portlets to provide the most commonly used functions of the larger office applications. Switching to these productivity applications for such users can provide real corporate savings by limiting the need for office suite licenses, not to mention the great way that portlets fit into the thin client strategy.

Lightweight Productivity Editors provide a server-centric solution based on portal technology. As served up by IBM Portal Server, the portlets allow users to log on from any machine that can connect to their individual portal site and manage documents that they've already created or stored. The user can also create more documents and store them in their portal document space, without leaving residual files or data on the client machine. This is a flexible solution for a client environment.

3.6 E-mail and calendaring

E-mail clients for Linux range from minimalist Web-based, mail-only clients to full-featured personal information management (PIM) applications. Several protocols are in use. The mail server and message repository that are currently in use, to some extent, determine the clients that can be used with it. The three most common types of mail servers that you are likely to encounter are:

- **Lotus Domino®**

  Lotus Domino is the IBM enterprise-level messaging and groupware suite from Lotus. It uses a proprietary mail protocol that can be used only by the Lotus Notes client, which is currently not available for Linux. It also provides a browser-based client that can handle messaging and calendaring much the same way that Notes can. In addition, you can configure Domino to allow access from standard POP3 mail clients.

- **Microsoft Exchange**

  Exchange is Microsoft's messaging and collaboration suite. At present, the only Linux mail client with Exchange support is Ximian Evolution with the commercial Ximian Connector product installed.

- **Standards based (POP, IMAP, Simple Mail Transfer Protocol (SMTP))**

  The majority of mail clients written for Linux support some combination of the POP3, IMAP, and SMTP mail protocols. When using the POP3 protocol, the messages are downloaded directly to the client and usually deleted from the server after retrieval.

POP3 generally imposes less overhead on the server. It is best suited for environments where users typically access their mail from a single personal workstation.

With the newer IMAP4 protocol, mail is stored centrally at the server. Users can create folders on the server to organize and manage their mail. Since mail remains at the server, IMAP users can access their mail from any system that has access to the mail server. It is well suited for environments where users roam between workstations. IMAP can become resource intensive on the server as users accumulate ever-larger mail stores.

Whereas POP and IMAP are protocols for receiving mail, SMTP is the standard protocol to send it. Several authentication methods are supported for each of these protocols. Support for a particular method (for example, passwords with or without SSL) may vary between servers and clients.

3.6.1 IBM Internet Mailbox portlet

The IBM Internet Mailbox portlet (formerly known as the Advanced e-mail for Productivity portlet) is a lightweight, standards based e-mail client. It was included along with the other lightweight productivity components in WebSphere Portal Server version 5.

In conjunction with Portal Document Manager (PDM), the e-mail portlet provides basic attachment handling capabilities. Documents can be imported into PDM from the file system, detached and stored in the document manager from this e-mail portlet, and attached to an e-mail from PDM. In addition, portal users can view Microsoft Office format attachments as HTML within the browser without requiring Microsoft Office or external viewers present on the local workstation. They can edit the attachments imported into PDM using the Lightweight Productivity Editors (see 3.5.1, “IBM Lightweight Productivity Editors” on page 78). As with all such document conversions, fidelity of imported Office documents is imperfect.

3.6.2 Lotus Workplace Messaging Version 1.1

Lotus Workplace Messaging™ is a standards-based (POP/IMAP/iCal) messaging solution built around WebSphere Application Server. The e-mail server is WebSphere Application Server based and uses DB2 as the backend mail repository.

This release also contains tools to migrate mail messages from Exchange and Domino into the Lotus Workplace Messaging message store. The server is administered through plug-ins to the standard WebSphere Application Server browser-based administration console. A robust browser-based mail and calendaring client is included and is WebSphere Application Server based.

3.6.3 Domino Web Access

Domino Web Access, formerly known as Lotus iNotes™, provides access to Lotus Notes (Domino) mail, calendaring, and discussion databases through a rich Web browser-based interface. It the current solution from IBM for Notes access on Linux clients.

You can use Domino Web Access as a stand-alone Web browser interface or as a collection of portlets available in IBM WebSphere Portal Server. Domino Web Access also allows users to view and manage their e-mail and view their calendars while offline using Domino Off-Line Services (DOLS). By allowing Web access to data on Domino servers, this product is a viable solution for thin clients, users who are roaming or can log on to multiple machines, and a strategy that involves a variety of Windows and Linux desktops.

To learn more about Domino Web Access, see:

http://www.lotus.com/products/inotes.nsf

3.6.4 Ximian Evolution

Ximian Evolution is a full-featured open source e-mail, calendaring, and PIM application. It is modeled closely after Microsoft Outlook and is GNOME-based. Ximian Evolution is included in most desktop-oriented distributions and in Ximian's Desktop offering.

Out of the box, Evolution supports standards-compliant mail servers such as POP3, SMTP, and IMAP4. With the addition of the commercial Ximian Connector, Ximian Evolution can be used as a full-fledged Microsoft Exchange 2000 client. Evolution may be the best solution for Basic Office or Advanced Office users for whom Web-browser-based solutions are inadequate. It may also be the best solution for customers who want to introduce Linux clients into an environment that has standardized on Microsoft Exchange.

For more information about Ximian Evolution, see:

http://www.ximian.com/products/evolution/

For more information about Ximian Connector, see:

http://www.ximian.com/products/connector/

3.6.5 Mozilla Mail

Mozilla includes a fairly robust, standards-compliant e-mail client as part of the Mozilla suite. The Mozilla mail client can handle attachments and rich text

(HTML) messaging and interfaces with servers using POP3, IMAP, and SMTP. It has the advantage of already being available and integrated with the Mozilla Web browser on any machine that has Mozilla installed. This mail client runs on any platform upon which Mozilla can install, including many types of Windows and Linux.

3.6.6 KMail

KMail is a POP3 and IMAP mail client that is included with the KDE Desktop environment. In addition to such standard features as rich text (HTML) and attachment handling, KMail includes robust security features. This includes support for digital signatures and several forms of encryption.

To learn more about KMail, see:

http://kmail.kde.org/

3.7 Virus detection and prevention

In general, Linux is a secure operating system. It has not suffered from the number of viruses and worms that other desktop operating systems have. Some of this is due to the inherent security of Linux. Nor has it been targeted by the developers of viruses and worms.

Linux is not without weaknesses and there are known vulnerabilities. However, because of the open source nature of Linux and the number of developers that can search for and find these vulnerabilities, most of them are addressed in an expeditious manner.

Many articles are available that promote the Linux and UNIX security models above other operating systems. Some of the advantages of a Linux and UNIX security model over those implemented by other operating systems are:

- Attachments in Linux e-mail do not have execute permissions. They must be launched by making a conscientious decision to execute them. Even scripting in HTML-based e-mail is turned off by default. The user must actively choose to trust the sender before the HTML is parsed and executed.
- Files in Linux are separated from system files and other users' data by user login. Even if a virus attacks a particular user, other users of the system and the system files themselves are safe.
- You can customize Linux systems and choose different underlying implementations of core subsystems, such as the desktop manager, e-mail systems, and other application suites. Because of this, it is more difficult for

virus writers to target a specific vulnerability that exists across a large number of users. This gives a much smaller target for a virus or worm creator.

Even if a worm successfully infects a company that implements a standard client strategy for all of its end users, the worm spreads relatively slowly because it may not hurt the Linux configurations of other companies. The warning from the original company and slow progression of the worm through susceptible systems provides a larger window of opportunity to protect against the further propagation of the infection.

- There is less integration between software components. Some companies may integrate their browsers or e-mail viewers with other applications such as office or chat products or even the operating system. This exposes these other products to the vulnerabilities of the integrated application. This is a rare scenario under Linux.
- As mentioned previously, many popular Linux programs are open source. Therefore, there are many more eyes combing through the code with a wide variety of perspectives. This provides a high probability of finding and fixing security holes before they can be exploited.

However, as with any operating system, the system administrator must use due diligence to ensure that systems are updated with the latest patches to remove known vulnerabilities. A straightforward way to reduce exposure is to avoid running unnecessary services on the client (for example, Web, File Transfer Protocol (FTP), and Telnet servers, and so on) in the first place. Another way is to run updated virus scans on e-mail and file servers.

Most Linux distributions include a firewall in the distribution. Many can generate a default configuration during installation. Most desktop-oriented distributions provide a graphical tool of some type to set up iptables using “canned” default configurations that are adequate for most environments. If you require a highly customized configuration, consult the iptables man page or the iptables HOWTO on the Web at:

http://www.linuxguruz.org/iptables/howto/

The following sections present a few representative examples of a variety of open source and commercial tools that are available to reduce susceptibility to viruses and hack attempts.

McAfee VirusScan Command Line Scanner for Linux

McAfee VirusScan Command Line Scanner for Linux brings antivirus protection into the Linux arena. This scanner is a native virus-scanning tool that is available on a wide variety of desktop and server platforms. Administrators can use standard UNIX task-scheduling tools, such as **chron** to automate the scanning of local- and network-mounted file systems. This tool also retrieves updated virus definitions over the network.

For more information about McAfee VirusScan Command Line Scanner for Linux, see:

`http://www.networkassociates.com/`

Symantec Antivirus Command Line Scanner 1.0

Symantec Antivirus Command Line Scanner 1.0 is another well-known product that is now available for Linux and other desktop and server platforms. Like other command line scanners on Linux, **`chron`** or similar tools are useful to automate the scanning of local- and network-mounted file systems. Symantec updates its protection capabilities over the network.

You can learn more about Symantec Antivirus Command Line Scanner 1.0 at:

`http://www.symantec.com/product/`

Tripwire

Tripwire is an open source file-activity monitor (commercial versions are available for other platforms including Windows) that can detect when critical files have been altered. For example, it may detect when /etc files have been changed, binaries have been replaced with trojans, and so on. It can take some administrator-specified action (for example, e-mail an alert to a system administrator) when an unexpected change occurs.

Tripwire is completely configurable by the administrator. It allows flexibility over what files are monitored and what actions are taken. Tripwire is included in some distributions, including Red Hat and Caldera.

You can learn more about Tripwire on the Web at:

`http://www.tripwire.com`

Kaspersky Corporate Suite

Kaspersky Labs is a provider for several environments that range from home users to corporate customers. Kaspersky Corporate Suite is a fully-supported product that offers workstation, file server, and Web server protection. It also offers daily antivirus database updates over the Internet using a minimum of bandwidth. The suite can also be centrally administered.

To learn more about Kaspersky Corporate Suite, see:

`http://www.kaspersky.com`

TrendMicro ServerProtect for Linux

TrendMicro Server Protect for Linux is an antivirus solution that you can deploy on server or client machines. It provides the same sets of functions that other antivirus products have such as manual or automatic updates pulled from the Internet, scanning macros and scripting in files, and scanning files for virus signatures. If outbreaks or infections occur, you can configure this product to send notifications of such events by e-mail or Simple Network Management Protocol (SNMP) to administrators. This product also allows for remote administration using such Web browsers as Mozilla or Internet Explorer.

For more information about TrendMicro ServerProtect for Linux, go to:

`http://www.trendmicro.com/`

F-Secure Anti-Virus Total Suite

F-Secure Corporation is a software security company that offers a wide variety of security products on many different platforms and in multiple languages. F-Secure has a small footprint and fast scanning abilities.

To learn more about the various F-Secure products, visit:

`http://www.f-secure.com`

3.8 Manageability

For years, programs have existed for remote access and control of UNIX, since UNIX and, in particular, X11, are well suited for these tasks. Similar capabilities have been included in Linux. Also the XFree96 X server works smoothly with programs for remote access such as Hummingbird® Exceed.

Administrators looks for several features when they consider remote management products. Among these features are remote application installation and configuration, backup and recovery of the desktop machine if needed, and reinstallation. Such requirements can be satisfied in a number of ways. The range starts with a simple terminal accessing the client hard drive and the ability to remotely run commands on the target machine. The range goes to a graphical representation of the subject desktop complete with control of the machine, including keyboard and mouse capturing.

Many situations exist where it is most expedient and efficient to use remote access in an enterprise environment. The most visible example where these products are in use today are in customer help centers. The option of using remote control products to resolve end-user issues can help to eliminate or drastically reduce the number of on-site appearances that a technician may need to make to make the desktop function again. Limiting such visits goes a long way toward lowering the total cost of ownership of a client platform.

Many applications are available that allow remote support and administration, a variety of which are mentioned here. For specific examples that use these and other remote management products, refer to Chapter 4, “Linux client administration” on page 91.

3.8.1 Telnet

Telnet is a program that allows login and command execution on a remote machine. It provides terminal-level access into a client workstation with no graphics capability. This makes it possible to edit or rebuild configurations and critical files.

Telnet is not a recommended method of remote access if there is a requirement for security. Passwords are transmitted in clear text that make the system vulnerable to packet-sniffing attacks.

3.8.2 SSH

SSH is a remote login program that allows command execution on a remote machine using secure, encrypted communications. You can use SSH in much the same way that you use Telnet. It can be a useful tool to rebuild clients that are not properly configured.

An administration client that has X Window System support can run X applications remotely from the client being administered (for example, configuration tools). You must run the **sshd** daemon on the client to enable this possibility.

SSH is included with most Linux distributions. It has emerged as a secure replacement for Telnet.

3.8.3 Hummingbird Exceed

Hummingbird allows management of Linux clients from Windows workstations. Hummingbird is an X server, terminal emulator, and a complete set of TCP/IP utilities for Windows machines. It has been used for X Window System remote access to UNIX systems from PC’s for years. This product includes SSH2 supporting X, Telnet, FTP, and password protection using the RC5 algorithm. This product is produced and sold by Hummingbird, Ltd.

3.8.4 Reflection X

Reflection is a product from WRQ, Inc. It is an X server that allows remote access to Linux clients, rendering and displaying graphical X Window System applications on a Windows machine.

Reflection has built in GLX support, so it can run OpenGL 3-D applications. It also supports multiple transports and includes a Linux connection template and script to automate the establishment of Linux sessions.

3.8.5 Webmin

Webmin provides a modular, Web-based administration console to remotely manage a Linux system. For example, Webmin can help to manage local printer queues, local user accounts, disk quotas, servers and services running at the client, and so on.

The Webmin architecture is open. This means that developers can extend Webmin by writing custom modules for specific administrator tasks.

Webmin is actively supported by the open source community. It runs on most Linux distributions and several other UNIX-like operating systems.

For more information, visit:

http://www.webmin.com

3.8.6 VNC

Virtual Network Computer (VNC) has two components. One is on the server, which can be a Linux client workstation running the VNC Server. The other is the client, which can be run on Windows, UNIX and Linux, OS/2, DOS, and Mac.

VNC is fully cross platform. There is a Java viewer for remote control of machines from any Internet browser that is Java-capable.

VNC is free. It can be redistributed under the terms of the GNU General Public License (GPL). The VNC server and client are included in most Linux distributions.

You can download VNC from the Web at:

http://www.realvnc.com

3.9 File systems

Linux already has several advantages over OS/2 and Windows in terms of reliability and availability. A chief advantage is the fact that you can apply system configuration changes and software updates (for example, security patches) without requiring a system reboot. However, you may need to restart affected applications and services.

Another feature of Linux that promotes reliability is its support for journaling file systems. Journaling file systems can reduce the likelihood of data loss in the event of a crash or power failure and typically recover (file system integrity check) faster after a forced shutdown. Using a journaling file system may, in theory, result in a slight (10% to 15%) input/output (I/O) performance degradation versus a non-journaled file system. In most client environments, this impact is not noticeable.

Some of the more common file systems for Linux are:

- **ext2** has been the standard Linux file system for many years and is non-journaling.
- **ext3** is an enhanced version of ext2 that adds journaling capabilities. Developed and used primarily by Red Hat, ext3 can add journaling capabilities to existing ext2 file systems. For more information, see:

  `http://www.redhat.com/support/wpapers/redhat/ext3/`
- **ReiserFS** is one of the first journaling file systems to gain widespread use on Linux and is relatively robust. ReiserFS is favored by SuSE and Mandrake. To learn more about ReiserFS, see:

  `http://www.namesys.com/v4/v4.html`
- **JFS** is an IBM-sponsored open source implementation of the journaled file system (JFS) used for many years with AIX and Warp Server for e-business. The Linux version of JFS is useful when migrating from OS/2 to Linux. You can mount and access OS/2 JFS volumes from Linux. You may need to modify file ownership and permissions on the OS/2 volume to access files.

  As of this writing, no mainstream distributions are using JFS as the default file system. We recommend that you do not use JFS for the /boot partition.

  You can learn more about JFS on the Web at:

  `http://www.ibm.com/developerworks/oss/jfs/`
- **XFS** is a journaling file system donated to open source by SGI. Among other features, XFS provides native support for extended attributes and POSIX ACLs. As of this writing, no mainstream distributions are using XFS as the default file system. However, it is included with the Mandrake distribution,

along with a version of Samba compiled with disk quota and Windows NT® ACL support when using XFS.

For more information about XFS, see:

http://oss.sgi.com/projects/xfs/

You can further enhance availability by devising a recovery system that allows a client image to be rapidly rebuilt (with minimal or no local IT expertise) in the event of software corruption, virus infection, or hardware failure. The ability to rebuild a client system from a bootable CD or remotely over the network minimizes the downtime of the client system. You can learn more about installation in Chapter 7, “Linux client installation” on page 145.

3.10 Financial device support

Many OS/2 users are in the banking, financial services, or retail industries. These end-point machines frequently have hardware devices attached via serial ports or Universal Serial Bus (USB). A single machine may have one or more of the following devices attached:

- Cash drawer
- Magnetic stripe reader and PIN pad
- MICR check reader
- Passbook or other specialized printer
- Barcode reader

In OS/2, these devices most likely communicated with applications using a different native device driver for each piece of hardware. Migrating these workstations to Linux or any other platform necessitates that these applications and device drivers at least need to be recompiled if not rewritten.

However, for many customers, now is the time to consider *Java/Extensions for Financial Services (J/XFS)*. J/XFS complements Microsoft’s Extensions Financial Services (XFS) standard. The goal of J/XFS is to encourage applications to be written in Java, to run the same application on many different platforms, including Linux. Because it is Java based, the application can be designed to be an applet. This allows for a thin-client model and central administration of the product.

J/XFS enables the application to be written with little or no knowledge of the specific device that the application is managing. It standardizes the functionality that is common for types of devices. For example, an application only needs to know that it is interfacing with a PIN pad, but does not need to know the brand or type of PIN pad.

On the device end of J/XFS, there needs to be a *device service* (think device driver) to interact between the device and J/XFS. It communicates with the device using the Java Communication APIs. The device service is written to the specific device and is usually provided by the device manufacturer.

Third-party companies, such as Dynasty Global, Inc. and LUTZWOLF Systems, GmbH, have already developed many device services for legacy equipment. They continue to partner with hardware manufacturers to develop J/XFS device services for old and new products, and to manage J/XFS environments and J/XFS application development. J/XFS device services are also available from IBM for legacy IBM equipment.

For more information about J/XFS, see:

http://www.jxfs.com/documentation.html

To learn more about Dynasty Global, see:

http://www.dynastyglobal.com

And for more information about LUTZWOLF Systems, see:

http://www.lutzwolf.com/

3.11 Summary

This chapter provided an overview of many of the functional requirements for a client system that are common among enterprises currently using OS/2. These functional requirements include various applications such as terminal emulators, Web browsers, and office suites. There are also functional requirements that are not associated with end user applications, but none the less are important for the support and management of client systems. Such requirements include support for management facilities, robust file systems, and specialized devices.

As you saw in this chapter, a wide range of products and solutions are available from a large number of vendors that can address these requirements. Customers need to evaluate their specific requirements and match those with the capabilities of these and other solutions to determine if a Linux-based solution is appropriate for their user base.

4

Linux client administration

This chapter explains how you can manage a Linux client in a complex network environment, including graphical administration, remote user assistance, and backups.

4.1 Local graphical client administration

Managing Linux from a command line can be hard work even for a skilled administrator, since configuration parameters often change from one Linux version to another and among distributions. To manage individual systems, graphical administration tools often provide a fast and good solution.

This section discusses two types of graphical administration tools:

- **Local and proprietary**: Each distribution runs its own tool, for instance, YaST2 on SuSE and Red Hat Config on Red Hat.
- **Remote and common**: This type allows centralized administration of different Linux distributions, even from a non-Linux machine, for such tools as Webmin.

4.1.1 SuSE YaST2

YaST2, Yet Another Setup Tool, is the configuration tool for SuSE. It has two front ends: ncurses (text) and QT (graphical).

YaST2 is started from a command line or from the system menu and requires a root user. If it is run by another user, it asks for a root password before starting.

To start YaST2 from a command line, simply type:

```
yast2
```

If the DISPLAY variable is not set, YaST2 starts in ncurses mode. If set, it starts in graphical mode.

Yast2 is made from individual modules. Running **yast2** starts the YaST2 control center from which any module can be accessed clicking over the right icon. You can start each module, instead, directly from the command line by passing the module name as a parameter:

```
yast2 module-name
```

Here *module-name* can be any module name obtained by running:

```
yast2 -l
```

Some useful examples are:

- **autoyast**: Creates an Extensible Markup Language (XML) file for SuSE unattended installation
- **backup**: Creates a backup of the current installation
- **bootfloppy**: Creates a boot floppy

- **disk**: Manages disk partitions, including Logical Volume Manager (LVM)
- **dns**: Manages the machine name and Domain Name System (DNS)
- **fax**: Prepares local workstation to send and receive faxes
- **firewall**: Configures the Linux local firewall (SuSE Firewall2)
- **host**: Edits the /etc/hostname file
- **hwinfo**: Gives some information about the hardware
- **inst_source**: Configures the places where SuSE looks for updates; can be either File Transfer Protocol (FTP), Hypertext Transfer Protocol (HTTP), Samba, Network File System (NFS), CD, DVD or local directories
- **lan**: Configures local area network (LAN) connections
- **language**: Changes the Linux language
- **ldap**: Configures the workstation to authenticate users with a Lightweight Directory Access (LDAP) server using an LDAP client
- **lvm_config**: Manages the LVM from volume groups to logical volumes
- **modem**: Configures telephone connections
- **nfs**: Configures NFS client connections
- **nfs_server**: Configures the local workstation as an NFS server and manages file system exports
- **online_update**: Starts YaST Online Update (YoU) to update SuSE from FTP or HTTP servers
- **powertweak**: Configures the system's advanced options
- **printer**: Manages printers
- **profile-manager**: Manages SCPM, which stores different system profiles

  These profiles are really the same system configuration, but SCPM attaches to each profile created user scripts that can change system configuration after or before starting or stopping the system.
- **restore**: Allows restoring a previously made backup
- **security**: Manages security options for the system
- **sound**: Configures sound
- **sw_single**: Installs or removes software packages that came with the distribution
- **sysconfig**: Changes the configuration options from the system such as the display manager used

A good trick to avoid password prompting or ncurses mode is to configure the root to start graphical tools where the user that started X may be located. To accomplish this, edit the common profile file and add these lines to the end:

```
if [ $USER = root ]
then
export DISPLAY=:0.0
else
xhost +local:root >/dev/null 2>&1
fi
```

The common profile file differs among distributions. For SuSE, it is /etc/bash.bashrc. For Red Hat, it is /etc/bashrc. SuSE recommends that you create an independent profile called /etc/bash.bashrc.local to avoid overwrites when updating the distribution.

Let’s look at some YaST2 screens. Figure 4-1 shows the main window.

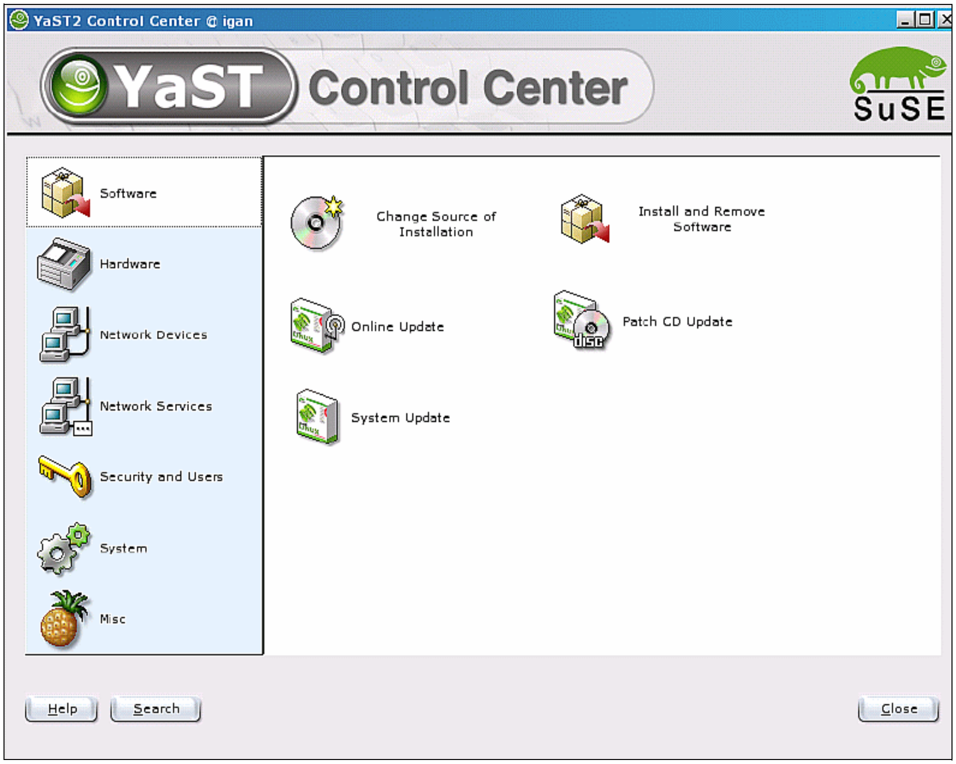


Figure 4-1 YaST2 main window

You can click the Hardware icon to access the Hardware configuration utility, which is shown in Figure 4-2.

