IBM eserver

IBM

Deploying Apache on IBM eserver BladeCenter

Installing Linux and IBM Director Agent

Installing Apache

Building a Web cluster

Rufus Credle
Eric Butler
Tim Verhoeven
David Green

Redpaper

ibm.com/redbooks

International Technical Support Organization

Deploying Apache on IBM eserver BladeCenter

November 2003

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

Second Edition (November 2003)

This edition applies to IBM @server BladeCenter (8677-1xx), IBM @server BladeCenter HS20 (8678-21x and 8678-41x), Red Hat Linux 7.3 and SuSE Linux Enterprise Server 8.

Contents

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 eserver™
Redbooks(logo) ™
eServer™
ibm.com®
pSeries®
xSeries®
zSeries®
AIX®
BladeCenter™
Chipkill™
Domino®
DB2®
DFS™
IBM®
Lotus Notes®
Lotus®
Netfinity®
Notes®
OS/2®
Predictive Failure Analysis®
PS/2®
Redbooks™
ServerProven®
Tivoli Enterprise™
Tivoli®
WebSphere®

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

Rational Software Corporation®
Rational®

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

As data centers have grown with the proliferation of Intel®-based servers over recent years, it is important to note that rack space and floor space can now be more efficiently used with the IBM @server BladeCenter™ HS20 servers. Attractive cost savings are also possible where a large number of rack installed servers are required since the equivalent number of 1U servers will be much more expensive.

This IBM® Redpaper describes how to set up and configure Linux and Apache on the IBM @server BladeCenter and provides some pointers on what applications to use to manage the installation. We describe building a Web cluster using Linux, Apache, Linux Virtual Server and discuss keepalived as well. We also describes the functionality of the IBM @server BladeCenter in this type of environment.

The team that wrote this Redpaper

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

Rufus Credle is a Senior I/T Specialist and certified Professional Server Specialist at the International Technical Support Organization, Raleigh Center. He conducts residencies and develops Redbooks™ about network operating systems, ERP solutions, voice technology, high availability and clustering solutions, Web application servers, pervasive computing, and IBM and OEM e-business applications, all running IBM @server xSeries® systems. Rufus's various positions during his IBM career have included assignments in administration and asset management, systems engineering, sales and marketing, and IT services. He holds a BS degree in business management from Saint Augustine's College. Rufus has been employed at IBM for 22 years.

Eric Butler is a Network Engineer in United States of America. He has three years of experience with Linux and Open Source software. He holds a degree in Electrical Engineering. He has worked at IBM for seven years. His areas of expertise include network monitoring, administration of Lotus® Domino®, AIX®, and Linux systems, DCE/DFS™, Apache, Samba, MRTG, Cricket, and xSeries and pSeries® hardware. Eric participated extensively in the writing of *Deploying Samba on IBM @server BladeCenter*, REDP3595.

Tim Verhoeven is a IT Professional in Belgium. He has four years of experience in Linux and Open Source software. He has worked at IBM for one year. He holds a degree in Electronics and Computer Science. His areas of expertise include Linux, Apache, Samba, WebSphere® Application Server, Java™ and J2EE, xSeries and zSeries® hardware and clustering. Tim participated extensively in the writing of *Deploying Apache on IBM @server BladeCenter*, REDP3588.

David Green is a Staff Engineer at IBM in Research Triangle Park, North Carolina and works in BladeCenter ecosystem development. He worked on the development of the Layer 2-7 GbE Switch Module and the Optical Passthrough Module for BladeCenter. He holds a Bachelor of Science degree in Information Systems from UNC-Greensboro. His areas of expertise include IBM Eserver BladeCenter, Fibre Channel, SANS and networking.

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

Tamikia Barrow, Diane O’Shea, Gail Christensen, Cecilia Bardy, Jeanne Tucker
International Technical Support Organization, Raleigh Center

Stephen Hochstetler
International Technical Support Organization, Austin Center

Rob Sauerwalt, Global Brand Manager and Team Lead - IBM @server Marketing
IBM Raleigh

AmyFreeman, Mgr, Brand and ISV Communications, xSeries Linux Marketing
IBM Raleigh

ITSO PC-5009 project team: David Robertson, David L Brown, Lowell Davis, Thomas Ternau
IBM Raleigh

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 will team with IBM technical professionals, Business Partners and/or customers.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you will 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 papers to be as helpful as possible. Send us your comments about this Redpaper 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. HQ7 Building 662
  P.O. Box 12195
  Research Triangle Park, NC 27709-2195

Summary of changes

This section describes the technical changes made in this edition of the Redpaper and in previous editions. This edition may also include minor corrections and editorial changes that are not identified.

Summary of Changes
for Deploying Apache on IBM @server BladeCenter
as created or updated on November 20, 2003.

November 2003, Second Edition

This revision reflects the addition, deletion, or modification of new and changed information described below.

New information

- Added 1.2.7, “BladeCenter Layer 2-7 GbE Switch Module” on page 13
- Added Chapter 7, “A brief introduction to Layer 4-7 Switching” on page 99
- Added information to 1.2.4 “BladeCenter 1200W Power Supply Module”
- Added URLs to the bibliography

Changed information

- None at this time

1

Introduction to IBM eServer BladeCenter technology and its advantages

In this chapter, we will introduce the IBM @server BladeCenter. This introduction includes an overview of the BladeCenter technology and hardware for the chassis and the blades; we also look at blade management and the options that are currently available. We will also discuss the advantages of blade servers over other servers in the market, where the blade servers fit into the market and why the market is driven towards blade servers.

1.1 Introduction to blade server technology

Blade servers are a relatively new technology that has captured industry focus because of its modular design, which can reduce cost with a more efficient use of valuable floor space, and its simplified management, which can help to speed up such tasks as deploying, reprovisioning, updating and troubleshooting hundreds of blade servers. All this can be done remotely with one graphical console using IBM Director systems management tools. In addition, blade servers provide improved performance by doubling current rack density. By integrating your resources and sharing key components, you will not only reduce cost but also increase availability.

1.1.1 IBM eServer™ BladeCenter and BladeCenter HS20 features

- The IBM @server BladeCenter has the following features:
  - Rack-optimized, 7U modular design enclosure: holds up to 14 hot-swap BladeCenter HS20 blades with up to six enclosures in a 42U rack.
  - Contains a high availability MidPlane supporting the hot-swap of individual blades.
  - Two 1200-watt, hot-swap power modules and support for two optional 1200-watt power modules: provides redundancy and power for robust configurations. See guidelines in 1.2.4, “BladeCenter 1200W Power Supply Module” on page 9.
  - Two hot-swap 325 CFM blowers and thermal sensors throughout to monitor and alert you to over-temperature conditions.
  - Management module: lets you manage and control components in the enclosure.
  - Optional hot-swap redundant Ethernet and Fibre Channel switch modules (supports up to four network switch modules).
  - Control panel : contains USB port and status LEDs.
- The IBM @server BladeCenter HS20 features are as follows:
  - Choose between 2.0 or 2.4 GHz(1) Xeon processors with quad-pumped 400 MHz front-side bus (FSB) and full-speed 512 KB ECC L2 caches.
  - Standard 512 MB system memory with Chipkill ECC support: supports 8GB maximum, when 2GB memory module becomes available.
  - Dual Broadcom Gigabit Ethernet controllers with teaming and failover support.
  - Integrated service processor: monitors critical components on each blade for remote and local systems management.
  - ATA-100 IDE controller: economical interface for up to two optional 40 GB IDE HDDs.
  - SCSI expansion connector: supports optional storage unit containing an Ultra320 RAID 1 SCSI controller and backplane support for two hot-swap HDDs.

Each of these features will be discussed in greater detail later in the paper, as well as a few other important points.

1.2 Technical overview

In this section, we will look at each of the key components individually and explain their function within the IBM @server BladeCenter and the BladeCenter HS20.

1.2.1 BladeCenter chassis

The IBM @server BladeCenter is a 7U modular chassis capable of housing up to 14 blade servers. The BladeCenter chassis allows individual blades to share resources such as power, switch, management and blower modules. The front view of the BladeCenter chassis is shown below in Figure 1-1. We can see the fourteen slots which, in this example, has been populated by one blade server and thirteen Processor Blade Fillers. These Processor Blade Fillers are required if a slot is not populated by a blade server or the HS20 SCSI Expansion Option, which will be discussed later, to provide proper airflow and system cooling. The BladeCenter Media Tray is also shown; it is located at the top of the chassis above the blade slots. This is shown in greater detail in Figure 1-2 on page 4.

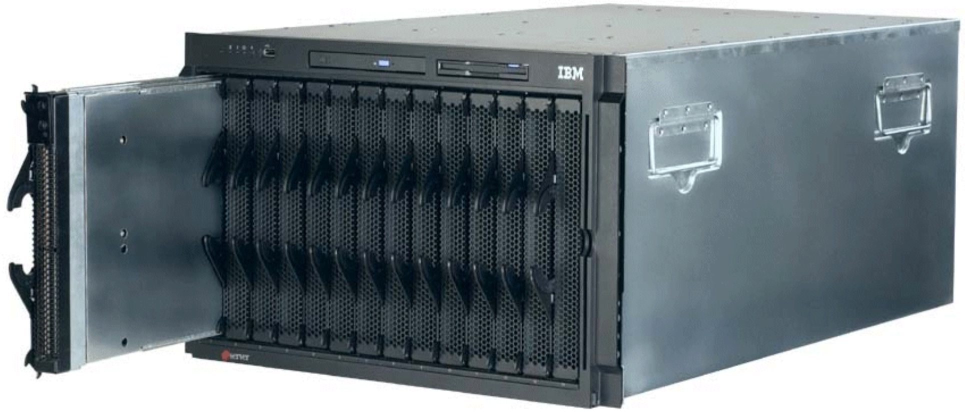


Figure 1-1 Front view of BladeCenter chassis

The Media Tray is a hot-pluggable unit that contains an interface card, CD-ROM and diskette drive. Mounted on the interface card is a USB 1.1 port and system information LEDs which provide status information for your IBM @server BladeCenter and BladeCenter HS20s.

There are five LEDs on the front panel. Figure 1-2 on page 4 illustrates the location of these LEDs. The five LEDs are:

Power This is a green LED which indicates the presence of power in the BladeCenter chassis. If this LED is not on, it could indicate one of the following conditions:

- There is no power to the chassis.
- The management module is not present or has failed.
- The LED has failed.

Location This LED is used to locate or identify the particular IBM @server BladeCenter. When on, this LED is bright blue and can be set to blink or remain constant. This LED will be on as a result of a request from the systems administrator via the Management Module or if a component requires maintenance.

Over- temperature This is an amber LED which will report any over-temperature conditions that occur either in the BladeCenter chassis or the blade servers. If an over-temperature condition occurs, the IBM @server BladeCenter may increase the speed of the blower to correct this, in which case the LED will automatically be turned off.

Information The information LED is also amber; this LED reports non-critical events. These events are recorded in the Error log. This LED must be manually switched off via the management module.

System Error Again, this is an amber LED which reports when a system error occurs. This LED reports errors such as a failure of a power, blower or switch modules. It will also be on if a system error occurs on a blade. The failed component's LED will also be on to help isolate the error.

These system LEDs are also located on the rear of the BladeCenter chassis under blower module 2.

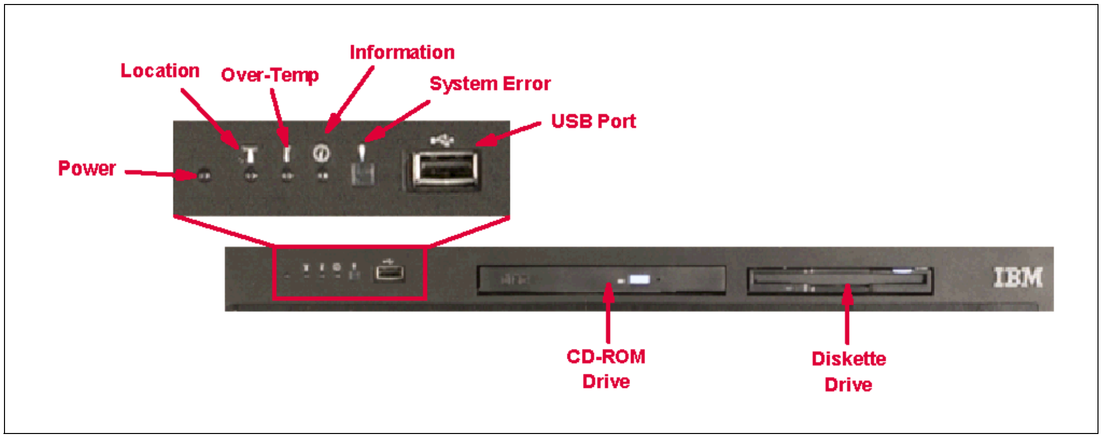


Figure 1-2 Media tray front view

As we discussed earlier, the BladeCenter chassis is capable of housing several hot swap and redundant components, such as:

- Power supply modules
- Switch modules
- Management modules
- Blower modules

In Figure 1-3 on page 5, you can see where each of these components should be located in the rear of the IBM @server BladeCenter.

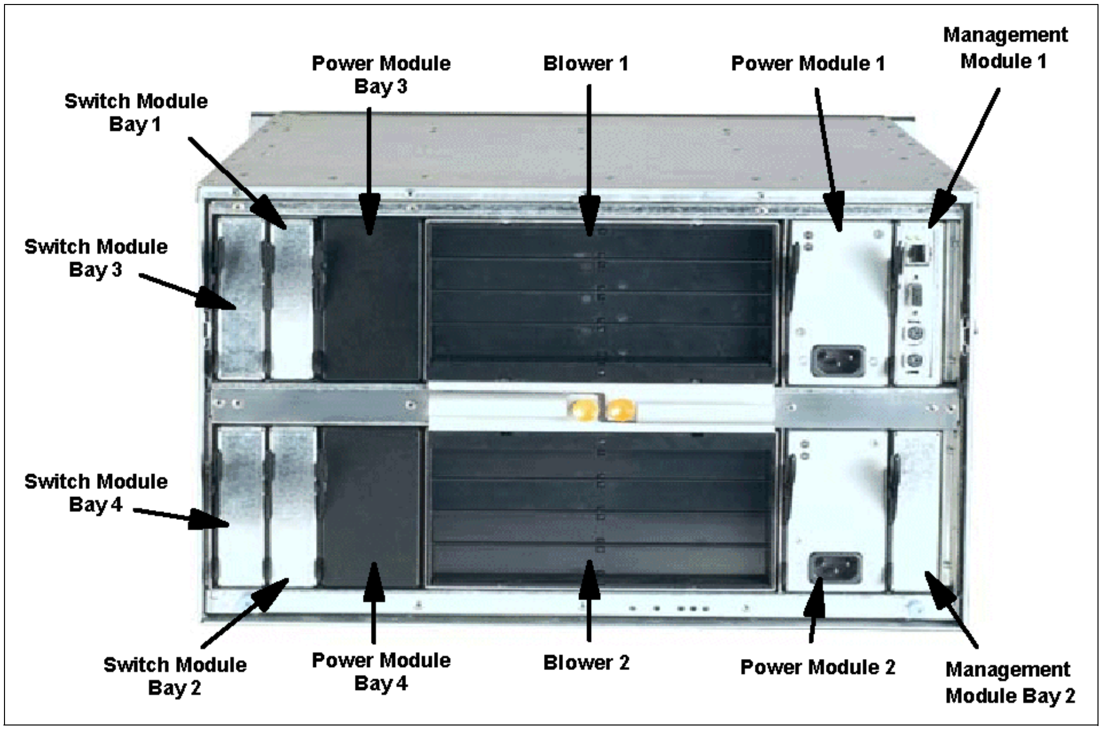


Figure 1-3 Rear view of BladeCenter chassis

The IBM @server BladeCenter automatically detects all blades and modules that are installed.

1.2.2 IBM eServer BladeCenter HS20

The IBM @server BladeCenter HS20 blades are high-throughput, two-way SMP-capable Xeon-based blade servers, highly scalable by adding memory and a second processor. Two Intel Xeon connectors are standard on the blade board to support installation of a second processor. High-speed, PC2100 DDR SDRAM is synchronized for up to 400 MHz processor-to-memory subsystem performance with current processors. There are four memory DIMM connectors; due to two-way interleaving, installation of memory options in pairs is required. Current memory options available are 256 MB, 512 MB and 1 GB size DIMMs, which support a minimum of 512 MB and a maximum of 4 GB of system memory. In the future, IBM will release a 2 GB memory option which will see the maximum system memory double from 4 GB to 8 GB.

Figure 1-4 on page 6 shows the BladeCenter HS20 with its top cover removed.

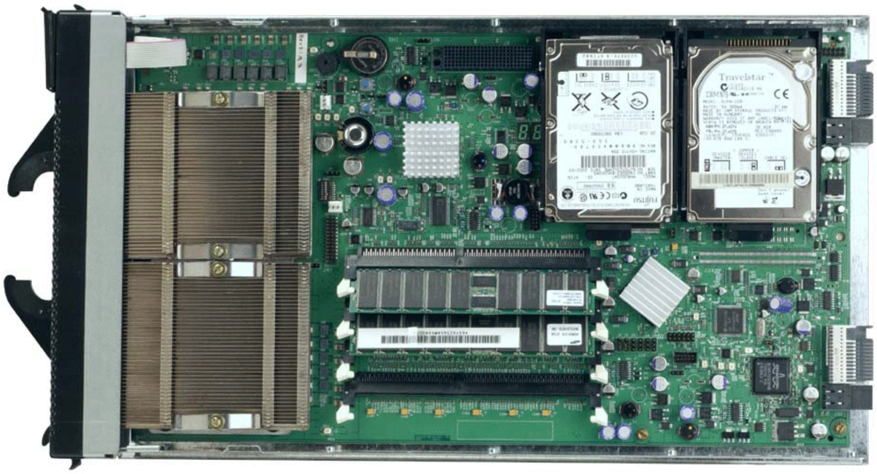


Figure 1-4 BladeCenter HS20 with top cover removed

Each BladeCenter HS20 has an integrated service processor on-board that communicates with the BladeCenter Management Module to enable blade server management remotely (see 1.2.3, “BladeCenter Management Module” on page 8 for more details on the management module). The service processor provides the following functions:

- Loader and OS Watch Timer
- Remote soft shutdown
- POST Watchdog Timer
- Light Path support
- VPD support
- PFA for VRM/CPU/Memory/HD
- Numeric based Error log
- ASM interconnect support (RS485)
- Environmental querying and alerts (TEMP/Voltages)
- Automatic Server Restart
- Remote Power On/Off
- In-band support for UMS/Director
- I2C interface to core logic (CSB5 chip)
- Local Environmental Monitoring
- Local LED control
- RS-485 interface to the Management Modules

The BladeCenter HS20 also has two integrated Ethernet controllers for redundancy. If redundancy is required, you must install Ethernet switch modules in switch module bays 1 and 2 (see 1.2.6, “BladeCenter 4-Port Ethernet Switch Module” on page 12 for more details).

Each controller is auto-sensing and will connect at the appropriate rate, even if the transfer rate is 10 Mbps, 100 Mbps or 1000 Mbps. The controller will also set the appropriate duplex state.

The Ethernet controller is capable of providing several teaming options that increase throughput and fault tolerance. In your blade server, a team consists of the two Ethernet controllers to utilize the options below:

Adapter fault tolerance (AFT) Provides automatic redundancy for your Ethernet controllers. You can configure either one of the integrated Ethernet controllers as the primary Ethernet controller. If the primary link fails, the secondary controller takes over. When the primary link is restored to an operational state, the Ethernet traffic switches back to the primary Ethernet controller.

Adaptive load balancing (ALB) Enables you to balance the transmission data flow among the two controllers. ALB also includes the AFT option. You can use ALB with any 100BASE-TX/1000BASE-T switch.

Cisco Fast EtherChannel (FEC) Creates a team of two controllers to increase transmission and reception throughput. FEC also includes the AFT option. You can use FEC only with a switch that has FEC capability.

The BladeCenter HS20 has a control panel which is located at the top of the blade servers, as shown in Figure 1-5.

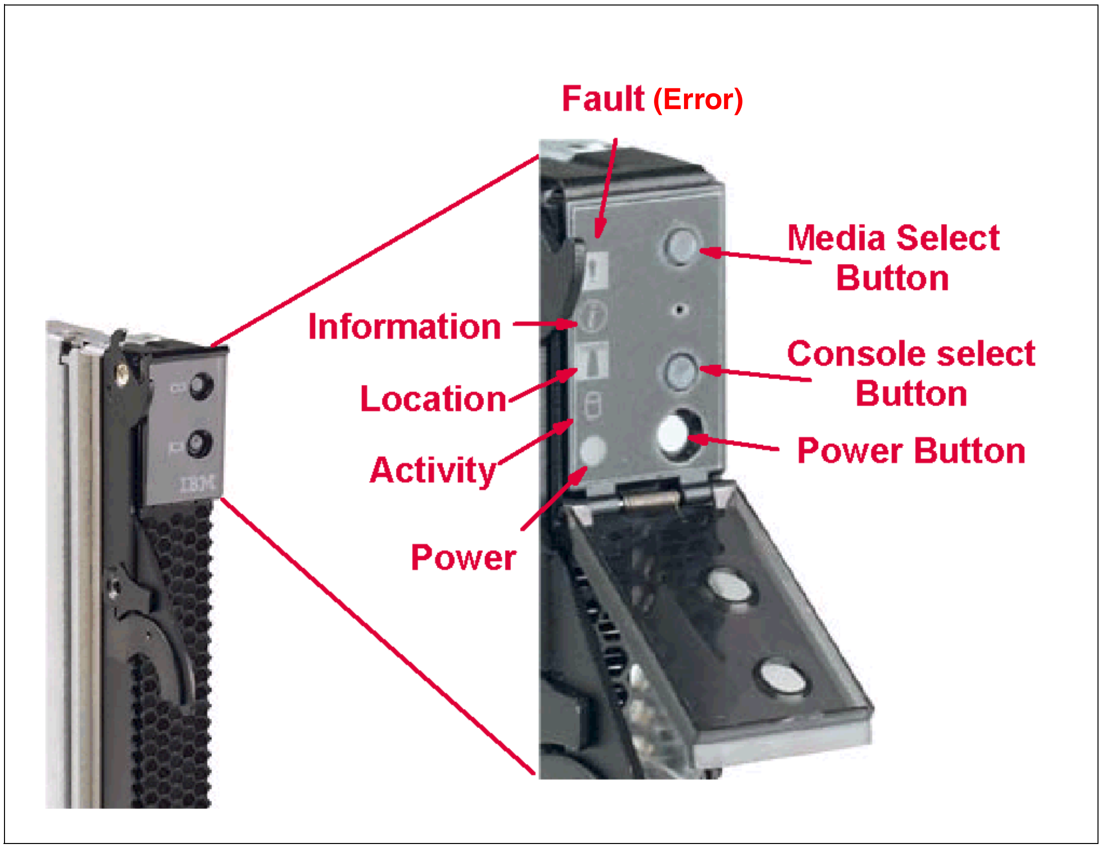


Figure 1-5 Blade server operators panel

Like the IBM @server BladeCenter's media tray, this control panel also has system information LEDs. The only difference is that the blade's panel also has control switches, which are detailed below.

Media-select button Press this button to associate the CD-ROM drive, diskette drive, and USB port with this blade server. This button lights when the ownership of the CD-ROM drive, diskette drive, and USB port transfers to this blade server.

Blade-error This LED is also known as the blade system-error LED. When this amber LED is on, it indicates a system error has occurred in the blade.

Information When this amber LED is on, it indicates information about a system error for this server has been placed in the BladeCenter System Error log.

Location This blue LED is turned on in response to a programmed condition, or remotely by the system administrator, to aid in blade identification for maintenance. The location LED on the IBM @server BladeCenter will be on also. Turn off the location LED after maintenance is complete.

Activity When this green LED is on, it indicates that there is activity in the blade server; this includes hard disk and network activity.

Power-on While the IBM @server BladeCenter has AC power, this green LED turns on and stays on when you turn on your blade server.

Console select button Press this button to associate the keyboard, mouse, and video ports with this blade server. This button lights when the ownership of the keyboard, mouse, and video transfers to this blade server.

Power-control button This button is located behind the control panel door. Press this button to manually turn the blade server on or off.

Note: The power-control button only has effect if the local power control option is enabled via the Management Module.

1.2.3 BladeCenter Management Module

The BladeCenter Management Module's primary function is to provide systems management for your IBM @server BladeCenter and blade servers, but it does have other important functions such as multiplexing the keyboard/video/mouse (KVM) to provide a local console for the individual blade servers and configuring the BladeCenter unit and switch modules. The management module communicates with all of the key components of the IBM @server BladeCenter including the switch, power and blower modules, as well as the blade servers themselves. The management module detects the presence, absence and condition of each of these components. A picture of the BladeCenter Management Module is shown in Figure 1-6 on page 9.

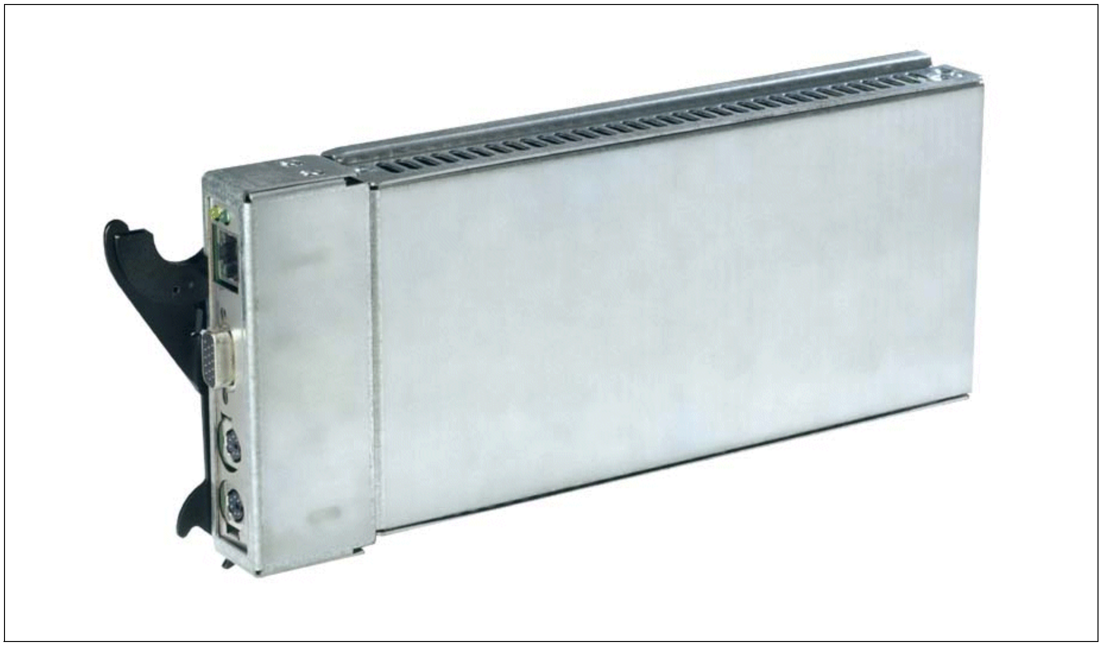


Figure 1-6 BladeCenter Management Module

The Management Module has a standard RJ45 connector for a 10/100MB Ethernet remote console connection, as well as two PS/2® connectors for keyboard, mouse and a 15-pin D-shell connector for video which are provided for the local console. Although the connectors for the keyboard and mouse are PS/2 type connectors, these devices are routed to a USB bus, enabling them to be switched between blades.

Note: The operating system in each blade must provide USB support for the blade server to recognize and make use of the keyboard and mouse.

The Management Module will retrieve and monitor critical information about the chassis and blade servers such as temperature, voltages, power supply, memory, fan and HDD status; this information will then be fed into an error and status log.

The manageability functions of the IBM @server BladeCenter are accessible via a Web GUI that is contained in the management module. This GUI allows you to view the status of, and control each blade server, which includes shutting down and restarting.