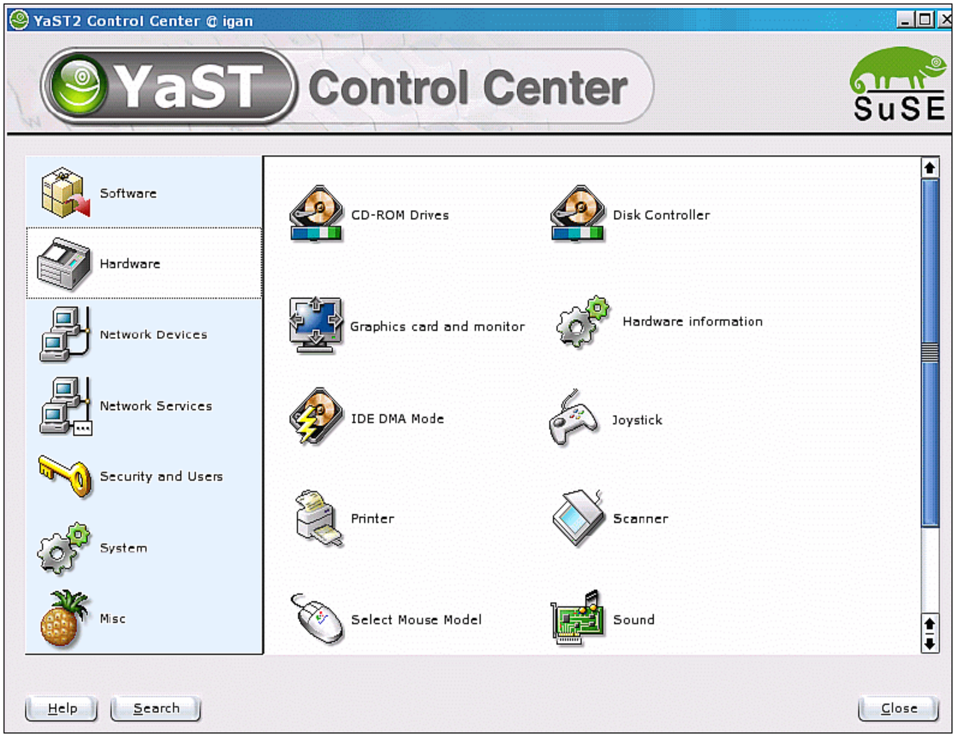
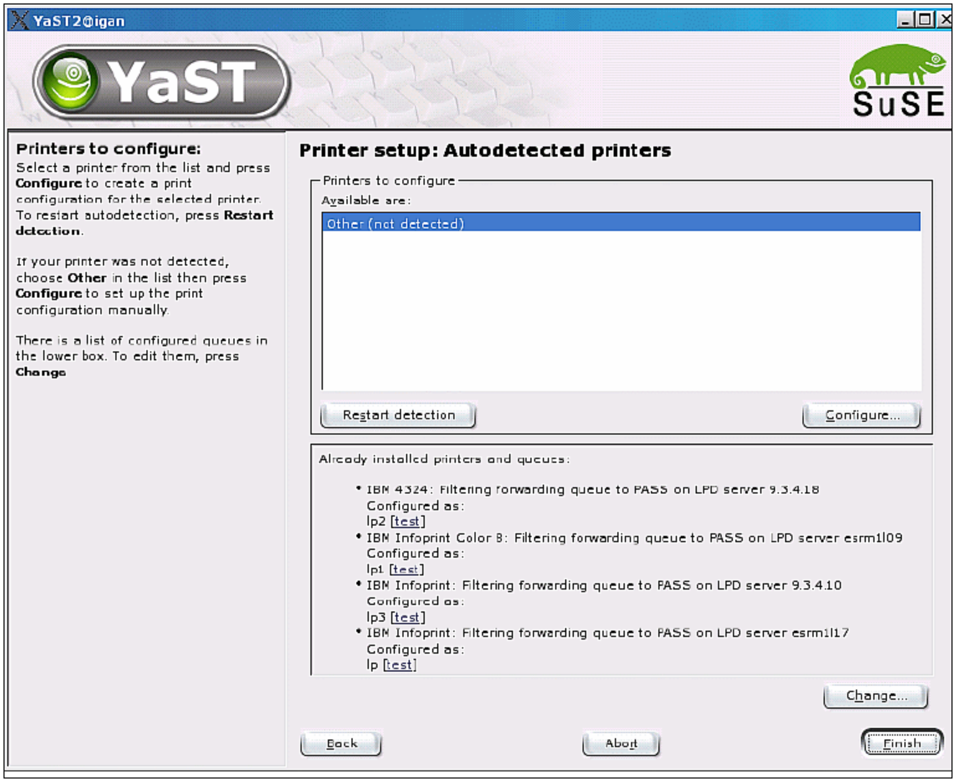


Figure 4-2 Hardware configuration utility

Or you can click the Printer icon to set up the Printer configuration. See Figure 4-3.



Printers to configure:
Select a printer from the list and press **Configure** to create a print configuration for the selected printer. To restart autodetection, press **Restart detection**.

If your printer was not detected, choose **Other** in the list then press **Configure** to set up the print configuration manually.

There is a list of configured queues in the lower box. To edit them, press **Change**

Printer setup: Autodetected printers

Printers to configure
Available are:
Other (not detected)

Already installed printers and queues:

- IBM 4324: Filtering forwarding queue to PASS on LPD server 9.3.4.18
  Configured as:
  lp2 [test]
- IBM Infoprint Color 8: Filtering forwarding queue to PASS on LPD server esrm1l09
  Configured as:
  lp1 [test]
- IBM Infoprint: Filtering forwarding queue to PASS on LPD server 9.3.4.10
  Configured as:
  lp3 [test]
- IBM Infoprint: Filtering forwarding queue to PASS on LPD server esrm1l17
  Configured as:
  lp [test]

Figure 4-3 Printer setup

From YaST2, you can start many configuration modules one by one. They are all independent so closing one of them doesn't close any of the other modules.

4.1.2 Red Hat administration

Red Hat executes its management applications as independent modules that can be started by typing their names. You can see them from the KDE desktop by clicking the Start Here icon. This opens Konqueror (KDE navigator) to display the icons for the different applications.

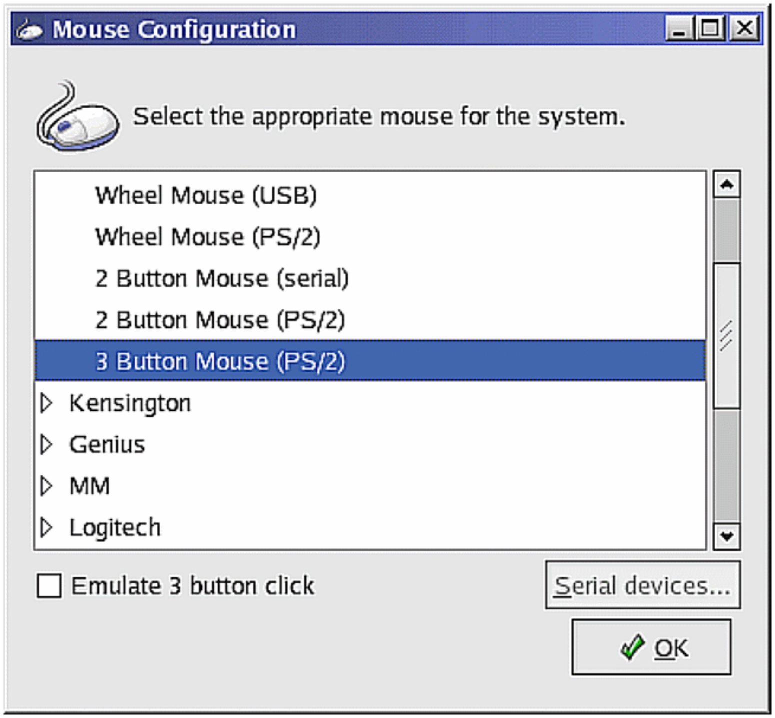
The programs that you need to run to configure each part of Red Hat are:

```
redhat-config-mouse
redhat-config-nfs
redhat-config-packages
redhat-config-printer
```

```
redhat-config-date
redhat-config-bind
redhat-config-securitylevel
redhat-config-services
redhat-config-keyboard
redhat-config-language
redhat-config-proc
redhat-config-users
redhat-rpm-config
redhat-config-network
redhat-config-httpd
redhat-config-samba
redhat-config-kickstart
redhat-config-rootpassword
redhat-config-soundcard
redhat-config-xfree86
redhat-config-network-tui
redhat-config-printer-gui
```

For example, to configure the mouse, you type the following command and then see the window shown in Figure 4-4:

```
redhat-config-mouse
```



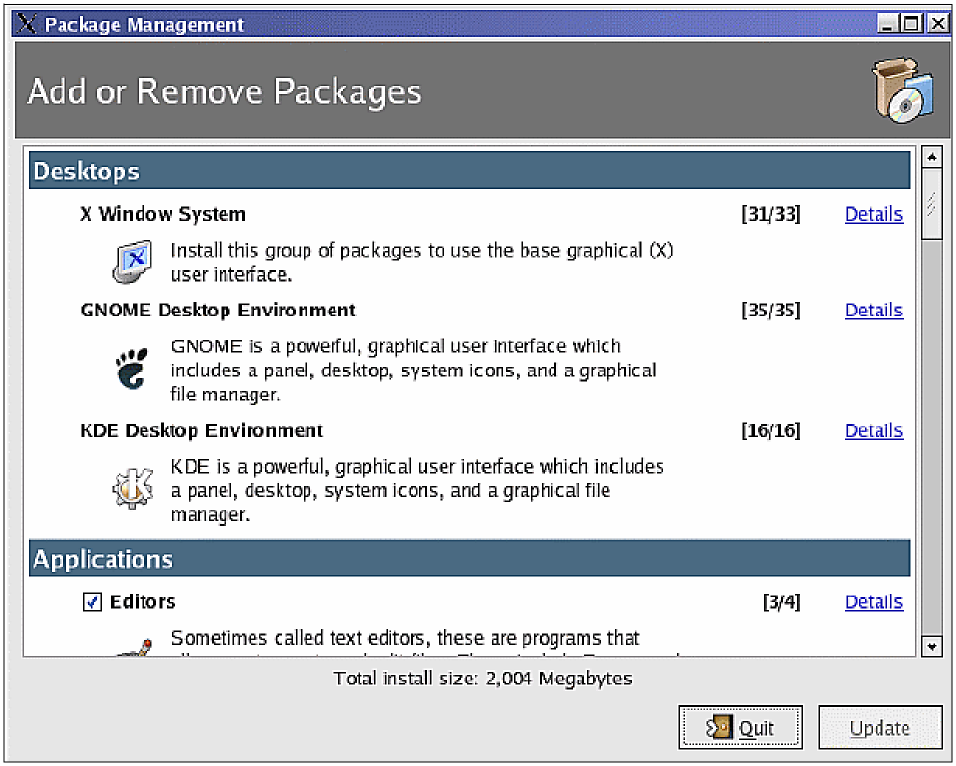
Mouse Configuration

Select the appropriate mouse for the system.

Figure 4-4 Redhat-config-mouse

Or to manage the software installed, type the following command and you see the Window shown in Figure 4-5:

```
redhat-config-packages:
```



Add or Remove Packages

Desktops

X Window System [31/33] Details

Install this group of packages to use the base graphical (X) user interface.

GNOME Desktop Environment [35/35] Details

GNOME is a powerful, graphical user interface which includes a panel, desktop, system icons, and a graphical file manager.

KDE Desktop Environment [16/16] Details

KDE is a powerful, graphical user interface which includes a panel, desktop, system icons, and a graphical file manager.

Applications

☑ Editors [3/4] Details

Sometimes called text editors, these are programs that

Total install size: 2,004 Megabytes

Figure 4-5 Redhat-config-packages

4.2 Remote graphical client administration

This section explains how to manage a Linux workstation remotely and graphically. Additionally, you can manage a remote workstation by starting YaST2 or RedHat-Config as a remote X application. It runs on the workstation but is displayed on the administration console (see 4.5.9, “Remote starting of graphical applications” on page 122).

4.2.1 Webmin

Webmin is a Web interface for UNIX systems administration. It works with Red Hat, SuSE, and many other UNIX systems.

Webmin consists of a Web server and several Common Gateway Interface (CGI) programs that update system configuration files. All are written in standard Perl 5.

Webmin comes with SuSE 8.2, but not with Red Hat 9. It is distributed under a BSD-like license. You can download the latest versions of Webmin from the Web at:

http://www.webmin.com

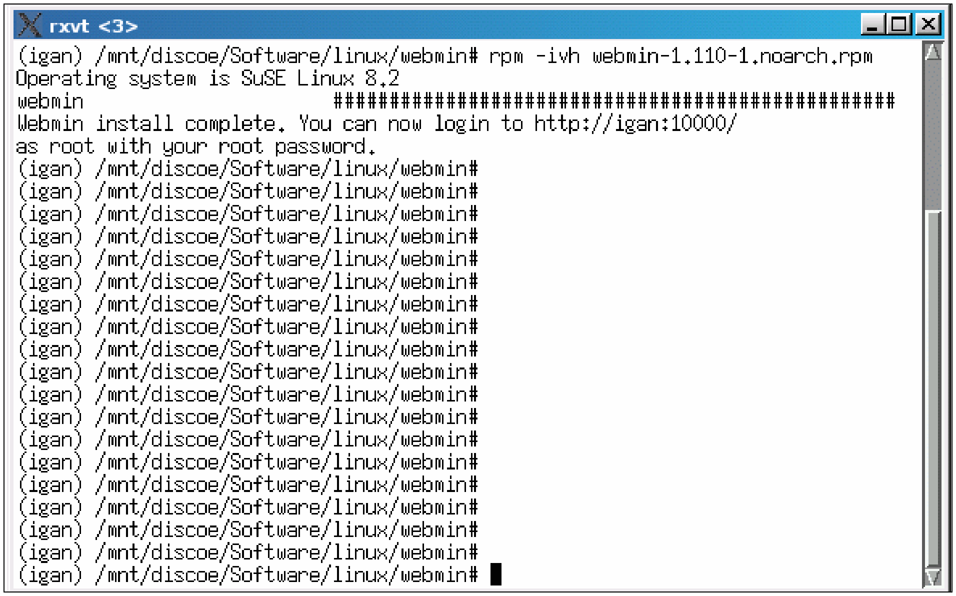
Webmin is made of modules. Anyone can write new modules to extend Webmin for their specific needs using Webmin application programming interfaces (APIs).

To install Webmin, first download the rpm package or the source files from the Webmin Internet site. Then run:

```
rpm -ivh webmin-xxx.rpm
```

Webmin needs another package to be installed, not as a prerequisite, but as something useful to it. Its name is *usermin*. Sermon provides a graphical interface for a user to manage their own environment.

Figure 4-6 shows the Webmin installation screen.



```
(igan) /mnt/discoe/Software/linux/webmin# rpm -ivh webmin-1.110-1.noarch.rpm
Operating system is SuSE Linux 8.2
webmin                      ##################################################
Webmin install complete. You can now login to http://igan:10000/
as root with your root password.
(igan) /mnt/discoe/Software/linux/webmin#
(igan) /mnt/discoe/Software/linux/webmin#
(igan) /mnt/discoe/Software/linux/webmin#
(igan) /mnt/discoe/Software/linux/webmin#
(igan) /mnt/discoe/Software/linux/webmin#
(igan) /mnt/discoe/Software/linux/webmin#
(igan) /mnt/discoe/Software/linux/webmin#
(igan) /mnt/discoe/Software/linux/webmin#
(igan) /mnt/discoe/Software/linux/webmin#
(igan) /mnt/discoe/Software/linux/webmin#
(igan) /mnt/discoe/Software/linux/webmin#
(igan) /mnt/discoe/Software/linux/webmin#
(igan) /mnt/discoe/Software/linux/webmin#
(igan) /mnt/discoe/Software/linux/webmin#
(igan) /mnt/discoe/Software/linux/webmin#
(igan) /mnt/discoe/Software/linux/webmin#
(igan) /mnt/discoe/Software/linux/webmin#
(igan) /mnt/discoe/Software/linux/webmin#
(igan) /mnt/discoe/Software/linux/webmin# █
```

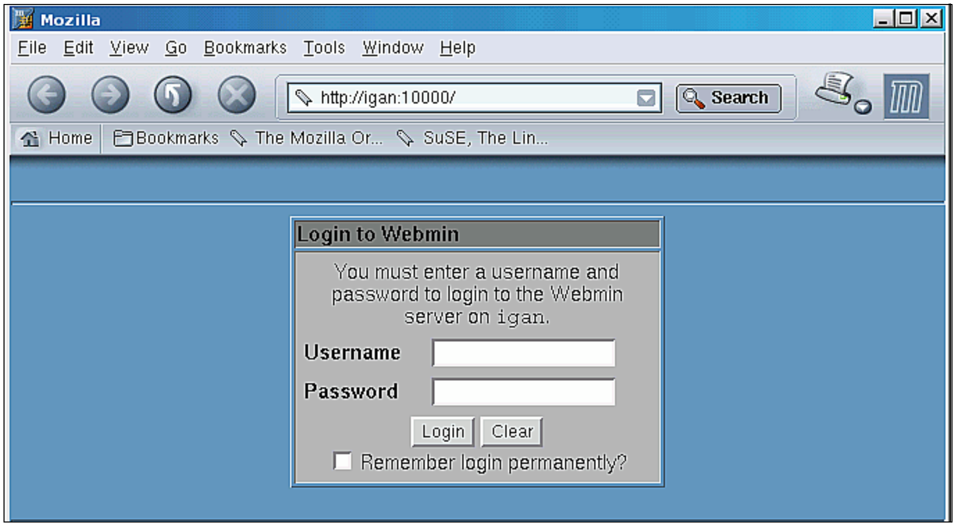
Figure 4-6 Webmin installation

Older versions of Webmin needed a script to run in order to configure it for the distribution, if it wasn't Red Hat. Version 1.110-1 of Webmin was tested in SuSE for this chapter. It didn't require any further configuration, only to install rpm.

You access Webmin from a Web browser, that can be on the same or a different machine than the one that is being administered. Simply point the browser to:

```
http://ip-address:10000
```

You are then prompted for a user and password as shown in Figure 4-7. The user should be root.



http://igan:10000/

You must enter a username and password to login to the Webmin server on igan.

□ Remember login permanently?

Figure 4-7   Webmin login

Then the Webmin main window (Figure 4-8) opens.

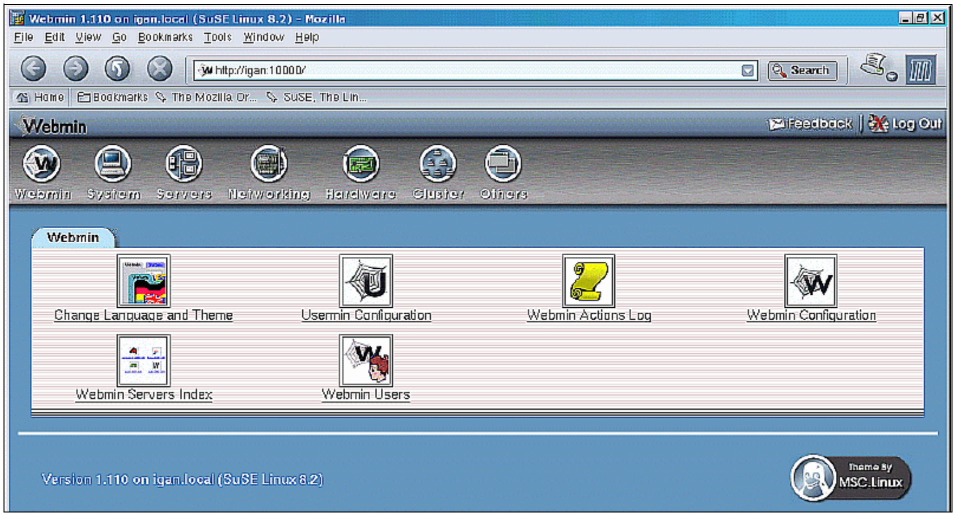


Figure 4-8 Webmin main window

The icons on the upper side (Figure 4-8) take the user to the particular system configuration tools. For example, system leads to Apache, CVS, DHCP, DNS, Fetchmail, MySQL, Samba, proxy, Sendmail and other servers configuration as shown in Figure 4-9.

> **Important:** Webmin returns errors if Samba is not installed and the user clicks the Samba icon. It does the same with each package that is not installed.

Click **Usermin Configuration**. If Usermin is not installed, it is automatically downloaded and installed.

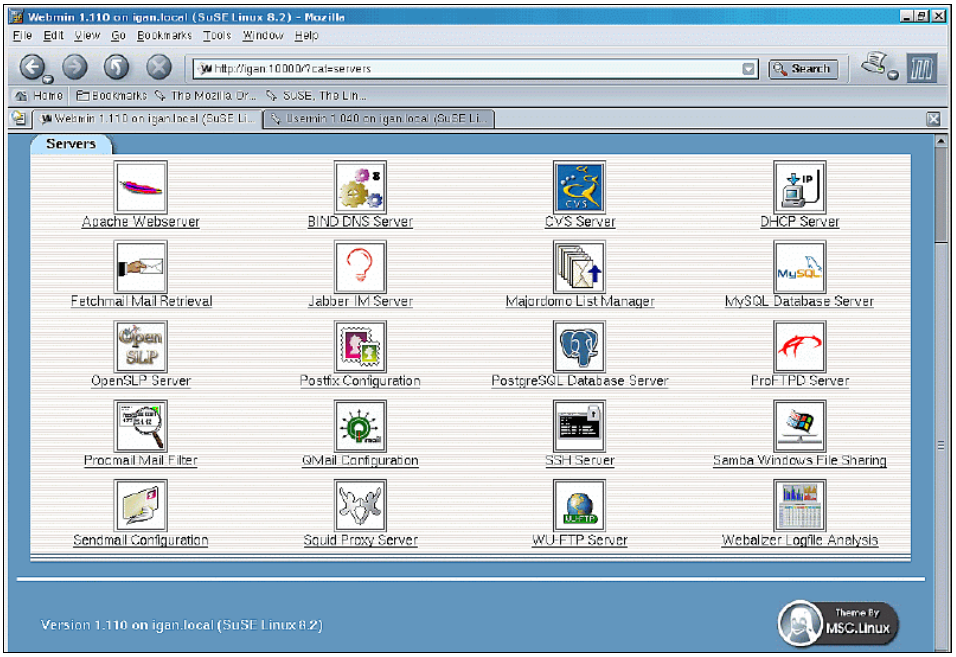
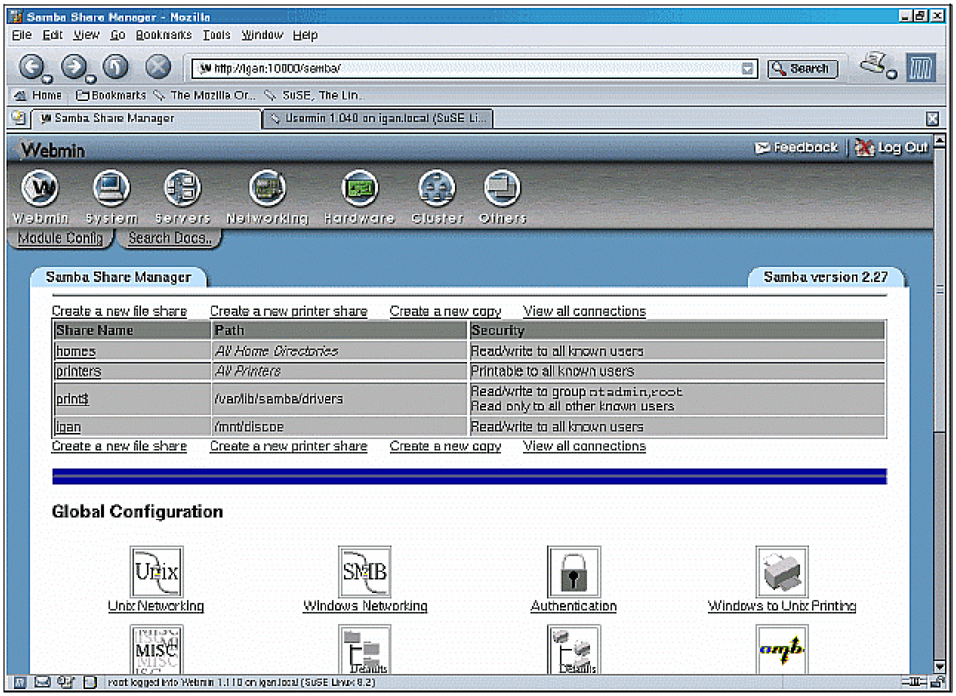

Webmin 1.110 on igan.local (SuSE Linux 8.2) - Mozilla
File Edit View Go Bookmarks Tools Window Help
http://igan:10000/?cat=servers
Search
Home Bookmarks The Mozilla Or... SuSE, The Lin...
Webmin 1.110 on igan.local (SuSE Li...
Usermin 1.040 on igan.local (SuSE Li...
Servers
Apache Webserver
BIND DNS Server
CVS Server
DHCP Server
Fetchmail Mail Retrieval
Jabber IM Server
Majordomo List Manager
MySQL Database Server
OpenSLP Server
Postfix Configuration
PostgreSQL Database Server
ProFTPD Server
Procmail Mail Filter
QMail Configuration
SSH Server
Samba Windows File Sharing
Sendmail Configuration
Squid Proxy Server
WU-FTP Server
Webalizer Logfile Analysis
Version 1.110 on igan.local (SuSE Linux 8.2)
Theme By MSC.Linux

Figure 4-9 Servers administration

Webmin sometimes calls other administration tools, which is the case for Samba. It links to Samba Web Administration Tool (SWAT) to manage Samba. However, in some cases, Webmin can also administer some aspects of Samba from other icons that don't use SWAT.

Figure 4-10 shows some of the Samba configuration options.