1.2.4 BladeCenter 1200W Power Supply Module

The standard BladeCenter chassis will ship with two 1200W Power Supply Modules, but, depending on the configuration of your IBM @server BladeCenter, you may require all four power modules. The standard two power modules provide power for the following components:

- Blade slots 1 through 6
- Blowers
- Management modules
- Switch modules
- Media tray

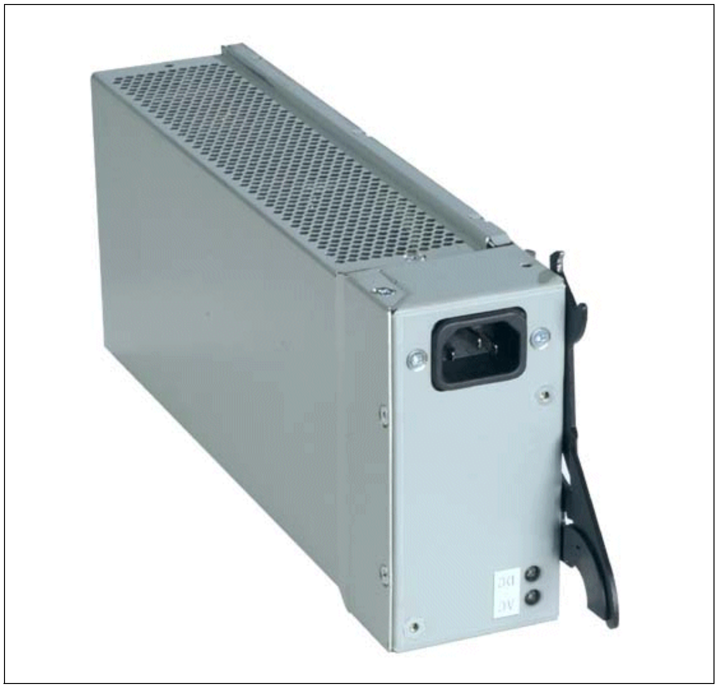


Figure 1-7 BladeCenter 1200W Power Supply Module

Power Modules 3 and 4 are required to provide power to blade slots 7 to 14. Figure 1-8 on page 11 shows how power is distributed by each power module. One power module is capable of providing enough power in the event of a power module failure. Power module 2 provides redundancy for power module 1 and power module 4 does the same for power module 3, although these power modules will effectively share the load under normal operating conditions. Supported configurations require either two or four power supplies, which is why when you order the optional BladeCenter 1200W Power Supply Module kit, you will receive two of the power modules.

Important: Nonredundant power is not supported in BladeCenter products. Power modules must always be present in power bays 1 and 2. When any blade server or option is in blade bay 7 through 14, power modules must be present in power bays 1 and 2, as well as in power bays 3 and 4. If a power module fails or an ac power failure occurs, BladeCenter units configured for redundant power operation, as described in this document, will operate in a nonredundant mode, and the blower modules will run at full speed. You must replace the failing power module or restore ac power as soon as possible to regain redundant power operation and to reset the blower modules to their normal operating speed.

BladeCenter power module upgrade guidelines

This section contains information that will help you determine whether you need to upgrade the power modules in your IBM @server BladeCenter unit when installing IBM @server BladeCenter HS20 blade servers.

As of the date of this printing, three BladeCenter power-module options are available: IBM BladeCenter 1200W Power Supply Module (part number 48P7052), IBM BladeCenter 1200W to 1400W Power Supply Upgrade Kit (part number 90P0197), and IBM BladeCenter 1800W Power Supply Module (part number 13N0570). Go to `http://www.ibm.com/pc/compat/` for information about ordering these options. Obtain and use the Technical Update with your BladeCenter and blade server documentation for future reference.

The Technical Update can be obtained from the following URLs:

```
http://www-1.ibm.com/support/docview.wss?uid=psg1MIGR-53353
ftp://ftp.software.ibm.com/pc/pccbbs/pc_servers_pdf/13n0308.pdf
```

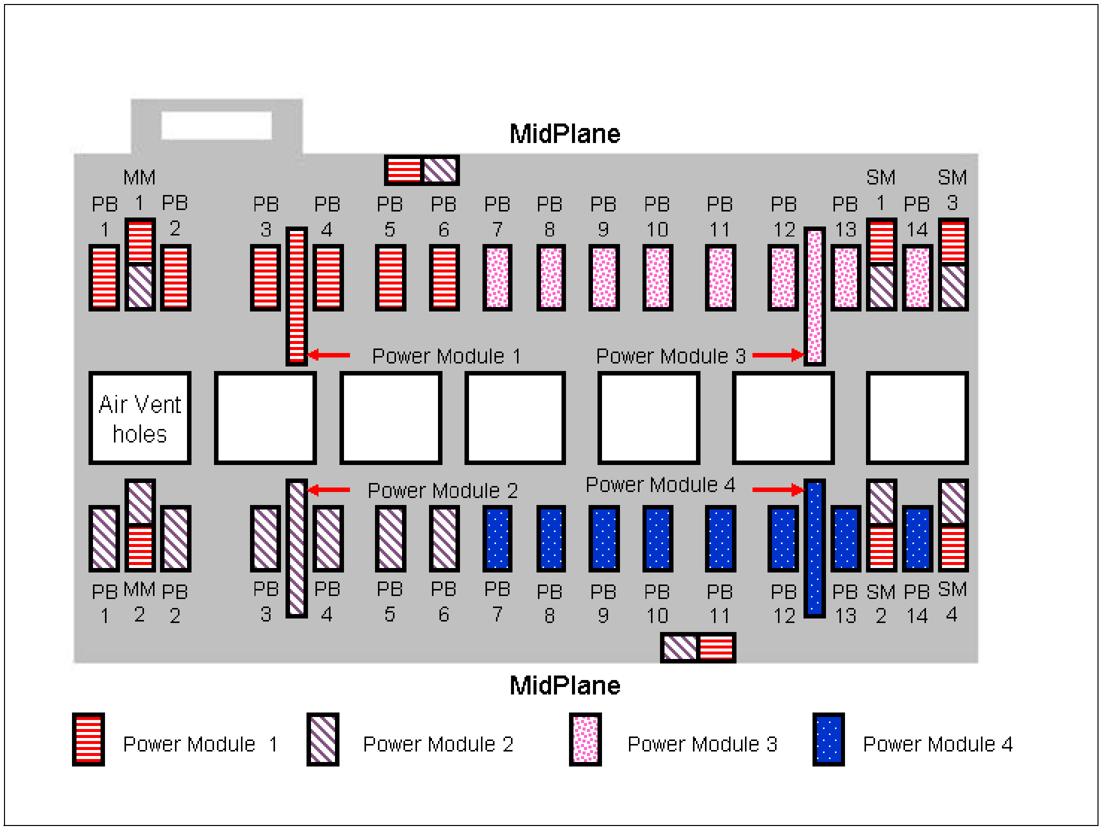


Figure 1-8 MidPlane power connector

1.2.5 BladeCenter blowers

The IBM @server BladeCenter ships with both hot-swap blowers which are required to provide adequate cooling; these blowers provide a total airflow of approximately 325 CFM; however, each blower will run at approximately 50% under normal conditions. If one blower fails, the other blower is capable of providing enough cooling for the entire chassis. However, the failed blower should be replaced within 48 hours. The speed of the blowers is controlled via the Management Module which receives signals from thermal monitors located in critical locations. There are two scenarios which will cause both blowers to function at full speed:

- The management module fails and the redundant module is not present.
- One of the power supply modules fails.

In Figure 1-9 on page 12, you can see that there are four fins which are currently closed since the blower is not operational. When operational, the blower draws the air from the front to the rear. Each blower also has an LED which will light up in the event of an error.

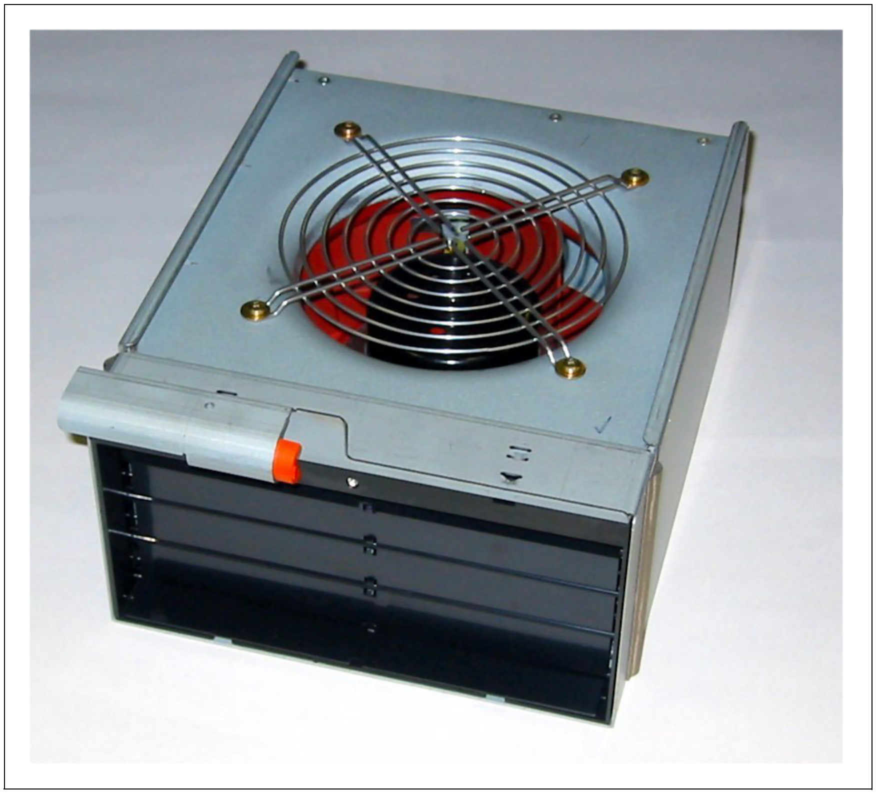


Figure 1-9 BladeCenter Blower Module

1.2.6 BladeCenter 4-Port Ethernet Switch Module

The Ethernet Switch Module (Figure 1-10 on page 13) has several purposes; in addition to providing network connectivity for the IBM @server BladeCenter and blades, it also provide interconnectivity between the blades and management modules. The Ethernet switch module does not come standard with the IBM @server BladeCenter; it is available for purchase as an option but it is required in most cases and should be installed in switch bay 1. This module is a fully functional Ethernet switch which has four 1000 Mbps ports externally, two internal 10/100 Mbps links to the management modules and fourteen 1000 Mbps links to the blades, which are also internal. Two of these switch modules can be added for redundancy. The switch module is configured via the Management Module.

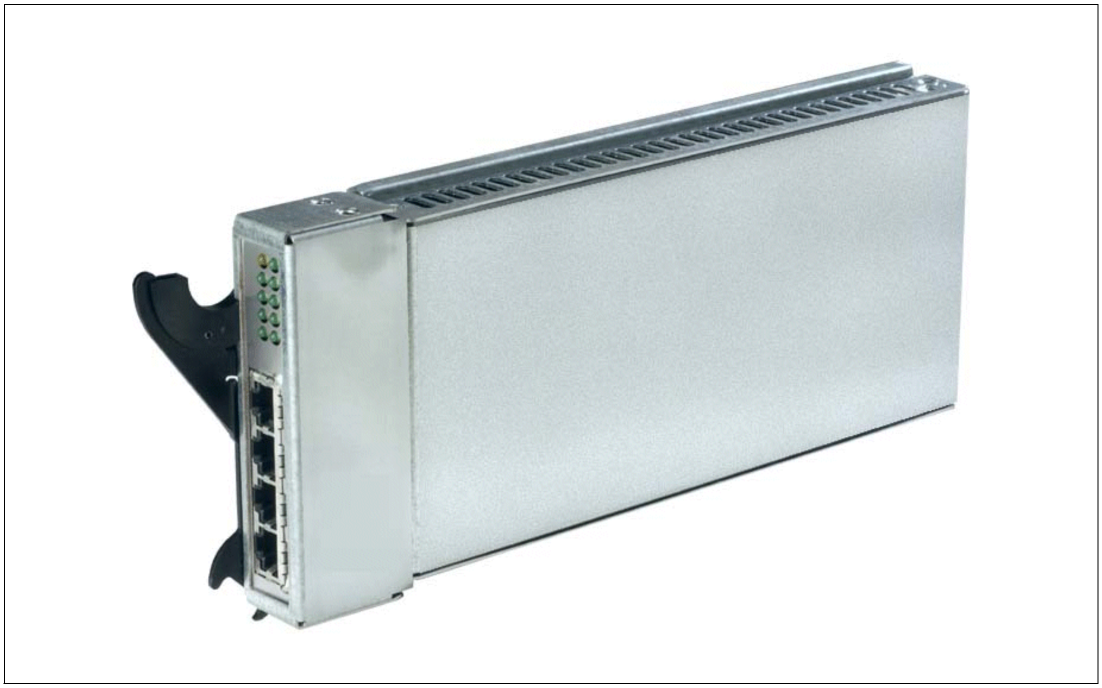


Figure 1-10 BladeCenter 4-Port Ethernet Switch Module

In Figure 1-10, the two LEDs at the top of the switch module indicate power-on and Ethernet switch errors. There are also LEDs next to each port which indicate Ethernet link and activity.

1.2.7 BladeCenter Layer 2-7 GbE Switch Module

The Ethernet switch module (Figure 1-11 on page 14) has several purposes. In addition to providing network connectivity for the BladeCenter unit and blade servers, it also provides interconnectivity between the blades and management modules. The Layer 2-7 GbE Switch Module does not come standard with the IBM @server BladeCenter. It is available for purchase as an option, but it is required in most cases and should be installed in switch bay 1. This module is a fully functional Ethernet switch that has four 1000 Mbps ports externally, two internal 10/100 Mbps links to the management modules and 14 1000 Mbps links to the blades, which are also internal. Two of these switch modules can be added for redundancy. The switch module is configured via the Management Module. For a more in-depth look at Layer 2-7 Switching, refer to the Redpaper *IBM @server BladeCenter Layer 2-7 Network Switching*, REDP3755.

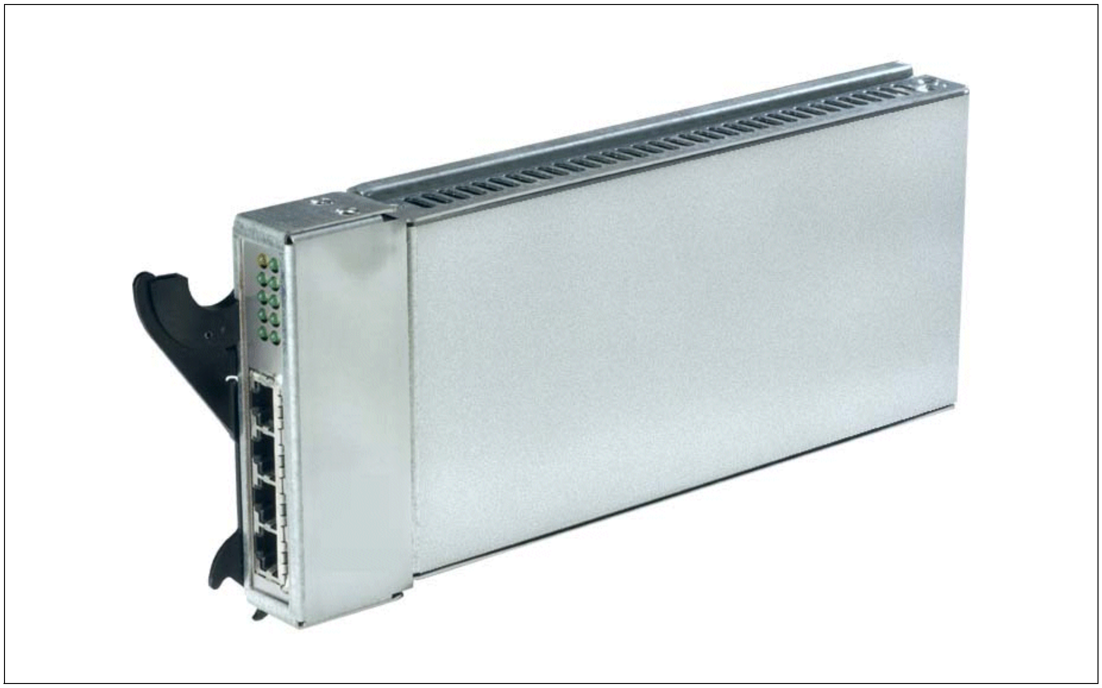


Figure 1-11 BladeCenter 4-port Ethernet Switch Module

In Figure 1-11, the two LEDs at the top of the switch module indicate power-on and Ethernet switch error. There are also LEDs next to each port that indicate Ethernet link and activity.

1.2.8 BladeCenter 2-Port Fibre Channel Switch Module

If you require Fibre Channel connectivity for your IBM @server BladeCenter, there are two things you are required to do:

1. Install the IBM HS20 Fibre Channel Expansion Card in each blade that requires an FC connection.
2. Install one BladeCenter 2-Port Fibre Channel Switch Module (Figure 1-12 on page 15) in switch bay 3. If redundancy is required, a FC switch module must be installed in switch bay 4.

Each port on the FC switch module is capable of supporting transmission speeds of either 1 or 2 Gbps and must auto-negotiate to match the speed of any connected devices. A Small Form factor Pluggable (SFP) transceiver is required to populate these ports; these SFP transceivers are hot pluggable. The FC switch module is also managed through the Management Module.

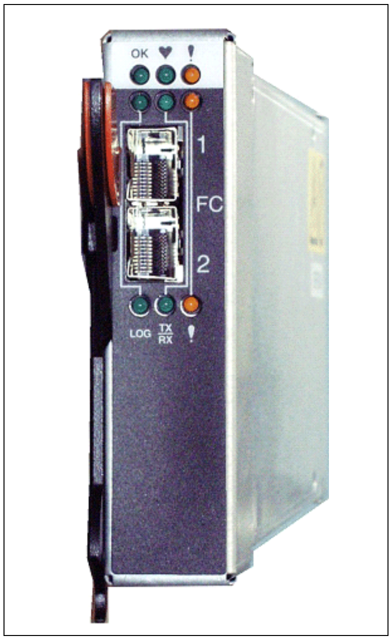


Figure 1-12 BladeCenter 2-Port Fibre Channel Switch Module

The storage options for the IBM @server BladeCenter are covered in Chapter 3, “Storage options for the BladeCenter” of the Redpaper *The Cutting Edge: IBM @server BladeCenter*, REDP3581.

1.2.9 BladeCenter Acoustic Attenuation Module

In environments where it is important to minimize sound emissions, there is a noise reduction option available that can be installed on the rear of the IBM @server BladeCenter. This option is called the BladeCenter Acoustic Attenuation Module (acoustic module or muffler). The acoustical noise emissions for a single BladeCenter unit configured with 14 IBM @server BladeCenter HS20 servers are:

- 7.4 bels (operating)
- 7.4 bels (idling)

The acoustical noise emissions for a single BladeCenter unit with a BladeCenter Acoustic Attenuation Module option installed and configured with 14 IBM @server BladeCenter HS20 servers are:

- 6.9 bels (operating)
- 6.9 bels (idling)

For visibility purposes, the system LEDs on the rear of the BladeCenter unit have been replicated on the acoustic module, as these are covered when this option is fitted. The acoustic module also fully covers the blower modules and partly covers the other modules installed.

Note: The BladeCenter Acoustic Attenuation Module requires approximately eight inches between the rear of the IBM @server BladeCenter and the rack door. For this reason, the NetBay 42 Enterprise Rack Cabinet should be used when installing the IBM @server BladeCenter with this option.

1.3 The advantages of IBM eServer BladeCenter

Unlike typical server architecture which scales up, the BladeCenter allows for the scale out approach, yet balances performance and density. In this section, we identify the advantages of the BladeCenter for your business, such as:

- Systems management
- High density computing
- Redundancy

1.3.1 Systems management

The systems management component for the BladeCenter is the combination of IBM Director, the Web interface and Rapid Deployment Manager (RDM). With these tools, you can reduce system outages, increase IT personnel productivity and reduce support costs.

IBM Director

IBM Director is a comprehensive workgroup hardware manager designed for use with IBM @server xSeries servers, PCs, notebooks and now IBM @server BladeCenter.

IBM Director provides a comprehensive view of the xSeries servers.

- Remote access to the server regardless of its status
- Resetting or cycling of the server
- Server inventory, AssetID, blade slot numbers utilized
- Monitoring and setting thresholds on the server events, including PFA
- Flash BIOS
- Monitoring and identifying potential performance bottlenecks

IBM Director allows you to reduce system outages, increase IT personnel productivity and reduce support costs.

These goals can be accomplished by:

- Monitoring server health
- PFA - Predictive Failure Analysis®
- Integration with enterprise system management environments

Monitoring server health

By setting thresholds on components within IBM Director, you can monitor the following:

- Operating system load
- POST time-out
- Voltage
- Temperature

PFA - Predictive Failure Analysis

Setting alerts on disk, memory and processors via PFA allows you to identify imminent component failure. PFA can monitor the following:

- Processors
- Memory

- Fans
- Power supplies
- SCSI HDDs

Integration with enterprise system management environments

IBM Director agents should be installed on each of the blade servers as this allows system management of the blades and also provides the advanced management capabilities of the xSeries servers to be accessed from such products as:

- Tivoli® Enterprise™ and Tivoli Netview
- Computer Associates CA Unicenter TNG
- HP OpenView
- Microsoft® SMS
- BMC Patrol
- NetIQ

This is an important consideration for organizations who have already spent time, money and resources on existing system management tools.

Web interface

The Web interface allows you to configure and remotely manage your IBM @server BladeCenter via a browser; this is built into the management module.

Supported browsers and required components are:

- Microsoft Internet Explorer 4.0 Service Pack 1, or later
- Netscape Navigator 4.72, or later (V6.0 is not supported)
- Java-enabled Web browser
- Support for JaveScript 1.2 or later
- HTTP 1.0, or later
- Minimum display resolution 800*600 pixels and 256 colors

Note: The Web interface does not support double-byte character set languages (DBCS).

The Management Module Ethernet port either receives a DHCP assigned address within two minutes of connecting to the LAN or it defaults to the following IP address 192.168.70.125 with a subnet address of 255.255.255.0. The default hostname is MMxxxxxxxxxxxx where xxxxxxxxxxxx is the MAC address.

Note: If multiple IBM @server BladeCenters are to be configured, only one can be assigned the default IP address of 92.168.70.125, otherwise IP address conflicts will occur.

If your DHCP server is not on the same subnet as the BladeCenter Management Module console, DHCP traffic needs to be allowed to traverse the router to the Management Module console to receive an DHCP assigned address, or IBM Director Server needs to be on the same subnet as the IBM @server BladeCenter.

The Web interface allows you to manage and check the status of each of the modules and blade servers. Below is a list of the functions and sub-functions available.

- Monitor
  - System status
  - Event log
  - Vital product data
- Blade Tasks
  - Power/restart
  - Firmware update
  - Configuration
- Switch tasks
  - Power/restart
  - Management
- MM console (Management Module)
  - General settings
  - Login profiles
  - Alerts
  - Network interface
  - Network protocols
  - Security
  - Configuration file
  - Fireware update
  - Restore defaults
  - Reset MM
- Log off

RDM - Rapid Deployment Manager

This powerful tool allows you to deploy system images to the blade servers from the administrator's console.

RDM supports the following environments:

- Windows® 2000
- Windows 2000 Advanced Server

The advantages of Rapid Deployment Manager are:

- Rapid deployment of Operating System image to the destination server
- Hot-spare blade option: “Hot-spare blades” on page 22.

1.3.2 High density computing

As data centers have grown with the proliferation of Wintel servers over recent years, it is important to note that rack space and floor space can be more efficiently taken up with the use of blade servers. A fully populated 42U rack is capable of holding six IBM @server BladeCenters and 84 dual processor blade servers, for a total of 168 CPUs per rack. This is twice the current density of a non-blade server. The IBM @server BladeCenter supports a minimum of one 4 port 1 GB Ethernet switch for up to fourteen blade servers. A total of four switch modules can be utilized within the IBM @server BladeCenter. This can be a combination of either Fiber Channel or Ethernet.

We recommend the use of blade servers for the following situations:

- Space constrained environments
- WebSphere
- Linux clusters
- Web caching
- Collaboration applications (Lotus Notes®, Microsoft Exchange and Citrix)
- Dynamic Web serving - Load balancing
- Firewall
- Telecommunications
- Active directory services
- Scientific and technical computing

These applications are typically processor and memory intensive, and so lend themselves to the scale out option rather then the scale up option. The scale out option is where the strength of the IBM @server BladeCenter becomes obvious. Due to the highly scalable range of components available with the BladeCenter unit, the blade server has a huge market.

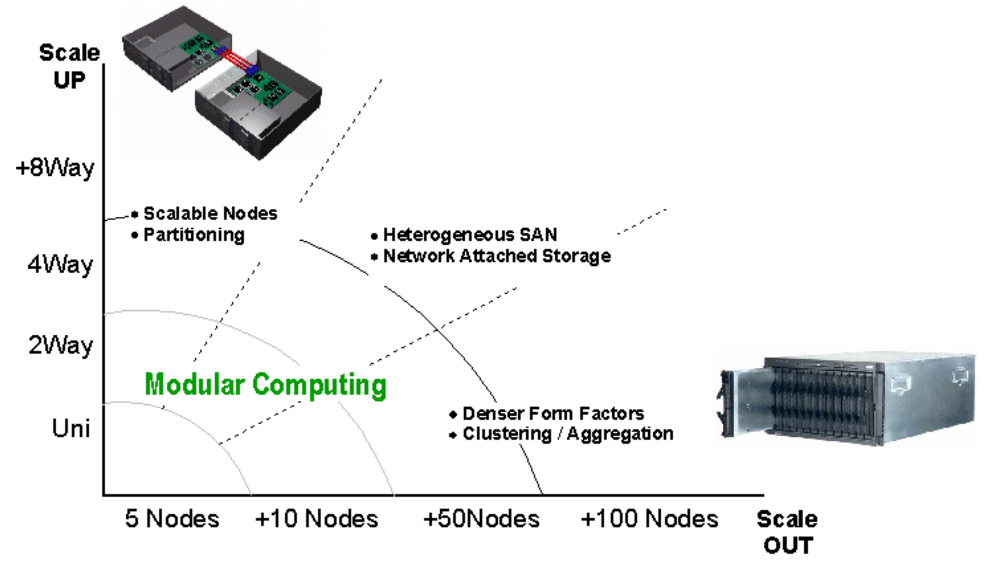


Figure 1-13 Scale up versus scale out

Attractive cost savings are also possible where a large number of rack installed servers are required. These savings run from power usage, to hardware procurement (due to not duplicating components), through to server management. Table 1-1 on page 20 demonstrates the input voltage required for different servers providing the same number of

CPUs. It also shows the heat output in BTUs, which can affect the cooling of your computer room and other equipment within the computer room.

Table 1-1 Power and BTU usage

	Number of Servers	Processor	Maximum load watts	BTU
IBM x330 4MX	14	Pentium® III 1.4GHz	3080	10502
IBM x335	14	Xeon 2.4GHz	4760	16231
IBM x342 - Dual Power	14	Pentium III 1.4GHz	5250	19821
IBM x345 - Dual Power	14	Xeon 2.4GHz	7000	23870
IBM @server **BladeCenter**	14	Xeon 2.4GHz	2500	8530

Note: The IBM @server BladeCenter in Table 1-1 is utilizing the onboard IDE controller.

Deploying an IBM @server BladeCenter is far simpler than deploying 14 1U servers and the associated cables. A single IBM @server BladeCenter only requires a KVM cable, one power cable, one Ethernet cable and a single system management cable. Even with a redundancy of these components, cabling can be reduced by eighty-three percent (see Table 1-2 for a typical example).

Table 1-2 Cable utilization

	1U Solution (2 - 42U Racks)	IBM @server BladeCenter Solution (2 - 42U Racks)
Power	84	12
Ethernet	168	48
KVM	84	6 (ACT)
Switch	6 (boxes)	1 (ACT)
KVM interconnect	5	0
Management connect	84	6
Total	431	73

The IBM @server BladeCenter is not intended to replace any of the xSeries suite of products, but rather to provide additional configuration options.

1.3.3 Redundancy

We all know failures occur occasionally; we also know that redundancy of key components increases system availability. Although previously, it was expensive to purchase redundant options for individual servers, the high availability features found in conventional xSeries rack servers have also been included within IBM @server BladeCenter. These features include:

- Hot-spare blades
- Processor

- Memory protection
- Disk mirroring
- MidPlane
- Hot-swap power and cooling modules
- Switch modules

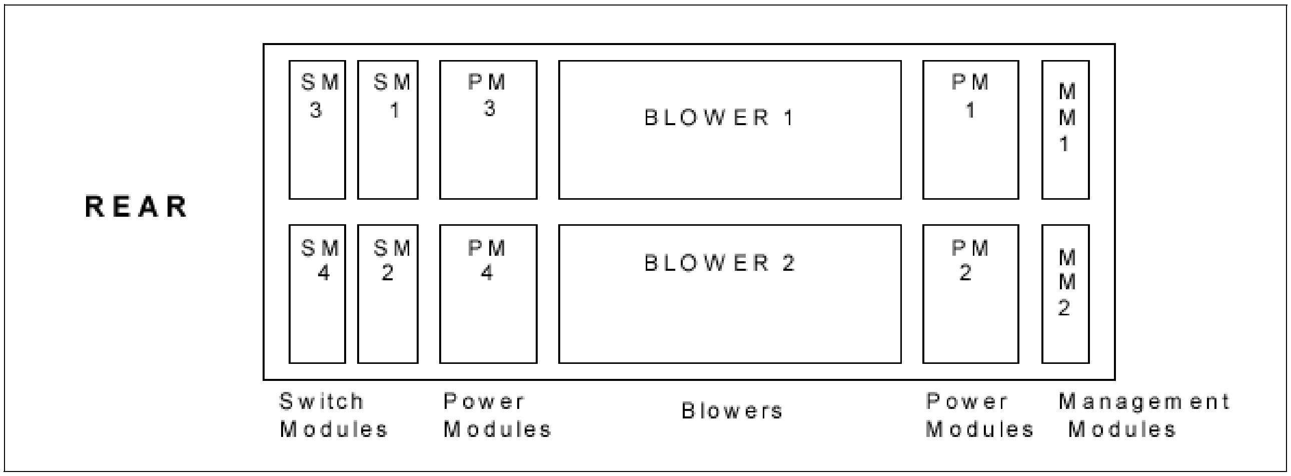


Figure 1-14 Module location on the rear of the IBM eserver BladeCenter

Table 1-3 on page 22 outlines which components can be redundant, hot swap, PFA or have Light Path diagnostics.

Table 1-3 Redundant, Hot-swap, PFA or Light Path

	Redundant	Hot Swap	PFA	Light Path
Blade	Yes	Yes	No	Yes
Processor	Auto Recovery	No	Yes	Yes
Memory	ECC w/Auto down size	No/Tool less	Yes	Yes
Power Supplies	Yes	Yes	Yes	Yes
Cooling Fans	Yes	Yes	Yes	Yes
Management Module	TBD	TBD	TBD	TBD
Switch Module	Yes	Yes	TBD	TBD
Backplane	Yes	No	No	No
Front Plane /Media Tray	No	Yes	No	No
CD ROM Drive	No	No	No	No
Floppy Drive	No	No	No	No
Hard Drive	Raid	Only with Blade Storage expansion (HS SCSI drives)	Yes	Yes
Power Cable	Yes	Yes	No	No

Hot-spare blades

MidPlane supports hot-spare blade servers; this operates in the same fashion as the hot-spare drive. By creating events within IBM Director, you can deploy an operating system to a hot-spare blade server automatically.

Processor

In a dual processor blade server, if a CPU fails, the following steps are taken by the system.

1. Force failed processor offline
2. Automatically re-boot server
3. Generate alerts
4. Continue operating with the working processor

Memory

There are four memory slots; memory must be installed in matching pairs. The following redundancy options are available:

- Chipkill™ ECC - provides correction for up to 4 bits per DIMM
- Memory hardware scrubbing - corrects soft memory errors automatically
- PFA - Creates alerts of imminent failure

Disk mirroring

Refer to Chapter 3, “Storage options for the BladeCenter” of the Redpaper *The Cutting Edge: IBM @server BladeCenter*, REDP3581 for hardware and software mirroring options of IDE and SCSI disks.

MidPlane

The middle plane provides connectivity between the blades and the modules at the rear of the BladeCenter unit. There are two connections on each blade server to independent middle planes for redundancy.

Hot-swap power and cooling modules

By sharing fans, power supplies, cables and other components within a BladeCenter unit and installing the redundant options for these, your organization can reduce the number of points of potential failure, thus increasing system availability.

Blowers

Two hot-swap blowers are standard in the IBM @server BladeCenter; the blower speed varies depending on the temperature. A failed blower needs to be replaced within 48 hours.

Note: An Acoustic Attenuation Module can be fitted to reduce noise.

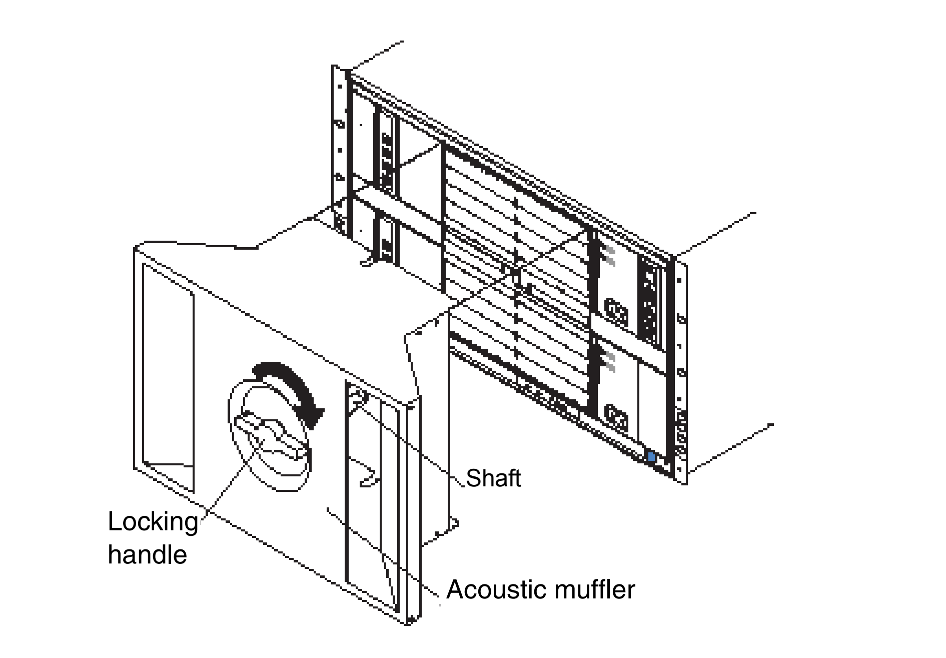


Figure 1-15 Acoustic Attenuation Module

Power

The IBM @server BladeCenter comes with two 220 volt 1200 watt hot-swap power modules in power bays 1 and 2. Table 1-4 on page 24 outlines the power module bays and their functions.

Table 1-4 Power module bays

Power module bays	Power module function
1	Provides power to all the BladeCenter modules in server bay slots 1-6.
2	Redundancy for Power module bay 1.
3	Provides power to all the BladeCenter modules in server bay slots 7-14.
4	Redundancy for Power module bay 3.

Note: A blade server in bay 6 with a SCSI storage expansion option requires a power module in power module bays 3 and 4 to support the SCSI storage expansion.

Cabling modules

The IBM @server BladeCenter is capable of managing four switch modules of either Ethernet or Fibre, the minimum requirement being a single Ethernet switch.

Ethernet

Two hot-swap 1 GB Ethernet four port switch models can be installed in switch module bays 1 and 2.

Fibre

Two hot-swap Fibre channel network interface switch modules can be placed in switch module bays 3 and 4.

Note: The IBM @server BladeCenter also includes a hot-swap media tray, which includes the CD-ROM and floppy drive.

2

The Apache HTTP Server project

The Apache Web server is probably the most popular project of the Apache Software Foundation. It started out in 1995 when a group of webmasters came together and created an updated version of the NCSA HTTP daemon, a public domain HTTP server that has not seen any new development since 1994. The first public release of the Apache HTTP Server was based on V1.3 of the NCSA HTTP daemon plus a set of patches from the Apache group.

The popularity of the Apache HTTP Server started to rise and since April of 1996 the Apache HTTP Server is the most popular Web server on the Internet. It is available for a multitude of platforms and because of its Open Source nature it is also used in other products. For example, IBM’s HTTP Server is based on the Apache code. Currently about 60% of all Web sites run a version of the Apache HTTP Server according to the NetCraft Web server survey; for details, we refer you to:

`http://www.netcraft.com/survey/`

In 2002, the Apache HTTP Server Project has taken a new step by releasing V2.0 of their Web server. By using a newly developed runtime layer Apache 2.0 is now capable of better using the features of the different platforms Apache runs on. For the complete list of new features in Apache 2.0, see:

`http://httpd.apache.org/docs-2.0/new_features_2_0.html`

2.1 Apache Software Foundation

The Apache Software Foundation is a non-profit corporation that has been created to provide organizational, legal, and financial support for the Apache Open Source software projects. The current list of Apache Software Foundation projects include:

- **HTTP Server**: commonly known as Apache httpd
- **APR**: the Apache Portable Runtime
- **Jakarta**: server-side Java
- **Perl**: dynamic Web sites using Perl
- **PHP**: server-side, HTML embedded scripting language
- **TCL**: dynamic Web sites using TCL
- **XML**: XML solutions focused on the Web
- **Conferences**: meetings of developers and users
- **Foundation**: administration and infrastructure management

For more information about the Apache Software Foundation and how you can contribute, visit the Apache Software Foundation Web site:

`http://www.apache.org/foundation`

2.2 Apache HTTP Server V1.3

Until the release of V2.0 of Apache, this was the latest stable version of Apache. Therefore, it is still the most used version of Apache to this date. Apache 1.3 is available for almost all UNIX® platforms and other platforms like Windows and Novell.

One of the great advantages of Apache is that you can use modules to extend the functionalities of your Apache Web server. The three most popular modules for Apache probably are:

- mod_ssl: an Open Source implementation of the SSL and TLS protocols that allow Apache to serve secured and encrypted Web pages.
- mod_perl: a Perl interpreter implemented within Apache that allows fast execution of Perl cgi's and extension of Apache with modules written in Perl.
- php: a scripting language that is especially suited for Web development and which can be embedded into your HTML files.

More Apache modules can be found at the Apache Module Registry at:

`http://modules.apache.org/`

Some of the key features of Apache are as follows:

- Implements the latest protocols, including HTTP/1.1 (RFC2068).
- Is highly configurable and extensible with third-party modules.
- Can be customized by writing "modules" using the Apache module API.
- Provides full source code and comes with an unrestrictive license.
- Runs on most versions of UNIX (including Linux) without modification.
- DBM databases for authentication, which allow you to easily set up password-protected pages with enormous numbers of authorized users, without bogging down the server. A

wide variety of SQL databases can be used for authentication too (using additional modules).

- Customized responses to errors and problems, which allow you to set up files, or even CGI scripts, which are returned by the server in response to errors and problems. For example, you can set up a script to intercept 500 server errors and perform on-the-fly diagnostics for both users and yourself.
- Multiple Directory Index directives, which allow you to "say" DirectoryIndex index.html index.cgi, which instructs the server to either send back index.html or run index.cgi when a directory URL is requested, whichever it finds in the directory.
- Unlimited numbers of aliases and redirect directives that may be declared in the config files.
- Content negotiation, the ability to automatically serve clients of varying sophistication and HTML level compliance, with documents that offer the best representation of information that the client is capable of accepting.
- Multi-homed servers, which allow the server to distinguish between requests made to different IP addresses (mapped to the same machine).

2.3 Apache HTTP Server V2.0

In April of 2002, the Apache Software Foundation announced the release of Apache V2.0.35 for general availability. Since then, Apache 2.0 has been the preferred version of Apache. The Apache 2.0 project was almost three year in the making before this general release.

The Apache 2.0 series contains a new set of features:

- UNIX threading

  On UNIX systems with POSIX threads support, Apache can now run in a hybrid multi process, multi threaded mode. This improves scalability for many, but not all configurations.
- New build system

  The build system has been rewritten from scratch to be based on `autoconf` and `libtool`. This makes Apache's configuration system more similar to that of other packages.
- Multiprotocol support

  Apache now has some of the infrastructure in place to support serving multiple protocols. mod_echo has been written as an example.
- Better support for non-UNIX platforms

  Apache 2.0 is faster and more stable on non-UNIX platforms such as BeOS, OS/2®, and Windows. With the introduction of platform-specific multi-processing modules (MPMs) and the Apache Portable Runtime (APR), these platforms are now implemented in their native API, avoiding the often buggy and poorly performing POSIX-emulation layers.
- New Apache API

  The API for modules has changed significantly for 2.0. Many of the module-ordering/-priority problems from 1.3 should be gone. 2.0 does much of this automatically, and module ordering is now done per-hook to allow more flexibility. Also, new calls have been added that provide additional module capabilities without patching the core Apache server.

- IPv6 support

  On systems where IPv6 is supported by the underlying Apache Portable Runtime library, Apache gets IPv6 listening sockets by default. Additionally, the `Listen`, `NameVirtualHost`, and `VirtualHost` directives support IPv6 numeric address strings (for instance, `Listen [fe80::1]:8080`).

- Filtering

  Apache modules may now be written as filters which act on the stream of content as it is delivered to or from the server. This allows, for example, the output of CGI scripts to be parsed for Server Side Include directives using the `INCLUDES` filter in mod_include. The module mod_ext_filter allows external programs to act as filters in much the same way that CGI programs can act as handlers.

- Multilanguage error responses

  Error response messages to the browser are now provided in several languages, using SSI documents. They may be customized by the administrator to achieve a consistent look and feel.

- Simplified configuration

  Many confusing directives have been simplified. The often confusing `Port` and `BindAddress` directives are gone; only the `Listen` directive is used for IP address binding; the `ServerName` directive specifies the server name and port number only for redirection and `vhost` recognition.

- Native Windows NT® Unicode support

  Apache 2.0 on Windows NT now uses utf-8 for all filename encodings. These directly translate to the underlying Unicode file system, providing multilanguage support for all Windows NT-based installations, including Windows 2000 and Windows XP. *This support does not extend to Windows 95, 98 or ME, which continue to use the machine's local codepage for filesystem access.*

- Regular Expression Library Updated

  Apache 2.0 includes the Perl Compatible Regular Expression Library (PCRE). All regular expression evaluation now uses the more powerful Perl 5 syntax.

- Integrated SSL and WebDAV support.

  mod_ssl and mod_dav are both new modules in Apache 2.0. mod_ssl is an interface to the SSL/TLS encryption protocols provided by OpenSSL. mod_dav implements the HTTP Distributed Authoring and Versioning (DAV) specification for posting and maintaining Web content.

- Improved HTTP proxy support.

  The proxy module has been completely rewritten to take advantage of the new filter infrastructure and to implement a more reliable, HTTP/1.1 compliant proxy. In addition, new `<Proxy>` configuration sections provide more readable (and internally faster) control of proxied sites; overloaded `<Directory "proxy:...">` configurations are not supported. The module is now divided into specific protocol support modules including proxy_connect, proxy_ftp and proxy_http.

3

Installation of Linux and IBM Director Agent

In this chapter, we will describe in detail the basic installation of Red Hat Linux 7.3, SuSE Linux Enterprise Server 8 (SLES 8) and IBM Director Agent on an IBM @server BladeCenter system.

3.1 Installation of Red Hat Linux 7.3

We will discuss three methods of installing Red Hat Linux on an IBM @server BladeCenter system:

- CD installation
- Network installation
- PXE boot installation

The only difference between these is the means used to start the install. Once the installation process is started, everything else is the same.

3.2 CD installation

This is the most direct and simplest method of installing Red Hat on a blade server. Since the HS20 Blade uses USB CD-ROM, you must make a special boot disk with USB support.

3.2.1 Creating a boot disk

Download the diskette image from:

http://people.redhat.com/msw/boot-usb-sleep-7.3.img

The diskette can be created on a Windows or Linux system. Using a Windows system, copy the rewrite utility for DOS on Red Hat Install CD 1 in \dosutils\ to the hard drive. In a command prompt, run **`rawrite`** `X:\images\boot.img a:` where `X` is the actual drive letter. Using a Linux system, run **`dd if=boot-usb-sleep-7.3.img of=/dev/fd0`**.

3.2.2 Installing Red Hat 7.3

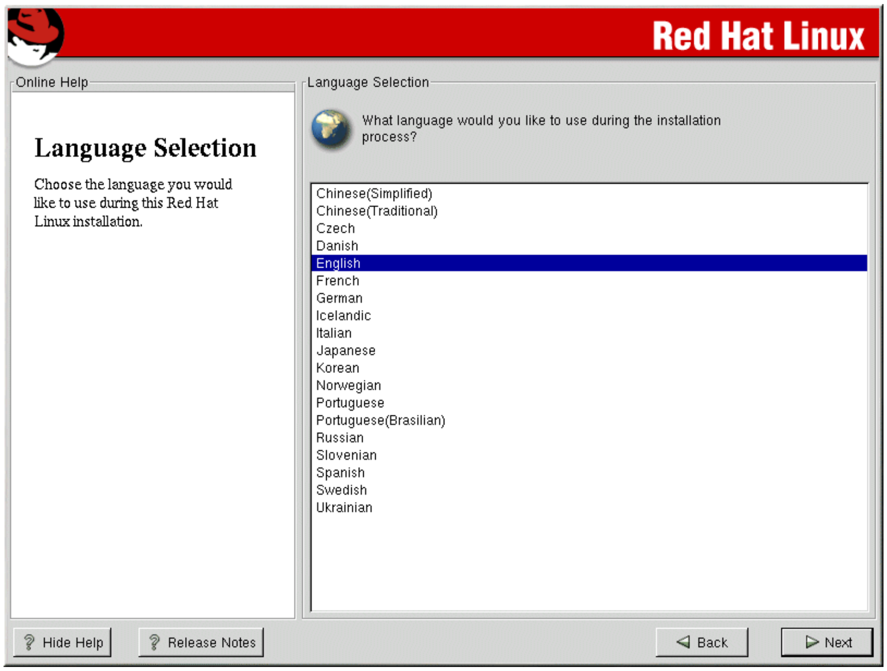
In this section, we will install Red Hat 7.3. Complete the following instructions:

1. Insert the boot diskette into the diskette drive and Red Hat Install CD 1 into the CD-ROM drive.
2. Power on the blade server and press the **Media Select** and **Console Select** buttons on the blade server.

.

Note: Do not switch the KVM from the blade server until the installation has proceeded to installing the packages after the About to Install window appears, otherwise the mouse will lose functionality.

3. At the Welcome to Red Hat Linux Version 7.3 window, press **Enter**.



Language Selection

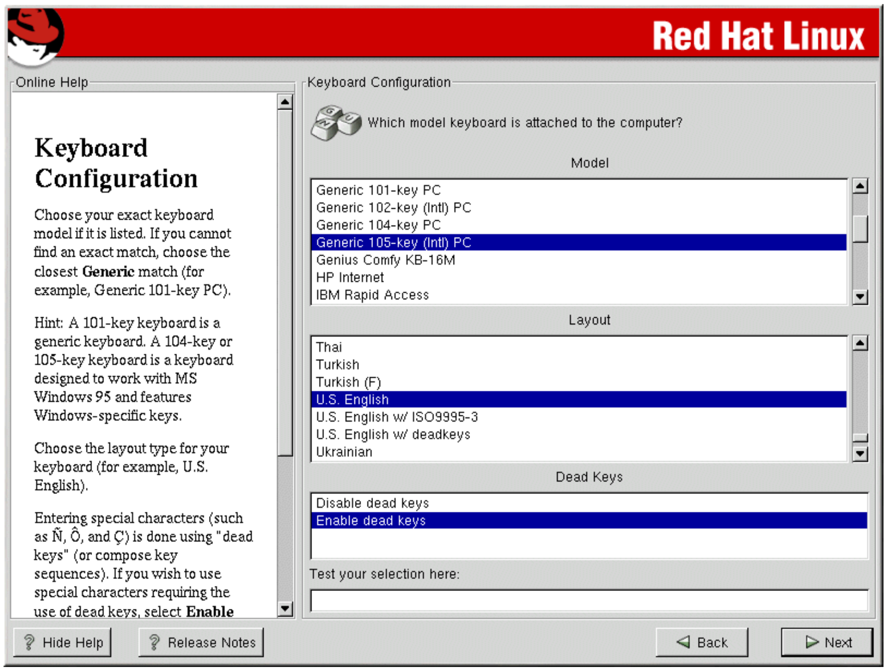
Choose the language you would like to use during this Red Hat Linux installation.

What language would you like to use during the installation process?

Chinese(Simplified)
Chinese(Traditional)
Czech
Danish
English
French
German
Icelandic
Italian
Japanese
Korean
Norwegian
Portuguese
Portuguese(Brasilian)
Russian
Slovenian
Spanish
Swedish
Ukrainian

Figure 3-1 Language selection

4. Select **English** and click **Next**.



Keyboard Configuration

Choose your exact keyboard model if it is listed. If you cannot find an exact match, choose the closest **Generic** match (for example, Generic 101-key PC).

Hint: A 101-key keyboard is a generic keyboard. A 104-key or 105-key keyboard is a keyboard designed to work with MS Windows 95 and features Windows-specific keys.

Choose the layout type for your keyboard (for example, U.S. English).

Entering special characters (such as Ñ, Ô, and Ç) is done using "dead keys" (or compose key sequences). If you wish to use special characters requiring the use of dead keys, select **Enable**

Which model keyboard is attached to the computer?

Model:
Generic 101-key PC
Generic 102-key (Intl) PC
Generic 104-key PC
Generic 105-key (Intl) PC
Genius Comfy KB-16M
HP Internet
IBM Rapid Access

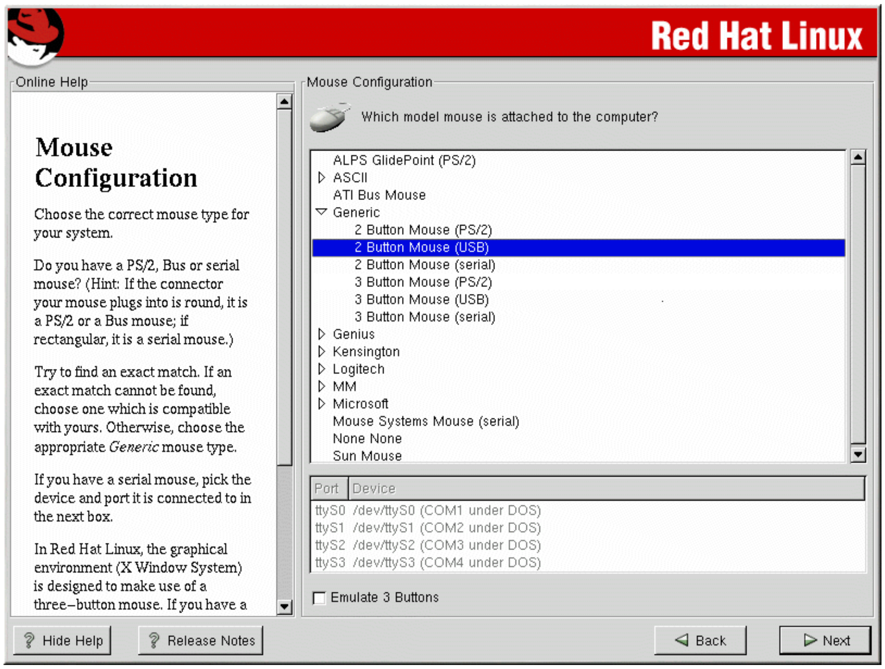
Layout:
Thai
Turkish
Turkish (F)
U.S. English
U.S. English w/ ISO9995-3
U.S. English w/ deadkeys
Ukrainian

Dead Keys:
Disable dead keys
Enable dead keys

Test your selection here:

Figure 3-2 Keyboard selection and configuration

5. Select the autoselected **USB Keyboard**, **US English**, **Enable dead keys**, and click **Next**.



Mouse Configuration

Choose the correct mouse type for your system.

Do you have a PS/2, Bus or serial mouse? (Hint: If the connector your mouse plugs into is round, it is a PS/2 or a Bus mouse; if rectangular, it is a serial mouse.)

Try to find an exact match. If an exact match cannot be found, choose one which is compatible with yours. Otherwise, choose the appropriate *Generic* mouse type.

If you have a serial mouse, pick the device and port it is connected to in the next box.

In Red Hat Linux, the graphical environment (X Window System) is designed to make use of a three–button mouse. If you have a

Which model mouse is attached to the computer?

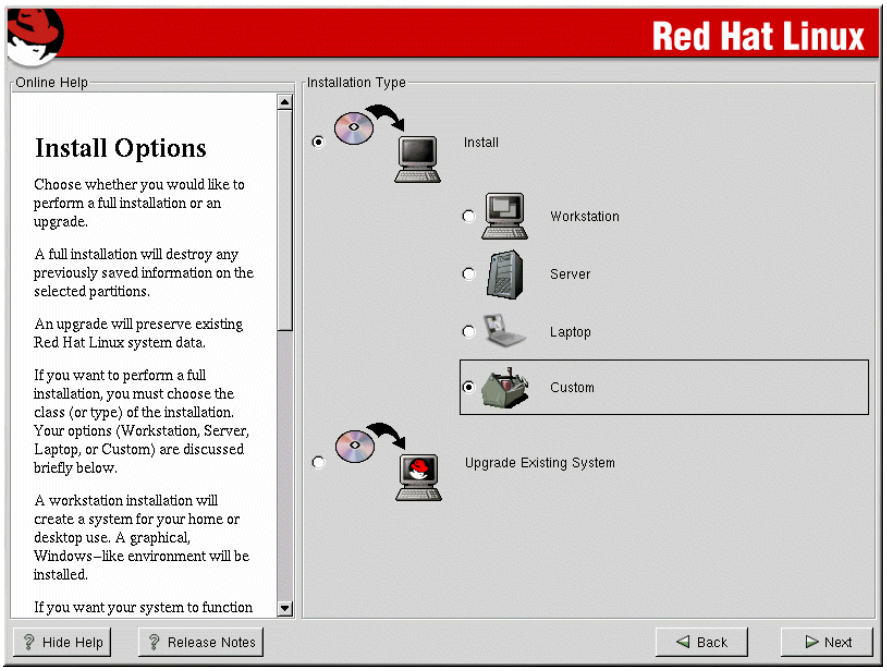
- ALPS GlidePoint (PS/2)
- ▷ ASCII
- ATI Bus Mouse
- ▽ Generic
  - 2 Button Mouse (PS/2)
  - **2 Button Mouse (USB)**
  - 2 Button Mouse (serial)
  - 3 Button Mouse (PS/2)
  - 3 Button Mouse (USB)
  - 3 Button Mouse (serial)
- ▷ Genius
- ▷ Kensington
- ▷ Logitech
- ▷ MM
- ▷ Microsoft
- Mouse Systems Mouse (serial)
- None None
- Sun Mouse

Port	Device
ttyS0	/dev/ttyS0 (COM1 under DOS)
ttyS1	/dev/ttyS1 (COM2 under DOS)
ttyS2	/dev/ttyS2 (COM3 under DOS)
ttyS3	/dev/ttyS3 (COM4 under DOS)

□ Emulate 3 Buttons

Figure 3-3 Mouse selection

6. Select the autoselected **2 Button Mouse (USB)** and click **Next**.



Install Options

Choose whether you would like to perform a full installation or an upgrade.

A full installation will destroy any previously saved information on the selected partitions.

An upgrade will preserve existing Red Hat Linux system data.

If you want to perform a full installation, you must choose the class (or type) of the installation. Your options (Workstation, Server, Laptop, or Custom) are discussed briefly below.

A workstation installation will create a system for your home or desktop use. A graphical, Windows–like environment will be installed.

If you want your system to function

● Install

○ Workstation

○ Server

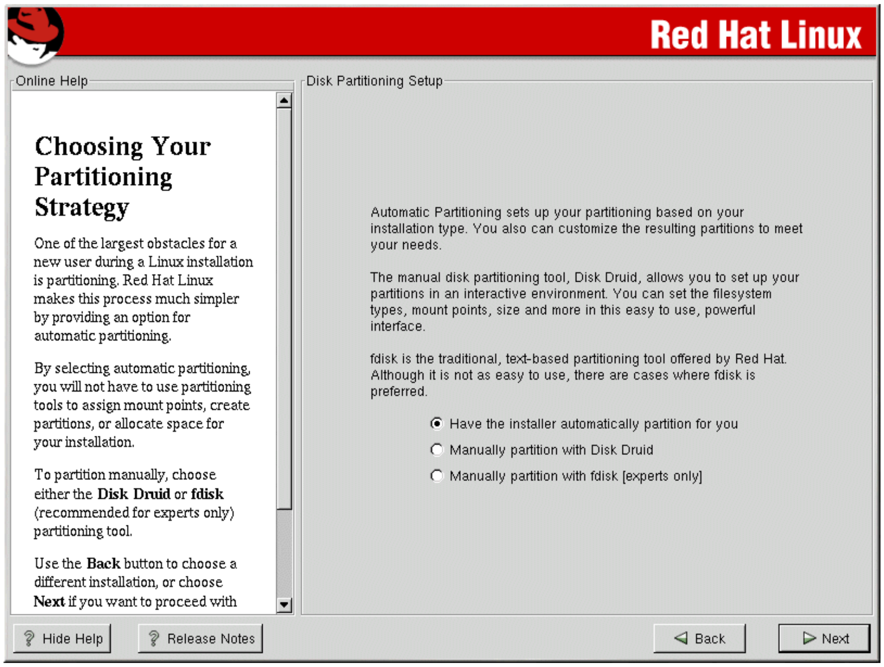
○ Laptop

● Custom

○ Upgrade Existing System

Figure 3-4 Install options

7. Select **Install** and **Custom**, and click **Next**.



Online Help

Choosing Your Partitioning Strategy

One of the largest obstacles for a new user during a Linux installation is partitioning. Red Hat Linux makes this process much simpler by providing an option for automatic partitioning.