Samba Share Manager - Mozilla
File Edit View Go Bookmarks Tools Window Help
http://igan:10000/samba/
Search
Home Bookmarks The Mozilla Or... SuSE, The Lin...
Samba Share Manager
Usermin 1.040 on igan.local (SuSE Li...
Webmin
Feedback
Log Out
Webmin System Servers Networking Hardware Cluster Others
Module Config
Search Docs..
Samba Share Manager
Samba version 2.27

Create a new file share   Create a new printer share   Create a new copy   View all connections

Share Name	Path	Security
homes	*All Home Directories*	Read/write to all known users
printers	*All Printers*	Printable to all known users
print$	/var/lib/samba/drivers	Read/write to group `ntadmin,root` Read only to all other known users
igan	/mnt/discoe	Read/write to all known users

Create a new file share   Create a new printer share   Create a new copy   View all connections

Global Configuration

Figure 4-10 Samba configuration through Webmin

You can start, stop, and check Webmin by running the following commands:

```
/etc/init.d/webmin start
/etc/init.d/webmin stop
/etc/init.d/webmin status
```

Usermin is the graphical tool for a user to administer their own environment. From Usermin, a user can manage mail, login scripts, and the ssh configuration; use a file manager from the browser; and perform other functions.

You call Usermin from a Web browser that can be on a different machine than the one that is being administered. To call it, enter the following Uniform Resource Locator (URL):

```
http://ip-address:20000
```

You are then prompted for a user ID and password. This time you can use any login user of the machine.

4.3 Tivoli Linux systems administration

Tivoli is a complete IBM framework that assists administrators in managing a complex IT environment. It performs distributed backups by applying backup policies, installs software and updates, and controls workstations and servers remotely. Using all the information that it gathers from the different systems, Tivoli provides views of the company's IT infrastructure to help management and administrators to take decisions. In short, Tivoli provides a wide range of capabilities that cover performance, availability, provisioning, configuration, security, and storage management.

Linux is a supported managed environment for most of the Tivoli applications. To describe all of the capabilities to manage Linux clients using Tivoli would require a redbook of its own. Visit the Tivoli pages at the following Web site for more information about the Tivoli suite of products:

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

4.4 Keeping Linux up-to-date

Linux versions change quickly, often once a year. The Linux versioning policy may vary for different distributions and from one version to the next. The choice of when to move to a newer version is often driven by business considerations more than by technical ones.

The following sections describe some of the facilities that are available to assist administrators in keeping Linux clients (and servers) up to date.

4.4.1 SuSE YoU

SuSE updates itself using the YaST2 Online Update (YoU) tool. There are several ways to start YoU:

- From the command line, type:

  ```
  you
  ```

- From YaST2, select **Online Update**.
- From the command line, enter this shortcut:

  ```
  yast2 online_update
  ```

YoU also starts updates from its scheduler. A root user is required to start YoU. Other users that start YoU are prompted for the root password.

If root is set to start graphic applications or root has started X, YoU starts in graphics mode. If not, it starts in ncurses (text) mode.

4.4.2 Red Hat Network

You can access Red Hat Network on the Web at:

http://rhn.redhat.com

Red Hat Network allows you to update a client from the Web and directly install, or download the necessary rpm packages to install later. You must register the first time you use it. It creates a system profile that is stored remotely.

Updating the system is a simple task. You simply click over the RedHat Network Alert Notification Tool icon on the tool bar. It shows which updates are found on the Red Hat Network and lets the user choose which ones to install. This icon changes to show if updates are pending.

From the command line, you can do this by entering:

```
up2date
```

4.4.3 Behind the scenes of automated updating

Both distributions differ in regard to automated updating.

Red Hat connects to Red Hat Network over Secured HTTP (HTTPS). In the first connection, you must register and a system profile is created. Up2date downloads an Secure Sockets Layer (SSL) certificate for HTTPS connections.

SuSE uses FTP or HTTP. YoU calls **wget** in the background to achieve this task. **wget** is a non-interactive tool that permits downloading in the background and completing of a previously broken FTP session. It may need some simple configuration by setting the HTTP and FTP proxies if needed, by exporting variables as follows:

```
export http_proxy=proxy-name:port
export ftp_proxy=proxy-name:port
```

4.5 User and workstation remote support

When choosing a client platform, the ability to provide remote administration and support is important. This section discusses some ways to perform remote workstation administration. Some of these facilities provide a graphical tool for system configuration and administration. Others allow the administrator to view what a user is doing and even to take control of the mouse and keyboard of a remote workstation.

4.5.1 Remote FrameBuffer protocol products

The Remote FrameBuffer (RFB) protocol is a relatively simple and light-weight protocol for sending display buffers that contain graphical content to be displayed on a remote display. There are several products that use this protocol to connect RFB clients to display applications that are actually running on the RFB server. A few of these are:

- VNC
- KRFB - KRCD
- TightVNC

4.5.2 Configuring a simple VNC

VNC is one of the applications that, based on RFB, you can use to remotely administer a machine in a graphical mode. This section explains a simple VNC configuration to access a VNC server from a VNC client. A password is required the first time it is started.

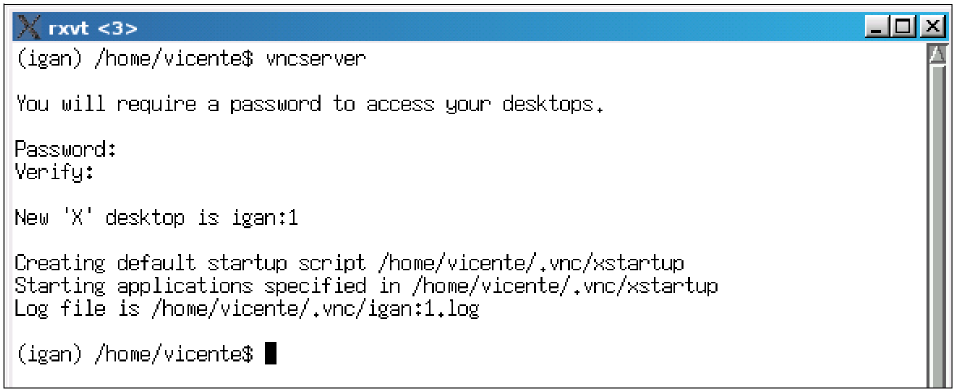
First, you must install the package VNC with its dependencies. This package contains the server and client software. The programs to be run are **`vncserver`** for the server and **`vncviewer`** for the client.

To start the VNC server, log into the workstation as a normal user. This can also be done by root, but it is a security exposure. Then type:

```
vncserver
```

You are prompted to enter a password. After you enter the password, VNC is ready for users to access it from a VNC client. This password is stored on the server. It is the same password to use to gain access from a VNC client regardless of the user that is logged on in the client system.

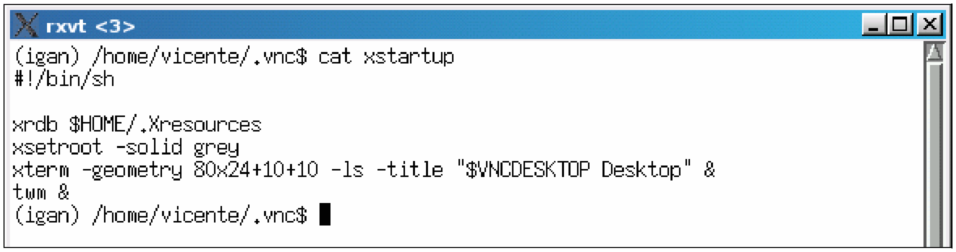
The window shown in Figure 4-11 represents the first time the VNC server is started from user *vicente*.



```
(igan) /home/vicente$ vncserver

You will require a password to access your desktops.

Password:
Verify:

New 'X' desktop is igan:1

Creating default startup script /home/vicente/.vnc/xstartup
Starting applications specified in /home/vicente/.vnc/xstartup
Log file is /home/vicente/.vnc/igan:1.log

(igan) /home/vicente$ █
```

Figure 4-11 Initial starting of VNC server

The VNC configuration creates a $HOME/.vnc directory where the configuration is stored. **xstartup** is used to define which applications to start when a VNC client logs in. Figure 4-12 shows a sample xstartup file.



```
(igan) /home/vicente/.vnc$ cat xstartup
#!/bin/sh

xrdb $HOME/.Xresources
xsetroot -solid grey
xterm -geometry 80x24+10+10 -ls -title "$VNCDESKTOP Desktop" &
twm &
(igan) /home/vicente/.vnc$ █
```

Figure 4-12 xstartup file for VNC server

The file starts a terminal and **twm**, a simple window manager. If the file is edited by deleting the line **twm &** and adding the following line, a KDE screen opens:

```
/usr/X11R6/bin/kde &
```

To connect from, for example, a Windows client running **vncviewer** (must be installed first), type the IP or hostname of the VNC server and its display number. The display number is shown when you start **vncserver**. If you forget the display number, you can enter:

```
ps -ef | grep Xvnc
```

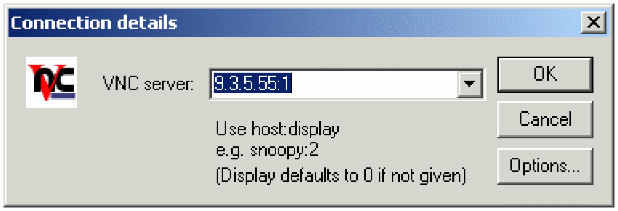
Then a line that contains the display number is shown such as:

```
Xvnc :1
```

This indicates that the display number is 1.

The following windows show the connection sequence.

First, you provide the server and port as shown in Figure 4-13.



VNC server: 9.3.5.55:1

Use host:display
e.g. snoopy:2
(Display defaults to 0 if not given)

Figure 4-13 VNC client connection: Host prompt

Then, you enter a password when prompted as shown in Figure 4-14.

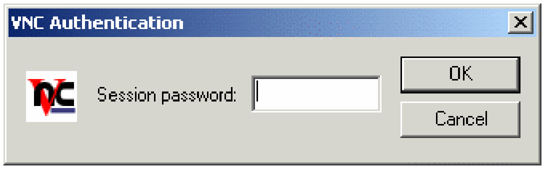


Figure 4-14 VNC client connection: Password prompt

If the password is correct, the user desktop window is shown (see Figure 4-15).

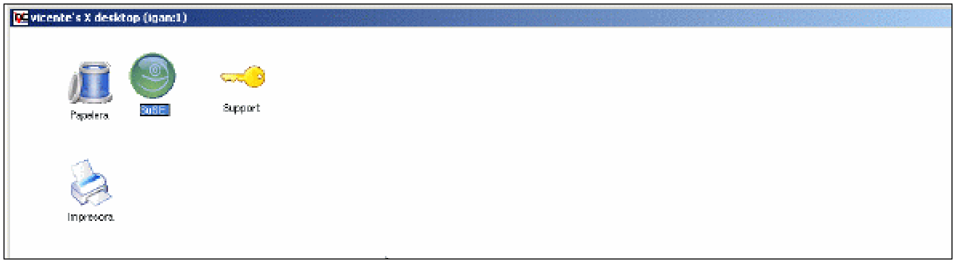


Figure 4-15 VNC client connected

The administrator (using the VNC client) can now perform any administration or support actions that are required on the user's system (the VNC server).

4.5.3 Configuring extended VNC

This section explains how to configure a VNC workstation for remote administration from the KDE Desktop Manager (KDM) entry point. This way, when a VNC client logs in, they see a KDM login panel instead of a KDE or GNOME session.

This configuration applies only to the SuSE distribution. It was tested on Version 8.2. Most of it also applies to later SuSE versions.

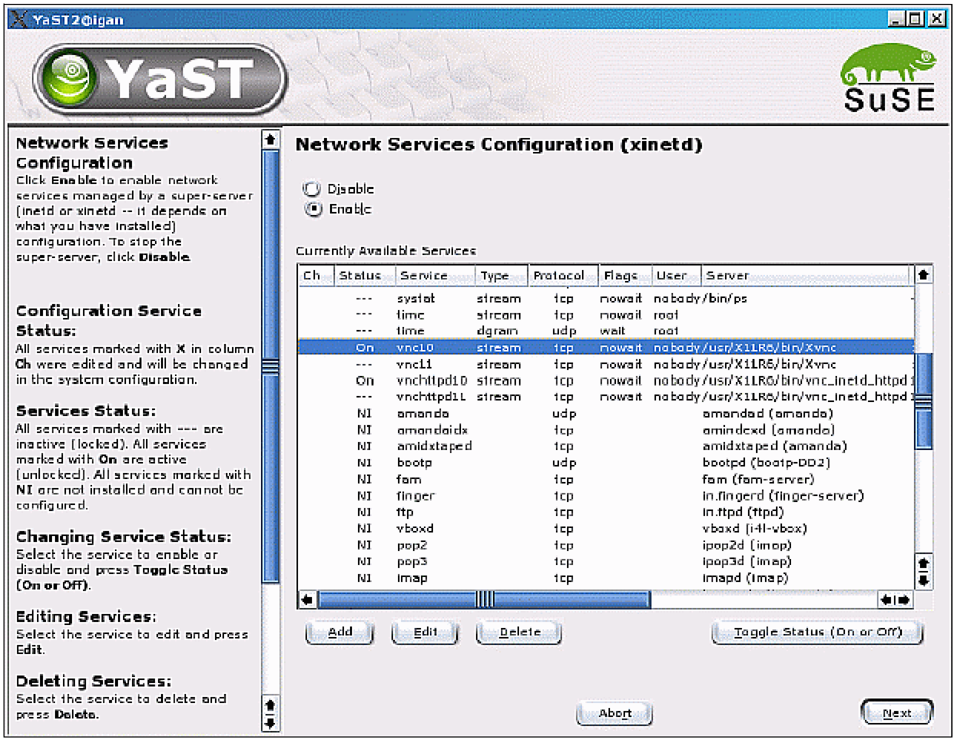
The VNC client does not prompt for a password. If it prompts for a password, you can press Enter or click the **OK** button without entering a password. After you connect to the VNC server, the KDE login appears. If a correct user and password are entered, a new X session is started with the desktop manager selected. KDM is the only display manager that allows this is at this point. From KDM, you can select any desktop manager.

The advantage of such a configuration is that you do not need to create a script and configuration file for each user to start the VNCServer. Nor do you need to maintain VNC passwords for each user.

Follow this example to see how to configure VNC this way:

1. Make sure that the package VNC is installed with its dependencies. This package contains the server and client software. The programs to be run are **`vncserver`** for the server and **`vncviewer`** for the client. In this configuration, the server is started as an xinetd service.
2. To configure xinetd, follow these steps:
   a. Start YaST2.
   b. In the left panel, select **Network Services**.
   c. In the main window, select **Network Services Configuration (xinetd)**.

3. On the Network Services Configuration window (Figure 4-16) that opens, the services *vnc10* and *vnchttpd10* must be activated. Click **Enable** to activate xinetd and click the services' **Toggle Status** button, one by one, to activate them. Click **Next**.



Network Services Configuration
Click **Enable** to enable network services managed by a super-server (inetd or xinetd -- it depends on what you have installed) configuration. To stop the super-server, click **Disable**.

Configuration Service Status:
All services marked with **X** in column **Ch** were edited and will be changed in the system configuration.

Services Status:
All services marked with --- are inactive (locked). All services marked with **On** are active (unlocked). All services marked with **NI** are not installed and cannot be configured.

Changing Service Status:
Select the service to enable or disable and press **Toggle Status (On or Off)**.

Editing Services:
Select the service to edit and press **Edit**.

Deleting Services:
Select the service to delete and press **Delete**.

Network Services Configuration (xinetd)

○ Disable
● Enable

Currently Available Services

Ch	Status	Service	Type	Protocol	Flags	User	Server
	---	systat	stream	tcp	nowait	nobody	/bin/ps
	---	time	stream	tcp	nowait	root	
	---	time	dgram	udp	wait	root	
	On	vnc10	stream	tcp	nowait	nobody	/usr/X11R6/bin/Xvnc
	---	vnc11	stream	tcp	nowait	nobody	/usr/X11R6/bin/Xvnc
	On	vnchttpd10	stream	tcp	nowait	nobody	/usr/X11R6/bin/vnc_inetd_httpd
	---	vnchttpd11	stream	tcp	nowait	nobody	/usr/X11R6/bin/vnc_inetd_httpd
	NI	amanda		udp			amandad (amanda)
	NI	amandaidx		tcp			amindexd (amanda)
	NI	amidxtaped		tcp			amidxtaped (amanda)
	NI	bootp		udp			bootpd (bootp-DD2)
	NI	fam		tcp			fam (fam-server)
	NI	finger		tcp			in.fingerd (finger-server)
	NI	ftp		tcp			in.ftpd (ftpd)
	NI	vboxd		tcp			vboxd (i4l-vbox)
	NI	pop2		tcp			ipop2d (imap)
	NI	pop3		tcp			ipop3d (imap)
	NI	imap		tcp			imapd (imap)

Figure 4-16 Xinetd configuration

4. Change to the /etc/xinetd.d directory and edit the vnc file to check that the correct services are activated. Example 4-1 shows an extract of the file with the configuration of the active services.

Example 4-1 Configuration extract file

```
# description: This serves out a VNC connection which starts at a KDM login
#   prompt. This VNC connection has a resolution of 1024x768, 16bit depth.
service vnc10
{
   socket_type     = stream
   protocol        = tcp
   wait            = no
   user            = nobody
   server          = /usr/X11R6/bin/Xvnc
   server_args     = :42 -inetd -once -query localhost -geometry 1024x768
-depth 8
   type   = UNLISTED
   port   = 5910
}
# description: This serves out the vncviewer Java applet for the VNC \
#  server running on port 5910, (vnc port 10).
service vnchttpd10
{
   socket_type     = stream
   protocol        = tcp
   wait            = no
   user            = nobody
   server          = /usr/X11R6/bin/vnc_inetd_httpd
   server_args     = 1024 768 5910 -depth 8
   type   = UNLISTED
   port   = 5810
}
```

The resolution has changed in the parameter -depth 8 (previously 16) for better performance.

5. Edit the /etc/sysconfig/displaymanager file and change the following lines as shown:

```
DISPLAYMANAGER_REMOTE_ACCESS="yes"
DISPLAYMANAGER_ROOT_LOGIN_REMOTE="yes"
```

6. Edit the /etc/opt/kde3/share/config/kdm/kdmrc file and change the next line on the stanza Xdmcp:

```
[Xdmcp]
Enable=true
```

7. To enable the changes, enter:

```
SuSEconfig --module xdm
```

8. Configure the display manager to respond to remote requests. You do this by editing several files. In the /etc/X11/xdm/xdm-config two file, you must modify two lines. The following example shows an extract of the file with the first line commented as it was and then modified, the third line added, and the last uncommented:

```
! DisplayManager.*.setup:/etc/X11/xdm/Xsetup
DisplayManager._0.setup:/etc/X11/xdm/Xsetup_0
DisplayManager.*.setup:/etc/X11/xdm/Xsetup_workstation
! SECURITY: do not listen for XDMCP or Chooser requests
! Comment out this line if you want to manage X terminals with xdm
DisplayManager.requestPort: 0
```

Now copy /etc/X11/xdm/Xsetup to Xsetup_0 and Xsetup_workstation to the same path.

9. Edit /etc/X11/xdm/Xaccess and uncomment the line:

   ```
   * any host can get a login window
   ```

10. Reboot the workstation or re-enter level 5 to make the changes take effect. Run the command sequence:

    ```
    init 3; sleep 5; init 5
    ```

11. Make sure that xinetd is up and running. Enter:

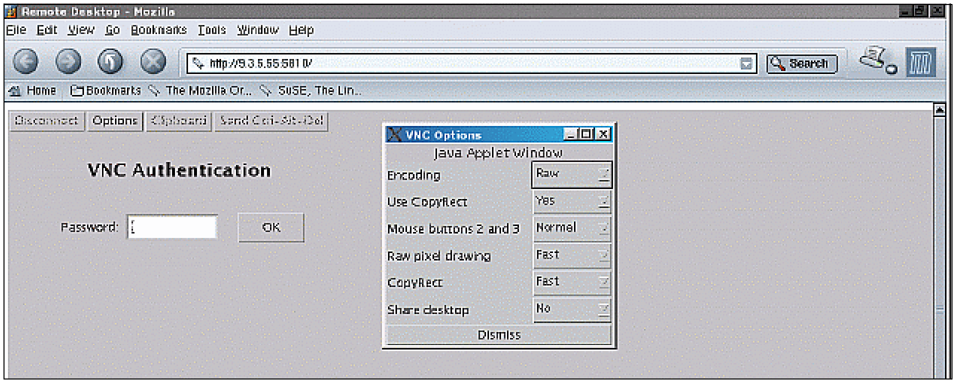
    ```
    /etc/init.d/xinetd status
    ```

    If it is not started, start it by entering:

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

Now the workstation serves VNC client connections and does so every time it restarts.

There are two ways to connect to the VNC server in this advanced configuration. The first way is to use vncviewer or any VNC client from Linux or other operating systems connecting to port 10 or 5910. The second way to connect to the VNC server in this configuration is from a Web client. For example, you enter `http://ip-address:5810` as shown in Figure 4-17.



http://9.3.5.55:5810/

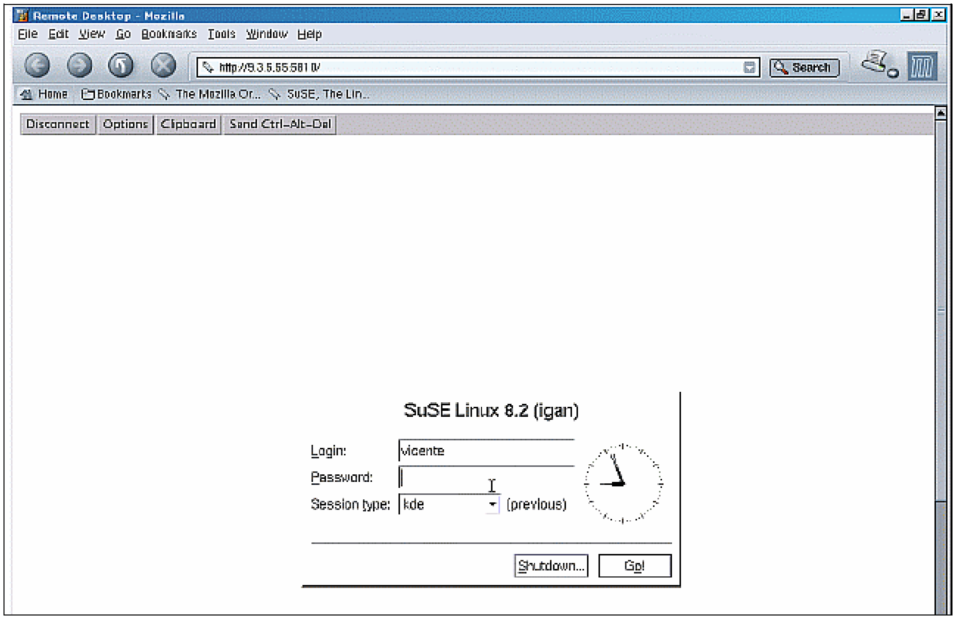
VNC Authentication

Password:

Encoding: Raw
Use CopyRect: Yes
Mouse buttons 2 and 3: Normal
Raw pixel drawing: Fast
CopyRect: Fast
Share desktop: No

Figure 4-17 VNC Web client connecting

You simply click **OK** without entering a password. Then you see the window shown in Figure 4-18. On this window, you type the user ID and password, select session type and click **Go!** to log in as in a local session.



SuSE Linux 8.2 (igan)

Login: vicente
Password:
Session type: kde (previous)

Figure 4-18 VNC Web client connected

4.5.4 KRFB

KRFB is a VNC clone that is installed with KDE. It is based on the RFB protocol. It comes packaged as *kdenetwork3-vnc*.

KRFB contains the **krfb** commands for the server and **krdc** commands to start the client. You can also start the client and server by using the KDE menu. You select **Internet -> Tools -> Desktop Sharing** for the server and **Remote Desktop Connection** for the client.

KRFB differs from VNC in that **krdc** can take control over the remote workstation keyboard and mouse, if this is granted. If not, it simply shows the remote display. You can use it for remote user support.

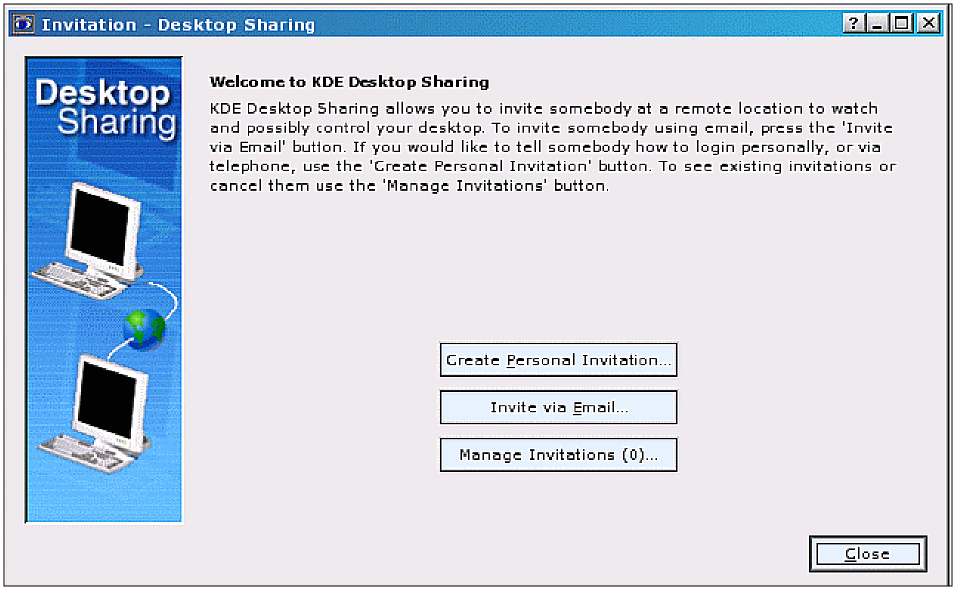
Let’s see an example:

1. In the remote workstation, start KRFB. You can do this from a terminal emulation by typing:

   ```
   krfb
   ```

   Or from the KDE menu, select **Internet -> Tools -> Desktop Sharing**.