By selecting automatic partitioning, you will not have to use partitioning tools to assign mount points, create partitions, or allocate space for your installation.

To partition manually, choose either the **Disk Druid** or **fdisk** (recommended for experts only) partitioning tool.

Use the **Back** button to choose a different installation, or choose **Next** if you want to proceed with

Disk Partitioning Setup

Automatic Partitioning sets up your partitioning based on your installation type. You also can customize the resulting partitions to meet your needs.

The manual disk partitioning tool, Disk Druid, allows you to set up your partitions in an interactive environment. You can set the filesystem types, mount points, size and more in this easy to use, powerful interface.

fdisk is the traditional, text-based partitioning tool offered by Red Hat. Although it is not as easy to use, there are cases where fdisk is preferred.

● Have the installer automatically partition for you

○ Manually partition with Disk Druid

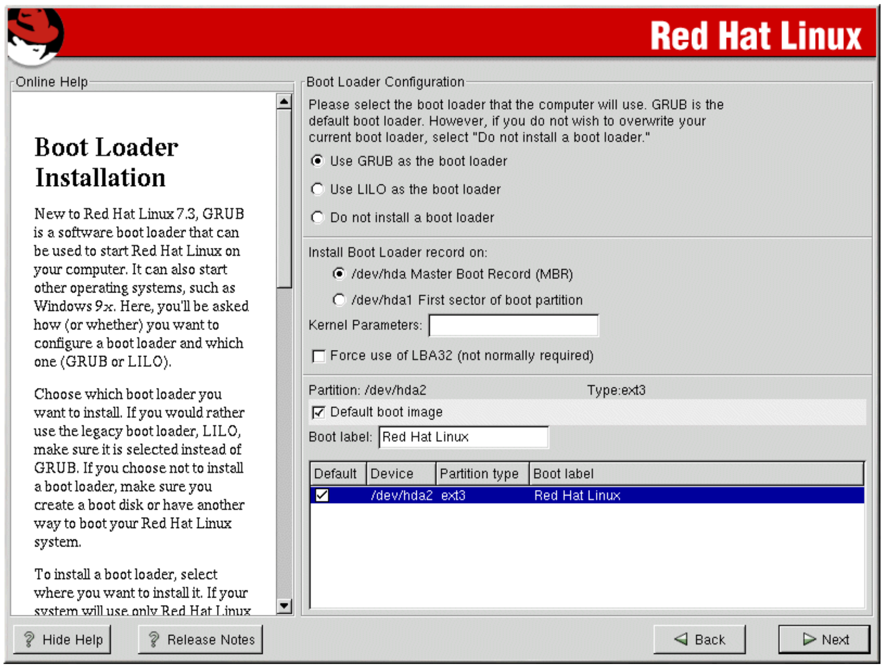
○ Manually partition with fdisk [experts only]

Figure 3-5 Disk partitioning

8. Partition the hard drive using either the automatic partiton (default choice) or manually. See:

   `http://www.redhat.com/docs/manuals/linux/RHL-7.3-Manual/install-guide/s1-diskpartsetup.html`

   for more information about partitioning the hard drive.



Online Help

Boot Loader Installation

New to Red Hat Linux 7.3, GRUB is a software boot loader that can be used to start Red Hat Linux on your computer. It can also start other operating systems, such as Windows 9x. Here, you'll be asked how (or whether) you want to configure a boot loader and which one (GRUB or LILO).

Choose which boot loader you want to install. If you would rather use the legacy boot loader, LILO, make sure it is selected instead of GRUB. If you choose not to install a boot loader, make sure you create a boot disk or have another way to boot your Red Hat Linux system.

To install a boot loader, select where you want to install it. If your system will use only Red Hat Linux

Boot Loader Configuration

Please select the boot loader that the computer will use. GRUB is the default boot loader. However, if you do not wish to overwrite your current boot loader, select "Do not install a boot loader."

● Use GRUB as the boot loader

○ Use LILO as the boot loader

○ Do not install a boot loader

Install Boot Loader record on:

● /dev/hda Master Boot Record (MBR)

○ /dev/hda1 First sector of boot partition

Kernel Parameters:

□ Force use of LBA32 (not normally required)

Partition: /dev/hda2 Type:ext3

☑ Default boot image

Boot label: Red Hat Linux

Default	Device	Partition type	Boot label
☑	/dev/hda2	ext3	Red Hat Linux

Figure 3-6 Boot loader options

9. Select **Use GRUB as the boot loader**, **/dev/hda Master Boot Record (MBR)**, **Default boot image**, and click **Next**.

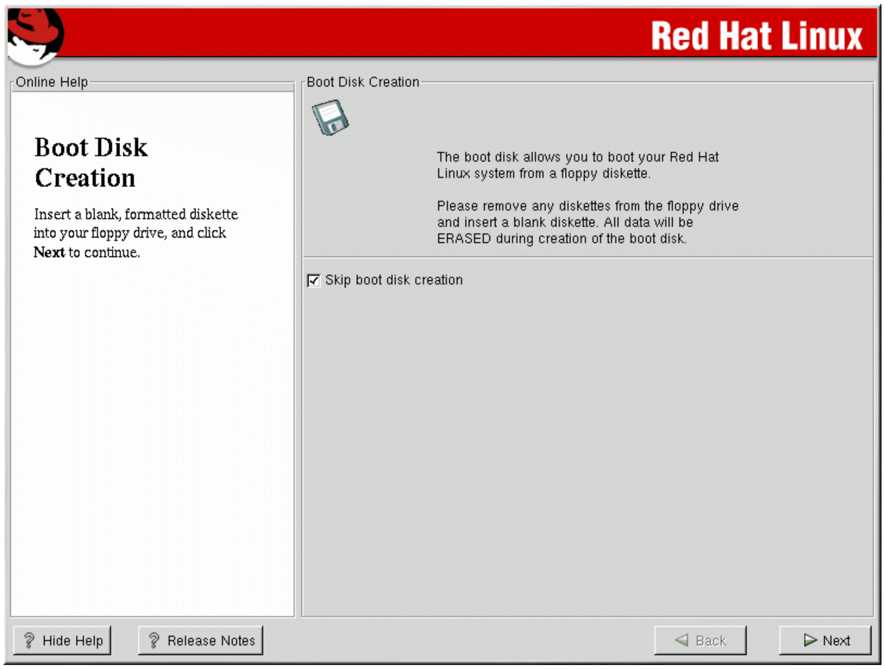
10. Do not select **Use a GRUB Password**; click **Next**.

11. On the eth0 page, deselect **Activate on boot**, click the **eth1** tab, and select **Configure using DHCP**, **Activate on boot,** then click **Next**.

12. Follow the Red Hat Linux 7.3 install instructions at

    http://www.redhat.com/docs/manuals/linux/RHL-7.3-Manual/install-guide/s1-firewallconfig.html

    to finish the install setup process.



Boot Disk Creation

Insert a blank, formatted diskette into your floppy drive, and click **Next** to continue.

The boot disk allows you to boot your Red Hat Linux system from a floppy diskette.

Please remove any diskettes from the floppy drive and insert a blank diskette. All data will be ERASED during creation of the boot disk.

☑ Skip boot disk creation

Figure 3-7 Boot disk creation

13. Select **Skip boot disk creation** and click **Next**.

    A boot diskette can be created with mkbootdisk after the diskette drive is defined; see list item 17 on page 35 for details.

14. Follow the Red Hat Linux 7.3 install instructions at

    http://www.redhat.com/docs/manuals/linux/RHL-7.3-Manual/install-guide/s1-xconfig.html

    to finish the install. We selected the following software groups.

    - Printing Support
    - Classic XWindow System
    - X Window System
    - GNOME
    - KDE
    - Network Support
    - Network Managed Workstation

- Utilities
- Software Developement
- Kernel Source

15. After the system is rebooted, log in as root.

16. If X11 is installed, edit /etc/X11/XF86Config-4.

    a. Add

    ```
    Section “ServerFlags”
       AllowMouseOpenFail
    EndSection
    ```

    to the top of the file.

    b. Save and exit.

17. The diskette drive is connected to the SCSI bus. It is configured to take the last ID on the SCSI bus. Since the blade server we are using has no SCSI hard drives, the diskette drive resides on /dev/sda. Therefore, we perform these steps to make the diskette drive available.

    a. Edit /etc/fstab and add

    ```
    /dev/sda/mnt/floppyautonoauto,auto0 0
    ```

    after all hard drive entries.

    b. Make the /mnt/floppy directory by running **`mkdir /mnt/floppy`**

    c. Mount the drive by running **`mount /dev/sda /mnt/floppy`**

18. Make a boot disk by running mkbootdisk and following the prompts.

19. Update the Ethernet drivers by editing /etc/modules so the entries for eth0 and eth1 look similar to those shown in Example 3-1.

Example 3-1 /etc/modules showing correct Ethernet drivers

```
alias parport_lowlevel parport_pc
alias eth0 bcm5700
alias eth1 bcm5700
```

3.2.3 Applying updates and installing the latest kernel

In this section, we will apply updates and the latest kernel to Red Hat.

1. Log on to the system as root.
2. Make a directory to store the updates in a file system which has at least 600 MB of free space. We made /usr/updates to store the updates.
3. Change to the updates directory.
4. Make *kernels* and *noInstall* directories.
5. Download all the current Red Hat 7.3 updates:

   a. Anonymous ftp to `updates.redhat.com`

   b. Use cd to go to 7.3/en/os

   c. Download all the files in the i386, i686, and noarch directories

6. Move all the kernel files into the kernels directory by typing

   ```
   mv kernel-* kernels
   ```

7. Move the i386 rpms that are replaced by i686 rpms into the noInistall directory by typing

   ```
   for i in $(ls *.i686.rpm); do mv ${i/i686.rpm/i386.rpm} noInstall; done
   ```

8. Install the updates by typing

   ```
   rpm -Fvh *.rpm
   ```

9. Install the kernel updates by:

   a. Typing **rpm -ivh kernels/kernel-2.4.18-18.7.x.i686.rpm** to install the uni processor kernel.

   b. Typing r**pm -ivh kernels/kernel-smp-2.4.18-18.7.x.i686.rpm** to install the smp kernel.

   c. Using **rpm -Fvh kerners/kernel-source-2.4.18-18.7.x.i386.rpm** to install the latest kernel source, if it is not already installed.

   d. Editing /boot/grub/grub.conf; change `default=2` to `default=0`, and save.

10. Reboot the system by typing `shutdown -r 0`.

11. Verify that system boots without any errors.

3.3 Network installation

Network installation is useful when you are installing multiple systems and do not want to have several copies of installation CDs. The system requirements for a Linux install server are:

1. 2 GB of free disk space
2. A network card
3. Anonymous ftpd or or NFS service

   This can be installed on a Linux system that satisfies requirements 1 and 2 above.

3.3.1 Setting up the NFS install server

A desktop system with enough free hard drive space was selected to be the install server. We transformed it into the install server by performing these steps.

1. Install nfs-utils rpm.
2. Make a /var/iso/RedHat directory.
3. Download Red Hat Linux 7.3 install iso images 1 through 3 and the md5 file.
   - rh-7.3-en-i386-cd1.iso
   - rh-7.3-en-i386-cd2.iso
   - rh-7.3-en-i386-cd3.iso
   - rh-7.3-en-i386.md5

Example 3-2 Output of md5sum check

```
[root@portal1 iso]# md5sum -c rh-7.3-en-i386.md5
rh-7.3-en-i386-cd1.iso: OK
rh-7.3-en-i386-cd2.iso: OK
rh-7.3-en-i386-cd3.iso: OK
md5sum: rh-7.3-en-i386-cd4.iso: No such file or directory
rh-7.3-en-i386-cd4.iso: FAILED open or read
md5sum: rh-7.3-en-i386-cd5.iso: No such file or directory
```

```
rh-7.3-en-i386-cd5.iso: FAILED open or read
md5sum: rh-7.3-en-i386-cd6.iso: No such file or directory
rh-7.3-en-i386-cd6.iso: FAILED open or read
md5sum: WARNING: 3 of 6 listed files could not be read
```

4. Check the files using **`md5sum -c rh-7.3-en-i386.md5`**.

   If any of the files are corrupt, delete them and download again. The output should be simular to that shown in Example 3-2 on page 36.

5. Check that other has read and execute permissions on /var/iso/RedHat and has read permission on all the iso files.

6. Edit /etc/export and add `/var/iso/RedHat *(ro)` to the end of the file.

7. Have NFS reload the export file using **`service nfs reload`**.

3.3.2 Setting up the FTP install server

A desktop system with enough free hard drive space was selected to be the install server. We transformed it into the install server by going through these steps.

1. Install anonftp rpm.
2. Make the /var/iso directory.
3. Download Red Hat Linux 7.3 install iso images 1 through 3 and the md5 file.
   - rh-7.3-en-i386-cd1.iso
   - rh-7.3-en-i386-cd2.iso
   - rh-7.3-en-i386-cd3.iso
   - rh-7.3-en-i386.md5

Example 3-3 Output of md5sum check

```
[root@portal1 iso]# md5sum -c rh-7.3-en-i386.md5
rh-7.3-en-i386-cd1.iso: OK
rh-7.3-en-i386-cd2.iso: OK
rh-7.3-en-i386-cd3.iso: OK
md5sum: rh-7.3-en-i386-cd4.iso: No such file or directory
rh-7.3-en-i386-cd4.iso: FAILED open or read
md5sum: rh-7.3-en-i386-cd5.iso: No such file or directory
rh-7.3-en-i386-cd5.iso: FAILED open or read
md5sum: rh-7.3-en-i386-cd6.iso: No such file or directory
rh-7.3-en-i386-cd6.iso: FAILED open or read
md5sum: WARNING: 3 of 6 listed files could not be read
```

4. Check the files using **`md5sum -c rh-7.3-en-i386.md5`**.

   If any of the files are corrupt, delete them and download again. The output should be similar to that shown in Example 3-3.

5. Make the 7.3, 7.3/cd1, 7.3/cd2, 7.3/cd3, 7.3/RedHat, 7.3/RedHat/RPMS directories in /var/ftp/pub.

Example 3-4 Mounting iso images

```
mount -o loop,ro -t iso9660 /var/iso/rh-7.3-en-i386-cd1.iso /var/ftp/pub/cd1
mount -o loop,ro -t iso9660 /var/iso/rh-7.3-en-i386-cd2.iso /var/ftp/pub/cd2
mount -o loop,ro -t iso9660 /var/iso/rh-7.3-en-i386-cd3.iso /var/ftp/pub/cd3
```

6. Mount the three install iso images to the cd directories.

7. Change to directory /var/ftp/pub/7.3.

8. Copy all the files in cd1/RedHat/base into RedHat/base using **`cp -r cd1/RedHat/base/RedHat`**.

9. Make symbolic links for in the rpm files in the RedHat/RPMS directory on cd1, cd2, and cd3 in /var/ftp/pub/RedHat/RPMS.

Example 3-5 Creating symbolic links

```
cd RedHat/RPMS
ln -s ../../cd1/RedHat/RPMS/*.rpm .
ln -s ../../cd2/RedHat/RPMS/*.rpm .
ln -s ../../cd3/RedHat/RPMS/*.rpm .
```

10. Configure the FTP server to start on reboot using the command:

```
chkconfig wu-ftpd on
```

11. Start the FTP server with the following command:

```
service xinetd restart
```

Example 3-6 Restarting xinetd

```
[root@portal1 etc]# service xinetd restart
Stopping xinetd:                                           [  OK  ]
Starting xinetd:                                           [  OK  ]
```

12. Verify that the FTP server is working by connecting the system using an FTP client and logging in as anonymous.

Example 3-7 Testing the anonymous FTP setup

```
[root]# ncftp 9.24.105.99
NcFTP 3.1.3 (Mar 27, 2002) by Mike Gleason (ncftp@ncftp.com).

Copyright (c) 1992-2002 by Mike Gleason.
All rights reserved.

Connecting to 9.24.105.99...
portal1 FTP server (Version wu-2.6.2-5) ready.
Logging in...
The response 'NcFTP@' is not valid
Next time please use your e-mail address as your password
   for example: joe@blade4.itso.ral.ibm.com
Guest login ok, access restrictions apply.
Logged in to portal1.
ncftp / >
```

13.

3.3.3 Creating the Network boot diskette

For the network installation on the IBM @server BladeCenter, we have to create a modified version of the normal Red Hat boot diskette. This is to include the driver for the blade server Ethernet adapter.

1. Install the kernel-BOOT-2.4.18-3.i386.rpm package from the Red Hat Linux CD 2 or from the Red Hat FTP site using **`rpm -ivh kernel-BOOT-2.4.18-3.i386.rpm`**.
2. Create a temporary work directory for all the files and two directories for mounting images by executing the commands:
   - `mkdir /tmp/newboot`
   - `mkdir /mnt/loop0`
   - `mkdir /mnt/loop1`
3. Copy the bootnet.img file from the images directory on Red Hat Linux CD1 to the work directory just created, cp /mnt/cdrom/images/bootnet.img /tmp/newboot.
4. Change to the working directory, cd /tmp/newboot.
5. Mount this image using the loopback device using **`mount -o loop /tmp/newboot/bootnet.img /mnt/loop0`**.
6. Copy the initrd.img file to the work directory and add .gz extension to the end by executing **`cp /mnt/loop0/initrd.img ./initrd.img.gz`**.
7. Unzip initrd.img.gz by executing **`gunzip initrd.img.gz`**.
8. Mount this RAMdisk image using **`mount -o loop initrd.img /mnt/loop1`**.
9. Copy the module-info and pcitable files to the working directory by executing:
   - **`cp /mnt/loop1/modules/module-info`**
   - **`cp /mnt/loop1/modules/pcitable`**
10. Edit module-info and add the lines shown in Example 3-8 to the bottom.

Example 3-8 Lines added to module-info

```
bcm5700
        eth
        "Broadcom BCM5700 10/100/1000 Ethernet adapter"
        line_speed "Line speed"
```

11. Save and close the file.
12. Edit the file pcitable and add the lines shown in Example 3-9 to the bottom.

Example 3-9 Lines added to pcitable

```
0x14e4  0x1644  "bcm5700"   "BROADCOM Corporation|NetXtreme BCM5700 Gigabit Ethernet"
0x14e4  0x1645  "bcm5700"   "BROADCOM Corporation|NetXtreme BCM5701 Gigabit Ethernet"
0x14e4  0x1646  "bcm5700"   "BROADCOM Corporation|NetXtreme BCM5702 Gigabit Ethernet"
0x14e4  0x1647  "bcm5700"   "BROADCOM Corporation|NetXtreme BCM5703 Gigabit Ethernet"
0x14e4  0x164d  "bcm5700"   "BROADCOM Corporation|NetXtreme BCM5702FE Gigabit Ethernet"
0x14e4  0x16a6  "bcm5700"   "BROADCOM Corporation|NetXtreme BCM5702X Gigabit Ethernet"
0x14e4  0x16a7  "bcm5700"   "BROADCOM Corporation|NetXtreme BCM5703X Gigabit Ethernet"
```

13. Save and close the file.
14. Copy these files over the original ones in the RAMdisk image by executing:
    - **`cp -f module-info /mnt/loop1/modules`**
    - **`cp -f pcitable /mnt/loop1/modules`**

15. Copy the modules.cgz file to the working directory and add the .gz extension to the end by executing **`cp /mnt/loop1/modules/modules.cgz ./modules.cpio.gz`**.
16. Unzip the file using **`gunzip modules.cpio.gz`**.
17. Make a directory named 2.4.18-3BOOT using **`mkdir 2.4.18-3BOOT`**.
18. Copy into this directory the bcm5700 module, installed on the system in step 1, by executing:

```
cp /lib/modules/2.4.18-3BOOT/kernel/drivers/addon/bcm5700/bcm5700.o \
./2.4.18-3BOOT/
```

19. Add this driver to the modules.cpio file by executing:

```
ls 2.4.18-3BOOT/* | cpio -o --append -H crc -F modules.cpio
```

20. Zip the modules.cpio file using **`gzip modules.cpio`**.
21. Copy this file over the original in the mounted RAMdisk by executing **`cp -f modules.cpio.gz /mnt/loop1/modules/modules.cgz`**.
22. Unmount the RAMdisk using **`umount /mnt/loop1`**.
23. Zip the RAMdisk by executing **`gzip initrd.img`**.
24. Copy the RAMdisk to the mounted boot diskette image by running **`cp -f initrd.img.gz /mnt/loop0/initrd.img`**.
25. Unmount the boot diskette image using **`umount /mnt/loop0`**.
26. Create a boot diskette from this image by placing a diskette into the floppy drive and executing **`dd if=bootnet.img of=/dev/fd0`**.

This install diskette is used to perform the network-based installations described in the next two sections.

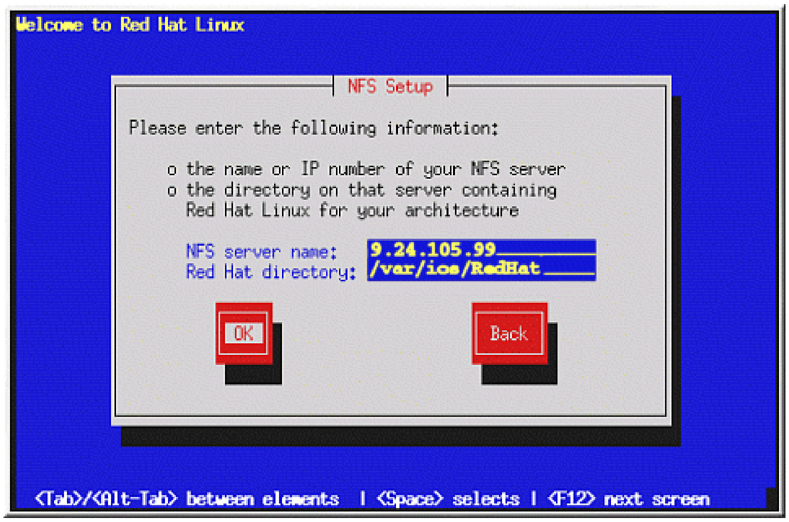
3.3.4 Installing from an NFS server

In this section, we will install Red Hat Linux via the NFS server. Complete the following instructions:

1. Insert the network install diskette created in 3.3.3, “Creating the Network boot diskette” on page 39 into the IBM @server BladeCenter diskette drive.
2. Power on the blade server and press the **Media Select** and **Console Select** buttons on the blade server.

> **Note:** Do not switch the KVM from the blade server until the installation has proceeded to installing the packages after the About to Install window has appeared, otherwise the mouse will lose functionality.

3. Choose **English**.
4. Choose **us**.
5. Choose **NFS image**.
6. Choose the interface you want to use. In our case, eth1 was connected to switch 1.
7. Accept the default Use dynamic IP configuration (BOOTP/DHCP); choose or type in the IP address, netmask, default gateway, and primary name server.



```
Welcome to Red Hat Linux

NFS Setup

Please enter the following information:

    o the name or IP number of your NFS server
    o the directory on that server containing
      Red Hat Linux for your architecture

      NFS server name:    9.24.105.99
      Red Hat directory:  /var/ios/RedHat

<Tab>/<Alt-Tab> between elements   | <Space> selects | <F12> next screen
```

Figure 3-8 NFS setup window

8. Type in the NFS sever IP address and the path to the iso files.

> **Note:** If the server cannot be reached and DHCP has been used for the client configuration, choose **Back** and type in the TPC/IP info. Then try again.

Follow the CD installation process detailed in step 3 on page 30.

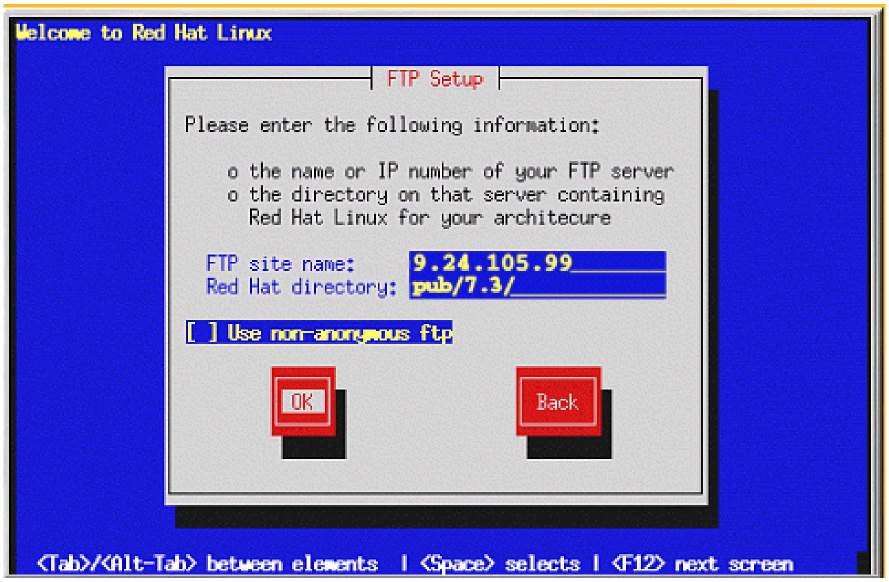
3.3.5 Installing the Linux operating system from an FTP server

In this section, we will install Red Hat Linux via the FTP Server. Perform the following instructions:

1. Insert the network install diskette created in 3.3.3, “Creating the Network boot diskette” on page 39 into the IBM @server BladeCenter diskette drive.
2. Power on the blade server and press the **Media Select** and **Console Select** buttons on the blade server.

> **Note:** Do not switch the KVM from the blade server until the installation has proceeded to installing the packages after the About to Install window has appeared, otherwise the mouse will lose functionality.

3. Choose **English**.
4. Choose **us**.
5. Choose **FTP**.
6. Choose the interface you want to use. In our case, eth1 was connected to switch 1.
7. Accept the default Use dynamic IP configuration (BOOTP/DHCP); choose or type in the IP address, netmask, default gateway, and primary name server.



```
Welcome to Red Hat Linux

                    FTP Setup

  Please enter the following information:

      o the name or IP number of your FTP server
      o the directory on that server containing
        Red Hat Linux for your architecure

    FTP site name:      9.24.105.99
    Red Hat directory:  pub/7.3/

  [ ] Use non-anonymous ftp

  <Tab>/<Alt-Tab> between elements  | <Space> selects | <F12> next screen
```

Figure 3-9 FTP setup window

8. Type in the IP address of the FTP server and directory path to RedHat relative to /var/ftp.

> **Note:** If the server cannot be reached and DHCP has been used for the client configuration, choose **Back** and type in the TPC/IP info. Then try again.

9. Follow the CD installation process described in step 3 on page 30.

3.4 Installation using PXE Boot

PXE stands for Pre eXecution Environment. This is a special function of the modern network adaptors that allows you to execute special pieces of code during the boot process. This is usually used to access files on other servers for booting itself, for example downloading the Linux kernel from another server into memory and then booting this kernel.

This enables computers to boot without the need for any local disk. We will use this system to start the installation process without the need for diskettes and CDs. When the blade server starts the installation using the PXE boot process, the installation is similar to the Network installation in the previous section.

3.4.1 Configuration of the PXE Boot server

We will use a boot server that contains the installation media and the necessary network services that allow the remote installation. We will use the blade server on which we installed Red Hat Linux in the CD installation process.

1. First, complete the server setup steps for an NFS install server described in 3.3, “Network installation” on page 36. In short, make sure you have a local copy of the Red Hat installation files and that these files are accessible using NFS. See section 3.3.1, “Setting up the NFS install server” on page 36 for details. Also make sure that the boot server has a static IP address.
2. Download or copy the DHCP server (the full name of the package, in our case, was dhcp-2.0pl5-8.i386.rpm) from the Red Hat CDs.

3. Install the DHCP server using **rpm -ivh dhcp-2.0pl5-8.i386.rpm**.
4. Copy the sample DHCP server configuration file to the /etc directory with **cp /usr/share/doc/dhcp-2.0pl5/dhcpd.conf.sample /etc/dhcpd.conf**
5. Edit the /etc/dhcpd.conf file to make it look like the file shown in Example 3-10.

Example 3-10 /etc/dhcpd.conf file for PXE boot

```
subnet 192.168.0.0 netmask 255.255.255.0 {
# --- default gateway
        option routers                192.168.0.1;
        option subnet-mask            255.255.255.0;

        option nis-domain             "itso.ral.ibm.com";
        option domain-name            "itso.ral.ibm.com";
        option domain-name-servers    192.168.0.1;

        option time-offset            -18000; # Eastern Standard Time

        option dhcp-class-identifier    "PXEClient";
        option vendor-encapsulated-options ff;

        range dynamic-bootp 192.168.0.128 192.168.0.255;
        default-lease-time 21600;
        max-lease-time 43200;

        }
}
```

Replace the IP addresses of the default gateway (`option routers`) and `domain-name-servers`, and the domain-names with the correct value in your setup.

6. Start the DHCP server using **service dhcpd start**. If you want, you can now test whether the DHCP server works.
7. Download the latest version of the PXE package from the updates section on the Red Hat FTP site or any of its mirrors. In our case, it was pxe-0.1-31.99.7.3.i386.rpm.
8. Install the PXE package using **rpm -ivh pxe-0.1-31.99.7.3.i386.rpm**.
9. Add the following lines to the end of the /etc/services file:

Example 3-11 Lines required in /etc/services for PXE boot

```
mtftp          1759/udp
pxe            67/udp
pxe            4011/udp
```

10. Add the following lines to /etc/xinetd.conf before the line `includedir /etc/xinetd.d`:

Example 3-12 Configuration of the PXE tftp server in /etc/xinetd.conf

```
service mtftp
{
        socket_type = dgram
        wait = yes
        user = root
        server = /usr/sbin/in.mtftpd
        server_args = /tftpboot
}
```

11. Add the following lines to the end of the /etc/rc.d/rc.local file.

Example 3-13 Commands in rc.local file needed for PXE boot

```
route add -host 255.255.255.255 eth0
route add -net 224.0.0.0 netmask 224.0.0.0 eth0
```

12. Copy the Linux kernel and initial ramdisk file for PXE boot from the first Red Hat Linux CD to the TFTP install section. In our case, it looked like Example 3-14.

Example 3-14 Copying of Linux kernel and initrd for PXE boot

```
cp /var/ftp/pub/7.3/cd1/images/pxeboot/vmlinuz /tftpboot/X86PC/UNDI/linux-install/linux.1
cp /var/ftp/pub/7.3/cd1/images/pxeboot/initrd-everything.img \
/tftpboot/X86PC/UNDI/linux-install/linux.2
```

13. Open the file /etc/pxe.conf in a editor and look for the following lines.

Example 3-15 Default configuration of PXE daemon in /etc/pxe.conf

```
[UseDHCPPort]
1
```

Change the `1` to a `0` and save this change.

14. Open the file /etc/init.d/pxe in a editor and look for the following lines.

Example 3-16 Changes to /etc/init.d/pxe

```
# Starts the pxe daemon
#
# chkconfig: - 56 54
```

Change the `56` to `98`. This will assure that the PXE daemon starts after the DHCP daemon.

15. Set up the DHCP and PXE services so that they will start up at boot using the following lines.

Example 3-17 Enabling dhcpd and pxe daemon at boot

```
[root@portal1 root]# chkconfig dhcpd on
[root@portal1 root]# chkconfig pxe on
```

16. Now reboot the server using **`shutdown -r now`**.
17. In the messages displayed in the startup sequence, see if both the DHCP and the PXE daemons start without any error messages.

Our PXE boot server is now ready for action.

3.4.2 Starting the installation

First, we have to configure the blade server so that it will use the PXE system in the startup boot sequence to allow for a remote boot.

If you want to have PXE always enabled as first in the boot sequence, perform the following steps.

1. Log in to the Web interface of the eServer BladeCenter management module.
2. Select **Blade Tasks -> Configuration**.
3. Click **Boot Sequence**.

4. Click the name of the blade server you want to edit.
5. Select **Network - PXE** as the first device; the rest depends on your personal preference.
6. Click **Save**.

If you want to use PXE only once for the installation, press **F12** at the BIOS startup window when is shows you the different functions keys you can use.

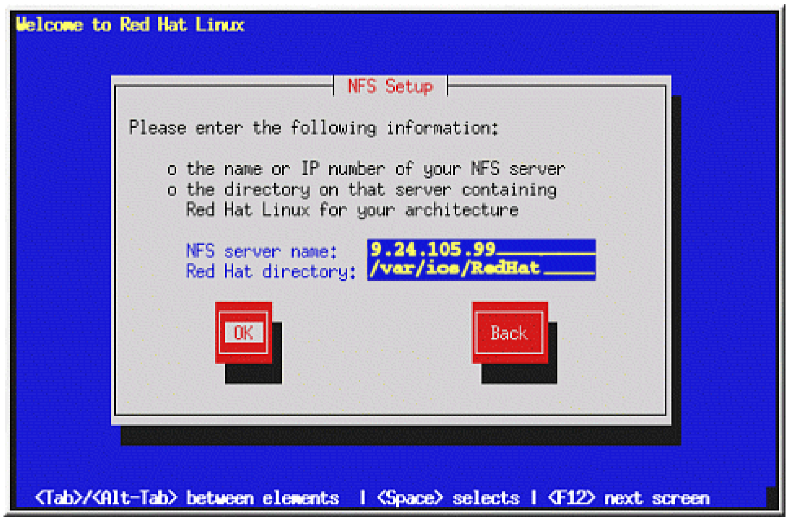
Now we will reboot the blade server so that the installation process can begin.

1. Log back in to the Web interface of the BladeCenter management module if you have logged out.
2. Click **Blade Tasks -> Power/Restart**.
3. Select the checkbox on the line of the blade server on which you will perform the installation.
4. Click **Restart Blade**.
5. Press the **KVM select** button on the front of the blade server to follow the installation process.

> **Note:** Do not switch the KVM from the blade server until the installation has gone on to installing the packages after the About to Install window appears, otherwise the mouse will lose functionality.

We will now begin the actual installation process.

1. Follow the boot sequence of the blade server until you see the message `Press F8 to view menu ....`
2. Select the line `Remote Install Linux` and press **Enter**.
3. When you see the line `Press any key to enter kernel parameters...`, press the **Enter** key twice. This will boot the kernel with the default parameters.
4. You should see some messages indicating that the kernel and the initrd are being downloaded and then the Linux kernel will boot.
5. Choose **English**.
6. Choose **us**.
7. Choose **NFS image**.
8. Choose the interface you want to use. In our case, eth1 was connected to switch 1.
9. Accept the default Use dynamic IP configuration (BOOTP/DHCP); choose or type in the IP address, netmask, default gateway, and primary name server.


Welcome to Red Hat Linux

NFS Setup

Please enter the following information:

o the name or IP number of your NFS server
o the directory on that server containing
  Red Hat Linux for your architecture

NFS server name: 9.24.105.99
Red Hat directory: /var/ios/RedHat

OK
Back
<Tab>/<Alt-Tab> between elements | <Space> selects | <F12> next screen

Figure 3-10 NFS install setup window

10.Type in the NFS sever IP address and the path to the iso files.

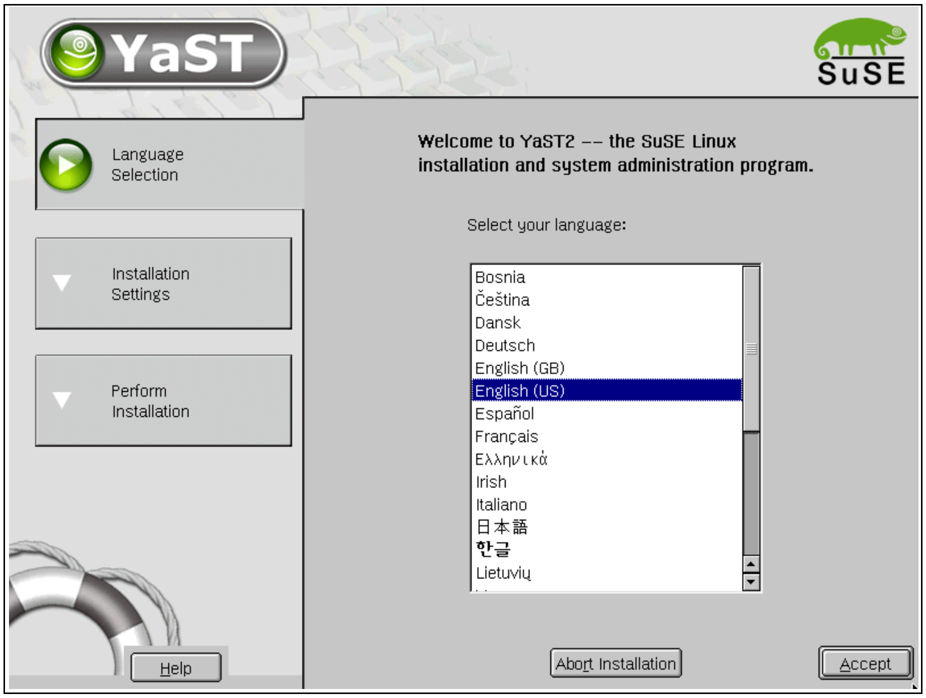
Note: If the server cannot be reached and DHCP has been used for the client configuration, choose **Back** and type in the TPC/IP info. Then try again.

11.Follow the CD installation process from step 3 on page 30.

3.5 Installation of SuSE Linux Enterprise Server 8

In this section, we will go over the installation of SuSE Linux Enterprise Server (SLES) V8 for the IA-32 platform. We will only handle the CD-ROM installation. You will find information about the other methods of installing SLES 8 in the *SLES Installation Guide*.

1. Start the server and insert the SuSE Linux Enterprise Server 8 for IA-32 CD 1 into the CD-ROM drive.
2. At the `boot options:` prompt, press **Enter**. You see a bar (`Loading Linux kernel`*)* and when it reaches 100% the Linux kernel will boot.
3. When the installer is started, you receive a dialog box containing the end user license agreement. Press **Enter** to accept.



Welcome to YaST2 -- the SuSE Linux installation and system administration program.

Select your language:

Figure 3-11 SLES Installation - Language Selection

4. For the IBM @server BladeCenter HS20, the mouse will not work initially when the installer starts. Follow these directions:

   a. Press **Alt + a** to select **English (US)**.

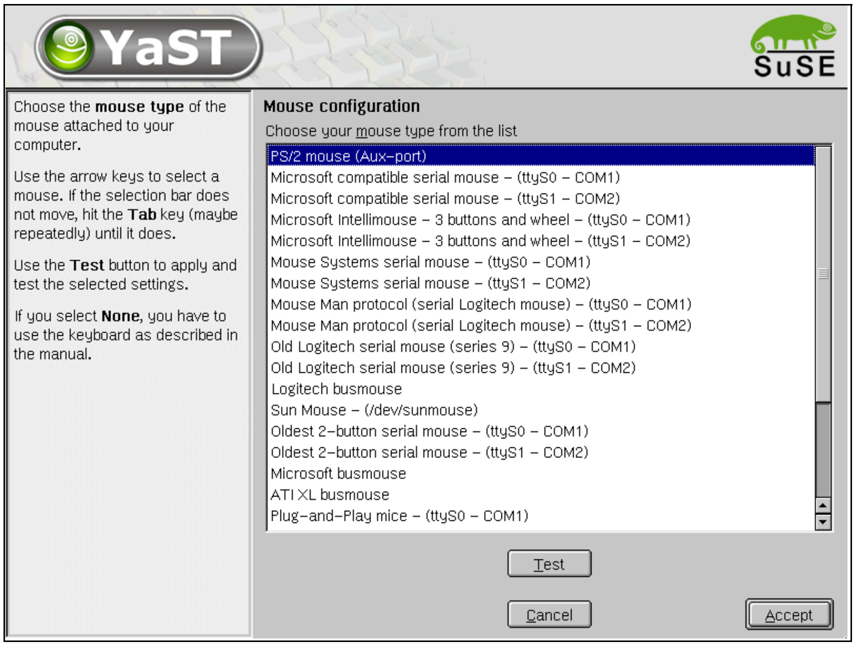
   b. If the `Please select` dialog box appears, press **Alt + o** to select **New installation**.

   c. Press **Alt + c** and scroll, with the arrow keys, to Mouse to change the mouse, then press **Enter**. Scroll down through the list to USB mouse and Press **Alt + t** (Test), to ensure that it is working with your mouse. Click **Accept** when finished.

5. At the Installation Settings display, your settings are preselected for the installation that is specific to your system. You may not need to alter many of these settings, but make sure you do follow the instructions under the Boot section. Click **Accept** when finished:

   – **Mode**: Do not change this.

   – **Keyboard layout**: If the language is set correctly, in most cases the correct keyboard setting will be automatically selected. Click **Keyboard Layout** to modify the keyboard mapping, if needed. Select the keyboard layout from the list. You can check that it works properly by typing in the Keyboard test box. Click **Accept** when finished.



Choose the **mouse type** of the mouse attached to your computer.

Use the arrow keys to select a mouse. If the selection bar does not move, hit the **Tab** key (maybe repeatedly) until it does.

Use the **Test** button to apply and test the selected settings.

If you select **None**, you have to use the keyboard as described in the manual.

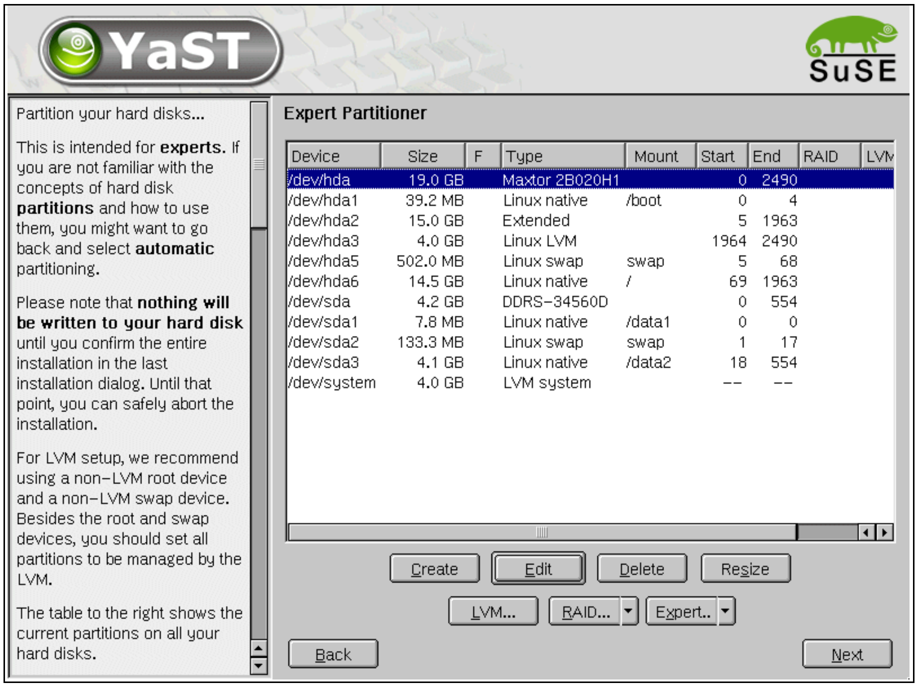
Mouse configuration

Choose your mouse type from the list

- PS/2 mouse (Aux-port)
- Microsoft compatible serial mouse – (ttyS0 – COM1)
- Microsoft compatible serial mouse – (ttyS1 – COM2)
- Microsoft Intellimouse – 3 buttons and wheel – (ttyS0 – COM1)
- Microsoft Intellimouse – 3 buttons and wheel – (ttyS1 – COM2)
- Mouse Systems serial mouse – (ttyS0 – COM1)
- Mouse Systems serial mouse – (ttyS1 – COM2)
- Mouse Man protocol (serial Logitech mouse) – (ttyS0 – COM1)
- Mouse Man protocol (serial Logitech mouse) – (ttyS1 – COM2)
- Old Logitech serial mouse (series 9) – (ttyS0 – COM1)
- Old Logitech serial mouse (series 9) – (ttyS1 – COM2)
- Logitech busmouse
- Sun Mouse – (/dev/sunmouse)
- Oldest 2-button serial mouse – (ttyS0 – COM1)
- Oldest 2-button serial mouse – (ttyS1 – COM2)
- Microsoft busmouse
- ATI XL busmouse
- Plug-and-Play mice – (ttyS0 – COM1)

Figure 3-12 SLES Installation - Mouse Configuration

- – **Mouse**: Click **Mouse** if you need to change the type of mouse you are using. Changing the mouse type is not necessary in most cases. Select the correct mouse type from the list and click **Test** to ensure that it is working with your mouse. Click **Accept** when finished.
- – **Partitioning**: Click **Partitioning** if you need to change the partitioning scheme for your drive(s). You will see YaST2's proposed partitioning scheme and can do one of the following:
  - Accept the suggestion by clicking **Next**.



Partition your hard disks...

This is intended for **experts.** If you are not familiar with the concepts of hard disk **partitions** and how to use them, you might want to go back and select **automatic** partitioning.

Please note that **nothing will be written to your hard disk** until you confirm the entire installation in the last installation dialog. Until that point, you can safely abort the installation.

For LVM setup, we recommend using a non-LVM root device and a non-LVM swap device. Besides the root and swap devices, you should set all partitions to be managed by the LVM.

The table to the right shows the current partitions on all your hard disks.

Expert Partitioner

Device	Size	F	Type	Mount	Start	End	RAID	LVM
/dev/hda	19.0 GB		Maxtor 2B020H1		0	2490		
/dev/hda1	39.2 MB		Linux native	/boot	0	4		
/dev/hda2	15.0 GB		Extended		5	1963		
/dev/hda3	4.0 GB		Linux LVM		1964	2490		
/dev/hda5	502.0 MB		Linux swap	swap	5	68		
/dev/hda6	14.5 GB		Linux native	/	69	1963		
/dev/sda	4.2 GB		DDRS-34560D		0	554		
/dev/sda1	7.8 MB		Linux native	/data1	0	0		
/dev/sda2	133.3 MB		Linux swap	swap	1	17		
/dev/sda3	4.1 GB		Linux native	/data2	18	554		
/dev/system	4.0 GB		LVM system		--	--		

Figure 3-13 SLES Installation - Expert Partitionner

- Modify the proposal by clicking **Base partition setup on this proposal** and then clicking **Next**.

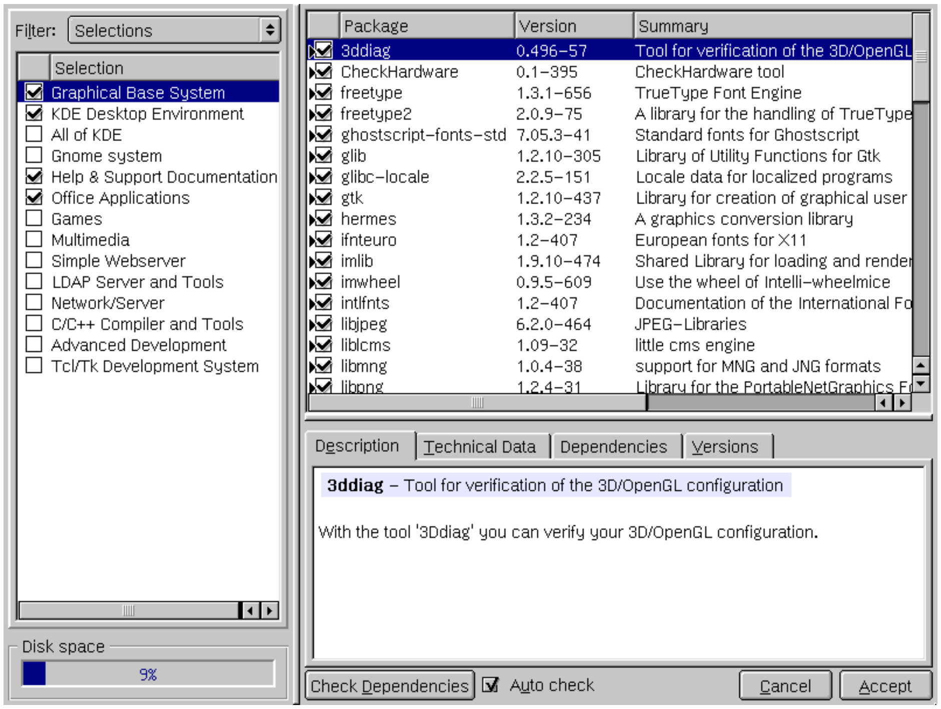
  The Expert Partitioner window opens; it allows you to view and manipulate the existing partitions on the hard drive(s).

  Click **Create** to add new partitions, **Delete** to delete partitions (highlight the partition you wish deleted first), **Edit** to make changes to existing partitions, and **Resize** to resize ext2, ext3, ReiserFS, XFS, and swap partitions.

  When finished, select **Next**.

- Discard the proposal by clicking **Create custom partition setup** and then **Next**. This will bring you to the Preparing Hard Disk - Step 1 window, where you can select the hard disk on which you want to install SuSE Linux Enterprise Server 8. Click **Next**. Depending on the current state of the hard disk, you will have the opportunity to erase the complete hard disk, delete certain partitions or install SLES 8 on the free space currently available on the disk.

  Or you can click **Custom partitioning -- for experts** and then click **Next**. This will bring you into the Expert Partitioner window as metioned in the previous point.



Filter: Selections

Selection
☑ Graphical Base System
☑ KDE Desktop Environment
□ All of KDE
□ Gnome system
☑ Help & Support Documentation
☑ Office Applications
□ Games
□ Multimedia
□ Simple Webserver
□ LDAP Server and Tools
□ Network/Server
□ C/C++ Compiler and Tools
□ Advanced Development
□ Tcl/Tk Development System

Package	Version	Summary
☑ 3ddiag	0.496−57	Tool for verification of the 3D/OpenGL
☑ CheckHardware	0.1−395	CheckHardware tool
☑ freetype	1.3.1−656	TrueType Font Engine
☑ freetype2	2.0.9−75	A library for the handling of TrueType
☑ ghostscript−fonts−std	7.05.3−41	Standard fonts for Ghostscript
☑ glib	1.2.10−305	Library of Utility Functions for Gtk
☑ glibc−locale	2.2.5−151	Locale data for localized programs
☑ gtk	1.2.10−437	Library for creation of graphical user
☑ hermes	1.3.2−234	A graphics conversion library
☑ ifnteuro	1.2−407	European fonts for X11
☑ imlib	1.9.10−474	Shared Library for loading and render
☑ imwheel	0.9.5−609	Use the wheel of Intelli−wheelmice
☑ intlfnts	1.2−407	Documentation of the International Fo
☑ libjpeg	6.2.0−464	JPEG−Libraries
☑ liblcms	1.09−32	little cms engine
☑ libmng	1.0.4−38	support for MNG and JNG formats
☑ libpng	1.2.4−31	Library for the PortableNetGraphics Fo

3ddiag – Tool for verification of the 3D/OpenGL configuration

With the tool '3Ddiag' you can verify your 3D/OpenGL configuration.

Disk space: 9%

☑ Auto check

Figure 3-14 SLES Installation - Software packages installation

– **Software**: Click **Software** to change the default software configuration (the default configuration installs most basic networking, administrative, and development tools needed). In the Software Selection window, select the software configuration you want. Click **Accept** after you have chosen the software you want.

  For a specialized selection, click **Detailed selection**. You can then select specific software using the package tool. When finished, click **Accept**.

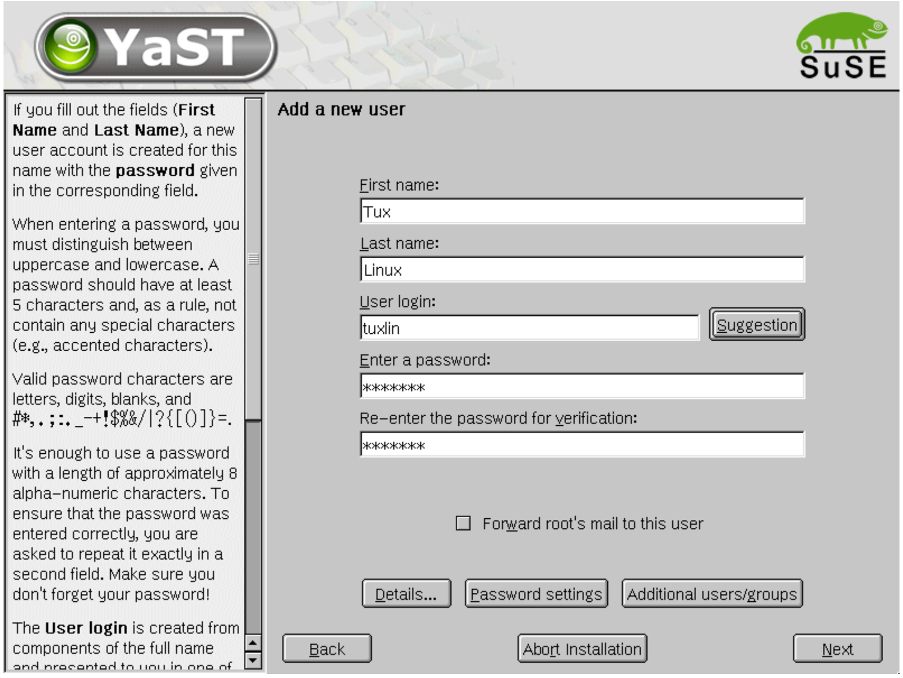
– **Booting**: Click **Booting** and make sure that **Write GRUB to the boot disk ('MBR on <device>')** is selected. Add the parameter `acpi=oldboot` under the Kernel boot.

- **Time zone**: Click **Time zone** to modify the default time zone for the installation. Select the correct time zone from the list. Under the *Hardware clock set to* section, select either local time or UTC, as appropriate. Click **Accept**.
- **Language**: Click **Language** to modify the default language for the installation. Select the needed language from the list and click **Accept**.

6. If all Installation Settings are to your preference, click **Accept** and click **Yes, install** at the Warning window.
7. You will be asked to insert the 'UnitedLinux Version 1.0' CD 1. Do so and click **OK**.
8. Later in the installation process, you will be asked to insert additional CDs to finish installing the packages. You will need the 'UnitedLinux Version 1.0' CD 2 and the 'SuSE SLES Version 8' CD1.
9. When all packages are installed, the installer will perform some post installation scripts and then reboots into the final installer section on the hard disk. Remove any CDs from the CD-ROM drive as the prompt indicates and press **Enter**.

Note: In our case, the mouse did not work in the next steps of the installation process. We had to use keyboard navigation.

10. At the *Password for 'root', the system administrator* window, type the administrative password that you want twice and click **Next**.



If you fill out the fields (**First Name** and **Last Name**), a new user account is created for this name with the **password** given in the corresponding field.

When entering a password, you must distinguish between uppercase and lowercase. A password should have at least 5 characters and, as a rule, not contain any special characters (e.g., accented characters).

Valid password characters are letters, digits, blanks, and #*,.;:._-+!$%&/|?{[()]}=.

It's enough to use a password with a length of approximately 8 alpha-numeric characters. To ensure that the password was entered correctly, you are asked to repeat it exactly in a second field. Make sure you don't forget your password!

The **User login** is created from components of the full name and presented to you in one of

Add a new user

First name: Tux

Last name: Linux

User login: tuxlin

Enter a password: ********

Re-enter the password for verification: ********

☐ Forward root's mail to this user

Figure 3-15 SLES Installation - Adding new users

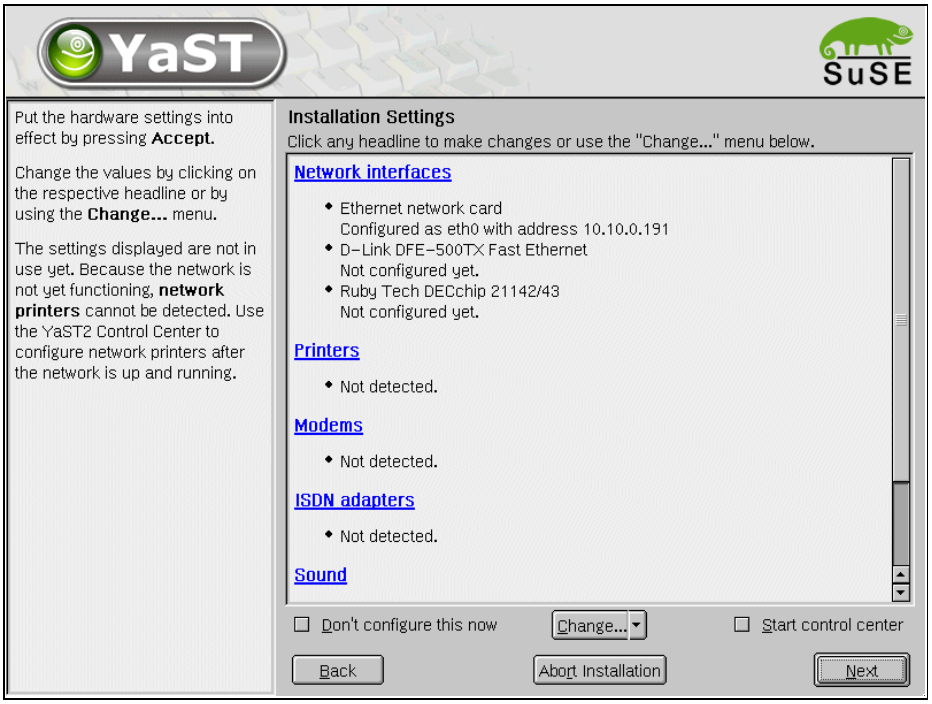
11. At the Add a New User window, type the information for one system user. You can also add additional users and groups or change the password settings by clicking the respective buttons to do so. When finished, click **Next**.