2. On the Welcome to KDE Desktop Sharing panel (Figure 4-19), click the **Create Personal Invitation** button.

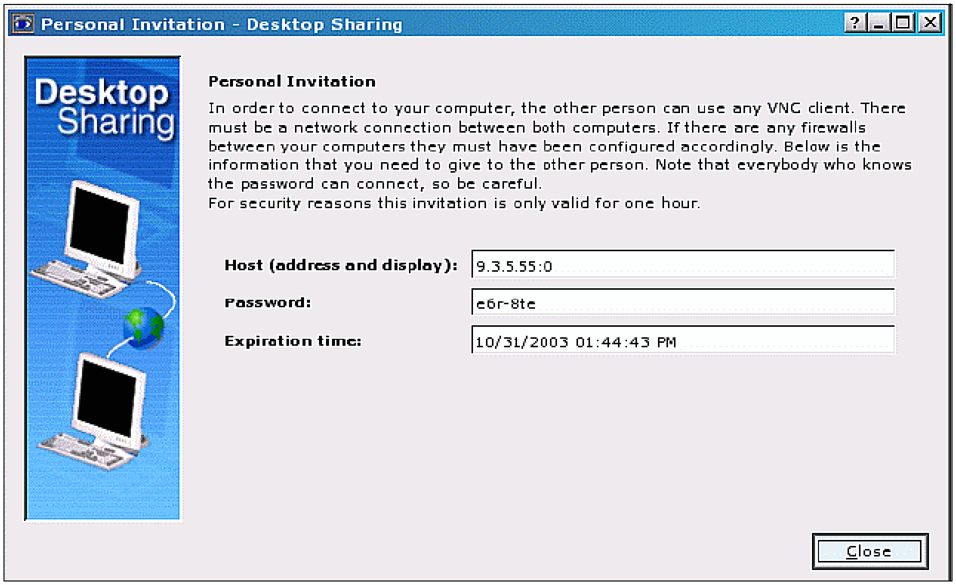


Welcome to KDE Desktop Sharing

KDE Desktop Sharing allows you to invite somebody at a remote location to watch and possibly control your desktop. To invite somebody using email, press the 'Invite via Email' button. If you would like to tell somebody how to login personally, or via telephone, use the 'Create Personal Invitation' button. To see existing invitations or cancel them use the 'Manage Invitations' button.

Figure 4-19 KRFB to create an invitation

3. On the Personal Invitation panel (Figure 4-20), click **Close**.



Personal Invitation - Desktop Sharing

Personal Invitation

In order to connect to your computer, the other person can use any VNC client. There must be a network connection between both computers. If there are any firewalls between your computers they must have been configured accordingly. Below is the information that you need to give to the other person. Note that everybody who knows the password can connect, so be careful.
For security reasons this invitation is only valid for one hour.

Host (address and display): 9.3.5.55:0

Password: e6r-8te

Expiration time: 10/31/2003 01:44:43 PM

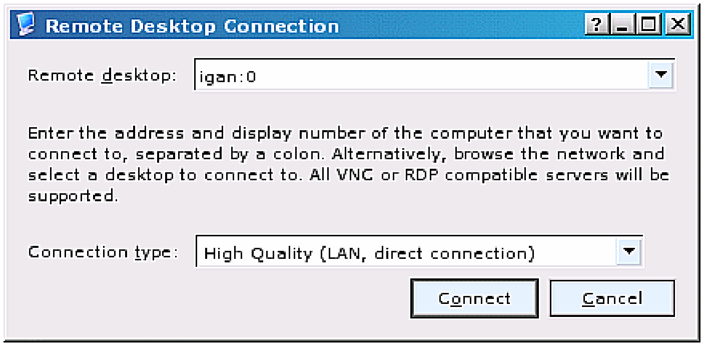
Figure 4-20 KRFB Personal Invitation panel

4. From the administrator workstation, open a VNC client. In this example, let's use KRCD. You start it from a terminal session by typing:

```
krcd
```

   Or from the KDE menu, select **Internet -> Tools -> Remote Desktop Connection**.

5. On the Remote Desktop Connection window (Figure 4-21), type the remote desktop name.



Remote Desktop Connection

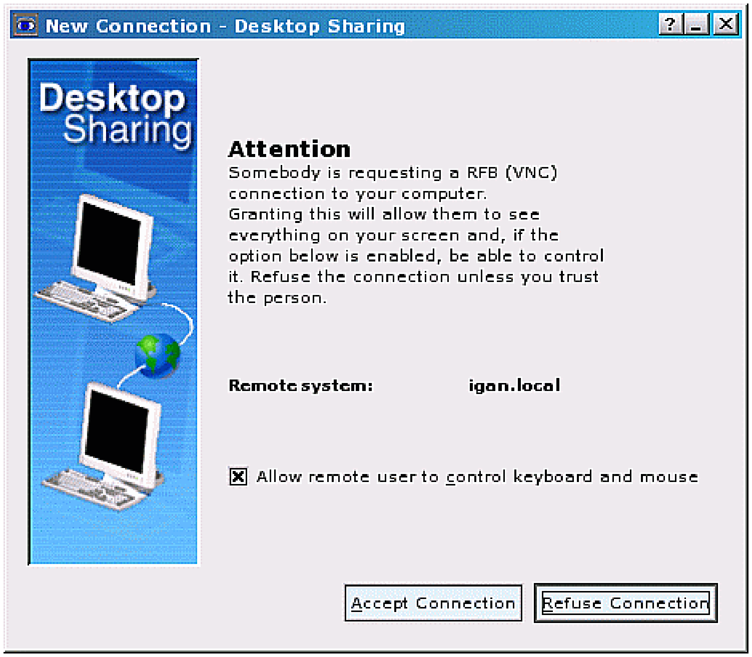
Remote desktop: igan:0

Enter the address and display number of the computer that you want to connect to, separated by a colon. Alternatively, browse the network and select a desktop to connect to. All VNC or RDP compatible servers will be supported.

Connection type: High Quality (LAN, direct connection)

Figure 4-21 Starting the KRCD connection

6. The Attention panel (Figure 4-22) appears on the server side. It advises of an incoming connection. In the example, remote control of the keyboard and mouse has been granted.



New Connection - Desktop Sharing

Attention

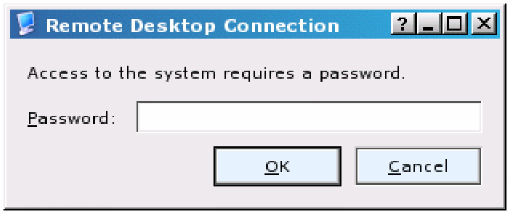
Somebody is requesting a RFB (VNC) connection to your computer. Granting this will allow them to see everything on your screen and, if the option below is enabled, be able to control it. Refuse the connection unless you trust the person.

Remote system: igan.local

☒ Allow remote user to control keyboard and mouse

Figure 4-22 KRFB connection advise

If you click **Accept Connection**, you are prompted for a password on the client side as shown in Figure 4-23.



Remote Desktop Connection

Access to the system requires a password.

Password:

Figure 4-23 KRDC password

Enter the password and click **OK**.

A VNC client window opens. In this case, it does so with the control of the mouse and keyboard from the administrator's console and from the remote workstation console.

4.5.5 TightVNC

TightVNC is another VNC server. You can download it from the Web in rpm format and install it on Red Hat 7.1 and later versions. Go to:

http://www.tightvnc.com/download.html

Source packages are also available so you can compile it for other Linux distributions.

TightVNC replaces other VNC software. To install it, you must first remove other VNC software packages:

```
rpm -e vnc
```

Then install tightvnc:

```
rpm -ivh tightvnc-xxx.rpm
rpm -ivh tightvnc-server-xxx.rpm
```

The commands to run TightVNC are the same as those to run VNC. **vncserver** starts a VNC server, and **vncviewer** starts a client connection.

4.5.6 IBM Desktop On Call

IBM Desktop On Call is a commercial product that is designed for remote administration of workstation and servers. There are versions for OS/2, Linux, and Windows, among others.

Desktop On Call has the same disadvantages as VNC and X in terms of network bandwidth consumption and screen resolution. It has a couple of advantages over them in terms of security. Security issues are integrated into Desktop On Call, and don’t rely on other layers to be configured separately. It is easier to configure a firewall to support Desktop On Call than it may be for VNC.

Desktop On Call comes as an RPM package for Red Hat or as a compressed tarball. You install the RPM version by running **rpm -ivh**. You install the tarball version as explained in the following step. This example is based on Version 5.0-1:

1. Create a temporary directory and decompress the tarball there:

   ```
   mkdir /opt/temp
   cp edtoc-5.0-1.tar.gz /opt/temp
   cd /opt/temp
   tar -zxvf edtoc-5.0-1.tar.gz
   mv edtoc /opt
   ```

2. The software is installed in the /opt/edtoc directory. Now, perform the configuration:

   a. Add the executables path to the PATH variable. You do this in the user or user's profile.

      - In Red Hat, you can set it in /etc/bashrc for all users.
      - In SuSE, you can add it to a new file called /etc/bash.bashrc.local.

      This file is executed by all users and is independent of the SuSE version. This means that no update of the system can overwrite it, so personal configurations are maintained.

      In this example, the variables that are added are:

      ```
      export PATH=/opt/edtoc/bin:$PATH
      export DTOC_PATH=/opt/edtoc
      ```

   b. Re-login as user *root* to load the new variables. Verify that they have the right values:

      ```
      echo $PATH
      echo $DTOC_PATH
      ```

   c. Create the Desktop On Call user password. Desktop On Call uses two different users: guest and other. In this example, we use user *vicente* and a password *pwdx*. Run the following commands to establish the passwords:

      ```
      $DTOC_PATH/bin/dtocpasswd vicente pwdx
      $DTOC_PATH/bin/dtocpasswd guest guest_pwd
      ```

   These passwords are not necessarily the users' system password. To change the password later, delete `$HOME/dtoc/.upasswd` and rerun the former command to re-create it.

3. Restart the X server in Linux. You can do this from user root by entering:

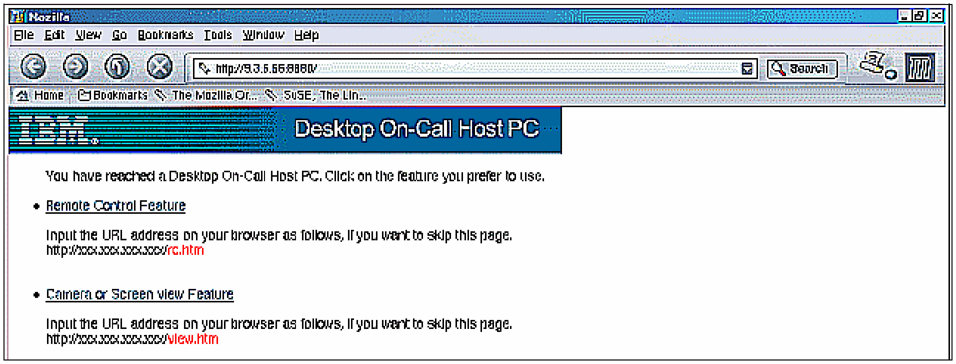
   ```
   init 3
   init 5
   ```

To manage Desktop On Call daemons, you can use the following commands located at $DTOC_PATH/bin:

- To start, run **dtoc**.
- To stop, run **dtocstop**.
- To check the status, run **dtocstat**.

The root user or Desktop On Call user whose password was created earlier can run these commands. It should be always the same user since **dtoc** creates the /tmp/.dtoc subdirectory, which is owned by the user that starts **dtoc**. If the user changes from *root* to another, errors appear, stating that the files in /tmp/.dtoc cannot be created or modified. In that case, delete `/tmp/.dtoc/*` and run **`dtoc`** again.

Desktop On Call is accessed from a Web navigator by using a URL such as `http://server_IP:8880`. On the page that opens (see Figure 4-24), click **Remote Control Feature**.



You have reached a Desktop On-Call Host PC. Click on the feature you prefer to use.

- Remote Control Feature

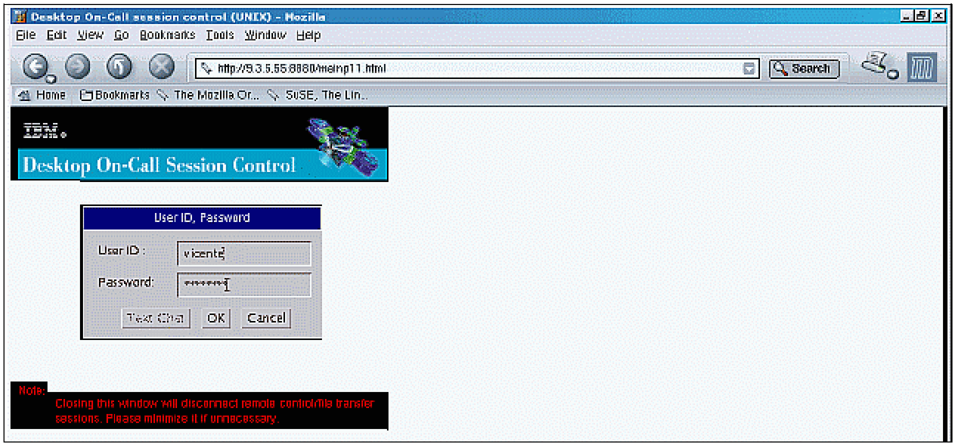
  Input the URL address on your browser as follows, If you want to skip this page.
  http://xxx.xxx.xxx.xxx/rc.htm

- Camera or Screen view Feature

  Input the URL address on your browser as follows, If you want to skip this page.
  http://xxx.xxx.xxx.xxx/view.htm

Figure 4-24 Desktop On Call client connection

On the login page (Figure 4-25) that opens, enter a user ID and password.

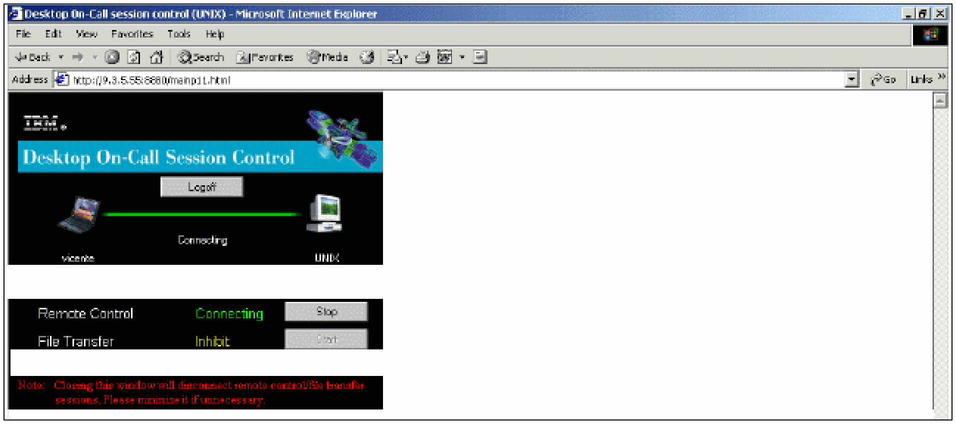

Desktop On-Call session control (UNIX) - Mozilla
File Edit View Go Bookmarks Tools Window Help
http://9.3.5.55:8880/mainp11.html
Search
Home
Bookmarks
The Mozilla Or...
SuSE, The Lin...
IBM
Desktop On-Call Session Control
User ID, Password
Text Chat
OK
Cancel

User ID : vicente
Password: ********

Note:
Closing this window will disconnect remote control/file transfer sessions. Please minimize it if unnecessary.

Figure 4-25 Desktop On Call client login

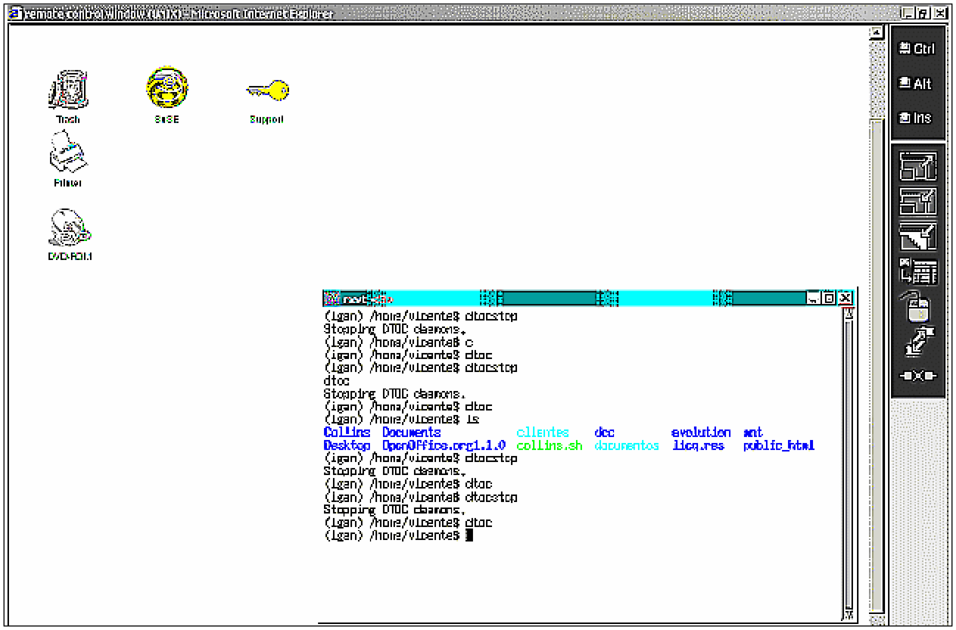
Now the connection in progress page is displayed for a few seconds. See Figure 4-26.


Desktop On-Call session control (UNIX) - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://9.3.5.55:8680/mainp1.html
Go Links
IBM
Desktop On-Call Session Control
Logoff
Connecting
vicente
UNIX
Remote Control
Connecting
Stop
File Transfer
Inhibit
Start

Note: Closing this window will disconnect remote control/file transfer sessions. Please minimize it if unnecessary.

Figure 4-26 Desktop On Call connection in process

Finally, the connection is made and you see the remote window (Figure 4-27).



```
(igan) /home/vicente$ dtocstop
Stopping DTOC daemons.
(igan) /home/vicente$ c
(igan) /home/vicente$ dtoc
(igan) /home/vicente$ dtocstop
dtoc
Stopping DTOC daemons.
(igan) /home/vicente$ dtoc
(igan) /home/vicente$ ls
Collins  Documents          clientes    doc         evolution  ant
Desktop  OpenOffice.org1.1.0  collins.sh  documentos  licq.res   public_html
(igan) /home/vicente$ dtocstop
Stopping DTOC daemons.
(igan) /home/vicente$ dtoc
(igan) /home/vicente$ dtocstop
Stopping DTOC daemons.
(igan) /home/vicente$ dtoc
(igan) /home/vicente$ 
```

Figure 4-27 Desktop On Call client connection complete

Now, the administrator and the user can both use their mouse and keyboard to work with the desktop. In case of contention, the administrator controls everything.

There are several icons on the right of the screen, shown separately in Figure 4-28. The first three icons are used as the Ctrl and Alt keys. For example, to type Ctrl-C on a remote terminal, click the **Ctrl** icon and then type `c`.

The next two icons increase or decrease the size of the remote desktop to the maximum of the navigator screen.

The two icons that follow change the color pattern of the screen, reducing it to 256 colors.

The mouse icon swaps the buttons of the local mouse. The next icon is for file transfer. It is not implemented in Linux since you can use FTP instead.

The last icon disconnects from the remote workstation.

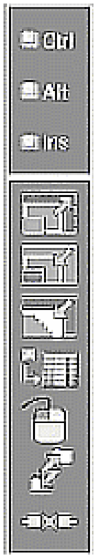


Figure 4-28 Desktop On Call icons

4.5.7 Citrix client

Citrix is a Linux (and other operating systems) graphical remote client to a Windows machine. You download Citrix client for Linux from the Web with the name ICA*Client-version_numbers*.rpm from:

http://www.citrix.com

You install Citrix client by running the following command on the directory /usr/lib/ICAClient:

```
rpm -ivh ICAClient-version_numbers.rpm
```

To launch Citrix client, enter:

```
/usr/lib/ICAClient/wfcmgr
```

When you launch Citrix client, it performs some configuration for the user and opens an administration window to create the necessary connections. This command is used later to reconfigure the Citrix client or to run connections. The configuration is stored in the $HOME/.ICAClient directory for each user configured.

Then, enter the following command to start the configured sessions:

```
/usr/lib/ICAClient/wfica
```

4.5.8 Remote login

Telnet is the normal text service used to access a UNIX server remotely. It has changed due to security concerns as user information goes through the network in plain text.

OpenSSH was created to avoid the security exposure of sending unencrypted user and password information over a network. Even if the network is private, for example, a corporate LAN, the risk of someone sniffing and obtaining password information is high enough to encourage the use of encrypted passwords.

OpenSSH substitutes the **telnet** command with **ssh** and substitutes the **ftp** command with **sftp**. These new commands encrypt user information before sending it to the servers. To start, stop, or show the status of the **ssh** daemon, called **sshd**, enter:

```
/etc/init.d/sshd start
/etc/init.d/sshd stop
/etc/init.d/sshd status
```

Other services are also replaced. Here are some simple examples of normal use, assuming a machine called *workstation* and a user called *user*:

- To log in remotely (Telnet replacement), type either of the following commands:

  ```
  ssh user@workstation
  ssh -l user workstation
  ```

- To copy a file (rcp replacement), enter:

  ```
  scp ./filename user@workstation:/destination_path
  ```

- To run a command (rsh replacement), enter:

  ```
  ssh user@workstation command
  ```

You are prompted for a password in each case.

4.5.9 Remote starting of graphical applications

If the administration workstation is a Linux machine or an X server is installed, the administrator can start applications, such as YaST2, on a remote workstation that is displayed on the administration workstation.

From an administration console, complete the following steps:

1. Log in the administration workstation in run level 5, that is, in graphical mode.
2. Open a graphical terminal (for instance, **konsole, rxvt, xterm**) and enter:

   ```
   xhost +
   ```

This enables the local display of X server graphics coming from a remote workstation.

3. Log in to the remote workstation (using **ssh** for example) and type:

```
export DISPLAY=administration_console_ip:0.0
```

The administrator can now issue commands that launch graphical applications, and a new window that displays that application appears on the administrator's workstation. The administrator can interact with the application as though it were running locally. However, depending on the application and the network speed, the application may perform slower than if the window were open on the system running the application.

4.6 Summary

This chapter provided a brief glimpse into managing remote Linux workstations and a look at some the tools and facilities available to manage such workstations. Many capabilities are available to securely perform remote administration and support. This is one of the values of using Linux as a client platform.

5

Coexistence considerations

Changing to a new client platform does not happen overnight. There is a need to support multiple client platforms in parallel. In fact, it is likely that this condition will exist to some extent indefinitely. Therefore, it is important to discuss how multiple client platforms, including OS/2, Linux and Windows can coexist and share resources.

This chapter discuss some of the key considerations related to coexistence such as file and printer sharing.

5.1 Samba

Samba is a multi-platform smb-over-ip file and printer sharing tool. To this extent, it is capable of replacing an OS/2 LAN server or Windows NT or 2000 server.

5.1.1 SWAT and xinetd configuration

Samba uses two daemons to provide its service and another to allow Web-based administration. It also uses the winbind daemon to authenticate Linux users in a Windows domain. These daemons are defined as follows:

- **smbd**: Provides file sharing and printing services using the Server Messaging Block (SMB) protocol.
- **nmbd**: This is the NetBIOS name server. It provides naming services to clients. It enables the capability to browse available resources using the Network Neighborhood facility in a Windows machine.
- **winbind**: This daemon allows a Samba client to authenticate users in a Windows NT domain. In Samba Version 3, it is also possible to authenticate through a Windows 2000 Directory Server.
- **SWAT**: The administration daemon is called Samba Web Administration Tool (SWAT) and is started via the xinetd daemon in Linux.

You can start Samba daemons from inetd (xinetd in later Linux versions) or as independent daemons.

The nmbd daemon is required to access the resources shared by Samba from an OS/2 or Windows-based client. For Linux clients, this daemon is not necessary.

You can also start Samba daemons from the init.d scripts by typing:

```
/etc/init.d/smb start
```

You can stop them or report their status by using a **`stop`** or **`status`** parameter.

The only way to start the SWAT administration daemon is from xinetd. After you configure and start SWAT, it provides an easy-to-use Web interface to administer the Samba server. Simply point a Web browser to:

```
http://hostIP:901
```

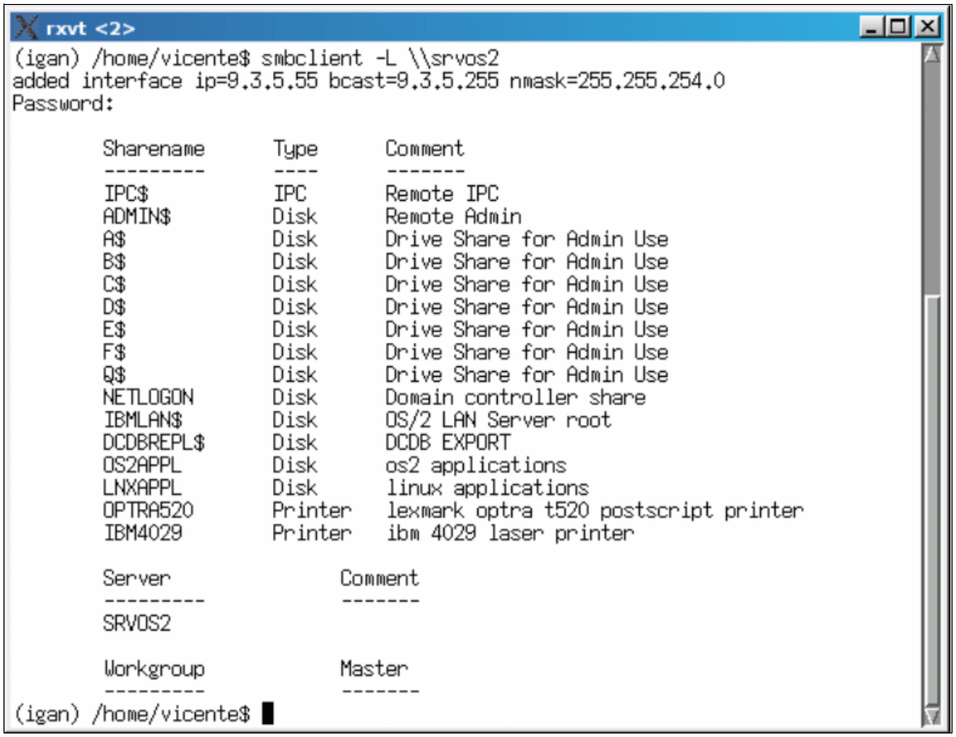
Here *hostIP* is the IP address of the Samba server.