Note: You need to add at least one user to let the installation continue.

12. The Desktop Settings window displays. Select one of the following options:
    - **Text mode only -- no graphical desktop** (this setting will not configure the graphical environment). If this setting is selected, click **Next** and continue to step 23.
    - **Graphical desktop environment** (this setting is the default selection). When making this selection, you will see some auto-detected settings for the graphical environment.
13. Click **Accept** to accept the settings and continue to step 23. To alter the settings, continue with step 15.
14. Click **Change** to change the graphic environment settings.
15. If your monitor was not detected, you will be prompted to start the configuration dialog to set up the display. Click **Yes** and then the **Properties** button. Select the appropriate vendor and model (for IBM Monitor specifications, go to: `ftp://ftp.pc.ibm.com/pcicrse/psref/mbook.pdf`). Click **OK** to return to SaX2. Click **Finish**.
16. For the mouse to work, choose the following in the mouse configuration dialog. At the **General** tab, select **USB-Mouse** and then **PS/2**. At the Port tab, select **/dev/input/mice**. Then click **OK**.

    Note: We needed to modify a configuration file after installation. See step 26.

17. If needed, select the correct resolution, color depth, and refresh rate (for your monitor) by clicking **Desktop** (on the left), **Color and Resolution**, and then the **Properties** button. It is possible to damage your hardware by making these settings too high. Do not make changes if you do not know the correct settings. Click **OK** and then **Finish** when done.
18. Click **Finalize >>**.
19. Click **Test...** at the informational window.
20. A window displays allowing you to fine tune the configuration with arrows (if the window does not appear, press **Ctrl+Alt+Backspace** to return to SaX2 and change the settings). If everything looks right, click **Save**.
21. Click **OK** at the dialog that confirms that the configuration is saved.
22. Click **Accept** at the Desktop Settings window if the settings are acceptable. If the settings are not acceptable, refer to step 15 again.
23. You may receive a warning at this time asking if you want to detect your printers. Select **Skip detection** unless you will be configuring a printer; in that case you should click **Yes**.



Put the hardware settings into effect by pressing **Accept**.

Change the values by clicking on the respective headline or by using the **Change...** menu.

The settings displayed are not in use yet. Because the network is not yet functioning, **network printers** cannot be detected. Use the YaST2 Control Center to configure network printers after the network is up and running.

Installation Settings

Click any headline to make changes or use the "Change..." menu below.

Network interfaces

- Ethernet network card
  Configured as eth0 with address 10.10.0.191
- D-Link DFE-500TX Fast Ethernet
  Not configured yet.
- Ruby Tech DECchip 21142/43
  Not configured yet.

Printers

- Not detected.

Modems

- Not detected.

ISDN adapters

- Not detected.

Sound

□ Don't configure this now
□ Start control center

Figure 3-16 SLES Installation - Hardware configuration

24. The Installation Settings display enables you to configure various types of hardware (the hardware configuration can also be done later with the program YaST2). For example, you can set up your network interfaces here. When satisfied, click **Next** to finish the installation.
25. The installer will then reboot into the installed Linux. The installation is finished now.
26. In our case, we needed to change the file /etc/sysconfig/hotplug to make the mouse work in X. Search for the line that start with `HOTPLUG_USB_STATIC_MODULES=...` And add `mousedev` to the list at that line. At the next reboot, the mouse should work in X.

    If the server starts up by default in X, you can switch to a console terminal with **Crtl+Alt+F1** or in KDE you can use **Alt+F1** to access the menu for a terminal.

3.6 Installation of IBM Director Agent

In this section, we will give you a short overview of what IBM Director is and how to install the IBM Director Agent part on a Linux system. For more information about IBM Director and system management of the IBM @server BladeCenter, we refer you to the Redpaper *IBM @server BladeCenter Systems Management*, REDP3582.

3.6.1 Overview of IBM Director V4

IBM Director is a comprehensive workgroup hardware manager designed for use with IBM @server xSeries servers, PCs, notebooks and now IBM @server BladeCenter.

IBM Director V4 includes support for your IBM @server BladeCenter server, enabling you to manage, deploy and monitor your system much more efficiently. IBM Director includes features such as self-management and proactive and predictive tools which provide higher levels of availability and reliability.

The IBM Director software is made up of three components:

- IBM Director Server
- IBM Director Agent
- IBM Director Console

A different combination of these components is required for each of the hardware groups in your IBM Director environment. The management server must contain all three of these components. The IBM Director Console must be installed on the management console or any system from which a system administrator will remotely access the management server. The IBM Director Agent must be installed on each system you intend to manage.

IBM Director Server

The IBM Director Server is the main component of IBM Director software. The server component contains the management data, the server engine, and the application logic. IBM Director Server provides basic functions such as:

- Discovery of the managed systems
- Storage of configuration and management data
- Inventory database
- Event listening
- Security and authentication
- Management console support
- Administrative tasks

IBM Director comes with the Microsoft Jet database engine but other database applications can be used in larger IBM Director management solutions. On Linux installations, the PostgreSQL database server is supported in addition to DB2® and Oracle that are supported on both Windows and Linux.

The IBM Director Console and IBM Director Agent are automatically installed when you install the IBM Director server. Every IBM @server xSeries comes with an IBM Director Server license.

IBM Director Agent

The IBM Director Agent allows the Director server to communicate with systems on which it is installed. The agent provides the server with management data which can be transferred using TCP/IP, NetBIOS and IPX protocols. The IBM Directors Agent Web-based access can only be enabled on Windows operating systems. All IBM systems come with an IBM Director Agent license. Additional licenses can be purchased for non-IBM systems.

IBM Director Console

The IBM Director Console enables systems administrators to manage all systems which have an agent installed. This is done easily via the graphical user interface (GUI) by either a drop-and-drag action or a single click. Unlike the agent, the console and server communicate and transfer data using TCP/IP. The IBM Director Console does not require the Director agent to be installed unless you want to manage this system as well, in which case the agent must be installed separately. IBM Director Console does not require a license and can be installed on as many systems as needed.

3.6.2 Installation of IBM Director Agent on Linux

The section discusses the requirements and installation steps for the IBM Director Agent.

IBM Director Agent Requirements

Following are the hard- and software requirements for installing IBM Director Agent on Linux.

Table 3-1 Requirements for IBM Director Agent

	IBM Director Agent
CPU	Pentium 266+ MHz
Memory (RAM)	128 MB
Diskspace	220 MB
Supported Linux OS	Red Hat Linux, versions 7.1, 7.2, 7.3 Red Hat Linux Advanced Server, V2.1 SuSE Linux, versions 7.2, 7.3, 8.0, 8.1

Note: The above values for the hardware are the minimum requirements. For performance reasons and depending on the size of your Director management solution, these values may need to be increased. These requirements are in addition to any other software requirements that may be running on the system.

Installation steps

Complete the following steps to install IBM Director Agent 4.

Note: Some packages from the IBM Director Agent require that you install the source of the kernel that you currently have running. You can find the kernel source for both Red Hat and SuSE in the kernel-source package.

1. Insert the Director 4 CD into the CD-ROM drive.
2. If the CD-ROM drive automounts, go to step 3. Otherwise, type the following command and press **Enter**:

   ```
   mount /mnt/cdrom
   ```

   Where `/mnt/cdrom` is the mount point of the CD-ROM drive.
3. Type the following command and press **Enter**:

   ```
   cd /mnt/cdrom/director/agent/linux/
   ```

   Where `/mnt/cdrom` is the mount point of the CD-ROM drive.
4. Copy the i386/ folder that contains the IBM Director Agent for Linux code to a place on the harddisk:

   ```
   cp -a i386/ /usr
   ```

5. The IBM Director Agent will be installed in the /opt directory. Depending on how you partitioned your hard disk, it could be that there is not enough space on the partition that contains the /opt directory. If this is the case, then follow these commands to transfer the /opt directory to another partition:

   a. `mv /opt /usr/opt`

   b. `ln -s /usr/opt /opt`

   We transferred the /opt directory to the partition holding the /usr directory.

6. Open a text editor and modify the dirinstall script. This file is located in the i386/ directory you copied in step 4 and is fully commented.

   You can specify the location of the .rpm files (if they are located in a different directory from the dirinstall script), select the IBM Director Extensions you want to install, and choose log file options.

   > **Note:** In our installations, the ibmlm78 package would not install. Therefore, we disabled this by changing the line `set LM78 Driver=1` to `=0`.

7. Save the modified installation script.

8. Allow the installation script to be executed with the command **`chmod a+x dirinstall`**.

9. Type the following command and press **Enter**:

   ```
   ./dirinstall
   ```

   Where `dirinstall` is the name of the installation script. This will start the installation process.

10. To enable encryption, type the following command and press **Enter**:

    ```
    /opt/IBM/director/bin/cfgsecurity
    ```

11. To start IBM Director Agent, type the following command and press **Enter**:

    ```
    /opt/IBM/director/bin/twgstart
    ```

12. To unmount the CD-ROM drive, type the following and press **Enter**:

    ```
    umount /mnt/cdrom
    ```

    Where `/mnt/cdrom` is the mount point of the CD-ROM drive.

Your IBM Director Agent is now installed. For more information on how to use this IBM Director Agent, refer to the IBM Director documentation and the *IBM* @server *BladeCenter Systems Management* Redpaper, REDP3582.

4

Installation of Apache

In this chapter, we will go over the installation of Apache into our IBM @server BladeCenter environment. We will cover the installation of V1.3 as well as V2.0 of Apache.

For each version of Apache we will go into the installation of Apache itself as well as the SSL, Perl and PHP modules.

We will perform all the installations on the Red Hat Linux 7.3 server that we have installed in the previous chapter. In addition, we will mention the differences between the Red Hat installation and the SuSE Linux Enterprise Server installation.

4.1 Installation of Apache HTTP Server V1.3

In this section, we will perform the installation of V1.3 of the Apache HTTP Server on an IBM @server BladeCenter.

1. Depending on the options you selected during the installation of Red Hat Linux, it could be that the Apache HTTP Server is already installed. You can verify this in the following way.

Example 4-1 Check to see what version of Apache is installed

```
[root@blade1 root]# rpm -qa | grep apache
apacheconf-0.8.2-2
apache-1.3.23-11
apache-devel-1.3.23-11
[root@blade1 root]#
```

   If you see a line that looks like `apache-1.3.23-11` then Apache is already installed on your server.

2. Check on the Red Hat Errata page for V7.3 (`http://rhn.redhat.com/errata/rh73-errata.html`) that the version you have installed is the latest recommended version. If there are newer versions available then download those from the Red Hat FTP site or any of their mirrors.

   For SLES, you need to contact SuSE for the location of the updates.

3. You can install the updates, if there are newer versions, as follows.

Example 4-2 Upgrade of Apache

```
[root@blade1 root]# rpm -Uvh apache-1.3.23-14.i386.rpm apache-devel-1.3.23-14.i386.rpm
Preparing...                ########################################### [100%]
   1:apache                 ########################################### [ 50%]
   2:apache-devel           ########################################### [100%]
[root@blade1 root]#
```

   If you receive no errors in the update process then the Apache HTTP Server is correctly installed and ready for use.

> **Note:** The installation of the updates should be done logged in as the root user.

4.1.1 Installation of the SSL, Perl and PHP modules

On a Red Hat Linux 7.3 system, these three modules also come as RPM packages. Their installation is therefore similar to that of Apache itself. On SLES, the PHP package is called mod_php4 and not php.

1. Check whether any of these modules is installed, and if so, what version is installed.

Example 4-3 Module and version installation check

```
[root@blade1 root]# rpm -qa | grep mod_ssl
mod_ssl-2.8.7-4
[root@blade1 root]# rpm -qa | grep mod_perl
[root@blade1 root]# rpm -qa | grep php-
asp2php-0.76.2-1
php-pgsql-4.1.2-7.3.4
php-devel-4.1.2-7.3.4
asp2php-gtk-0.76.2-1
php-imap-4.1.2-7.3.4
```

```
php-ldap-4.1.2-7.3.4
php-4.1.2-7.3.4
[root@blade1 root]#
```

In our case, we see that the Perl module is not installed, so we need to install it.

2. Check on the Red Hat Linux 7.3 Errata page to see if there are any updates required for any of these modules. If so, download these from the Red Hat FTP site or any of their mirrors. Again, for SLES contact SuSE for the exact location.
3. If you need to install a module that does not have an update, you can find it on your Red Hat Linux CDs or on the Red Hat FTP site.
4. Install the missing module(s) and the updates you might have. In our case, we have to install the Perl module and a update for the SSL module.

Example 4-4 Upgrade/Installation of Apache modules

```
[root@portal1 root]# rpm -Uvh mod_ssl-2.8.7-6.i386.rpm mod_perl-1.26-5.i386.rpm
Preparing...                ########################################### [100%]
   1:mod_ssl                ########################################### [ 50%]
   2:mod_perl               ########################################### [100%]
[root@portal1 root]#
```

> **Note:** The php module is made up of different rpm packages. This is because php itself also uses a system of modules that allow you to extend the capabilities of php. The core php package is named `php-<version number>.`
> On SLES, everything is integrated into one package.

If you see no errors in the installation process then the modules were installed correctly and our Apache setup is now ready to go.

4.1.2 Testing of Apache and mod_ssl

Now that we have a complete setup of our Apache server, let us see if it works. The default configuration of Apache as delivered in the Red Hat RPM package does not need to be changed for our basic tests.

Let us start it up. Most Linux distributions use a standard way of starting and stopping services like Apache. This is done using scripts located in the directory init.d. On Red Hat Linux 7.3 the complete path is /etc/init.d, and this is also the default location on many other Linux distributions. In there you should also find a script called httpd. This script is the one that will allow us to start and stop Apache.

Red Hat Linux has a set of commands that allows easy management of these services. To start and stop services, we use the **`service`** command. To get a list of what options we have for the Apache service script, we use this:

Example 4-5 Starting of Apache

```
[root@blade1 root]# service httpd
Usage: httpd {start|stop|restart|condrestart|reload|status|fullstatus|graceful|help|
configtest}
[root@blade1 root]# service httpd start
Starting httpd:                                            [  OK  ]
[root@blade1 root]#
```

You see that without an argument the **service** command will tell us what the options are. So we start Apache using **service httpd start**. You should get an OK back as shown in the example. This means that Apache has started without errors and is running now.

On SLES, the easiest way to start and stop Apache is by using the **apachectl** command. For more information about this command, see 5.1.2, "The apachectl tool" on page 68.

So now you can open a browser and put the domain name or IP address of your server into the URL bar of your browser. You should see a test page come up. If so, Apache is working.

The default Red Hat Apache configuration file also includes the necessary statements to enable the mod_ssl module. This allows Apache to have encrypted communications with the browser. To test this, simply change the `http://` section of the URL to `https://`. In our example, this then would be `https://blade1/`.

On SLES, you need to edit the file /etc/sysconfig/apache and set the line `HTTPD_SEC_MOD_SSL=no` to `=yes`. Then run the command **SuSEconfig**. This will update the Apache configuration files. Finally, run the command **cd /usr/share/doc/packages/mod_ssl; ./certificate.sh** as root and answer all the questions asked.

This last command will set up a certificate for Apache. When this command was run, you were also asked to type in a pass phrase. Now when you start Apache on SLES with **apachectl startssl**, you will be asked to enter that same pass phrase.

When you enter the changed URL, you will receive an error indicating that there is something wrong with the certificate of the server. The exact error/warning/alert depends on your browser. You should accept the warning and say you want to continue.

The page you requested is the same, so you cannot see the page itself that is sent over a secured link. But the `https://` URL and the closed lock symbol in the status bar of your browser indicate that the page was send in an encrypted form.

4.1.3 Testing of mod_perl

When installed using the Red Hat RPM packages, the mod_perl module is not enabled by default.This will be the first step, then we will create a little test script to see if our Perl code is being executed by mod_perl.

On SLES, mod_perl is enabled by default. To find the correct place to put your Perl script with SLES, see the Apache configuration file at /etc/httpd/httpd.conf. The Apache server root in SLES is /srv/www.

1. Open the Apache configuration file (called httpd.conf) in an editor. You will probably have to do this as root. On Red Hat Linux systems, this will be in the /etc/httpd/conf directory. In the configuration file, look for the following text.

Example 4-6 Configuration of mod_perl in /etc/httpd/conf/httpd.conf

```
#
# If the perl module is installed, this will allow execution of mod_perl
# to compile your scripts to subroutines which it will execute directly,
# avoiding the costly compile process for most requests.
#
#<IfModule mod_perl.c>
#    Alias /perl /var/www/perl
#    <Directory /var/www/perl>
#        SetHandler perl-script
#        PerlHandler Apache::Registry
#        Options +ExecCGI
#    </Directory>
#</IfModule>
```

2. In the editor, remove the # character from each line starting with the `<IfModule...` line and ending with the `</IfModule>` line. The # character is commonly used in Linux as an indicator for a comment line.
3. Save your changes and exit the editor.
4. Then let Apache reload the new configuration file by typing `service httpd reload` on the command line (still as root user).
5. Go to the /var/www directory using **`cd /var/www`**. In this directory, you should put all the different document types that Apache will serve.
6. Create a new `perl` directory with **`mkdir perl`**. This directory should contain all your Perl code for your Web site.
7. Create a new file called test.pl in the Perl directory you just created. Add the following content into this file.

Example 4-7 Test Perl code for mod_perl

```
use strict;
my $r = shift;
$r->send_http_header('text/html');
$r->print("It worked!!!\n");
```

8. Because this Perl code is will be executed by the mod_perl module, we need to allow the execution of our test Perl script by using **`chmod a+x test.pl`**. This allows everyone to execute your script.
9. In a browser type the URL `http://<servername>/perl/test.pl`, replacing `<servername>` with the actual DNS name or IP address of your server. You should see a white page with the text `It worked!!!`. If so, then mod_perl is working.

4.1.4 Testing of PHP

The final module that we will test is PHP. PHP allows you to extend regular HTML files by putting in code into your HTML files that will be executed every time the file is send to a browser.

PHP is enabled by default in Red Hat Linux as well as in the SLES installation. So we only need to create a test file that has some PHP code in it.

1. Go to the directory /var/www/html for Red Hat and /srv/www/htdocs for SLES. This directory contains all HTML files for Apache and since PHP is embedded into HTML files, this is the place to put them.
2. Create a new file called test.php using your favorite editor. We use the extension `.php` so we can differentiate between normal HTML files and the ones that also contain PHP code.
3. Put the line `<?php phpinfo(); ?>` into the test.php file and save the changes. Everything between the `<?php` and `?>` tags will be executed by the PHP module as being PHP code.
4. In a browser type the URL `http://<servername>/test.php`, replacing `<servername>` with the actual DNS name or IP address of your server. You should see a long page that begins with showing you the version of PHP you are using. If so, then PHP is working.

   If you have installed any extra PHP modules on Red Hat Linux, such as php-ldap, then browse through the whole page and see if these modules are listed.

4.2 Installation of Apache HTTP Server V2.0

The easiest way to install the Apache 2.0 is to install Red Hat Linux 8.0. It includes Apache 2.0 plus the SSL, Perl and PHP modules compiled for this version. Since we are using Red Hat Linux 7.3 on our blade servers, we will have to use another way to install Apache 2.0 since you cannot have the Apache RPM packages from Red Hat 8.0 and Red Hat 7.3 installed at the same time.

For SuSE users, the SuSE Linux 8.1 Professional edition contains both Apache 1.3 and Apache 2.0.

We will compile Apache 2.0 from the source in a separate directory so that it does not interfere with the Apache 1.3 installation. The SSL module comes with a base Apache source package. The Perl and PHP modules will also be installed from source.

> **Note:** You still can run only one Apache at a time because standard HTTP and HTTPS ports are required by Apache. If you want both versions running at the same time, you will have to configure one Apache to use non-standard ports.

4.2.1 Installation of Apache HTTP Server and the SSL module

1. Download the latest version of the `apache2` source from:

   `http://www.apache.org/dist/httpd`

   We downloaded the httpd-2.0.43.tar.gz package.
2. Extract the source with **`tar -xzf httpd-2.0.43.tar.gz`**.
3. Go into the source directory: **`cd httpd-2.0.43`**.
4. Configure the source with: **`./configure --with-mpm=prefork --enable-modules=all --enable-mods-shared=all --enable-cgi --enable-ssl --enable-proxy --enable-so --with-ssl=/usr/include/`**.

   We configure Apache to build all modules as shared modules, also to build the SSL and proxy modules. We also choose the **prefork** processing module. This is a non-threaded module. This is because the Perl in Red Hat 7.3 is still V5.6. Perl 5.6 does not support multi-threading; for that, you need to have Perl V5.8.
   If you do not want to build mod_perl, you can choose a threaded processing module like *worker*.

   SLES 8 includes Perl 5.8, so you can build a threaded processing module with mod_perl.
5. Start the compilation with the **`make`** command. This could take a while to finish.
6. If the compilation was successful, you can install Apache 2.0 with **`make install`**.

Apache 2.0 will now be installed in the directory /usr/local/apache2. This way of installing Apache will not create a script for starting and stopping Apache in the /etc/init.d directory. This means you cannot use the **`service`** and the **`chkconfig`** commands.

4.2.2 Installation of the Perl module

1. Download the latest version of the Perl module from `http://perl.apache.org/dist/`. You will need to get the 2.0 or later version of mod_perl. The older versions (1.x) are not compatible with Apache 2.0. In our case the package was named mod_perl-2.0-current.tar.gz.
2. Extract the mod_perl source with **`tar -xzf mod_perl-2.0-current.tar.gz`**.

3. Go into the source directory, cd mod_perl-1.99_07/.
4. Configure mod_perl with: **`perl Makefile.PL MP_AP_PREFIX=/usr/local/apache2/ MP_INST_APACHE2=1`**.

   We point mod_perl to our Apache 2.0 directory and tell it to ignore the already installed V1.3 of Apache and its modules.
5. Compile mod_perl with **`make`**.
6. Test the newly created module with **`make test`**.
7. If the test completed with no errors, you can install mod_perl with **`make install`**.

Now mod_perl has been installed into our Apache 2.0 directory.

4.2.3 Installation of the PHP module

1. Download the latest version of the PHP source from http://www.php.net/downloads.php. In our case, the package was called php-4.2.3.tar.gz.
2. Extract the PHP source code with **`tar -xzf php-4.2.3.tar.gz`**.
3. Go into the source directory with **`cd php-4.2.3`**.
4. Configure the PHP module with the command as in Example 4-8.

Example 4-8 Configuration of PHP module

```
./configure --prefix=/usr/local/apache2/ --with-config-file-path=/usr/local/apache2/conf/
--enable-force-cgi-redirect --disable-debug --enable-pic --disable-rpath
--enable-inline-optimization --with-bz2 --with-db3 --with-curl --with-dom=/usr
--with-exec-dir=/usr/local/apache2/bin --with-freetype-dir=/usr --with-png-dir=/usr
--with-gd --enable-gd-native-ttf --with-ttf --with-gdbm --with-gettext --with-ncurses
--with-gmp --with-iconv --with-jpeg-dir=/usr --with-openssl --with-png --with-pspell
--with-regex=system --with-xml --with-expat-dir=/usr --with-zlib --with-layout=GNU
--enable-bcmath --enable-exif --enable-ftp --enable-magic-quotes --enable-safe-mode
--enable-sockets --enable-sysvsem --enable-sysvshm --enable-discard-path
--enable-track-vars --enable-trans-sid --enable-yp --enable-wddx --without-oci8
--with-pear=/usr/local/apache2/pear --with-imap --with-imap-ssl
--with-kerberos=/usr/kerberos --with-ldap --with-mysql=/usr --with-pgsql
--enable-memory-limit --enable-shmop --enable-versioning --enable-calender --enable-dbx
--enable-dio --enable-mcal --with-apxs2=/usr/local/apache2/bin/apxs.
```

   We have used a large number of options; you are free to remove or add options to suit your environment. Some of these options also require that the devel RPM package(s) for that option be installed. If you see errors in the configuration process, either install the necessary devel package or remove the option.

5. Compile the PHP module with **`make`**.
6. If the compilation completed with no errors, install the PHP module with **`make install`**.

Our Apache installation is complete now. The next task will be to configure Apache and test it.

4.2.4 Configuration and testing of Apache

In the /usr/local/apache2 directory, you will now find our complete Apache 2.0 environment. We will call this directory the Apache 2.0 root. In this section, all references for files and directories will be against this root unless explicitly mentioned.

Let us start with the configuration.

1. Go to the conf/ directory and open the file httpd.conf in a editor.
2. Scroll down until you come to a set of lines all beginning with `LoadModule`. At the end of this section, add the line `LoadModule perl_module modules/mod_perl.so`. This will load the Perl module.

   You should verify that the line `LoadModule php4_module modules/libphp4.so` is also present in that section. In our case it was. This will load the PHP module.

   The SSL module, as part of the basic Apache 2.0 distribution, is already installed and configured.
3. Scroll further down until you come to the title `### Section 3: Virtual Hosts`. Edit this section so that it looks like the following.

Example 4-9 Configuration of mod_perl and php in httpd.conf

```
#
# Bring in additional module-specific configurations
#
<IfModule mod_ssl.c>
    Include conf/ssl.conf
</IfModule>

# mod_perl configuration

PerlModule Apache2

Alias /perl/ /usr/local/apache2/perl/
<Location /perl/>
    SetHandler perl-script
    PerlResponseHandler ModPerl::Registry
    PerlOptions +ParseHeaders
    Options +ExecCGI
</Location>

# php configuration
AddType application/x-httpd-php .php .php4 .php3 .phtml
AddType application/x-httpd-php-source .phps


### Section 3: Virtual Hosts
```

   The bold section mark the things that we had to add. Close the httpd.conf file , saving the changes you made.

4. Copy the file php.ini-recommended from the source directory of PHP (were you did the configure, make, etc.) and copy if into the conf/ directory with the name php.ini.
   In our case, it looked like this: `cp /root/apache/php-4.2.3/php.ini-recommended /usr/local/apache2/conf/php.ini`.
5. If you want to use the SSL module with Apache 2.0, then you will need to generate a key for use in the encryption process. In you do not intend to use SSL, you can skip to the next step.

   The easiest way to get this key is to copy one from the Apache 1.3 distribution. If you want a real key, follow the instructions in the SSL module documentation.

To copy the Apache 1.3 key, we did this:

Example 4-10 Copying the Apache 1.3 key

```
[root@portal1 root]# cp -a /etc/httpd/conf/ssl.crt/ /usr/local/apache2/conf/
[root@portal1 root]# cp -a /etc/httpd/conf/ssl.key/ /usr/local/apache2/conf/
```

Your Apache 2.0 configuration is now complete. The next step will be to create two small test scripts for mod_perl and PHP and to test everything.

1. In the Apache 2.0 root , create a directory perl/ using **`mkdir perl`**.
2. In this directory, create a file called test.pl ; it should have the same content as in Example 4-7 on page 61.
3. In the directory htdocs/ create a file called test.php and put the line `<?php phpinfo(); ?>` into it.
4. Start Apache 2.0 with the command **`./bin/apachectl startssl`**. Remember that you should do this in the Apache 2.0 root. If you do not want to use SSL in the command, replace `startssl` with `start`.
5. Open a browser and type the name of your server in the URL bar. You should see a test page for Apache.
6. In the URL bar of your browser, replace `http://` with `https://` to test SSL. You should see the same page again, but this time *encrypted* as indicated by the dialogs that may appear and the closed lock in the status bar of your browser.
7. Add `test.php` or `/test.php` behind the URL. Now you should see an information page from PHP. At the top, the version of PHP is indicated.
8. In the URL, replace `test.php` with `perl/test.pl`. You now should see a page with the message `It worked !!!`.

If you were able to see every test page, then your Apache 2.0 setup is configured and working correctly.

5

Apache management

In this chapter, we will discuss three ways to configure and manage your Apache server. They are:

- Using command-line tools
- Using the Webmin Web based interface
- Using the Red Hat GUI tool

5.1 Command-line management of Apache

In this section, we will go over some useful command-line tools that you can use to manage the Apache Web server.

5.1.1 The httpd.conf file

This is the main configuration file of Apache and therefore the place to make changes to the behavior of Apache. The location this file in our case is /etc/httpd/conf/httpd.conf for the Apache 1.3 installation on Red Hat, /etc/httpd/httpd.conf for the Apache 1.3 installation on SLES and /usr/local/apache2/conf/httpd.conf for the Apache 2.0 installation.

The httpd.conf file is separated into three sections:

- Global environment
- Main server configuration
- Virtual hosts

In the Global environment section, you can control the Apache server as a whole. This, for example, means that you determine the modules to be loaded, and other options relative to how the Apache server itself will work.

In the Main server configuration, you configure the default server. This is the server that will respond if none of the virtual servers that are defined match the request of the client.

In the Virtual hosts section, you can define all the virtual servers that you want plus their options.

You can also include other files into the main Apache configuration using the `<Include>` directive. This is frequently used to put each virtual host configuration into a separate file. This makes it a bit easier to keep a clear overview.

For the complete documentation on the Apache configuration file syntax, we refer you to the Apache documentation at:

http://httpd.apache.org/docs/ (for Apache 1.3)
http://httpd.apache.org/docs-2.0/ (for Apache 2.0)

5.1.2 The apachectl tool

The apachectl script allows you to perform a couple of basic functions and is similar to the functions of the service **`httpd`** command. The parameter given to apachectl will determine which function will be performed. The different parameters are:

- **start**: this starts the Apache HTTP server.
- **startssl**: this start the Apache HTTP server with the SSL module (only for SLES and Apache 2.0).
- **stop**: this stops Apache.
- **restart**: this restarts Apache.
- **status**: this gives a short overview of what Apache is doing.
- **fullstatus**: this gives a more detailed view of what Apache is doing.

- **graceful**: this restarts Apache without terminating processes. It is useful for loading configuration changes.
- **configtest**: this tests the Apache configuration file for any syntax errors.

In the default Red Hat configuration, the **`status`** and **`fullstatus`** commands do not work. This is because a command that apachectl depends on is not installed and because of some configuration statements in the Apache configuration file that need to be activated. Here is how you can get them to work.

1. The apachectl script look for lynx to display the status. However, Red Hat 7.3 comes by default with links, which is a replacement for lynx. Therefore, open the apachectl script in an editor. The default location is /usr/sbin/apachectl.
2. Look for the line: `LYNX="lynx -dump"`.
3. Replace `lynx` with `links`.
4. Save your changes and close the editor.
5. The status that will be shown by the apachectl script is actually a Web page that is generated by a module from Apache. You can also use a regular browser to see this page. On Red Hat, this status page is disabled by default. So we will need to enable it.
   Open the Apache configuration file in a editor.
6. Look for the line: `#<Location /server-status>`.
7. Remove the # character from every line between `#<Location /server-status>` and `</#Location>`, including these lines as well.
8. Save your changes and close the editor.
9. Load these configuration changes using **`apachectl graceful`**.
10. Test our changes by **`apachectl status`** or **`apachectl fullstatus`**.

You can also request a even more detailed overview by uncommenting the line `#ExtendedStatus On` in the Apache configuration file.

On SLES, there is no **`lynx`** or **`links`** command, so you cannot access the Apache status pages using apachectl. The server status pages are enabled by default, so you can use a normal browser to view these pages.

5.1.3 The ab tool

This is a basic benchmarking tool. It allows you to test a certain URL. You can define the number of requests it will make, how many it will make at the same time, and how long it will test for. It also knows the basic HTTP features such as authentication, cookies, GET/POST and headers. A example of a command and some of the resulting data follows.

Example 5-1 Example of the ab tool

```
[root@blade1 root]# ab -n 1000 -c 20 http://blade1/server-status

...

Requests per second:    1042.75 [#/sec] (mean)
Time per request:       19.18 [ms] (mean)
Time per request:       0.96 [ms] (mean, across all concurrent requests)
Transfer rate:          2109.05 [Kbytes/sec] received

Connnection Times (ms)
              min  mean[+/-sd] median   max
Connect:        0     4    1.4      4     9
```

```
Processing:    10    13   15.2     12   488
Waiting:        6    13   15.2     12   487
Total:         10    18   15.2     17   493

...
```

Here we requested the URL `http://blade1/server-status` a thousand times with twenty requests at the same time. We also put some of the returning data in the example and you can see that we had approximately 1050 requests per second.

The exact configuration options can be found on the manual page that you can view with the **man ab** command.

5.1.4 The dbmmanage, htdigest and htpasswd commands

These three commands are all used to create and manage files that contain data used in authentication of users. They basically allow you to put in a password linked with a user.

For more information about the different ways to perform access control in Apache, we refer you to the Authentication, Authorization, and Access Control chapter of the Apache documentation at http://httpd.apache.org/docs/howto/auth.html for Apache 1.3 or http://httpd.apache.org/docs-2.0/howto/auth.html for Apache 2.0.

5.1.5 The logresolve and rotatelogs commands

These commands allow you to manage the logs created by Apache.

With **logresolve** you can do a DNS lookup for all the client IP addresses in the Apache logs. Apache can do this on its own, but this take time and is therefore usually turned off. With **logresolve** you can do this at a later time. **Logresolve** uses a hash table so it remembers IP addresses it has already looked up and then creates a new log file with the DNS names instead of the IP addresses.

With **rotatelogs** you can have Apache automatically create a new log file at a certain interval without having to stop Apache for this. The two options are the name of the log file to be written and the lapse of time after which to create a new log file.

The exact syntax and options can be found in the manual pages of these commands. You can access the manual pages with the commands **man logresolve** or **man rotatelogs**.

5.1.6 Other tools and commands

There are also some standard UNIX/Linux tools that you can use for managing Apache. For example, with **ps aux** and **top** you can see what Apache is doing on a process level. With **netstat -an** you can see what network connections are open and in what state they are.

With **free**, **iostat**, **vmstat** and **mpstat** you can get information about the Linux system in general in terms of memory, CPU usage, I/O activities, etc.

For more information about each of these tools, we refer you to their manual page.

5.2 Web-based management using Webmin

Webmin is an Open Source Web-based interface for system administration for UNIX. Support is present for nearly all major Linux, BSD and UNIX systems. Using any browser that supports tables and forms (and Java for the File Manager module), you can set up user accounts, Apache, DNS, file sharing and so on.

Webmin consists of a simple Web server and a number of CGI programs which directly update system files like /etc/inetd.conf and /etc/passwd. The Web server and all CGI programs are written in Perl V5, and use no non-standard Perl modules.

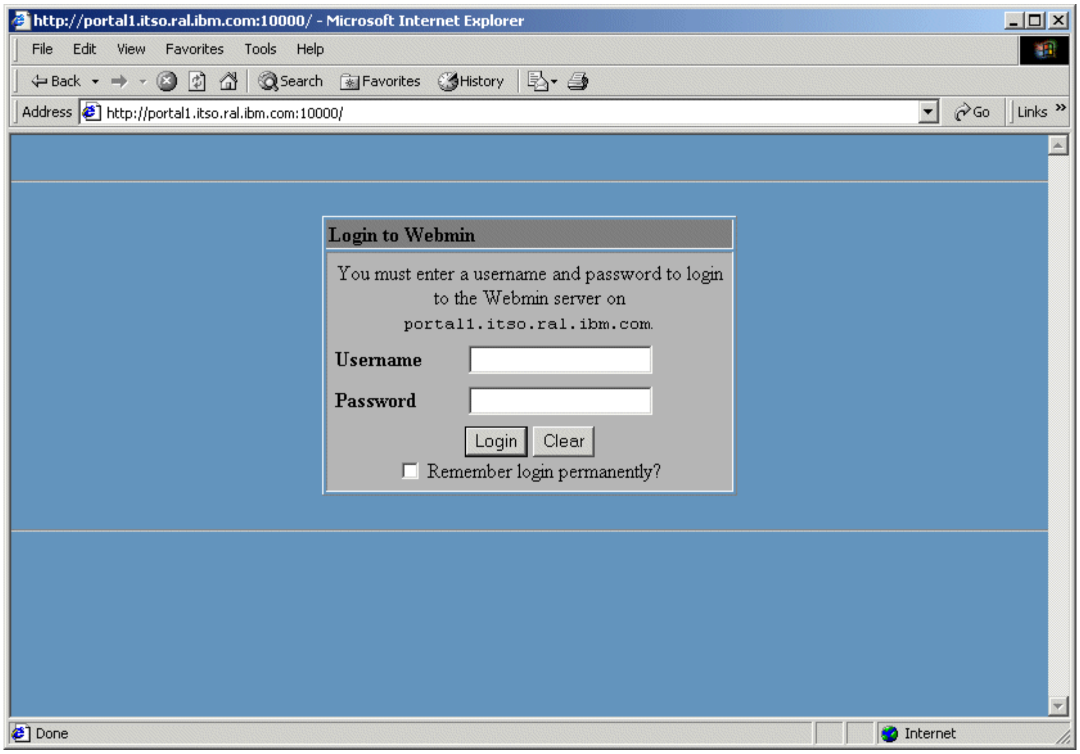
Webmin is available in different languages and uses modules to extend its functionality. This means that you can add extra modules besides the default ones or create new modules yourself.
You can find more information and download the latest version of Webmin at:

`http://www.Webmin.com`

5.2.1 Installation of Webmin

In this section, we begin with the installation of Webmin. Perform the following steps:

1. Download the latest RPM version of Webmin from the Webmin Web site. In our case, the RPM was called Webmin-1.030-1.noarch.rpm.
2. Install the RPM using **`rpm -ivh Webmin-1.030-1.noarch.rpm`**.
3. You should see a message that the install is complete and that you can surf to the server's name and log in as user root.
4. Open a browser and use following URL: `http://<servername>:10000/`, replacing `<servername>` with the actual DNS name or IP address of your server.
   Webmin listens to port 10000 instead of the default HTTP port (80) and that explains the :10000 in the URL.
5. You should see a window similar to Figure 5-1 on page 72.



http://portal1.itso.ral.ibm.com:10000/ - Microsoft Internet Explorer

Address: http://portal1.itso.ral.ibm.com:10000/

Login to Webmin

You must enter a username and password to login to the Webmin server on portal1.itso.ral.ibm.com.

Username

Password

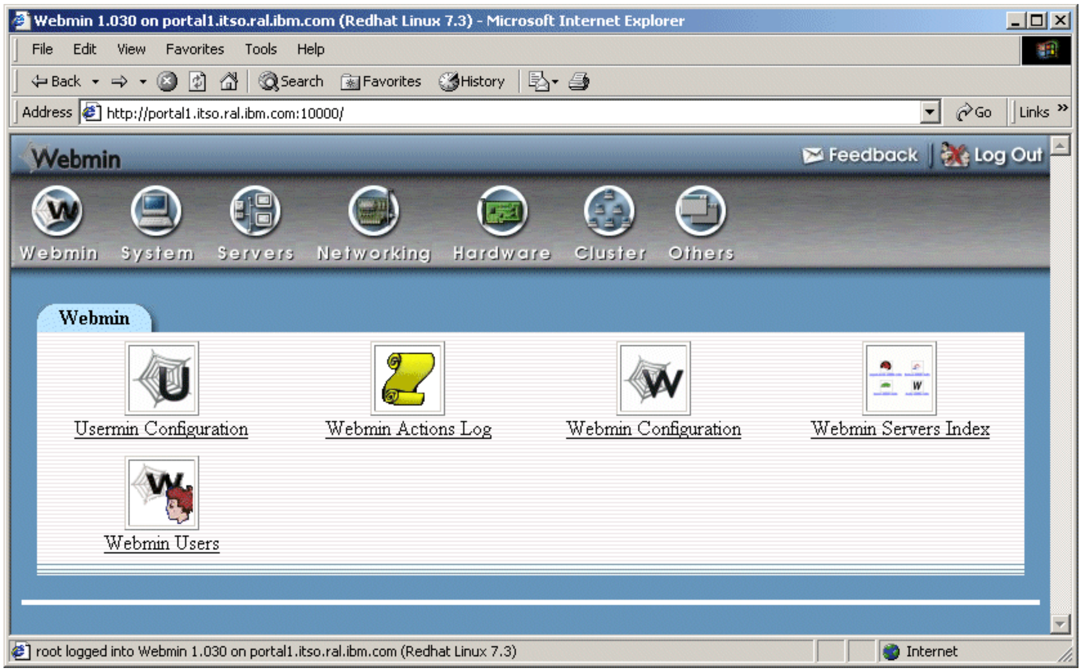
Login Clear

□ Remember login permanently?

Done

Figure 5-1 Login window of Webmin

6. In the Username field, enter `root` and in the Password field enter your root password. Then click **Login**. You should see a window similar to Figure 5-2. If you are able to access, then Webmin is working correctly. The next thing that we will do is enable Webmin to use SSL to secure the data sent and received.



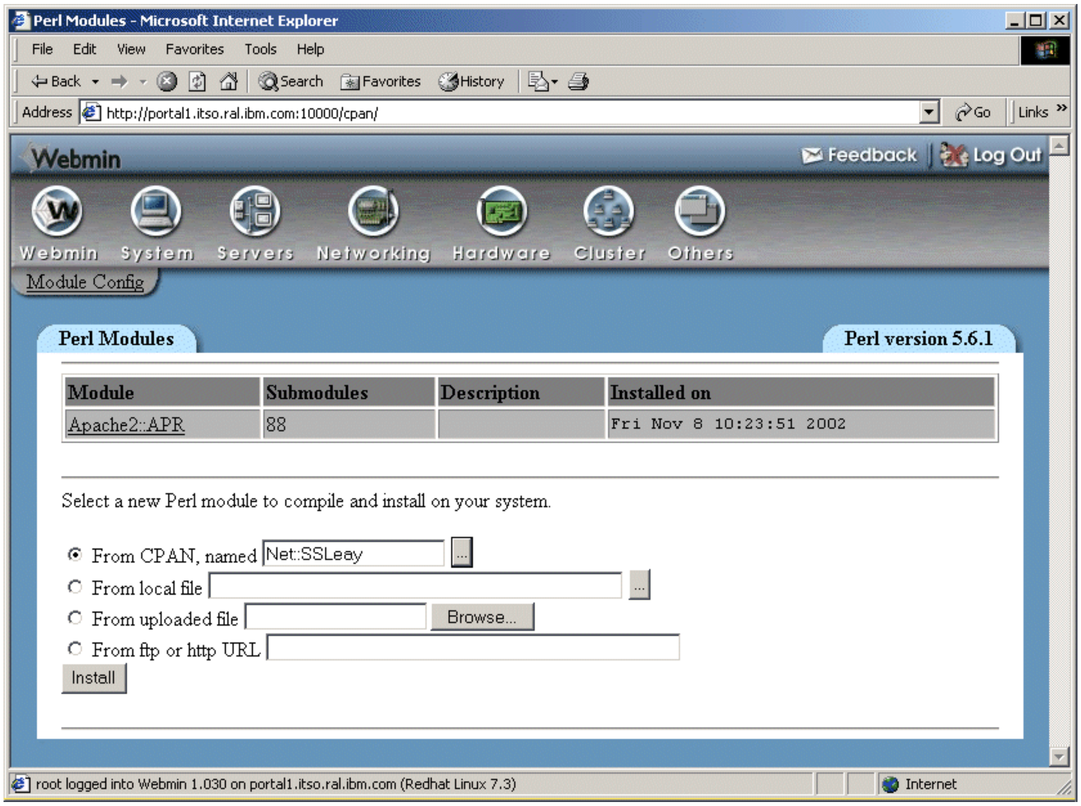
Webmin 1.030 on portal1.itso.ral.ibm.com (Redhat Linux 7.3) - Microsoft Internet Explorer

Address: http://portal1.itso.ral.ibm.com:10000/

root logged into Webmin 1.030 on portal1.itso.ral.ibm.com (Redhat Linux 7.3)

Figure 5-2 Default page of Webmin

7. Click **Others** and then click **Perl Modules**. You will see a window similar to Figure 5-3.


Perl Modules - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address http://portal1.itso.ral.ibm.com:10000/cpan/
Go Links
Webmin
Feedback Log Out
Webmin System Servers Networking Hardware Cluster Others
Module Config
Internet

Perl Modules

Perl version 5.6.1

Module	Submodules	Description	Installed on
Apache2::APR	88		Fri Nov 8 10:23:51 2002

Select a new Perl module to compile and install on your system.

● From CPAN, named Net::SSLeay

○ From local file

○ From uploaded file Browse...

○ From ftp or http URL

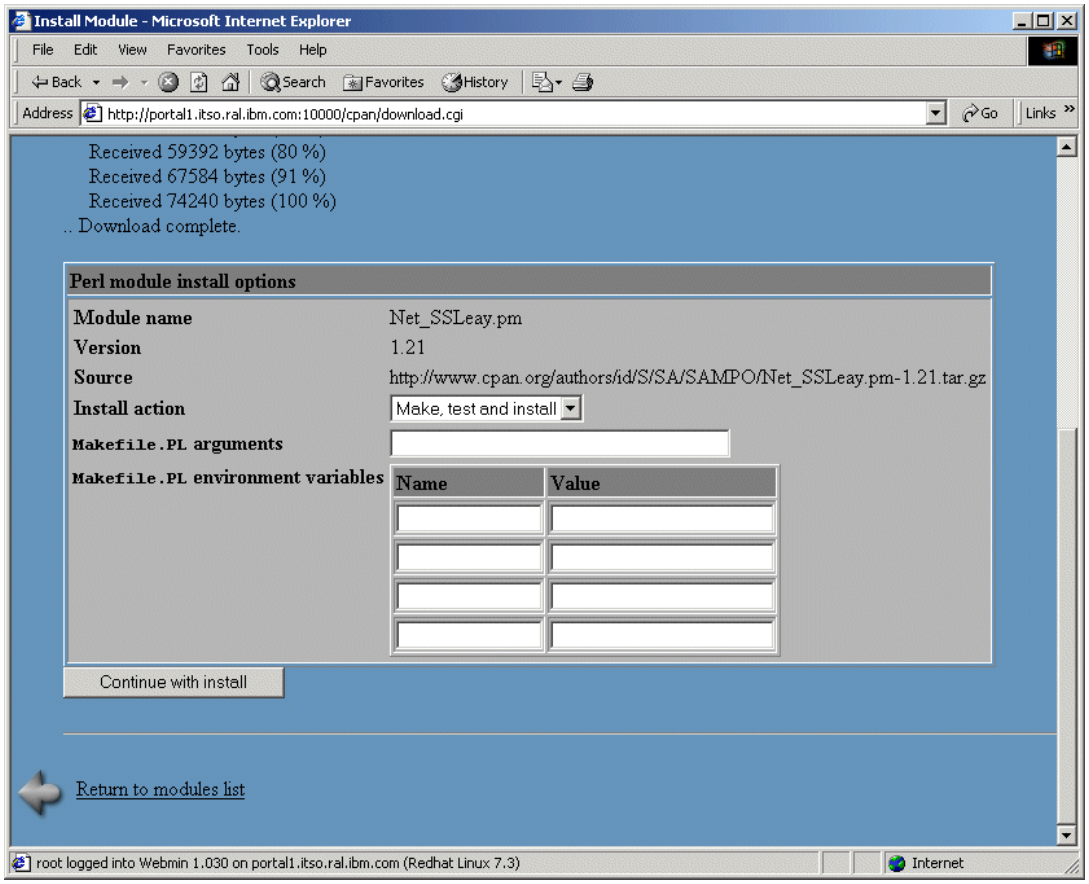
Install

root logged into Webmin 1.030 on portal1.itso.ral.ibm.com (Redhat Linux 7.3)

Figure 5-3 Perl Modules window

8. Fill in the field From CPAN, named with the value `Net::SSLeay`. Click **Install**.

> **Note:** Your server needs to have access to the Internet for this. If you need to pass through proxy servers, you can enter these in the **Webmin -> Webmin Configuration -> Proxy Servers** page. Otherwise, download the Net:SSLeay Perl module using other means and use any of the other options that Webmin offers you.


Install Module - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
History
Address
http://portal1.itso.ral.ibm.com:10000/cpan/download.cgi
Go
Links
root logged into Webmin 1.030 on portal1.itso.ral.ibm.com (Redhat Linux 7.3)
Internet

```
    Received 59392 bytes (80 %)
    Received 67584 bytes (91 %)
    Received 74240 bytes (100 %)
.. Download complete.
```

Perl module install options

Module name	Net_SSLeay.pm
Version	1.21
Source	http://www.cpan.org/authors/id/S/SA/SAMPO/Net_SSLeay.pm-1.21.tar.gz
Install action	Make, test and install
`Makefile.PL` arguments	
`Makefile.PL` environment variables	

Name	Value

Continue with install

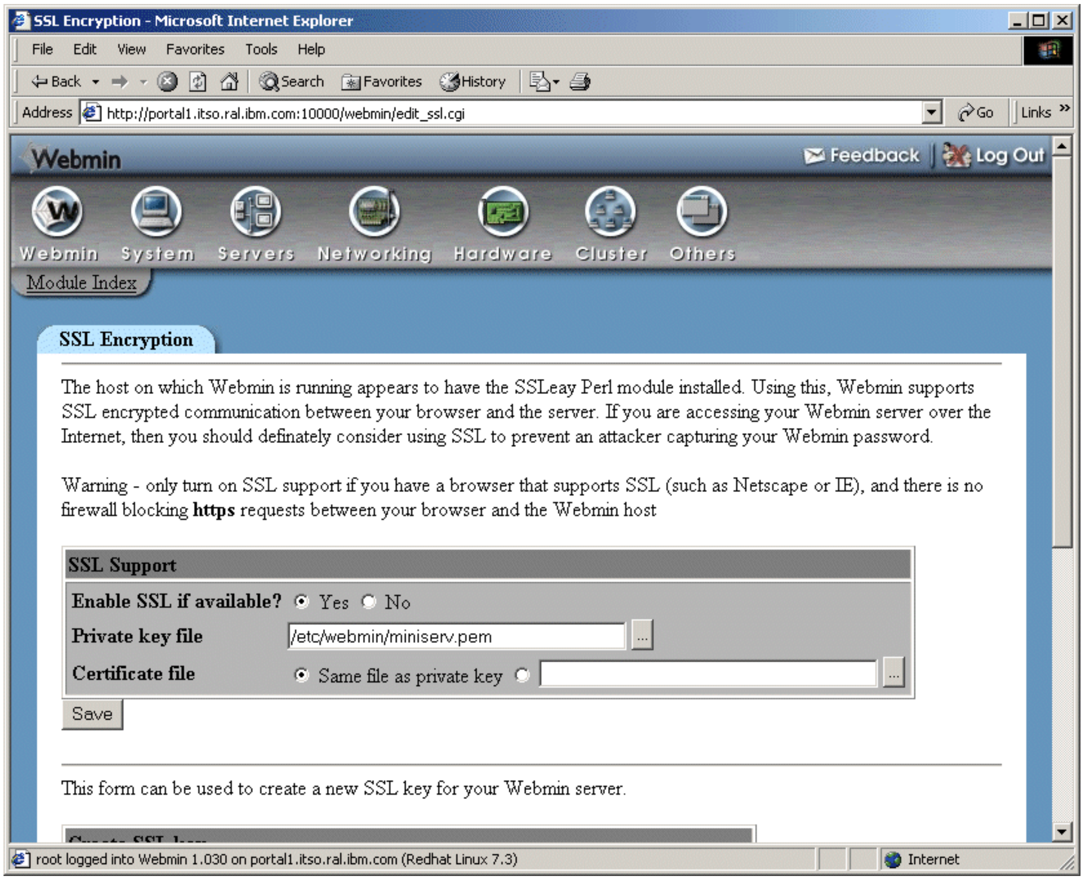
Return to modules list

Figure 5-4 Installation of Net_SSLeay module

9. Click **Continue with install**.

10. Wait until you see `Make, test and install of Net_SSLeay.pm successful` at the bottom of the Web page. The module is now correctly installed. This can take a while, but in our case we had to retry the installation.

11. Click **Webmin** (top of the page) **-> Webmin Configuration -> SSL Encryption**.


SSL Encryption - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address
Go
Links
Webmin
Feedback
Log Out
Webmin System Servers Networking Hardware Cluster Others
Module Index
Internet

http://portal1.itso.ral.ibm.com:10000/webmin/edit_ssl.cgi

SSL Encryption

The host on which Webmin is running appears to have the SSLeay Perl module installed. Using this, Webmin supports SSL encrypted communication between your browser and the server. If you are accessing your Webmin server over the Internet, then you should definately consider using SSL to prevent an attacker capturing your Webmin password.

Warning - only turn on SSL support if you have a browser that supports SSL (such as Netscape or IE), and there is no firewall blocking **https** requests between your browser and the Webmin host

SSL Support	
Enable SSL if available?	● Yes ○ No
Private key file	/etc/webmin/miniserv.pem
Certificate file	● Same file as private key ○

Save

This form can be used to create a new SSL key for your Webmin server.

root logged into Webmin 1.030 on portal1.itso.ral.ibm.com (Redhat Linux 7.3)

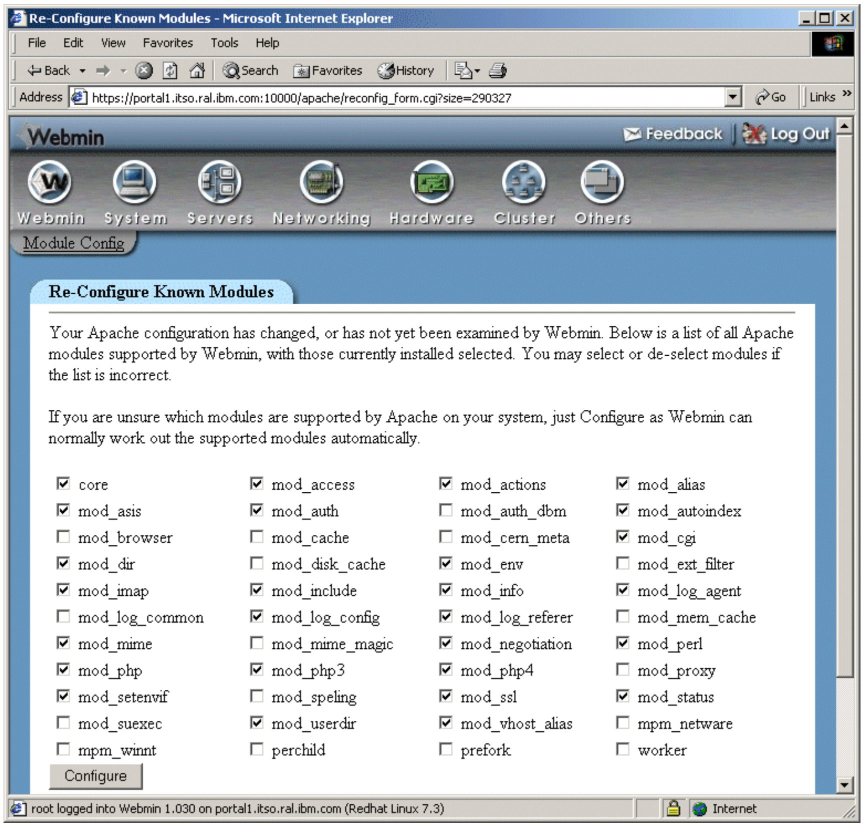
Figure 5-5 Webmin SSL settings

12. Click **Save**.
13. Webmin will now reload using SSL/HTTPS. This means that you might receive a dialog box with a warning from your browser about the certificate; accept it.

Your Webmin management interface is now ready.

5.2.2 Using Webmin to manage Apache

Webmin offers a lot of possibilities but we will only zoom in on the Apache module from Webmin. If you want to learn more about Webmin then we suggest you just play with it or consult the documentation on the Webmin home page.


Re-Configure Known Modules - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address https://portal1.itso.ral.ibm.com:10000/apache/reconfig_form.cgi?size=290327
Go Links
Webmin
Feedback
Log Out
Webmin System Servers Networking Hardware Cluster Others
Module Config

Re-Configure Known Modules

Your Apache configuration has changed, or has not yet been examined by Webmin. Below is a list of all Apache modules supported by Webmin, with those currently installed selected. You may select or de-select modules if the list is incorrect.