5.1.2 Connecting from Linux to OS/2 LAN servers

In the case where OS/2 servers are still in place, but Linux clients appear, it may be necessary to access the OS/2 server from a Linux client. To connect, assume an OS/2 server called *srvos2*, a local Linux mount point */mnt/appl*, a share called *lnxappl*, and a user called *fred* with password *wilma*. With these variables in place, you enter the command:

```
smbmount //srvos2/lnxappl /mnt/appl -o username=fred,password=wilma
```

Figure 5-1 shows how you see the shared SMB resources in Linux.



```
(igan) /home/vicente$ smbclient -L \\srvos2
added interface ip=9.3.5.55 bcast=9.3.5.255 nmask=255.255.254.0
Password:

        Sharename      Type      Comment
        ---------      ----      -------
        IPC$           IPC       Remote IPC
        ADMIN$         Disk      Remote Admin
        A$             Disk      Drive Share for Admin Use
        B$             Disk      Drive Share for Admin Use
        C$             Disk      Drive Share for Admin Use
        D$             Disk      Drive Share for Admin Use
        E$             Disk      Drive Share for Admin Use
        F$             Disk      Drive Share for Admin Use
        Q$             Disk      Drive Share for Admin Use
        NETLOGON       Disk      Domain controller share
        IBMLAN$        Disk      OS/2 LAN Server root
        DCDBREPL$      Disk      DCDB EXPORT
        OS2APPL        Disk      os2 applications
        LNXAPPL        Disk      linux applications
        OPTRA520       Printer   lexmark optra t520 postscript printer
        IBM4029        Printer   ibm 4029 laser printer

        Server               Comment
        ---------            -------
        SRVOS2

        Workgroup            Master
        ---------            -------
(igan) /home/vicente$
```

Figure 5-1 OS/2 shared resources as seen from a Linux Samba client

5.1.3 Connecting from OS/2 to a Linux Samba server

In some cases, it is desirable to connect from an OS/2 client to a Linux-based Samba server. To achieve this, you must edit the following two files in OS/2:

- **c:\ibmcom\rfcnames.lst**

  The content of this file must be the name and the IP address of the Samba server, for example:

  ```
  "SambaServer" 9.3.5.55
  ```

- **c:\ibmcom\rfcbcst.lst**

  This is the broadcast list. This file is only necessary if the client and server are located in different subnets. The content of this file must be the name of the Samba server followed by its broadcast IP address. That address is obtained, as root user, by typing **ifconfig**, and looking in the output for a string such as:

  ```
  Bcast:9.3.5.255.
  ```

  The content of this file is:

  ```
  "SambaServer" 9.3.5.55
  ```

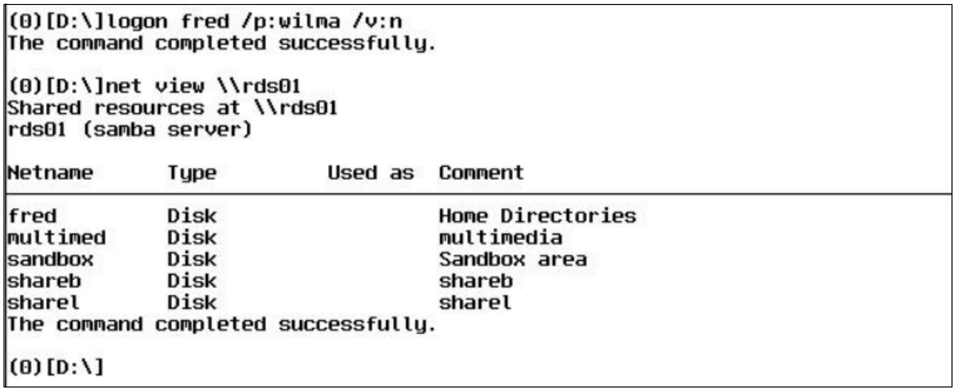
As shown in Figure 5-2, to log on from the OS/2 workstation, simply log on. For example, you enter:

```
logon fred /P:wilma /V:N
```

The N at the end means that you do not need to make any verification. If it was /V:D, you would log on into the domain. If it was /V:L, you would log on locally.

Now, you enter the **net view** command to show the resources on the network also as shown in Figure 5-2:

```
net view \\rds01
```



```
(0)[D:\]logon fred /p:wilma /v:n
The command completed successfully.

(0)[D:\]net view \\rds01
Shared resources at \\rds01
rds01 (samba server)

Netname      Type         Used as  Comment
-------------------------------------------------------
fred         Disk                  Home Directories
multimed     Disk                  multimedia
sandbox      Disk                  Sandbox area
shareb       Disk                  shareb
sharel       Disk                  sharel
The command completed successfully.

(0)[D:\]
```

Figure 5-2 OS/2 NetBIOS example

To access the shares, use the **net use** command to connect to the Samba shared file system as a local disk:

```
net use z: \\rds01\sandbox
```

5.1.4 Connecting from Linux to Linux Samba servers

If Linux-based Samba servers exist in a mixed client environment, you may want Linux clients to access the Linux Samba server. Although there are other ways to do this (such as using Network File System (NFS)), administrators may prefer Samba for performance reasons. Also it may be easier for them to manage only Samba and not both Samba and NFS.

To connect to a Samba server, a Linux workstation requires you to install a package—the Samba client package, usually called *samba-client*. The program that you need to run is **smbmount**.

The process is like a normal SMB connection, for example:

```
smbmount //SambaServer/share0 /mnt/share0 -o username=user-name
```

You prompted for the user's password in the Samba server. This password is the Samba password as established by the Samba administrator. It is not necessarily the Linux login password.

You can also connect using the **mount** command as shown here:

```
mount -t smbfs //SambaServer/share0 /mnt/share0 -o
username=usr-name,password=user-password
```

To avoid typing this command, you can add an entry at the end of the /etc/fstab file as shown here:

```
//SambaServer/share0 /mnt/share0 smbfs
noauto,username=usr-name,password=user-password 0 0
```

To connect, mount as a normal file system (from the command line or via an initialization script):

```
mount /mnt/share0
```

However, the file /etc/fstab is readable so anybody with access to the client system can read the password stored in it. It is better to use a credentials file. In the example, it can be /root/.smbpasswd with this content:

```
username=usr-name
password=user-password
```

The file must have read permission only for root, so run:

```
chmod 600 /root/.smbpasswd
```

Finally, modify the /etc/fstab file to as shown here:

```
//SambaServer/share0 /mnt/share0 smbfs credentials=/root/.smbpasswd 0 0
```

5.1.5 Connecting from Linux clients to Windows servers

The Samba client can use **smbmount** to mount exported file systems onto local directories. No configuration is needed, only installing samba-client rpm package as described previously.

To connect from a Linux client to a Windows server, assuming a share called share0, enter the command:

```
smbmount //WindowsServer/share0 /mnt/share0 -o username=user-name
```

If the users are not local, but rather Windows NT domain users, the command that you enter is slightly different (the domain name must be added):

```
smbmount //WindowsServer/share0 /mnt/share0 -o username=user-name/domain-name
```

It is also possible to connect using the **mount** command as shown in this example:

```
mount -t smbfs //WindowsServer/share0 /mnt/share0 -o
username=user-name,password=user-password
```

To avoid typing such a long command, you can add an entry at the end of the /etc/fstab file as shown here:

```
//WindowServer/share0 /mnt/share0 smbfs
noauto,username=usr-name,password=user-password 0 0
```

To connect, mount as a normal file system:

```
mount /mnt/share0
```

As in the previous section, the file /etc/fstab is readable so anyone can read the password stored in it. It is better to use a credentials file. In the example, it can be /root/.smbpasswd with this content:

```
username=domain\usr-name
password=user-password
```

Notice that the format and order of the user name and domain are different than the previous example.

The file must have read permission only for root, so you must enter:

```
chmod 600 /root/.smbpasswd
```

Finally, modify /etc/fstab to contain:

```
//WindowServer/share0 /mnt/share0 smbfs credentials=/root/.smbpasswd 0 0
```

5.1.6 Connecting from Windows workstations to Linux Samba servers

Connections from Windows clients are made as usual, but you must perform some customization in the server. Early versions of Samba were tricky to configure with password encryption enabled. Therefore, users usually modified the Windows registry to use only unencrypted passwords.

However, now it is possible and recommended to activate support for encrypted passwords on the Samba server and add user definitions to Linux and Samba. No configuration is needed on the Windows client.

You configure the server as explained here:

1. Edit /etc/samba/smb.conf and check the following entry:

   ```
   encrypt passwords=Yes
   ```

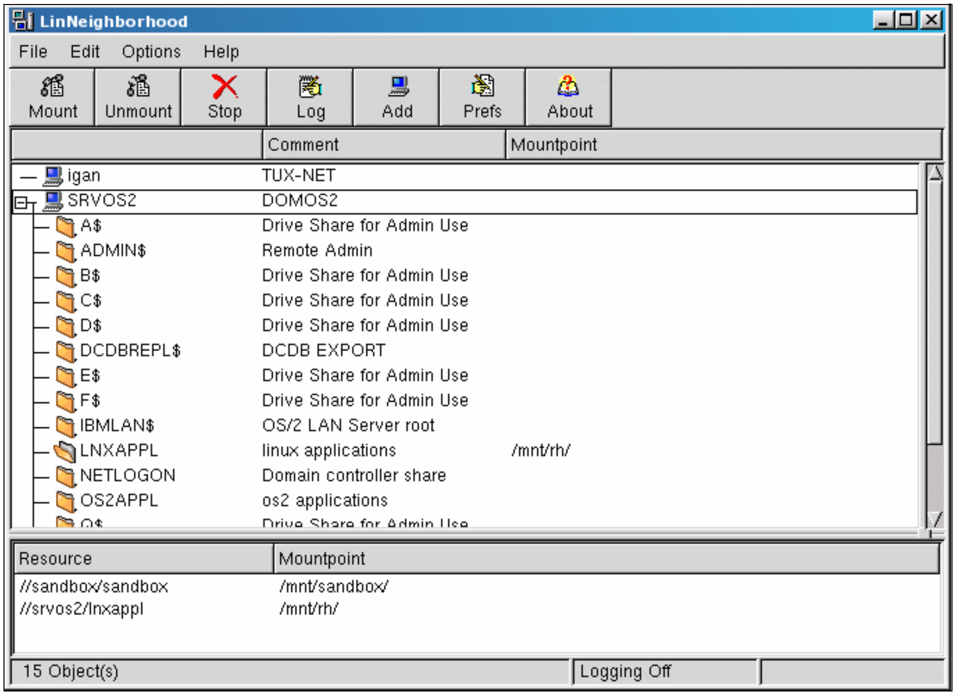
2. Create Samba users on Linux.
3. Define Samba Linux users as Samba users by running, for each user:

   ```
   smbpasswd -a user-name user-samba-password
   ```

If the access is made from Windows 95 or 98, the currently logged on user is the user ID and password used to mount Samba shared disks. In other versions of Windows, the user can specify a different user ID and password to access the Samba server.

5.1.7 LinNeighborhood

LinNeighborhood is an X Windows Server graphical application. It is similar to Windows Network Neighborhood that runs Samba client commands in the background and shows their results graphically as shown in Figure 5-3.



	Comment	Mountpoint
igan	TUX-NET	
SRVOS2	DOMOS2	
A$	Drive Share for Admin Use	
ADMIN$	Remote Admin	
B$	Drive Share for Admin Use	
C$	Drive Share for Admin Use	
D$	Drive Share for Admin Use	
DCDBREPL$	DCDB EXPORT	
E$	Drive Share for Admin Use	
F$	Drive Share for Admin Use	
IBMLAN$	OS/2 LAN Server root	
LNXAPPL	linux applications	/mnt/rh/
NETLOGON	Domain controller share	
OS2APPL	os2 applications	
Q$	Drive Share for Admin Use	

Resource	Mountpoint
//sandbox/sandbox	/mnt/sandbox/
//srvos2/lnxappl	/mnt/rh/

15 Object(s) Logging Off

Figure 5-3 LinNeighborhood main window

When you select Options -> Browse entire network from the menu bar, the network is scanned and new machines appear.

5.2 NFS

In cases where Samba is not desired, files can be shared through Network File System. NFS is the standard and most mature file sharing protocol, and now is in Version 4. It was initially developed by Sun Microsystems. It is standard, well understood, and supported robustly on different platforms. It uses the traditional UNIX permissions schema, but over a network.

5.2.1 NFS Version 4 improvements

The NFS Version 4 protocol supports file locking, stronger security, compound operations, client caching, and internationalization. It has been designed to work well on the Internet.

The goals of NFS Version 4 include:

- Better Internet access and performance
- The ability for clients and servers to negotiate security
- Common features for cross-platform interoperability
- Acceptance of standard protocol extensions

For more information about rfc3530, see:

http://www.ietf.org/rfc/rfc3530.txt

5.2.2 Working with NFS

This section describes a simple NFS configuration. NFS is supported by various NFS servers, which is included in the distributions and installed as a module with the kernel. You must enable NFS support (as it is by default) in the Linux kernel. To manage NFS, you must install the **nfs-utils** package.

To start, stop, and check NFS in SuSE, you run:

```
/etc/init.d/nfsserver start
/etc/init.d/nfsserver stop
/etc/init.d/nfsserver status
```

In Red Hat, the commands are similar:

```
/etc/init.d/nfsserver start
/etc/init.d/nfsserver stop
/etc/init.d/nfsserver status
```

You can configure NFS through the graphical tools, **redhat-config-nfs** or **yast2 nfs**, or you can configure it manually. We explain the manual configuration. When you understand it, graphical configuration is trivial. The rest of NFS is equivalent in both Red Hat and SuSE distributions.

Manual configuration involves these steps:

1. Export a file system or directory on the server side. To do this, edit the /etc/exports file and add a line for each exported directory. For example, to export /mnt/homes/fred, add this line:

   ```
   /mnt/homes/fred/ *(ro,root_squash,sync)
   ```

2. Start the NFS server:

   ```
   /etc/init.d/nfsserver start
   ```

Now you can mount the server from an NFS client by entering this command:

```
mount -t nfs server-ip:/mnt/homes/fred /home/fred
```

This way the home directory of the user *fred* is centralized in a remote NFS server.

To mount automatically each time the client is started, add a line in the /etc/fstab as in this example:

```
server-ip:/mnt/homes/fred /home/fred nfs defaults 0 0
```

To test it, enter:

```
mount -a
```

If you cannot mount the server, due to security issues not supported by the NFS client software or version, export, it as we did previously, in the /etc/exports file, with the option *insecure*:

```
/mnt/homes/fred/ *(ro,root_squash,sync,insecure)
```

Now user fred can log on to the client machine. He see his files but cannot modify them because the ID on the client, the owner of the exported directory, and the user who wants it mounted are different. To overcome this, you can:

- Give all permissions to all of the files before you export them. This allows all users to modify this directory. This is typically not a good solution.
- Export the directory with the rw option:

  ```
  /mnt/homes/fred/ *(rw,root_squash,sync)
  ```

If the client user is *root*, this doesn't work. It only works if the write permission for all the users are granted in the exported directory. Allowing this is dangerous, but you can do so using the *no_root_squash* option in the /etc/exports file:

```
/mnt/homes/fred/ *(rw,no_root_squash,sync)
```

5.3 FTP

There are many File Transfer Protocol (FTP) servers and FTP clients. FTP servers share files over a network. The files are stored in a file system and are accessed by FTP clients with a user and a password. After they are authenticated, the user can download files from the server and upload files to the server. They can do this only if the user has the necessary permissions on the accessed directory and over the accessed files.

FTP servers have evolved their security from simple access controls to more complex ones. They have also evolved from plain user and password information travelling through the network to encrypted user information.

OpenSSH was created to avoid the security exposure of sending unencrypted user and password information over a network. Even if the network is a private one, that is a corporate LAN, the risk of somebody sniffing and obtaining password information is high enough to encourage the use of encrypted passwords. OpenSSH substitutes the **telnet** command with **ssh** and the **FTP** command with **sFTP**.

Either way, the FTP protocol is well known and supported across a wide range of platforms. Interoperability between platforms is typically not an issue.

5.4 Printing

Another key aspect to coexistence is the sharing of printers. To learn about setting up network printer definitions for Linux clients, see 2.4, “Printing” on page 43.

5.5 Summary

This chapter covered some of the coexistence considerations for environments that will have a mixture of client types. This may be especially important during a phased transition. Using such facilities as Samba and NFS, you can create file servers, and users of different client types can still gain secure access to files.

6

Migration considerations

This redbook discusses the capabilities of Linux that make it a viable candidate as a client platform to replace OS/2 clients. It is not intended to be a migration guide. However, there are a few considerations that are worth examining in relationship to a possible migration, as discussed in this chapter.

6.1 Domain logons

The concept of a domain logon as it is known within OS/2 does not exist in the Linux Samba world. Samba provides the ability to have a primary domain controller and member servers, but no backup domain controllers. Linux clients do not currently provide a method of authenticating using standard Server Messaging Block (SMB) protocols as you see with OS/2.

To authenticate with a server at logon, you must use Lightweight Directory Access Protocol (LDAP). The examples in this chapter use:

- IBM Tivoli Directory Server V5.2

  http://www.ibm.com/software/tivoli/products/directory-server/

- Samba 3

  http://www.samba.org

6.1.1 Authenticating with LDAP

For a client to authenticate with LDAP, you change its client settings. This is normally done during the installation. You add the following line, for example, to add to the “ks.cfg” response file as described in Chapter 7, “Linux client installation” on page 145:

```
#System authorization information
auth  --useshadow --enableldap --enableldapauth --ldapserver 10.2.2.40
--ldapbasedn "o=redbook test,c=us" --enablecache
```

Within a user logon profile, you can run a batch program to perform user-specific actions. In our case, we mount a home directory for the user.

6.1.2 Automounting a share at console logon

To allow the automounting of shares at logon, we use a pluggable access module (PAM) module called *pam_mount.so*. We use Version 0.9.6-1.

To set up this module, log in as `root`. Then issue the following commands from the directory where you saved the file. Depending on the environment, you may need to install some dependencies. If so, when the following command fails, it suggests packages that are required but not currently installed:

```
rpm -ivh pam_mount-0.9.6-1.i386.rpm
```

After you install this version, you may need to edit some files. The locations shown in our examples may be specific to Red Hat.

In the /etc/security directory, you see the pam_mount.conf file. You can add automount shares to this file as shown in the following example:

```
volume * smb cidrds data /home/&/data     uid=&,gid=&,dmask=0750 - -
```

In this example, we mount share “data” from SMB server cidrds to the mount point /home/liz/data/. The asterisk (*) that follows the volume states that this mount is for all users who authenticate with the system. The ampersand character (&) represents the user ID of the user who is authenticating on the system.

After you make these changes, you need to edit the /etc/pam.d/login file. The default file looks like this:

```
#%PAM-1.0
auth       required   pam_securetty.so
auth       required   pam_stack.so service=system-auth
auth       required   pam_nologin.so
account    required   pam_stack.so service=system-auth
password   required   pam_stack.so service=system-auth
session    required   pam_stack.so service=system-auth
session    optional   pam_console.so
```

Add the following two lines to the bottom of the file:

```
auth       optional   pam_mount.so use_first_pass
session    optional   pam_mount.so
```

The switch on the first line, `use_first_pass`, allows the mount to happen using the password that was used to authenticate with the local system. If the passwords are different, then this is handled by reading the details included in the pam_mount.conf file.

6.1.3 Automounting for roaming users

This section describes the client login environment we set up in our test environment. Using Linux, we want to emulate the following abilities of an OS/2 client environment:

- Logon shares
- Locked down desktop
- Dynamically add or remove icons and alter user’s desktops

We achieved this by using:

- Red Hat Workstation 3 (any distribution will work)
- An LDAP authentication server
- A preconfigured KDE desktop

- The pam_mount module
- A Samba 3.0 server

Client configuration

Since we are authenticating with the LDAP server, we do not have any locally defined users on the system. We installed the pam_mount module as discussed in the previous section by adding the statements to the login file as shown. We added the following two lines to pam_mount:

```
volume * smb cidrds ldesktop /home/&/Desktop uid=&,gid=&,dmask=0770 - -
volume * smb cidrds taskbar /usr/share/config uid=&,gid=&,dmask=0770 - -
```

The first mount is to the Samba share “ldesktop”. It creates the users /home/*any user*/Desktop directory. This means that the user’s desktop icons exist on the server and allow us to add or remove icons for an individual user or group of users.

The second mount, which is to the Samba share “taskbar”, mounts at the /usr/share/config directory. This directory contains the configuration files for the KDE kicker task bar, which we preconfigured for our users.

Server configuration

Our distribution server (discussed in Chapter 7, “Linux client installation” on page 145) is also configured as an LDAP server and a Samba server. Example 6-1 shows our slapd.conf file, which is the main LDAP configuration file. We only show the bottom of the file since the rest is default and comments. The complete details to set up an LDAP server are outside the scope of this redbook.

Example 6-1 The slapd.conf file

```
#######################################################################
# ldbm database definitions
#######################################################################

database        ldbm
directory       /var/lib/ldap

suffix          "dc=linuxdomain,dc=net"
rootdn          "cn=root,dc=linuxdomain,dc=net"
rootpw          password

# Indices to maintain
index   objectClass,uid,uidNumber,gidNumber,memberUid   eq
index   cn,mail,surname,givenname                       eq,subinitial
```

Example 6-2 shows user “fred” who we exported to demonstrate the settings used in our user configuration.

Example 6-2 Configuration settings for user “fred”

```
dn: cn=fred flinstone, ou=users, dc=linuxdomain,dc=net
shadowMin: 0
sn: flinstone
loginShell: /bin/bash
userPassword:: e0NSWVBUfTMzaUMyUkVkUnN4NnM=
uidNumber: 104
gidNumber: 100
shadowFlag: 0
shadowExpire: -1
shadowMax: 999999
uid: fred
objectClass: top
objectClass: person
objectClass: posixAccount
objectClass: shadowAccount
gecos: fred flinstone
shadowLastChange: 10877
cn: fred flinstone
shadowInactive: -1
homeDirectory: /home/fred
shadowWarning: 7
```

Example 6-3 shows the settings we used in the smb.conf file for our two shares.

Example 6-3 smb.com file settings

```
[ldesktop]
        comment = "locked desktop"
        path = /hda7/ldesktop
        read only = no
        force user = root
        force group = ldesktop
        directory mask = 0775
        create mask = 0755

[taskbar]
        comment = "taskbar"
        path = /hda7/taskbar
        read only = yes
        force user = root
        force group = ldesktop
        directory mask = 0775
        create mask = 0755
```

Within Samba, create users to have the same user ID and password as within the LDAP server.

By configuring this environment, the administrator can now dynamically add or remove icons from the user's desktop, create new folders for them, or remove access to them.

6.1.4 GNOME roaming

You can achieve the same affect for the GNOME desktop. However, you must make the following changes to the mount commands:

```
volume * smb cidrds ldesktop /home/&/.gnome-desktop uid=&,gid=&,dmask=0770 - -
```

Create a new share for the GNOME gconf database and populate it with preconfigured environment data as discussed in 2.3, "GNOME desktop" on page 36. The following example shows where to mount the newly created share "gconf" on the local system:

```
volume * smb cidrds gconf /etc/gconf uid=&,gid=&,dmask=0770 - -
```

6.2 Extended attributes support and Samba

Not all Linux file systems support OS/2 extended attributes. If an OS/2 client is connected to a Samba server, for instance to back up client files before migrating a workstation to Linux, failures may occur when trying to copy data to the Samba server unless the partition is a journaled file system (JFS) partition. If the extended attributes cannot be written to the target file system, then strip the extended attributes from the files by using **eautil**, which is shipped with OS/2, for example:

```
eautil afilewitheas.cmd /s
```

This creates a the EAS directory in the current directory where the stripped extended attributes are placed in case they need to join with the files again in the future.

6.3 Migrating OS/2 data to Linux

The Linux kernel has support built into it for the traditional OS/2 file systems. This allows the ability to mount OS/2 drives within Linux to migrate any required data that the user has stored locally on their system. The following sections discuss each of the file systems—FAT, JFS, and HPFS—in turn.

6.3.1 Migrating data located on OS/2 FAT formatted systems

The FAT file system, as one of the first file systems for PCs, has supported mounting into the Linux file-system structure for years. Linux has the ability to read and write to the drive. The following command shows an example of mounting a FAT partition to Linux, where *hdb1* is the second hard disk in the machine containing the user's data:

```
mkdir /mnt/os2fatdata
mount -t vfat /dev/hdb1 /mnt/os2fatdata
```

6.3.2 Migrating data located on OS/2 JFS formatted systems

The title of this section is somewhat misleading. OS/2 JFS is fully compatible with Linux JFS. If an OS/2 workstation has a JFS data partition, and the workstation is reinstalled with Linux leaving the JFS data partition intact, you can mount the JFS partition into Linux and access it fully to read and write to the partition.

The following command shows an example of mounting an OS/2 JFS data partition to Linux, where *hdb2* is the second hard disk in the machine and the JFS file system is on the second partition on the disk:

```
mkdir /mnt/os2jfsdata
mount -t jfs /dev/hdb2 /mnt/os2jfsdata
```

6.3.3 Migrating data located on OS/2 HPFS formatted systems

Recent Linux kernels, by default, no longer contain the required code that allows the mounting of HPFS drives as read-only to allow for the copying of data to a different file system. Information about how to achieve this is dependant on the kernel version that is being used. Review the HPFS documentation found in the kernel source for more information. In our kernel, this is located in /usr/src/linux-2.4.21-4.EL/Documentation/filesystems/hpfs.txt.

This file details the usage of HPFS within the current kernel and how to add support.

6.3.4 Migrating data located on OS/2 HPFS386 formatted systems

A few clients run HPFS386 for improved performance, but HPFS386 is not supported on Linux. If it is required to mount an HPFS386 drive to migrate the data, we recommend that you strip the drive of its HPFS386 features using **`prepacl`**, for example:

```
prepacl d:\ /P /N
```

This command specifies to remove the ACLs and not to save them. After you perform this action, you can mount the drive as an HPFS drive as described previously.

6.4 Summary

This chapter gave a brief survey of a few items that may relate to a migration effort between OS/2 and Linux. It provided some helpful information of which an administrator should be aware when planning for a transition to Linux.

7

Linux client installation

A factor associated with the adoption of a new client operating system across an enterprise is how easily you can deploy and maintain it. This chapter describes one method to set up a Linux distribution server. This distribution server enables the installation of targets in an unattended state based on response files. Many of the conventions that are described in this chapter are based on OS/2's Configuration, Installation, and Distribution (CID) procedures. We designed this method of installation to be similar to current OS/2 CID procedures. This makes the transition easier for enterprises that currently deploy OS/2.

The method and examples shown in this chapter assume a Red Hat server as the distribution server. A similar mechanism can be created for SuSE or other distributions. The server platform is independent of the client platforms to be installed. Our example is based on Red Hat 9.0, but can be adapted for other versions.

To perform many of the actions described in this book, we use IBM Object REXX for Linux. REXX is a batch programing language and is used heavily in OS/2. REXX is available for z/OS®, OS/2, PC-DOS, Windows, and now Linux.

For information about how to obtain Object REXX for Linux from IBM, see:

http://www.ibm.com/software/awdtools/obj-rexx/linux/

Other implementations of REXX for Linux are also available.

Using the installation method that we describe allows the installation of Linux systems as either servers or clients. These procedures were tested with both Red Hat and SuSE distributions. The focus of this redbook is on client systems, which is where the majority of the emphasis is. However, you can also use the same procedures to install Linux server systems.

The distribution server uses File Transfer Protocol (FTP), Hypertext Transfer Protocol (HTTP), and optionally Dynamic Host Configuration Protocol (DHCP). If you are performing unattended installations with a network boot capability, such as Preboot Execution Environment (PXE), a trivial FTP (TFTP) server is also required.

7.1 Scenario

The scenarios that we discuss in this chapter are based on working with three machines: two client targets and one server. The server automates the deployment of client systems. The following scenario uses Red Hat V9.0 for both the server and sample client installations. The version and distribution used for the server are completely independent of the client images to be installed.

The first step in the process is to set up a deployment server. This process is described in the following section.

7.2 Creating a Rapid Deployment Server

This section describes the method that we used to set up and configure what we call a Rapid Deployment Server (RDS). You can perform several of these steps using alternate methods.

The RDS server should have a minimum of 128 MB of memory, a network adapter card, and a hard disk or array that allows 20 GB for storage. The installation process that we describe destroys any data that is currently on the server's disk drive. You should connect the system to the network during its setup and installation.

7.2.1 Partitioning the disk on the RDS

Although you can partition the disk space of a system when installing a Linux environment, we partition the disk in a separate step using a bootable DOS diskette. Based on our OS/2 experiences, we divide the first physical disk in four parts. The first part is a primary partition (hda1). The remaining part is an extended partition (hda2) that is extended and divided into three logical partitions (LPAR): hda5, hda6, and hda7.

The first partition hda1 is our maintenance partition. In this partition, we place a small recovery image. This allows us to check the system and possibly reinstall it without needing diskettes and CDs. hda5 is our “root” partition, hda6 is the “swap”, and hda7 is the data partition.

On data partition (hda7), we create symbolic links from the /home, /var, and other directories. This prevents us from losing installation images and other critical data in case we need to reinstall the root partition for any reason.

The sizes of these partitions depends on the total size of the hard disk. As a standard rule, we create hda1 (our maintenance partition) with a size of 24MB, hda5 (our root) with a size of 2048 MB, and the swap partition with a size of 512 MB. The rest is our data partition (hda7).

The right side in Figure 7-1 represents the target disk partitioning for our deployment server. The left side shows a typical partitioning for an OS/2 CID server for comparison.

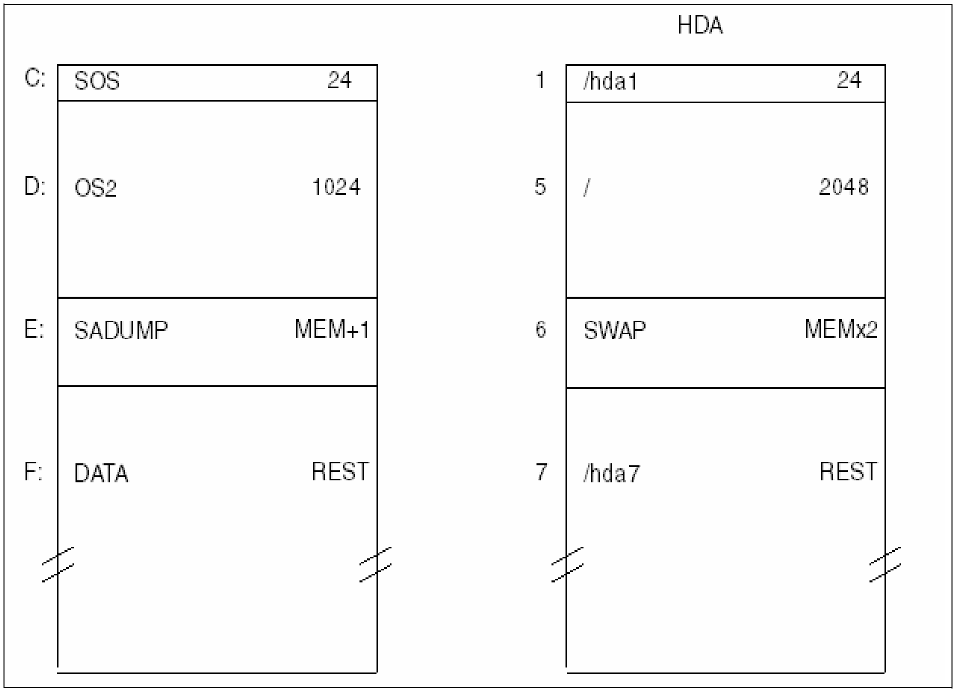


Figure 7-1 RDS disk partitioning

7.2.2 Creating an advanced DOS boot diskette

We require a DOS bootable diskette for some later steps. Use your own licensed copy of DOS from an existing system or download a copy of freedos from:

http://www.freedos.org

Copy the following files into the diskette:

- **aefdisk.exe**: You can download this file from:

  http://www.aefdisk.com

- **xfdisk.exe**: You can download this file from:

  http://www.mecronome.de/xfdisk

Create a file named PREP.BAT on the diskette as shown in Example 7-1.

Example 7-1 PREP.BAT file

```
@echo off
if .%1 == . goto syntax
if .%2 == . goto syntax
if .%3 == . goto syntax
aefdisk 1 /mbr
aefdisk 1 /delall
aefdisk 1 /pri:%1:83 /pri:%2:83 /pri:%3:82
aefdisk 1 /freesize
aefdisk 1 /delall
aefdisk 1 /pri:%1:83 /ext:0 /log:%2:83 /log:%3:82 /log:%freesize%:83
aefdisk 1 /show

goto end

:syntax
echo ! error: invalid options
echo.
echo . usage: prep size1 size2 size3
echo.
echo . sample: prep 24 1024 256

:end
```

This batch file performs the partitioning that we described earlier. You can modify it to meet any specific requirements.

Note: This disk that you created is used again in 7.2.2, “Creating an advanced DOS boot diskette” on page 148, and 7.3.3, “Installing a target” on page 166. Do *not* discard it.

7.2.3 Creating a CID-enabled Linux boot diskette with response file

We now create a Linux boot diskette and add the ks.cfg (KickStart) response file to the diskette. In a Red Hat environment, you can use a KickStart configuration file to provide responses to installation options. This allows for a fully automated installation.

If this Linux Server installation is the first Linux system in the environment, then you need to create the boot diskette on an existing OS/2 (or Windows) system. If a Linux system already exists, then skip the following OS/2 instructions and go straight to “Creating the boot disk from Linux”.

Creating the boot disk from OS/2

To create the book disk from OS/2, follow these steps:

1. Confirm that the loaddskf.exe program is installed on the OS/2 system. If it is not installed, it is available on any OS/2 CD-ROM. Copy it to a directory referenced by the OS/2 PATH environment variable.
2. Insert the Red Hat 9.0 CD 1 into the OS/2 system CD-ROM drive.
3. Insert a formatted diskette into the diskette drive.
4. Enter the following command, where A: is the diskette drive and Q: is the CD-ROM drive:

   ```
   loaddskf q:\images\bootdisk.img a:
   ```
5. Continue with the section “Editing the diskette” on page 150.

Creating the boot disk from Linux

To create the book disk from Linux, follow these steps:

1. Insert the Red Hat 9.0 CD-ROM 1 into the system CD drive.
2. Mount the CD to the file system:

   ```
   mount /dev/cdrom /mnt/cdrom
   ```
3. Insert a blank diskette and enter the following command:

   ```
   dd if=/mnt/cdrom/images/bootdisk.img of=/dev/fd0
   ```
4. Unmount the CD-ROM by entering the following command:

   ```
   umount /mnt/cdrom
   ```

Editing the diskette

Copy the ks.cfg file shown in Example 7-2 to the diskette.

Example 7-2 Kickstart ks.cfg file for installing Linux RDS server

```
# kickstart file for server "rds040"

authconfig --enableshadow --enablemd5
bootloader --location "mbr" --useLilo
firewall --disabled
install
keyboard "us"
lang "en_US"
langsupport --default "en_US.UTF-8"
mouse "generic3ps/2" --device "psaux"
network --device "eth0" --hostname "rds040" --ip "10.2.1.40" --bootproto //
"static" --netmask "255.255.0.0" --gateway "10.2.1.1"
rootpw "password"
reboot
text
timezone "Etc/GMT+1"

# sourcepath
cdrom

# desktop
skipx

# diskprep
bootloader --location "mbr" --useLilo
part "/hda1" --fstype "ext3" --size="1" --grow --onpart "hda1"
part "/" --fstype ext3 --size="1" --grow --onpart "hda5"
part "swap" --size="1" --grow --onpart "hda6"
part "/hda7" --fstype "ext3" --size="1" --grow --onpart "hda7"

%packages --resolvedeps
@ X Window System
@ GNOME Desktop Environment
caching-nameserver
bind
dhcp
expect
ftp
httpd
iptraf
lftp
mc
```

```
mozilla
pdksh
openssh-server
openldap
openldap-clients
openldap-servers
php
samba
samba-client
samba-common
samba-swat
tcpdump
tftp-server
unzip
vnc-server
vsftpd
xinetd

-Canna
-aspell-ca
-aspell-da
-aspell-de
-aspell-en-ca
-aspell-en-gb
-aspell-es
-aspell-fr
-aspell-it
-aspell-nl
-aspell-no
-aspell-pt
-aspell-pt_BR
-aspell-sv
-fonts-ja
-fonts-KOI8-R-100dpi
-fonts-KOI8-R
-man-pages-cs
-man-pages-da
-man-pages-de
-man-pages-es
-man-pages-fr
-man-pages-it
-man-pages-ja
-man-pages-ko
-man-pages-pl
-man-pages-ru
-openoffice
-openoffice-libs
-sendmail
-ttfonts-ja
```

```
-ttfonts-ko
-ttfonts-zh_CN
-ttfonts-zh_TW

%pre

%post
```

The diskette that automates the installation of the Linux RDS server is now complete.

7.2.4 Installing the server

To install the server, follow these steps:

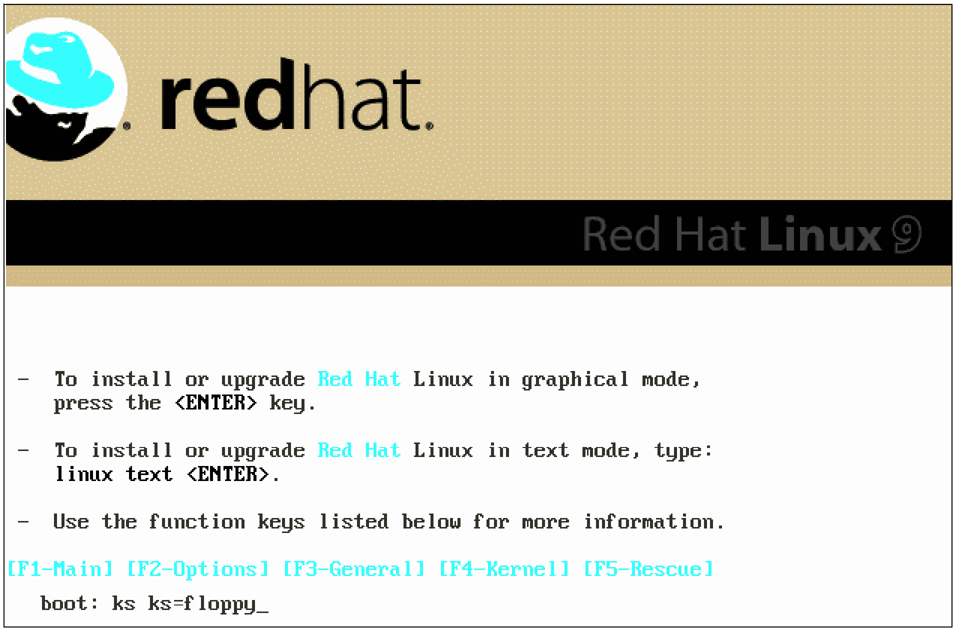
1. Boot the server with the Advanced DOS boot diskette and enter the following command:

   ```
   prep 24 2048 512
   ```

   > **Important:** This batch file does not ask for confirmation. Be aware that it *destroys* any existing partitions on the hard disk.

2. When the partitioning is finished, remove the diskette and place it in a safe location since we use it later.
3. Insert the bootable Linux diskette that we created in the previous section.
4. Boot the new Rapid Deployment Server.
5. After a successful boot, you see the screen shown in Figure 7-2. As shown in Figure 7-2, type:

   ```
   ks ks=floppy
   ```



```
 -  To install or upgrade Red Hat Linux in graphical mode,
    press the <ENTER> key.

 -  To install or upgrade Red Hat Linux in text mode, type:
    linux text <ENTER>.

 -  Use the function keys listed below for more information.

[F1-Main] [F2-Options] [F3-General] [F4-Kernel] [F5-Rescue]
  boot: ks ks=floppy_
```

Figure 7-2 Initial Red Hat installation screen

6. The Red Hat 9.0 installation now starts using the Anaconda installation program. As required, it prompts the user for the second and third CD-ROMs. Insert the CDs as prompted.
7. At the end of the installation, remove the diskette and CD and press the Enter key to reboot the server.
8. After the server is rebooted, log on using the user ID `root` and the password `password` (the default password defined in the ks.cfg file).
9. Check the network settings by entering the following command from a terminal shell:

   ```
   ip a
   ```

   The output should appear similar to this example:

   ```
   1: lo: <LOOPBACK,UP> mtu 16436 qdisc noqueue
       link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
       inet 127.0.0.1/8 brd 127.255.255.255 scope host lo
   2: eth0: <BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast qlen 100
       link/ether 00:09:6b:e9:d4:dd brd ff:ff:ff:ff:ff:ff
       inet 10.2.1.40/16 brd 10.2.255 scope global eth0
   ```

10. Enter the following command:

    ```
    ip r
    ```

The output should appear similar to the following example:

```
10.2.0.0/16 dev eth0  scope link
169.254.0.0/16 dev eth0  scope link
127.0.0.0/8 dev lo  scope link
default via 10.2.1.1 dev eth0
```

The IP output is based on the address range that we use as defined in the ks.cfg file.

The Red Hat Linux 9 server is now installed. The following sections show additional steps that are required to prepare the server for use.

7.2.5 Creating the CID directory structure

This section explains how to create the CID directory structure on the new RDS server. You enter the following commands from a terminal shell:

```
mkdir -p /hda7/sharel/bin/
mkdir -p /hda7/sharel/csd/
mkdir -p /hda7/sharel/img/
mkdir -p /hda7/sharel/pxe/
mkdir -p /hda7/sharel/mir/
mkdir -p /hda7/sharel/rsp/

mkdir -p /hda7/shareb/lnx/
mkdir -p /hda7/shareb/log/
```

The directory structure that we used is based on a well-known directory structure used in OS/2 products, such as CID and NVDM/2. The two main directories in our structure are *sharel* and *shareb*. All the other directories are located underneath these. We describe each of these in the following sections.

/sharel directory

The /sharel directory is set as a read-only directory for our targets during their distribution. This directory contains images and executables that are required for the installation. As you see later, we access this directory by using the FTP server as a symbolic link to /var/ftp.

The subdirectories of /sharel are:

- **/sharel/bin** contains the executable files for various REXX and bash procedures.
- **/sharel/csd** contains fixpacks and updates to be applied to the target systems.
- **/sharel/img** contains installable images. For example, RPMs and the relevant Linux distribution.

- **/sharel/mir** contains the files and subdirectories to be copied to the target. You can use it for programs that do not have an installation program of their own or if pre-configured files or data needs to be copied to the target.

  For example, by creating the extra subdirectories, as shown in the following example, and creating the file kickerrc, this file is automatically copied into the /usr/share/config directory on the target machine:

  ```
  /sharel/mir/usr/share/config/kickerrc
  ```

- **/sharel/pxe** is our TFTP tree. It is for the TFTP server that we use in combination with our network boot installation using PXE. The pxe directory is actually a symbolic link to the /tftpboot directory.
- **/sharel/rsp** contains all common response files that are used to install individual modules and applications.

/shareb directory

The /shareb directory is set with read and write permissions for only one user. In our environment, we create a user called *cidusr*. We access the relevant subdirectory structure by a specific targeted machine.

You can establish the following directory to support the installation of client systems. The lnx directory is used for Linux installations. You can use this same server to install other platforms such as OS/2 and Windows (not described in this redbook).

```
/shareb/lnx
/shareb/pws (used by OS/2 targets)
/shareb/win (used by windows targets)
```

The subdirectories within the /shareb/lnx directory are based on the names of the clients that are to be installed. You can set these names to anything, but our naming convention is *b01c183*, which signifies branch 01 client 183.

In the root of each target directory, for example /shareb/lnx/b01c183, we place the following files and directories:

- /shareb/lnx/b01c183/ks.cfg

  ks.cfg is the kickstart response file to install the Red Hat distribution. It contains the list of Red Hat packages to be installed or excluded.

  cidagent.ini is a file that explains to the CID agent what to do after the standard installation of the base products.

- /shareb/lnx/b01c193/matrix.xml

  matrix.xml is the response file for installing the SuSE distribution. It contains the list of SuSE packages to install or exclude.

- /shareb/lnx/b01c183/log

  The *log directory* contains the log files for each installation step. You can use the contents of this directory to remotely check completion of an installation or to help debug errors.

- /shareb/lnx/b01c183/cfg

  The *cfg directory* contains optional configuration files copied during the CID installation from the server to the target. For example, to have a custom resolv.conf file in the target's /etc directory, create the directory /file in the ./cfg directory:

  ```
  /etc/resolv.conf
  ```

7.2.6 Copying the RDS tools to the directory structure

This section explains how to copy the RDS tools that we developed to the server. These tools are available on an *as is* basis. To download these files, see the instructions in Appendix B, "Additional material" on page 187. The tools and sample files are provided in a ZIP file.

1. Expand this file into a temporary directory for example:

   ```
   cd /tmp
   unzip ./rdstools.zip
   ```

2. Copy the contents of the expanded file as shown here:

   ```
   cp /tmp/sharel/bin/* /hda7/sharel/bin
   cp /tmp/sharel/csd /hda7/sharel/csd
   cp /tmp/sharel/rsp /hda7/sharel/rsp
   cp /tmp/sharel/pxe /hda7/sharel/pxe
   ```

What we copied

Example 7-3 shows our desired directory tree. Much of this was created by the copy commands that we described previously. This tree assumes that other packages, such as Object REXX, and other tools are installed. You can find the descriptions of the following REXX files in 7.5, "REXX procedures used by RDS" on page 173.

Example 7-3 Directory tree

```
/hda7/sharel
    |-- bin
    |   |-- batchcmd
    |   |-- cid2
    |   |-- cidacl
    |   |-- cidagent
```

```
|   |-- cidupd
|   |-- cube
|   |-- ip2x
|   |-- ipllog
|   |-- m90
|   |-- net2
|   |-- rdscopy
|   |-- sos-rh90
|   |-- sos-rh90-raid
|   |-- sos-rh90-scsi
|   |-- sos-rhas30
|   |-- sos-rhws30
|   |-- syslevel
|   |-- tl
|-- csd
|   |-- rh90
|-- img
|   |-- apt
|   |-- java
|   |-- linneighborhood
|   |-- mirrordir
|   |-- orexx
|   |-- partimage
|   |-- rh90
|   |   |--RedHat
|   |       |-- RPMS
|   |       |-- base
|   |-- rhws30
|   |   |-- RedHat
|   |       |-- RPMS
|   |       |-- base
|   |-- rpl
|   |-- samba
|   |-- stated
|   |-- suse90
|   |   |-- boot
|   |   |   |-- loader
|   |   |-- content
|   |   |-- directory.yast
|   |   |-- docu
|   |   |-- dosutils
|   |   |-- media.1
|   |   |-- media.2
|   |   |-- media.3
|   |   |-- media.4
|   |   |-- media.5
|   |   |-- suse
|   |       |-- i586
|   |       |-- setup
|   |-- tftp
|   |-- webmin
|-- mir
|   |-- bank
|   |-- tools
|-- pxe
|   |-- cfg
|   |-- dos
|   |-- memtest
|   |-- msg
|   |-- pxelinux.0
|   |-- pxelinux.cfg
|   |-- rh90
|   |-- rhas30
|   |-- rhws30
|   |-- sos
|   |-- suse90
|-- rsp
    |-- apt-rh90.rsp
    |-- csd-rh90.rsp
    |-- customrds.rsp
    |-- java140.rsp
    |-- java141.rsp
    |-- lilo.rsp
    |-- linneighborhood.rsp
    |-- ltsp-30.rsp
    |-- moveparts.rsp
    |-- partimage-rh90.rsp
    |-- qsystem.rsp
    |-- samba3-rh90.rsp
    |-- sos-rh90.rsp
    |-- sos-rhas30.rsp
    |-- sos-rhws30.rsp
    |-- sosraid-rh90.rsp
    |-- sosscsi-rh90.rsp
    |-- srvtools.rsp
    |-- stated.rsp
    |-- tools.rsp
    |-- webmin.rsp
```

7.2.7 Copying preconfigured scenarios to the RDS server

Within the directory where the files were unzipped (tmp), there are some preconfigured scenarios that we use in this redbook.

Enter the following command from a terminal session:

```
cp /tmp/shareb/lnx /hda7/shareb/lnx
```

A second subdirectory under /shareb is /shareb/log, where common log files are written. Example 7-4 shows a list of the files copied into these directories. We describe these files as they are used by the distribution process discussed in the following sections.

Example 7-4 Files copied into the directories

```
/shareb
   |-- lnx
   |   |-- b01c183
   |   |   |-- cfg
   |   |   |-- cidagent.ini
   |   |   |-- ks.cfg
   |   |   |-- log
   |   |-- b01c193
   |   |   |-- base.xml
   |   |   |-- cfg
   |   |   |-- log
   |   |   |-- state.fil
   |   |-- ...
   |       |-- ...
   | |-- ...
```

7.2.8 Copying the installable images

We now copy all the images from the Red Hat 9.0 distribution CDs to the Rapid Deployment Server:

1. Create new subdirectories under /hda7/sharel/img/rh90:

   ```
   mkdir -p /hda7/sharel/img/rh90/RedHat/base
   mkdir -p /hda7/sharel/img/rh90/RedHat/RPMS
   ```

2. Locate the three Red Hat 9.0 CDs.
3. From the command shell, mount the first CD:

   ```
   mount /dev/cdrom /mnt/cdrom
   ```

4. Copy the required files:

   ```
   cp -fRv /mnt/cdrom/RedHat/base /hda7/sharel/img/rh90/RedHat/base
   cp -fRv /mnt/cdrom/RedHat/RPMS/* /hda7/sharel/img/rh90/RedHat/RPMS
   ```

5. Unmount the first CD to eject it by issuing the following command and then insert the second CD:

   ```
   umount /mnt/cdrom
   ```

6. Mount the second CD:

```
mount /dev/cdrom /mnt/cdrom
```

7. Copy the required files:

```
cp -fRv /mnt/cdrom/RedHat/RPMS/* /hda7/sharel/img/rh90/RedHat/RPMS
```

8. Unmount the second CD to eject it by entering the following command and then insert the third CD:

```
umount /mnt/cdrom
```

9. Mount the third CD:

```
mount /dev/cdrom /mnt/cdrom
```

10. Copy the required files:

```
cp -fRv /mnt/cdrom/RedHat/RPMS/* /hda7/sharel/img/rh90/RedHat/RPMS
```

11. Unmount the third CD and remove it:

```
umount /mnt/cdrom
```

The image copy is now complete.

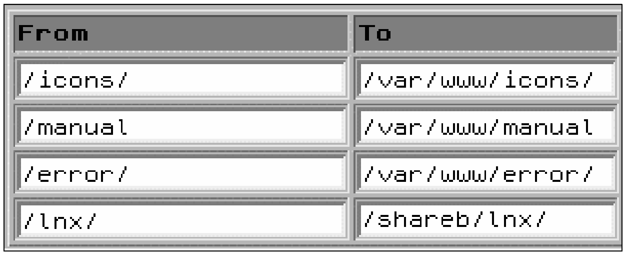
7.2.9 Configuring the services on the server

The following sections explain the changes to make to the FTP and HTTP services.

Configuring the HTTP server

In our environment, we use the Apache Web Server and Webmin to make our configuration changes. For more information about Webmin, see 4.2.1, “Webmin” on page 98.

The first change we make is to the default directories. In the "Document directory aliases" section of Webmin, we make the changes shown in Figure 7-3.



From	To
/icons/	/var/www/icons/
/manual	/var/www/manual
/error/	/var/www/error/
/lnx/	/shareb/lnx/

Figure 7-3 Mapping directory aliases

The changes result in modifications to the /etc/httpd/config/httpd.conf file.

After these changes, restart the HTTP server to allow the configuration changes to take effect. From a terminal shell on the server, enter:

```
cd /etc/init.d/
httpd restart
```

At this point, you can test the server by attempting to access the server using such Uniform Resource Locators (URLs) as:

```
http://10.2.1.40
http://10.2.1.40/lnx/b01c183/ks.cfg
```

Configuring the FTP server

In our environment, we use the vsFTP server. To configure the FTP server, follow these steps:

1. Change the default directory of the FTP server. From a terminal shell, enter the following command:

   ```
   mv -f /var/ftp /var/ftp.org
   ```

2. Link the /hda7/sharel distribution directory to the FTP server using a symbolic link by entering the following command:

   ```
   ln -fs /hda7/sharel /var/ftp
   ```

3. Edit the /etc/vsftpd/vsftpd.conf file and uncomment the following line by removing the leading hash “#”:

   ```
   #xferlog_file=/var/log/vsftpd.log
   ```

4. Save and exit the file.
5. Start the FTP server by entering the following command from a terminal shell:

   ```
   /etc/init.d/vsftpd start
   ```

6. Ensure that the FTP server starts at the next boot and subsequent boots by entering the following from a terminal shell:

   ```
   chkconfig vsftpd on
   ```

7.2.10 Creating user IDs and groups

The scripts used to drive the installation depend on a user ID called *cidusr*. We must create this user and associated group.

1. Make a new group, called *cidgroup*. Enter the following command from a terminal shell:

   ```
   groupadd cidgroup
   ```

2. Add a new user called *cidusr* with a password of *12345*. Enter the following from a terminal shell:

   ```
   useradd cidusr -p "$(perl -e 'print crypt("12345","am");')"
   ```

   This is the user ID that is specified in /hda7/sharel/bin/cid2, which is needed later in our CID installation.

3. Make this new user a member of the group *cidgroup*:

   ```
   usermod -G cidgroup cidusr
   ```

4. Change the home directory for user *cidusr* to */hda7/shareb/lnx*:

   ```
   usermod -d /hda7/shareb/lnx cidusr
   ```

5. Apply the required access permissions to the directory:

   ```
   chown -R root:cidgroup /hda7/shareb/
   chmod -R 775 "/hda7/shareb/"
   ```

6. Test the FTP server. For example, using a browser access:

   ```
   ftp://10.2.1.40
   ```

   Then attempt to log on using the *cidusr* ID.

7.3 Installing a target workstation using a bootable CD

The following sections explain how to install a target installation from the Rapid Deployment Server that we configured by using a bootable CD.

7.3.1 Overview of the installation process

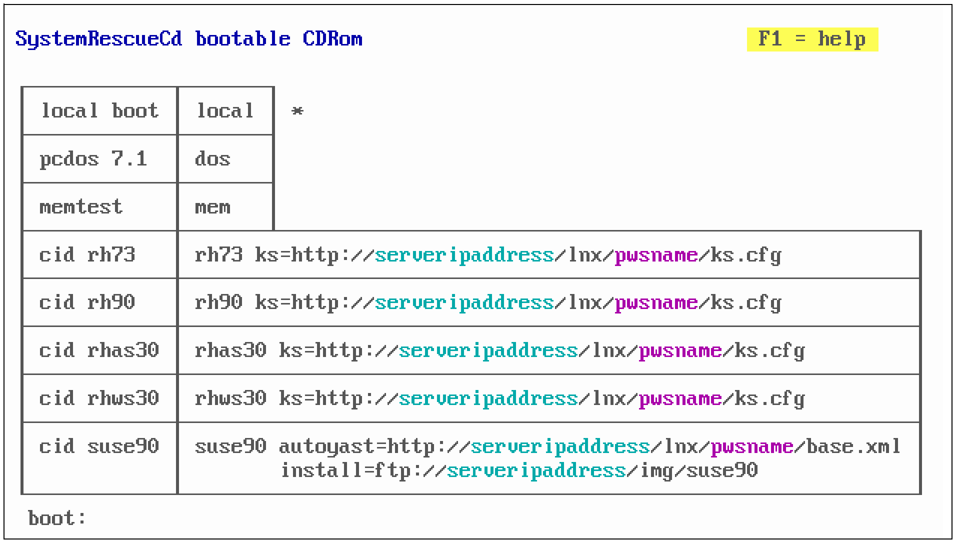
This installation process uses a configured bootable CD. We show how to create this in the next section. During the boot process, a TCP/IP address is requested from a DHCP server.

The target is then connected to the server. You download the configuration files and begin the installation process. After you complete the base installation, the REXX procedures take over and install any additional packages, fixes, and configuration changes that are required.

7.3.2 Creating bootable CID enabled CD-ROM

This section explains how to build a bootable CD for the unattended Linux distribution. The goal is to create a bootable CD that displays a menu that allows for the choice of Linux distribution to be used, either Red Hat, SuSE, or PC-DOS (a useful tool for initial partitioning).

Figure 7-4 shows a sample menu screen that the installer sees when booting the target system from the CD. We explain the usage of this menu after the step-by-step guide about how to create the bootable CD.



```
SystemRescueCd bootable CDRom                                   F1 = help

 local boot  | local  *
 pcdos 7.1   | dos
 memtest     | mem
 cid rh73    | rh73 ks=http://serveripaddress/lnx/pwsname/ks.cfg
 cid rh90    | rh90 ks=http://serveripaddress/lnx/pwsname/ks.cfg
 cid rhas30  | rhas30 ks=http://serveripaddress/lnx/pwsname/ks.cfg
 cid rhws30  | rhws30 ks=http://serveripaddress/lnx/pwsname/ks.cfg
 cid suse90  | suse90 autoyast=http://serveripaddress/lnx/pwsname/base.xml
             |        install=ftp://serveripaddress/img/suse90
 boot:
```

Figure 7-4 Initial menu from bootable CD

The following steps explain how to create the bootable CD-ROM on a Linux system:

1. To create a bootable CD image that results in the launching of an installation process from the server, we use a package called *syslinux*. Retrieve the syslinux package from the Web at:

   http://syslinux.zytor.com/

   The package comes in the formats .tar.gz, .bar.bz2, and zip.

2. Expand the package into a temporary location, for example:

   ```
   mkdir /home/temp
   cd /home/temp
   tar -xzvf syslinux.tar.gz
   ```

3. If you are using Linux, check for the *cdrecord* package. If this package is not installed, download the RPM from the Linux distribution vendor's download page or install from the CD.

4. Make a new directory on the hard disk:

   ```
   mkdir /cdfolder
   ```

5. Make the following subdirectories:

```
mkdir /cdfolder/dos
mkdir /cdfolder/isolinux
mkdir /cdfolder/rh73
mkdir /cdfolder/rh90
mkdir /cdfolder/rhas30
mkdir /cdfolder/rhws30
mkdir /cdfolder/suse90
```

6. Copy the *memdisk* file from the syslinux package into /cdfolder/dos:

```
cp /home/temp/syslinux/memdisk /cdfolder/dos
```

7. Locate the advanced DOS boot diskette that we created in 7.2.2, “Creating an advanced DOS boot diskette” on page 148.
8. Create an image of the bootable disk by entering the following command:

```
dd if=/dev/fd0 of=/cdfolder/dos/pcdos.img
```

9. Copy isolinux.bin from the syslinux package into /cdfolder/isolinux:

```
cp /temp/syslinux/isolinux.bin /cdfolder/isolinux/
```

10. Copy the following files into the /cdfolder/isolinux directory:
    - boot.msg
    - help.msg
    - isolinux.cfg

    The boot.msg and help.msg files are text files that represent the menu that is displayed to the installer and the help message that is displayed if the installer presses the F1 key. The format of these files and any special characters are defined in the syslinux documentation.

    Example 7-5 shows the isolinux.cfg file. You can also create this file with an ASCII editor.

Example 7-5 isolinux.cfg

```
prompt 1
timeout 100
display boot.msg
default local
F1 help.msg
F3 boot.msg

label local
        localboot 0

label dls
        kernel /dos/memdisk
        append initrd=/dos/pcdos.img

label mem
        kernel /memtest/memtest

label sos
        kernel /sos/vmlinuz
       append devfs=nomount ramdisk_size=2600 initrd=/sos/rescue.gz vga=normal
root=/dev/ram0 rw

label rh73
        kernel /rh73/vmlinuz
        append initrd=/rh73/initrde.img devfs=nomount ramdisk_size=8192 vga=788

label rh90
        kernel /rh90/vmlinuz
        append initrd=/rh90/initrde.img devfs=nomount ramdisk_size=10240
vga=788

label rhas30
        kernel /rhas30/vmlinuz
        append initrd=/rhas30/initrde.img devfs=nomount ramdisk_size=10240
vga=788

label rhws30
        kernel /rhws30/vmlinuz
        append initrd=/rhws30/initrde.img devfs=nomount ramdisk_size=10240
vga=788

label suse90
        kernel /suse90/linux
        append initrd=/suse90/initrd devfs=nomount ramdisk_size=65535 vga=788
```

11. For each corresponding Linux distribution, copy the following files from the first CD of the distribution:

    – Red Hat 7.3:

    ```
    cp /mnt/cdrom/images/pxeboot/vmlinuz /cdfolder/rh73/vmlinuz
    cp /mnt/cdrom/images/pxeboot/initrd.img /cdfolder/rh73/initrde.img
    ```

    – Red Hat 9.0

    ```
    cp /mnt/cdrom/images/pxeboot/vmlinuz /cdfolder/rh90/vmlinuz
    cp /mnt/cdrom/images/pxeboot/initrd.img /cdfolder/rh90/initrde.img
    ```

    – Red Hat Advanced Server 3.0:

    ```
    cp /mnt/cdrom/images/pxeboot/vmlinuz /cdfolder/rhas30/vmlinuz
    cp /mnt/cdrom/images/pxeboot/initrd.img /cdfolder/rhas30/initrde.img
    ```

    – Red Hat Advanced Workstation 3.0:

```
cp /mnt/cdrom/images/pxeboot/vmlinuz /cdfolder/rhws30/vmlinuz
cp /mnt/cdrom/images/pxeboot/initrd.img /cdfolder/rhws30/initrde.img
```

– SuSE 9.0:

```
cp /mnt/cdrom/boot/loader/linux /cdfolder/suse90/linux
cp /mnt/cdrom/boot/loader/initrd /cdfolder/suse90/initrd
```

At this point, the CD-ROM is configured with all the bootable images and the directory structure is ready to be created as an ISO file ready for burning.

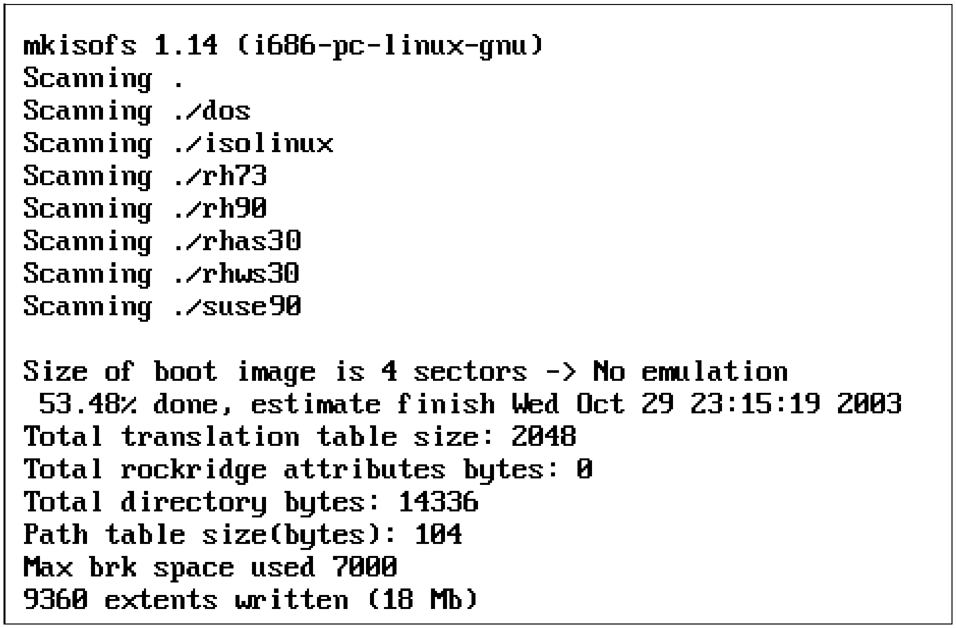
12. Change to the /cdfolder directory:

```
cd /cdfolder
```

13. Enter the following command:

```
mkisofs -v -b isolinux/isolinux.bin -c isolinux/boot.cat -no-emul-boot
-boot-load-size 4 -boot-info-table -o ../lincid.iso ."
```

You see the screen shown in Figure 7-5.



```
mkisofs 1.14 (i686-pc-linux-gnu)
Scanning .
Scanning ./dos
Scanning ./isolinux
Scanning ./rh73
Scanning ./rh90
Scanning ./rhas30
Scanning ./rhws30
Scanning ./suse90

Size of boot image is 4 sectors -> No emulation
 53.48% done, estimate finish Wed Oct 29 23:15:19 2003
Total translation table size: 2048
Total rockridge attributes bytes: 0
Total directory bytes: 14336
Path table size(bytes): 104
Max brk space used 7000
9360 extents written (18 Mb)
```

Figure 7-5 Output of the mkisofs command

Now that an ISO image file is created, burn a CD using this ISO image.

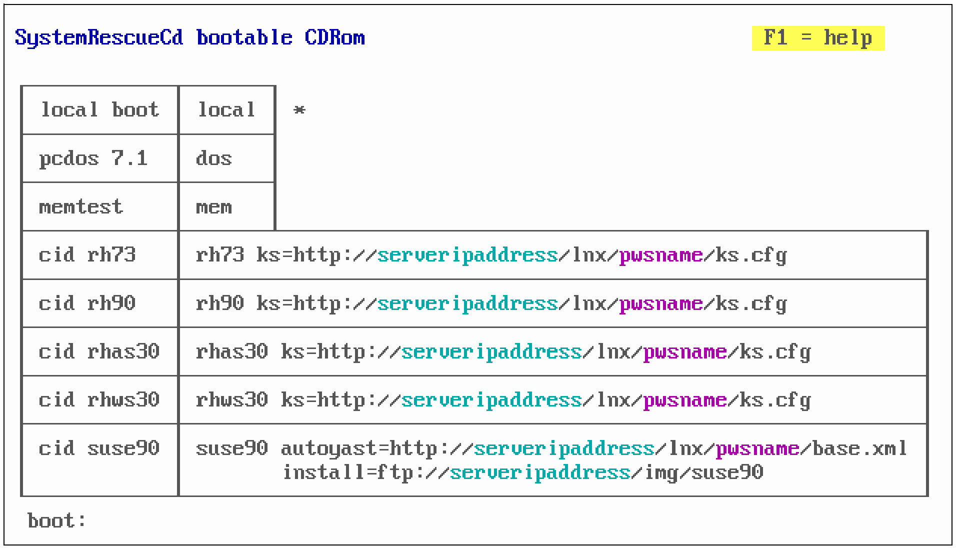
7.3.3 Installing a target

We are now ready to install a target system:

1. Boot the new bootable CID-enabled Linux CD on the target system. In our case, it is system B01C183.
2. Wait for the menu screen as shown in Figure 7-6. At the boot: prompt at the bottom of the screen, type:

   ```
   rhws30 ks=http://10.2.1.40/lnx/b01c183/ks.cfg
   ```

   This loads the Red Hat kernel and access the kickstart configuration file (ks.cfg) from the HTTP server (/shareb/lnx/b01c183).



```
SystemRescueCd bootable CDRom                                   F1 = help

 local boot   local     *
 pcdos 7.1    dos
 memtest      mem
 cid rh73     rh73 ks=http://serveripaddress/lnx/pwsname/ks.cfg
 cid rh90     rh90 ks=http://serveripaddress/lnx/pwsname/ks.cfg
 cid rhas30   rhas30 ks=http://serveripaddress/lnx/pwsname/ks.cfg
 cid rhws30   rhws30 ks=http://serveripaddress/lnx/pwsname/ks.cfg
 cid suse90   suse90 autoyast=http://serveripaddress/lnx/pwsname/base.xml
                     install=ftp://serveripaddress/img/suse90
boot:
```

Figure 7-6 Boot menu

3. After the ks.cfg is downloaded, Anaconda retrieves all packages via FTP, as specified in the ks.cfg (/sharel/img/rhws30) file.

   When the installation starts, you can remove the CD from the drive at any time. The installation continues to completion.

   Near the end of the installation, notice that Anaconda jumps to the *post* section.
4. Download *cid2* and continue the installation with the help of the REXX procedures.
5. At the end of the installation, the workstation reboots, as specified in the ks.cfg (reboot). Ensure that the reboot statement in the ks.cfg file exists and is not commented out.

The first Linux CID installation is complete.

7.4 Installing targets using RDS and PXE

The previous sections explain how to set up a Rapid Deployment Server and create a CD image that you can for an automated installation. However, this process still requires someone to boot the CD and enter an option at the menu.

The following sections explain how to install target systems using the Rapid Deployment Server by using a PXE-enabled system. This allows for the installation of clients without needing a CD or other bootable image to be physically present at the target system.

7.4.1 Introduction to the PXE protocol and its function

PXE allows a system to retrieve its network information, bootstrap image, and boot sequence from a network server. PXE is often used as a standard for remote booting. It is useful for unattended installations of systems or diskless workstations.

Although we do not discuss PXE and all its capabilities, we explain a simple and common setup where many the necessary facilities are included in the same server (the DHCP server). To learn more about the capabilities of PXE, refer to the PXE 2.1 specification at:

ftp://download.intel.com/ial/wfm/pxespec.pdf

Running a PXE boot environment

To run a PXE boot environment, you need the following components on the network:

- A DHCP server (DHCP or BOOTP daemon required)
- A TFTP server
- One or more PXE clients (PXE-enabled Network Interface Cards (NICs))

The DHCP server can be located anywhere on the network as long as it is reachable by the PXE clients. The DHCP server provides the PXE client with:

- An IP address
- A subnet mask
- A gateway address (optionally)

If the DHCP server identifies the request as a PXE client request, it sends additional information to the PXE client. If the machine is recognized as a PXE client, the server provides the PXE client with the server (the TFTP server) and the exact location of a Network Bootstrap Program (NBP). As we see, this additional information is all that is necessary for the client to access our Rapid Deployment Server and have a complete environment installed.

Network Bootstrap Program

After bootstrapping the NBP, the behavior of the PXE client depends on the NBP logic. A variety of NBPs are available on the Web. For more information about several free NBPs, see:

http://clic.mandrakesoft.com/documentation/pxe/ch06.html

We use the pxelinux bootstrap, which is part of the syslinux package distributed with Red Hat Linux. If for some reason the NBP cannot be found or the TFTP server is not available, the PXE client continues with the normal boot sequence.

The pxelinux NBP

The pxelinux bootstrap contacts the TFTP server and looks for a syslinux compliant boot configuration file. This file name is based on the PXE client IP address.

For example, if the IP address is 10.1.2.3, it looks for the following files in order:

```
/pxelinux.cfg/0A010203
/pxelinux.cfg/0A01020
/pxelinux.cfg/0A0102
/pxelinux.cfg/0A010
/pxelinux.cfg/0A01
/pxelinux.cfg/0A0
/pxelinux.cfg/0A
/pxelinux.cfg/0
/pxelinux.cfg/default
```

The syslinux boot configuration file determines how the PXE client will boot, which media to use, and any parameters that it needs to do this.

We can specify that it should boot from a local medium (a hard disk) or to boot from a kernel-image that is provided by the TFTP server. Since we install Red Hat in an unattended mode, the kernel-image and RAM disk that we boot from is the one that is used to install Red Hat. Instead of starting the normal interactive installation, we provide a kickstart file with the kernel boot parameters, the same as was done with the boot CD.

For more information about the capabilities of pxelinux or syslinux, read the documentation that comes with the syslinux distribution. On a recent version of Red Hat, go to /usr/share/doc/syslinux-1.75/pxelinux.doc.

Making the client aware it is to be installed

To make the client aware that it needs to be installed, we create a per-workstation directory on one of our Rapid Deployment Servers, for example:

```
/shareb/lnx/b01c183
```

Within this directory, we find either the Red Hat Linux Kickstart file *ks.cfg* or the SuSE Linux *base.xml* response file and a file called *state.fil*.

Here is an example of the state.fil file:

```
[CLIENT STATE]

State = HYBRID
PxeCfgfile = rhws30
Ipaddress = 10.2.1.183
MacAddress = 00:09:6B:E9:D4:DD
Servername = 10.2.1.40
Pwsname = b01c183
```

You can set the option `State` to either of the following options:

```
State = HYBRID
State = NEW
```

If the state is set to HYBRID, then no action is performed. If the state is set to NEW, then the system re-installs itself.

This state.fil file is read by a REXX procedure called STATED, which automatically configures the TFTP server and associated files.

For more information about STATED and what it does, refer to 7.5, “REXX procedures used by RDS” on page 173.

Enabling PXE on a target

Whether you can use PXE depends on the hardware and BIOS of the target systems. Only some NICs are PXE-enabled. If the NIC is PXE aware, then it may need to be enabled in the system BIOS or within the NIC itself using tools supplied with the NIC.

Ensure the NIC is enabled for network boot and that it is designated as the default within the boot sequence. The following boot sequence provides the most flexibility in most scenarios:

1. Network
2. CD-ROM
3. Hard disk

7.4.2 Additional server configuration steps for PXE targets

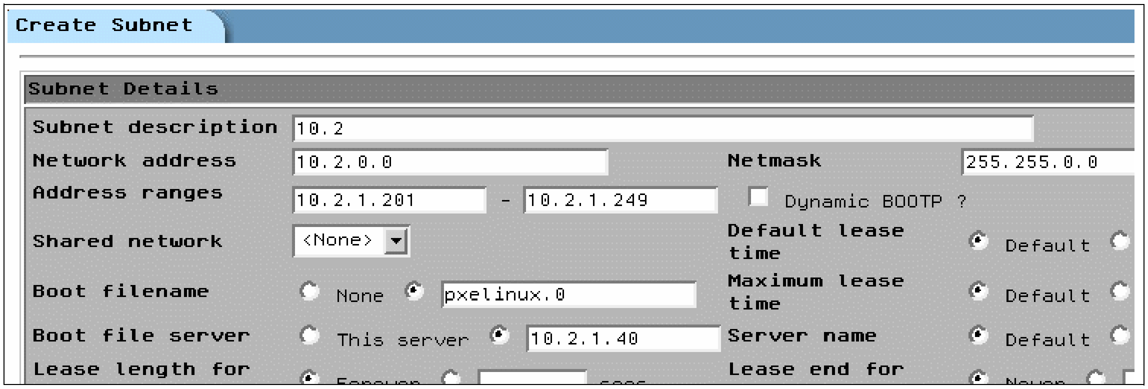
As discussed in the previous section, to distribute installations via PXE, you need to perform some additional steps. This section explains how to configure a DHCP server, a TFTP server, and the PXE settings on an RDS Linux server. You can add PXE settings to any DHCP server. We cover only setting up a Linux DHCP server. You can export the PXE settings easily to most other configurations.

DHCP server configuration

Follow these steps to set up the PXE-configured DHCP server:

1. Connect to the server using a browser and login to Webmin.
2. Click the **Servers** icon and then select the **DHCP Server** icon.
3. Click **Add a new subnet**.
4. On the Create Subnet page (Figure 7-7), under Subnet Details, complete the fields as shown in the example.

> **Note:** Enter the values based on the IP range of the specific environment. The range shown here is an example only.



Create Subnet

Subnet Details

Subnet description: 10.2

Network address: 10.2.0.0 | Netmask: 255.255.0.0

Address ranges: 10.2.1.201 - 10.2.1.249 | □ Dynamic BOOTP ?

Shared network: <None> | Default lease time: ● Default ○

Boot filename: ○ None ● pxelinux.0 | Maximum lease time: ● Default ○

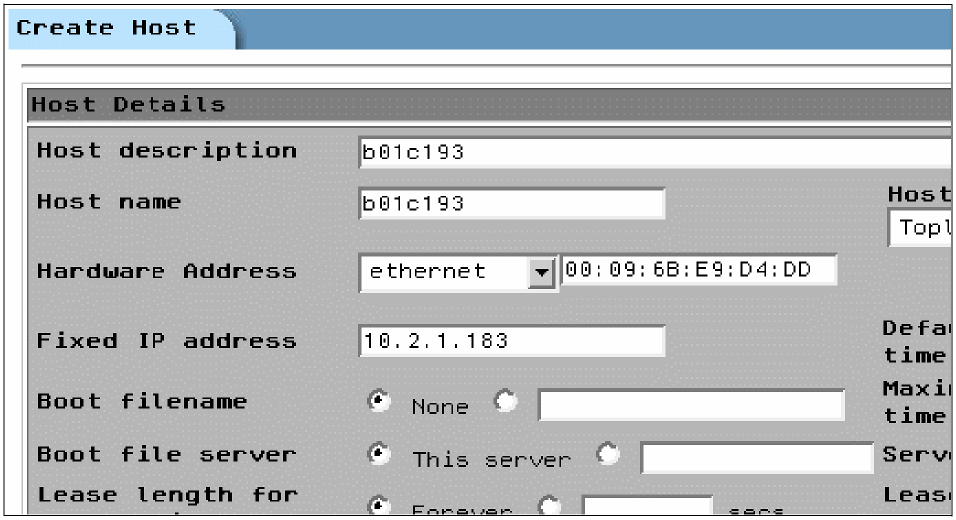
Boot file server: ○ This server ● 10.2.1.40 | Server name: ● Default ○

Lease length for ● Forever ○ secs | Lease end for ● Never ○

Figure 7-7 Configuring the IP range

5. Click the **Create** button at the bottom of the window.
6. Click the **Add a new host** button.

7. On the Create Host page (Figure 7-8), under Host Details, enter the information as shown in the example. Set the Hardware Address field entry to the MAC address of your PXE target workstation.



Create Host

Host Details

Host description: b01c193

Host name: b01c193

Hardware Address: ethernet 00:09:6B:E9:D4:DD

Fixed IP address: 10.2.1.183

Boot filename: ● None ○

Boot file server: ● This server ○

Lease length for: ● Forever ○ secs

Figure 7-8 Setting the MAC address

8. Click the **Create** button on the bottom of the window.
9. Click **Edit Client Options**.
10. Change the Dynamic DNS update style from the default to **Ad-hoc**.
11. Click **Save** on the bottom of the window.
12. Click the **Start Server** icon on the bottom of the window.

For reference, the file that was just edited via Webmin is /etc/dhcpd.conf. The administrator can do this manually if they are comfortable with managing DHCP configuration files.

Configuring the TFTP server

To configure the TFTP server and associated files, follow these steps:

1. Enable the tftpd server in the /etc/xinet.d/tftp file. Example 7-6 shows the changed file. You only need to make one change. That is set `disable = yes` to `disable=no`.

Example 7-6 Changed /etc/xinet.d/tftp file

```
# default: off
# description: The tftp server serves files using the trivial file transfer \
#       protocol.  The tftp protocol is often used to boot diskless \
#       workstations, download configuration files to network-aware printers, \
#       and to start the installation process for some operating systems.
service tftp
{
        disable                 = no
        socket_type             = dgram
        protocol                = udp
        wait                    = yes
        user                    = root
        server                  = /usr/sbin/in.tftpd
        server_args             = -s /tftpboot
        per_source              = 11
        cps                     = 100 2
        flags                   = IPv4
}
```

2. Start the tftpd server:

   ```
   /etc/init.d/xinetd restart
   ```

3. Test the TFTP server, for example, by entering the following commands:

   ```
   tftp 10.2.1.40
   >bin
   >get pxelinix.0 (you should see “Received 11156 bytes in 0.1 seconds”)
   >quit
   ```

Installing a target using PXE

After you configure everything as we explained, you install a client.

1. Configure the state.fil file to indicate:

   ```
   State = NEW
   ```

2. Power on the client.

Then in a few minutes, the installation is complete.

7.5 REXX procedures used by RDS

This section outlines and describes some of the primary REXX procedures used in our distribution process. All of these procedures are available from our FTP site. See Appendix B, “Additional material” on page 187, for information about how to obtain these REXX procedures and sample files.

7.5.1 CIDAGENT

The main procedure used during our CID installation performs the following actions:

- Creates a /var/cid directory for the temporary download files
- Creates a /var/log/cid directory for the log files
- Gets system environment variables such as IP address, Rapid Deployment Server IP address, workstation name, and so on
- Writes these variables into /var/cid/cid.ini
- Gets, via FTP, the cidagent.ini file stored in the per-workstation directory, for example /shareb/lnx/b01c183
- Reads and interprets the cidagent.ini file and executes the various commands indicated in that file and checks for proper return codes
- Upload a log file with the result after each execution
- Retrieve and copy all files beneath the config directory of the per-workstation directory, for example /shareb/lnx/b01c183/cfg
- If the state.fil file exists, obtains and updates it in the per-workstation directory

  This file is used in a PXE boot environment. Among other things, it determines whether a new installation should occur during the next boot.
- Uploads all log files collected during the unattended Installation into the log directory beneath the per-workstation directory, for example /shareb/lnx/b01c183/log

BATCHCMD

This procedure is used during our CID installation. It takes as parameters a response file and a log file as the following example shows:

```
batchcmd /r:csd-rh90.rsp /l:csd.log
```

When this program is executed, it performs the following actions:

- Downloads the response file from the /sharel/rsp directory
- Reads and interprets that file and execute each appropriate line, checking for proper return codes
- In case of errors, it writes "`errors found`" to the terminal (TTY1) and detailed error information to TTY5

Then you upload the log file to the per-workstation directory, for example:

```
/shareb/rds090/log/csd.log
```

CUBE

This procedure replaces strings in the ASCII files. Here is an example of the syntax to use:

```
cube /etc/lilo.conf "vga=788" with "vga=791"
```

CIDUPD

This procedure goes back to the Rapid Deployment Server. It downloads and updates the files specified within the REXX program.

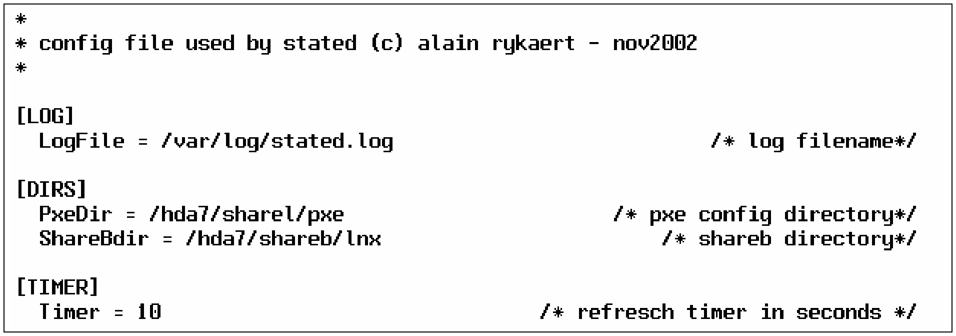
STATED

The state daemon is a procedure that runs as a detached process by running:

```
/etc/init.d/stated start
```

It is designed to run on the server to aid with the PXE automation. This process executes another REXX procedure, /bin/stated2, that performs the following actions:

- Reads the configuration /etc/stated.conf file and monitors:
  - The directory where the PXE-related files are stored, such as:
    - The PXE template directory /sharel/pxe/cfg
    - The Linux-related directories /sharel/pxe/rhws30, /sharel/pxe/suse90, etc.
    - The PXE configuration directory /sharel/pxe/pxelinux.cfg
  - The per-workstation directories where we store our target response files ks.cfg and a file called state.fil
  - The timer value informing our daemon how many seconds it waits before the next monitoring cycle
  - A log file to store information, for example /var/log/stated.log (Figure 7-9)



```
*
* config file used by stated (c) alain rykaert - nov2002
*

[LOG]
  LogFile = /var/log/stated.log                          /* log filename*/

[DIRS]
  PxeDir = /hda7/sharel/pxe                      /* pxe config directory*/
  ShareBdir = /hda7/shareb/lnx                       /* shareb directory*/

[TIMER]
  Timer = 10                               /* refresch timer in seconds */
```

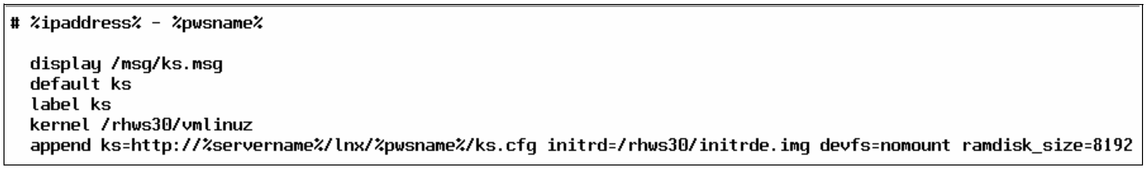
Figure 7-9 Example of the stated.conf file

- Deletes all files beneath the /sharel/pxe/pxelinux.cfg directory except for the default file
- Loops forever with the timer interval
- Gets all target names where the state.fil file is stored
- Reads and interprets each state.fil that it finds and reads such information as:
  - The system state which can be *new* or *hybrid* as described earlier
  - Workstation name, for example B01C183
  - Workstation IP address, for example 10.2.1.183
  - Rapid Deployment Server IP address, for example 10.2.1.40
  - Configuration file, for example rhws30
  - MAC address, for example 00:09:6B:E9:D4:DD
- If the daemon reads state.fil and finds a *new* state, it translates the decimal IP address to a hexadecimal address. For example, 10.2.1.183 becomes 0A0201B7.

  > **Tip:** This is accomplished by using another REXX procedure called *ip2x* to calculate the value.

- If a new state is found, it copies the template file /sharel/pxe/cfg/rhws30 into the /sharel/pxe/pxelinux.cfg directory with the hexadecimal address 0A0201B7 that was converted earlier.

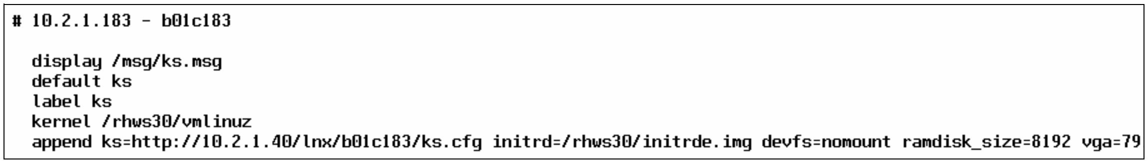
  Figure 7-10 shows the template file for a Red Hat PXE installation.



```
# %ipaddress% - %pwsname%

  display /msg/ks.msg
  default ks
  label ks
  kernel /rhws30/vmlinuz
  append ks=http://%servername%/lnx/%pwsname%/ks.cfg initrd=/rhws30/initrde.img devfs=nomount ramdisk_size=8192
```

Figure 7-10 Template pxelinux.cfg file

The daemon customizes this file. It fills in such information as workstation name and the rapid deployment server IP address that results in a configured file as shown in Figure 7-11.



```
# 10.2.1.183 - b01c183

  display /msg/ks.msg
  default ks
  label ks
  kernel /rhws30/vmlinuz
  append ks=http://10.2.1.40/lnx/b01c183/ks.cfg initrd=/rhws30/initrde.img devfs=nomount ramdisk_size=8192 vga=79
```

Figure 7-11 Customized pxelinux.cfg

- When an installation completes, it changes the state in the state.fil file from *new* to *hybrid*. When the daemon finds a hybrid state in the state.fil file, it deletes the /sharel/pxe/pxelinux.cfg/0A0201B7 file so that the workstation reads the /sharel/pxe/pxelinux.cfg/default file. This forces the workstation to boot to the local disk. That means that the configuration of the workstation's network adapter can always have the PXE option set. The default file is:

  ```
  #hybrid mode
     display /msg/boot.msg
     label local
     default local
     localboot 0
  ```

  The daemon logs all its activities into the /var/log/stated.log log file. To stop the stated daemon, enter:

  ```
  /etc/init.d/stated stop
  ```

  To check if the stated daemon is still running, enter:

  ```
  /etc/init.d/stated status
  ```

NET2

This procedure is used to emulate the OS/2 **net** command. It is named **net2** because Samba 3 introduced a *net* program. At the time of writing this redbook, the **net2** program can:

- Add, delete, and change users and passwords, for example:

  ```
  net2 user fred wilma /add"
  ```

- Browse for active sessions:

  ```
  net2 session
  ```

- Start, stop, or obtain an overview of all server services:

  ```
  net2 start
  net2 start dhcpd
  net2 stop httpd
  ```

SYSLEVEL

This procedure is an OS/2 look-a-like program to the **syslevel** command. It browses all products and their versions from a list specified within the program:

```
# syslevel
[kernel]
kernel-2.4.20-20.9
[samba]
samba-3.0.0-1
```

TL

This procedure collects information about the network configuration, similar to viewing lantran.log in OS/2. It shows the:

- Network type, manufacturer, and version
- Network status
- IP configuration
- DNS configuration
- Route configuration

CR

This procedure looks for CRLF or 0D0A'x in configuration files. It changes it back to a proper UNIX format by removing all of the carriage return ('0D'x) characters.

7.6 Summary

This chapter introduced a mechanism to easily install a large number of Linux clients. It is based on the same principles as the OS/2 CID mechanism. Therefore, OS/2 administrators that are familiar with using CID to install OS/2 clients may find this process familiar and useful.

A

Basic Linux for OS/2 users

For those of you who are familiar with OS/2, but not yet familiar with Linux, this appendix covers a basic introduction to Linux in comparison to a traditional OS/2 system. It includes an overview of:

- OS/2 commands and their counterparts within Linux
- Basic file system considerations

OS/2 commands and their Linux counterparts

Several Linux commands are similar or identical to either the name or function of OS/2 commands. Table 7-1 shows common OS/2 commands with their Linux counterparts.

Table 7-1 OS/2 commands versus Linux commands

Command purpose	OS/2 command	Linux command
Change to the parent directory	cd.. or cd ..	cd ..
Change to the root directory	cd\ or cd \	cd /
Clear the screen	cls	clear
Close a command prompt window	exit	exit
Compare the contents of two files	fc	diff
Copy a file	copy	cp
Create a directory	mkdir or md	mkdir
Delete Files	del or erase	rm
Display file contents a screen at a time	more	more or less
Display the amount of available memory	mem	free
Display/Change the date	date	date
Display/Change the time	time	date
Echo output to the screen	echo	echo
Find a file by filename	dir /s	find / -name “filename” -print
Find a file by other attributes	dir	find
Find a text string in a file	find	grep
Format a floppy disk	format	mke2fs or mformat
List files in a directory	dir	ls
Move files	move	mv
Rename a file	ren	mv
Show the current directory path	cd	pwd (present working directory)
Kernel version	bldvel os2krnl	uname -a
Check the path	path	echo $PATH

Command purpose	OS/2 command	Linux command
list environment variables	set	set

Basic file system considerations

The following sections provide a brief overview of the file systems. It includes the similarities and differences between those supported by OS/2 and Linux systems.

OS/2 file systems

The traditional file systems that you can use with OS/2 are FAT, HPFS, and HPFS386. With the release of OS/2 Warp Server for e-business, OS/2 users were introduced to IBM journaled file system (JFS) for OS/2. JFS sets itself apart from traditional OS/2 file systems in two ways:

- It isn’t bootable.
- It records all changes that are made to it, for example, database access, file moves and copies, and so on.

If OS/2 traps rendering the file system dirty, at any point, on reboot the JFS drive replays its journal log and brings the system up immediately without data loss or corruption.

IBM JFS for OS/2 and Linux

In December 1999, IBM took a snapshot of the JFS source code for OS/2. At this point, the developers started to port this code to Linux. The JFS source code was later made available as open source.

The first beta was released in late 2000. The first official 1.0.0 release was available in mid-2001. As of writing this redbook, JFS V1.1.4 is available and is the sixty-seventh release of IBM Enterprise JFS technology port to Linux.

Linux file system concepts

The most popular native Linux file systems are ext2, ext3, jfs, reiserfs, and xfs. These Linux file systems introduce new features and concepts that may be new to a traditional OS/2 user.

Hard disk device naming conventions

Several Linux standard conventions exist for named hard disk devices (HDDs).

IDE

If our first hard disk hda is divided into two partitions, then the naming convention is Disk0 = hda. The first partition is labeled hda1. The second partition is labeled hda2. Consider the following convention:

- Disk0 = hda
- Disk1 = hdb
- Disk2 = hdc
- Disk3 = hdd

The directory /hdb1 represents a second hard disk that is installed in our workstation, instead of this being represented by a second drive letter for example D:. We mounted it into our root file system "/" into the directory name of our choosing, which in this case is /hdb1.

SCSI

The Small Computer System Interface (SCSI) naming convention can change depending on the vendor, the device driver, and the distribution. Here is an example that is used by some IBM controllers. It is the same as the previous convention with the first letter changed to represent a SCSI drive:

- Disk0 = sda
- Disk1 = sdb
- Disk2 = sdc
- Disk3 = sdd

If our first hard disk sda is divided into two partitions, then the naming convention is Disk0 = sda. The first partition is labeled sda1. The second partition is labeled sda2.

Where have my drive letters gone?

The Linux operating system has no concept of drive letters as we are used to seeing in OS/2. Linux gives the impression that its entire file system is located in one place. This is achieved by using a flat directory structure for all partitions. Within Linux, the file system starts from the root, which is represented by a "/". Every directory, device, and partition are located underneath root "/".

Devices

Even devices used by Linux, such as the parallel port, CD-ROM, and floppy drive, are represented as files in the Linux file system. These are located within the /dev directory. However, the devices cannot be accessed via this directory. It is simply a listing of the available devices to the workstation. To access one of these devices, for example a CD-ROM drive, you need to mount this device to another directory within the Linux file system.

Mounting

In the previous section, we discussed how devices are defined to the Linux operating system. The following example shows how to mount your CD drive to the Linux file system to access your CD's data:

```
mount /dev/cdrom /mnt/cdrom
```

In earlier releases of Linux, this was a much more complicated process and you had to specify many additional parameters. Now most releases of Linux allow an even shorter form of the mount command as follows:

```
mount /mnt/cdrom
```

Note: Your default mount directory depends on the Linux distribution.

The **mount** command is used for more than accessing your CD-ROM, floppy and ZIP drives. It is also used to add extra hard disk drives, NFS shares, SMB shares, and so on. The following command shown an example of mounting an OS/2 LAN server alias to a file system:

```
mount -t smbfs -o username=userid,password=password //os2server/data
/mnt/os2serverdata
```

Although this share exists on an OS/2 server, it now appears as though it is part of the local Linux file system.

For more information about the **mount** command from a Linux workstation in a terminal shell, enter:

```
man mount
```

Umounting

To retrieve a CD from the CD-ROM drive, you must unmount it from the Linux file system as shown here:

```
umount /mnt/cdrom
```

Symbolic links

The directory /hda7 is a large partition that was created on the first hard disk as a separate data area. We can mount this second partition to the file system. In Figure A-1, notice that /hda7 is mentioned twice: /hda7 and /sharel -> /hda7/sharel. This is known as a symbolic link. A symbolic link is a way to make an alias for a file or directory to another.

Also in Figure A-1, notice the directory /sharel. When the user changes into this directory, they actually change into the /hda7/sharel directory:

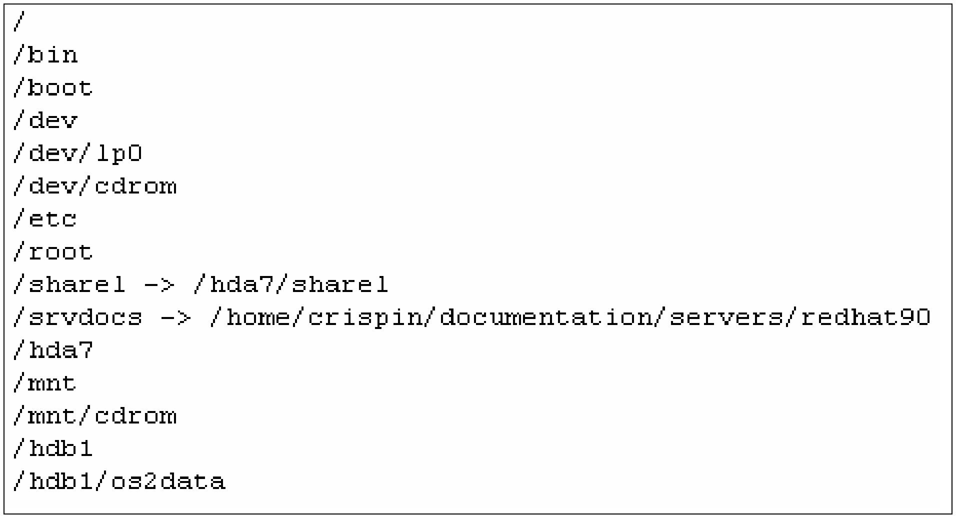
```
cd /sharel
```

Any file or directory can be made into a symbolic link. To create a symbolic link for the /home/crispin/documentation/servers/redhat90 directory, enter the following command:

```
ln -s /home/crispin/documentation/servers/redhat90 /srvdocs
```

This makes it easier to access that directory, since it can be referred to simply as /srvdocs.

If a directory listing is now created, for example with the **ll** command, the user now sees the screen shown in Figure A-1.



```
/
/bin
/boot
/dev
/dev/lp0
/dev/cdrom
/etc
/root
/sharel -> /hda7/sharel
/srvdocs -> /home/crispin/documentation/servers/redhat90
/hda7
/mnt
/mnt/cdrom
/hdb1
/hdb1/os2data
```

Figure A-1 Directory listing showing symbolic link

Text files in Linux

There is a difference between Linux and OS/2 and Windows clients that can affect coexistence, especially when sharing files is the format of text-based files. An end of line is marked in Linux as simply a line feed (LF) character. In Windows and OS/2, it is marked with a Carriage Return (CR) and an LF character.

If text files are shared or ported between OS/2 systems and Linux clients, then you may need to remove the extra CR character. There are several ways to do this such as using **`dos2unix`**.

According to dostext.txt, you can also use the **`tr`** command to transform a text file to a Linux formatted text file by entering:

```
tr -d “\r“ <dostext.txt> linuxtext.txt
```

Summary

This appendix provided a short introduction to some of the commands and methods for dealing with file systems in a Linux environment.

B

Additional material

This IBM Redbook refers to additional material that can be downloaded from the Internet.

Locating the Web material

The Web material associated with this redbook is available in softcopy on the Internet from the IBM Redbooks Web server. Point your Web browser to:

`ftp://www.redbooks.ibm.com/redbooks/SG246621`

Alternatively, you can go to the IBM Redbooks Web site at:

`ibm.com``/redbooks`

Select the **Additional materials** and open the directory that corresponds with the redbook form number, SG246621.

Using the Web material

The additional Web material that accompanies this redbook includes the following files:

File name	*Description*
SG246621.zip	Rapid Deployment Server sample files

System requirements for downloading the Web material

To use this additional material, you must have the Linux operating system.

How to use the Web material

Create a subdirectory (folder) on your workstation. Then unzip the contents of the Web material zip file into this folder.

For information about these files and how to use them, see Chapter 7, “Linux client installation” on page 145.

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

IBM Redbooks

For information about ordering these publications, see “How to get IBM Redbooks” on page 191.

- *OS/2 Server Transition*, SG24-6631
- *Linux Handbook: A Guide to IBM Linux Solutions and Resources*, SG24-7000

Online resources

Within this redbook, many URLs were provided where the reader could gain more information about a variety of technologies and products. The following list of references is a subset of those provided in the redbook, and primarily includes key IBM resources and sites associated with various open organizations:

- IBM Software

  `http://www.ibm.com/software`
- IBM WebSphere Host On-Demand

  `http://www.ibm.com/software/webservers/hostondemand/`
- IBM Java information

  `http://www.ibm.com/java`
- JFS

  `http://www.ibm.com/developerworks/oss/jfs`
- Lotus products

  `http://www.ibm.com/software/lotus`
- Omni print drivers

  `http://www.ibm.com/developer/opensource/linux/projects/omni/`
- Tivoli products

  `http://www.ibm.com/software/tivoli`

- ALSA

  http://www.alsa-project.org

- Blackdown

  http://www.blackdown.org

- Epiphany

  http://epiphany.mozdev.org

- GAIM

  http://gaim.sourceforge.net

- Galeon

  http://galeon.sourceforge.net

- GNOME

  http://www.gnome.org

- Helix

  http://www.helixcommunity.org

- Internet Printing Protocol

  http://www.pwg.org/ipp

- KDE

  http://www.kde.org

- KMail

  http://kmail.kde.org

- Konqueror

  http://www.konqueror.org

- LPRng

  http://www.lprng.org

- Mozilla

  http://www.mozilla.org

- NFS

  http://www.ietf.org/rfc/rfc3530.txt

- Samba

  http://www.samba.org

- tn520

  http://tn5250.sourceforge.net/

How to get IBM Redbooks

You can search for, view, or download Redbooks, Redpapers, Hints and Tips, draft publications and Additional materials, as well as order hardcopy Redbooks or CD-ROMs, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

IBM Redbooks

OS/2 to Linux Client Transition

(0.2”spine)
0.17”<->0.473”
90<->249 pages

OS/2 to Linux Client Transition

Gain information about using Linux as a client operating system

Discover how to replace OS/2 clients with Linux

Learn about helpful tools and tips

This IBM Redbook provides information related to the viability of Linux as a client platform. It targets technical personnel who are involved in evaluating Linux as a possible client platform. It also targets administrators and support personnel who are responsible for supporting client systems. This redbook can also be helpful to anyone who is evaluating the potential of using Linux for enterprise client systems. However, the key focus is on environments where OS/2 is currently used.

Many enterprises have been using OS/2 as a stable platform for critical enterprise client applications. However, as those enterprises look to the future, they look for a platform on which they can build a strategy that is open, standards-based, secure, and provides a cost-effective solution. Linux has become successful as a server platform in many of these same enterprises. It comes as no surprise that these enterprises also want to evaluate the possibility of including Linux for many of their client systems.

This redbook describes platform and functional considerations for choosing Linux as a client platform. It examines techniques and facilities for administering Linux clients, coexistence of Linux clients with other platforms, and a technique to easily install Linux clients based on the well-known OS/2-based CID methodology.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

SG24-6621-00 ISBN 0738498173