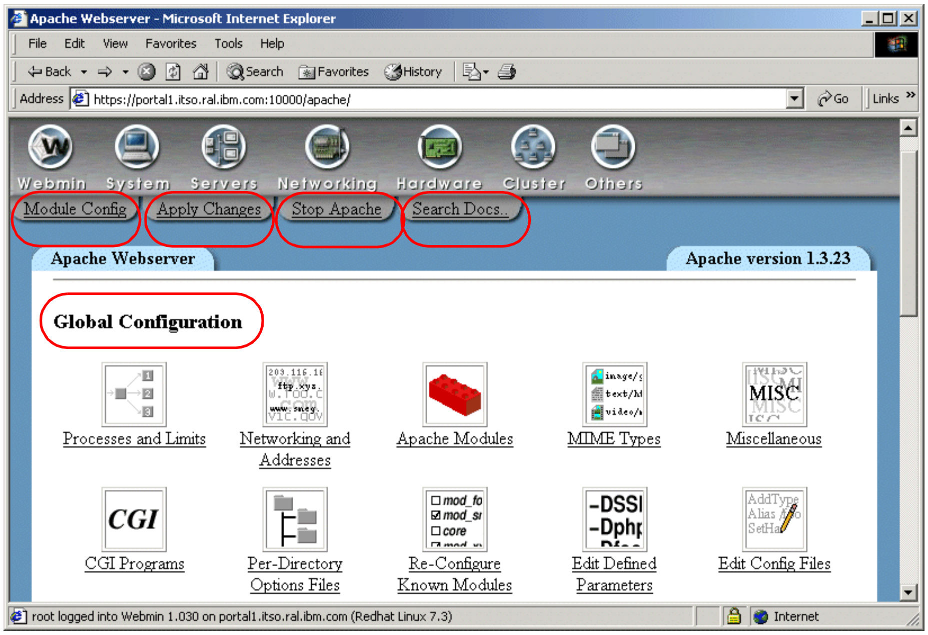
If you are unsure which modules are supported by Apache on your system, just Configure as Webmin can normally work out the supported modules automatically.

☑ core	☑ mod_access	☑ mod_actions	☑ mod_alias
☑ mod_asis	☑ mod_auth	□ mod_auth_dbm	☑ mod_autoindex
□ mod_browser	□ mod_cache	□ mod_cern_meta	☑ mod_cgi
☑ mod_dir	□ mod_disk_cache	☑ mod_env	□ mod_ext_filter
☑ mod_imap	☑ mod_include	☑ mod_info	☑ mod_log_agent
□ mod_log_common	☑ mod_log_config	☑ mod_log_referer	□ mod_mem_cache
☑ mod_mime	□ mod_mime_magic	☑ mod_negotiation	☑ mod_perl
☑ mod_php	☑ mod_php3	☑ mod_php4	□ mod_proxy
☑ mod_setenvif	□ mod_speling	☑ mod_ssl	☑ mod_status
□ mod_suexec	☑ mod_userdir	☑ mod_vhost_alias	□ mpm_netware
□ mpm_winnt	□ perchild	□ prefork	□ worker

root logged into Webmin 1.030 on portal1.itso.ral.ibm.com (Redhat Linux 7.3)

Figure 5-6 Re-configure known modules page

Go to the Apache module. From the Webmin menu, click **Servers** -> **Apache Webserver**. If this is the first time that you open the Apache module, then you will see a window similar to Figure 5-6. It asks you which modules from Apache you have installed. Click **Configure**; Webmin usually finds the correct ones. You will see a window similar to Figure 5-7.


Apache Webserver - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address https://portal1.itso.ral.ibm.com:10000/apache/
Go Links
Webmin System Servers Networking Hardware Cluster Others
Module Config
Apply Changes
Stop Apache
Search Docs..

Apache Webserver

Apache version 1.3.23

Global Configuration

Processes and Limits
Networking and Addresses
Apache Modules
MIME Types
Miscellaneous
CGI Programs
Per-Directory Options Files
Re-Configure Known Modules
Edit Defined Parameters
Edit Config Files

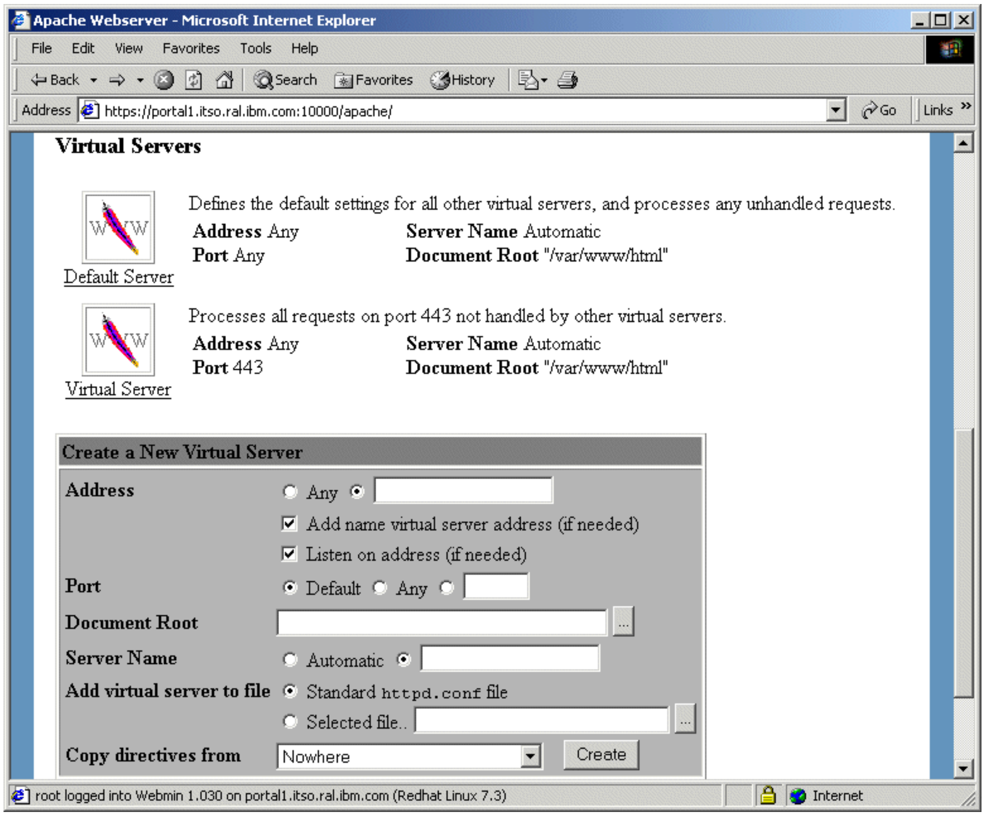
root logged into Webmin 1.030 on portal1.itso.ral.ibm.com (Redhat Linux 7.3)

Figure 5-7 Webmin Apache module - main section, part 1

Now you will see the main page of the Apache module. At the top, you can configure the Webmin Apache module with Module Config. You can restart Apache (Apply Changes), stop and start Apache (Stop Apache) and search for documentation on Apache (Search Docs.).

In the Global Configuration section, you can change the configuration of global Apache parameters such as which modules have to be loaded, which ports it has to listen on, and how many processes it has to start.

The Edit Config Files button allows you to edit the actual Apache configuration file; this is similar to using a editor.


Apache Webserver - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address
Go
Links
Default Server
Virtual Server
Create

https://portal1.itso.ral.ibm.com:10000/apache/

Virtual Servers

Defines the default settings for all other virtual servers, and processes any unhandled requests.

Address Any **Server Name** Automatic
Port Any **Document Root** "/var/www/html"

Processes all requests on port 443 not handled by other virtual servers.

Address Any **Server Name** Automatic
Port 443 **Document Root** "/var/www/html"

Create a New Virtual Server

Address	○ Any ●
	☑ Add name virtual server address (if needed)
	☑ Listen on address (if needed)
Port	● Default ○ Any ○
Document Root	
Server Name	○ Automatic ●
Add virtual server to file	● Standard `httpd.conf` file
	○ Selected file..
Copy directives from	Nowhere

root logged into Webmin 1.030 on portal1.itso.ral.ibm.com (Redhat Linux 7.3)

Internet

Figure 5-8 Webmin Apache module - main section, part 2

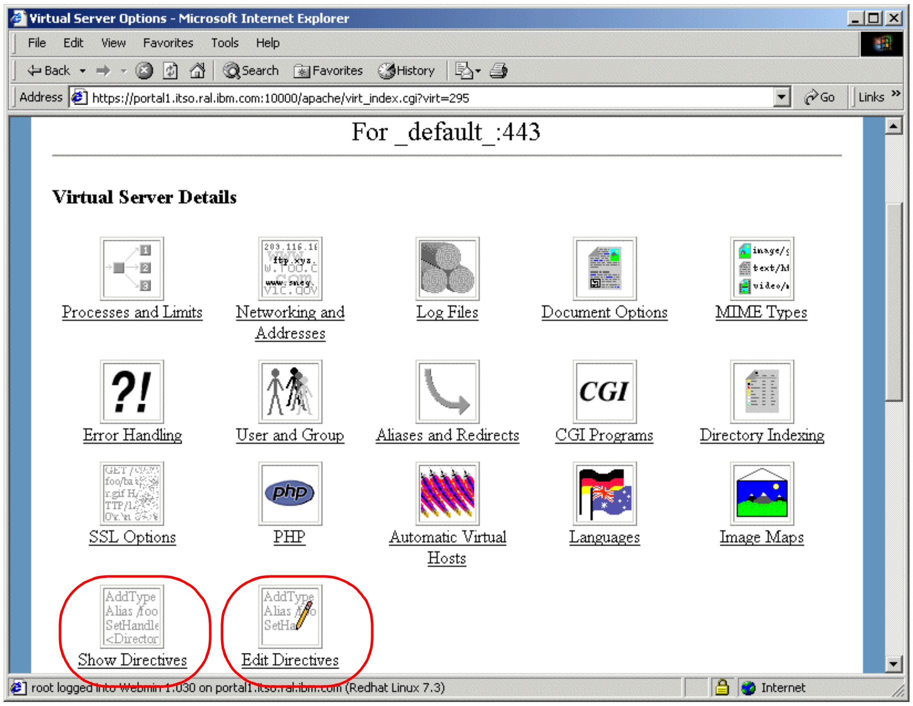
In the section Virtual Servers (Figure 5-8), you get an icon for each virtual server that is defined and the option to create a new virtual server.

Virtual servers are a technique that allows a single Web server to serve multiple Web sites, each with its own domain. There are two types of virtual servers. One is called IP-based virtual servers. The other is called name-based virtual servers.

With IP-based virtual servers, you need one IP address per Web site. The Web server needs to differentiate between the different virtual servers and uses the IP address requested by the client for that.

With name-based virtual servers, you only need one IP address per Web server. In this case, the client has to send a HTTP Host header is his request to allow the Web server to determine which virtual server is requested. This header is part of the HTTP/1.1 standard and is implemented in all modern browsers.

For more information, we refer you to the Apache documentation. In the Apache Virtual Host documentation, you will find more information about this and configuration. This documentation is at http://httpd.apache.org/docs/vhosts/ (*Virtual Host* is a synonym for Virtual Server).



Address: https://portal1.itso.ral.ibm.com:10000/apache/virt_index.cgi?virt=295

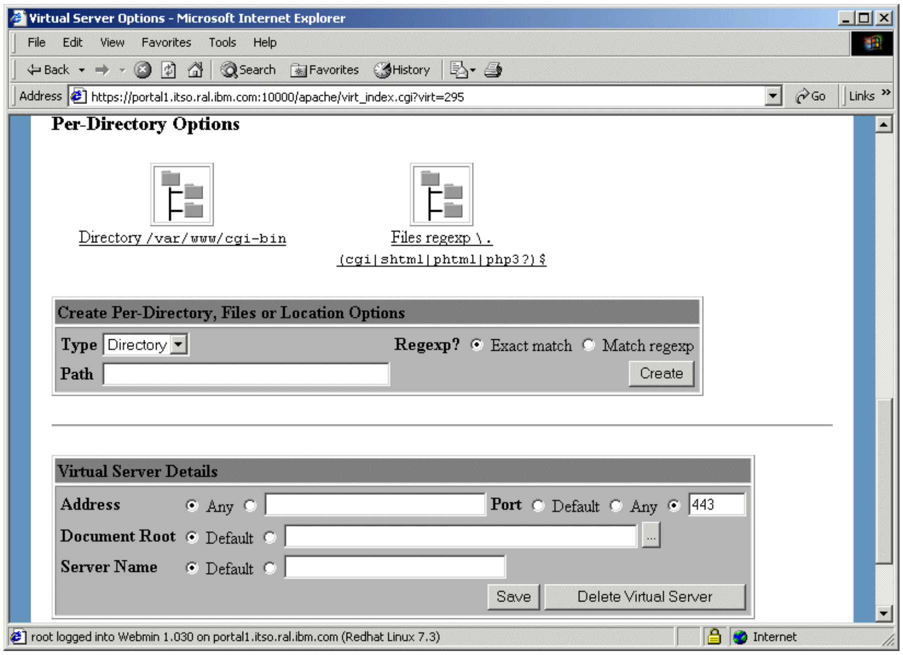
For _default_:443

Virtual Server Details

root logged into Webmin 1.030 on portal1.itso.ral.ibm.com (Redhat Linux 7.3)

Figure 5-9 Virtual server - main section

When you click the icon of a virtual server, you will get the virtual server details. Here, you can set the main options for each virtual server. With Show Directives and Edit Directives (Figure 5-9), you can see or edit the actual lines in the httpd.conf file.


Virtual Server Options - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address
Go
Links
Directory /var/www/cgi-bin
Files regexp \.(cgi|shtml|phtml|php3?)$
Create
Save
Delete Virtual Server
Internet

Address: https://portal1.itso.ral.ibm.com:10000/apache/virt_index.cgi?virt=295

Per-Directory Options

Create Per-Directory, Files or Location Options

Type Directory ▾ **Regexp?** ● Exact match ○ Match regexp

Path ______

Virtual Server Details

Address ● Any ○ ______ **Port** ○ Default ○ Any ● 443

Document Root ● Default ○ ______ ...

Server Name ● Default ○ ______

root logged into Webmin 1.030 on portal1.itso.ral.ibm.com (Redhat Linux 7.3)

Figure 5-10 Virtual server - directory options

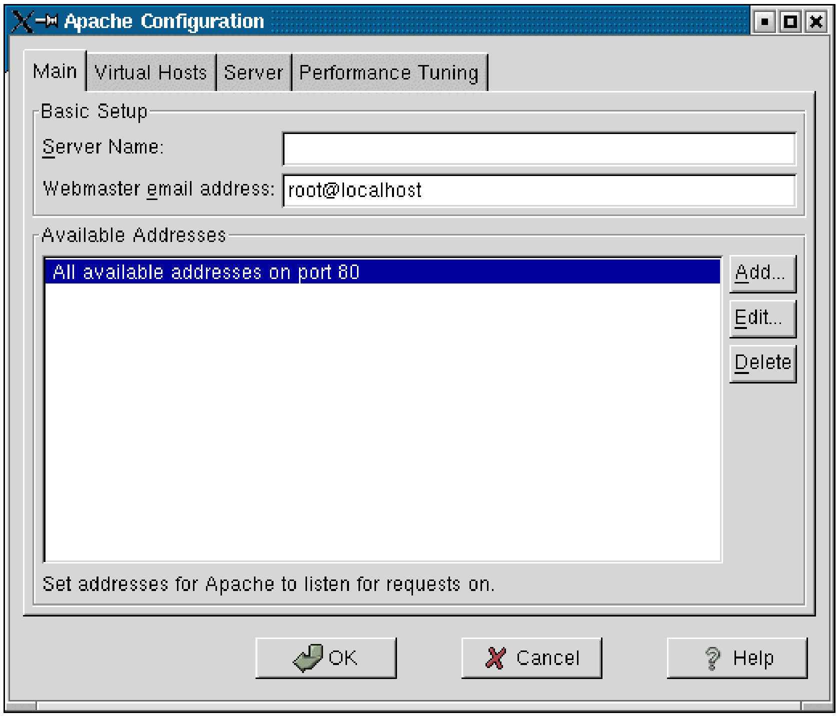
On the bottom half of the Virtual Server page, you have two sections.

The Per Directory Options (Figure 5-10 on page 78) allows you to edit and create options specific to a directory. This is mostly used for access control or enabling CGIs or other scripts for that directory. You can also use regular expressions in the path description of the directory option. This can for example be used to enable the execution of CGI scripts in all the directories that end with cgi-bin/.

The Virtual Server Details section shows you the parameters used in the description of this specific virtual server. In our case, the virtual server is identical to the default one, except that the options set in this virtual server only apply to connections made to port 443, which is the HTTPS or secured HTTP port.

5.3 GUI-based management

The Red Hat Linux Version 7.3 installation also includes a GUI-based program for configuring Apache 1.3. From the Webmin menu, click **System -> Apache Configuration**. Or you can start it from the command line with **`apacheconf`**.



Basic Setup

Server Name:

Webmaster email address: root@localhost

Available Addresses

All available addresses on port 80

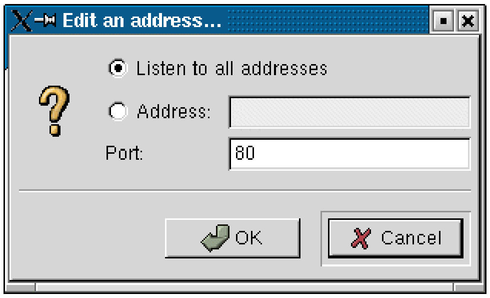
Set addresses for Apache to listen for requests on.

Figure 5-11 Red Hat's Apache Configuration tool - main window

In Figure 5-11 you can see the main window of this tool. There are four tabs that allow you to configure a specific part of Apache. We will review each of them.

Main tab

In this tab, you can set the name of the Web server, add the e-mail address of the Webmaster and define a list of addresses that Apache will listen to. Clicking the **Add...** or **Edit...** button will give you a dialog box similar to Figure 5-12 on page 80.


Edit an address...

● Listen to all addresses

○ Address:

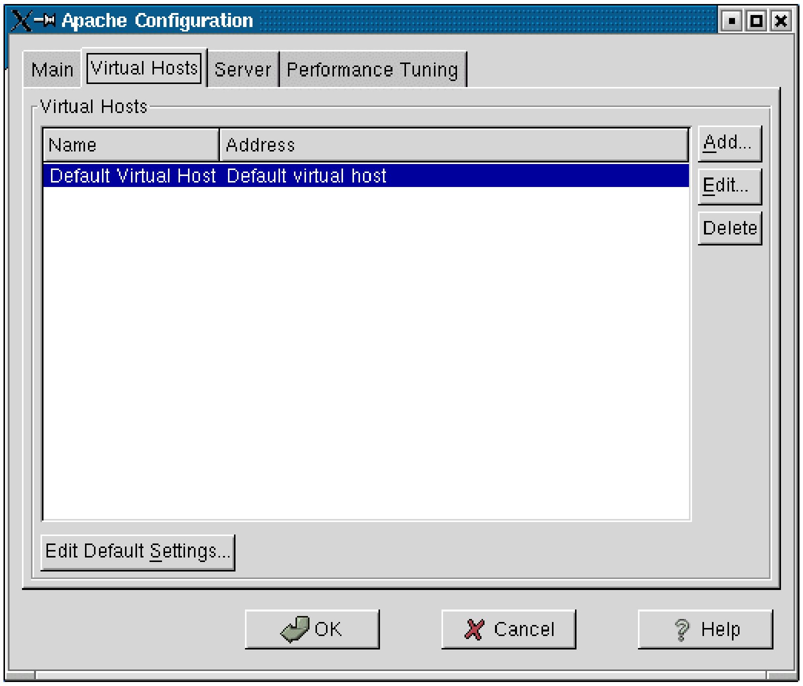
Port: 80

Figure 5-12 Listen to address dialog box.

Here you can specify which addresses Apache will listen to (either all of them or a specific one), and on what port.

Virtual Hosts tab

Under this tab (Figure 5-13), you will find a list of defined virtual servers and have the ability to add new ones, edit the existing ones and delete them. You can also edit the default settings of Apache. Both the Add..., Edit... and Edit Default Settings... buttons bring up a similar window.

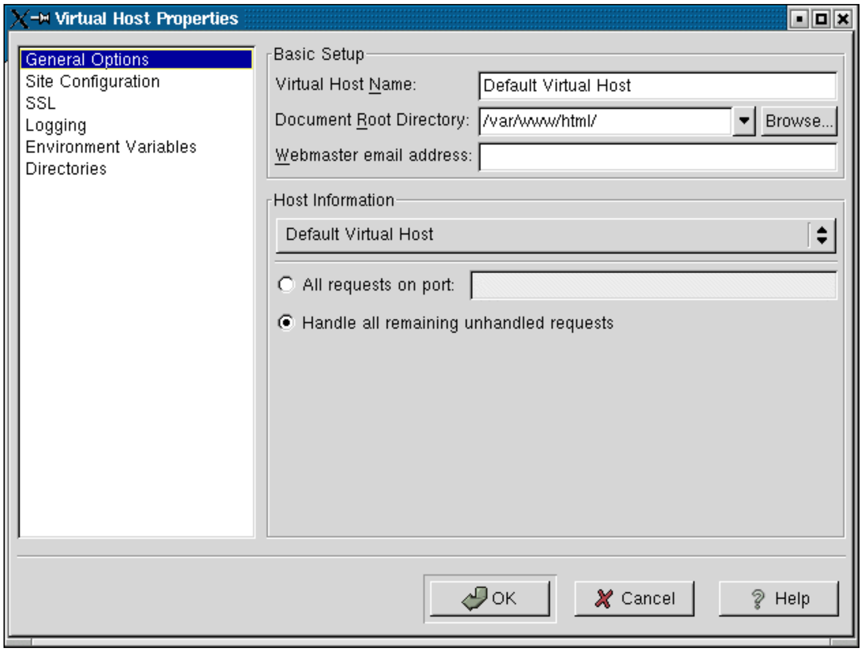

Apache Configuration

Name	Address
Default Virtual Host	Default virtual host

Figure 5-13 Virtual Hosts tab

In Figure 5-14 on page 81, you have access to all the directives specific to the virtual server you selected. This section is somewhat similar to the Webmin virtual server main window as seen in the previous section.

u



Virtual Host Name: Default Virtual Host

Document Root Directory: /var/www/html/

Webmaster email address:

Default Virtual Host

○ All requests on port:

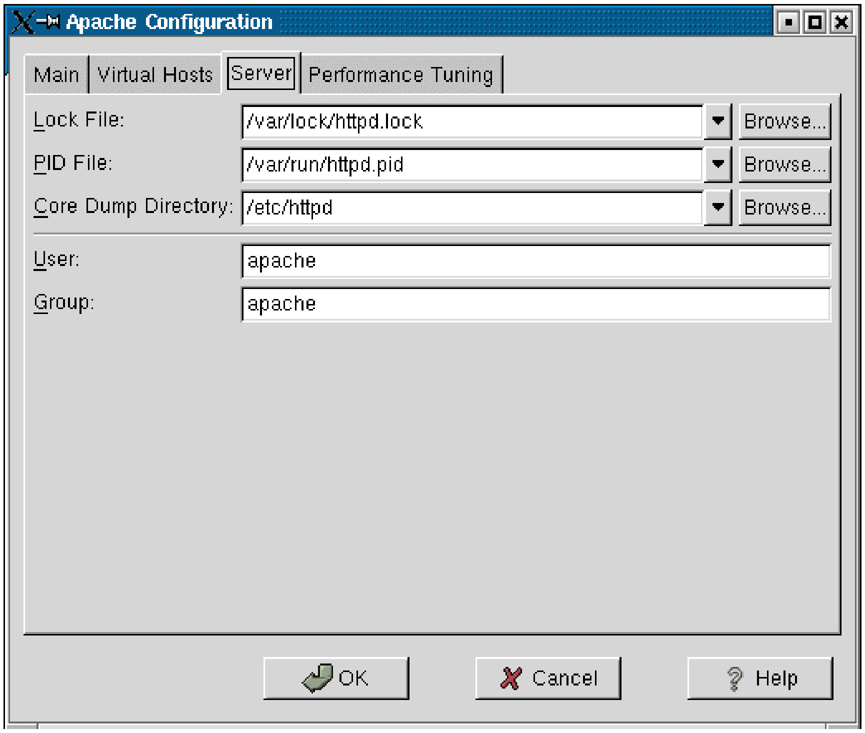
● Handle all remaining unhandled requests

Figure 5-14 Virtual Host properties

If you click **Help**, you will have access to the Apache Configuration tool (Figure 5-15 on page 82) help pages. The help pages describe what each option is. Or you can check the documentation of Apache itself for an exact reference to the options of each directive.

Server tab

As shown in Figure 5-15, you can edit some basic properties of the complete Apache environment.



Lock File: /var/lock/httpd.lock

PID File: /var/run/httpd.pid

Core Dump Directory: /etc/httpd

User: apache

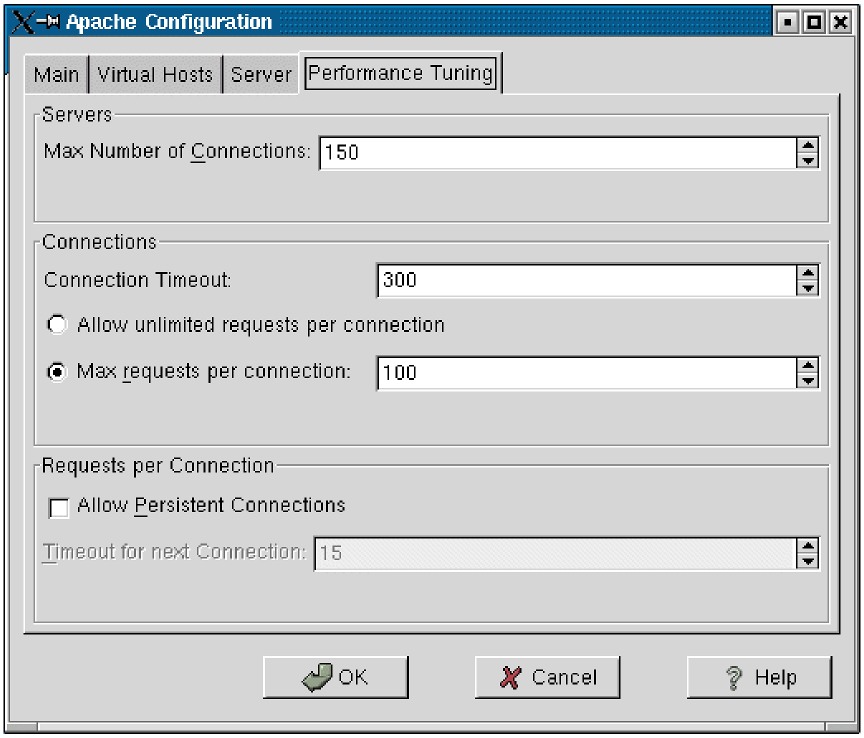
Group: apache

Figure 5-15 Apache Configuration tool - Server tab

Performance Tuning tab

This tab (Figure 5-16 on page 83) gives you access to the Apache parameters that can affect the performance of your server. Since this can be a complex matter, we refer you to the relevant section of the Apache documentation:

http://httpd.apache.org/docs/misc/perf-tuning.html



Servers

Max Number of Connections: 150

Connections

Connection Timeout: 300

○ Allow unlimited requests per connection

● Max requests per connection: 100

Requests per Connection

□ Allow Persistent Connections

Timeout for next Connection: 15

Figure 5-16 Apache Configuration tool - Performance Tuning tab

After you have made all your changes, you can click **OK** to save or click **Cancel** to ignore the changes you made.

6

Building a Web cluster

In this chapter, we will build a Web cluster using a set of blade servers in our IBM @server BladeCenter. The clustering will be done using two Open Source projects: Linux Virtual Server and Keepalived. The Web server used will of course be Apache.

We will build the Web cluster in two steps. The first one will be without high availability. The second step will provide high availability and use more nodes and services.

6.1 Overview of the cluster

Our cluster will be built using different parts. We will review each part and give a general overview.

6.1.1 The big picture

Figure 6-1 shows an overview of the cluster that we will set up. The two blade servers on the left will be called Director 1 and Director 2. They will provide the routing and load balancing for the real servers. We have two because we will also set up failover between the Directors.

Here, the name Director is not a reference to IBM Director. The name Director is used in the Linux Virtual Server and in the Keepalived documentation for the load balancer(s); it was decided to also use that name here.

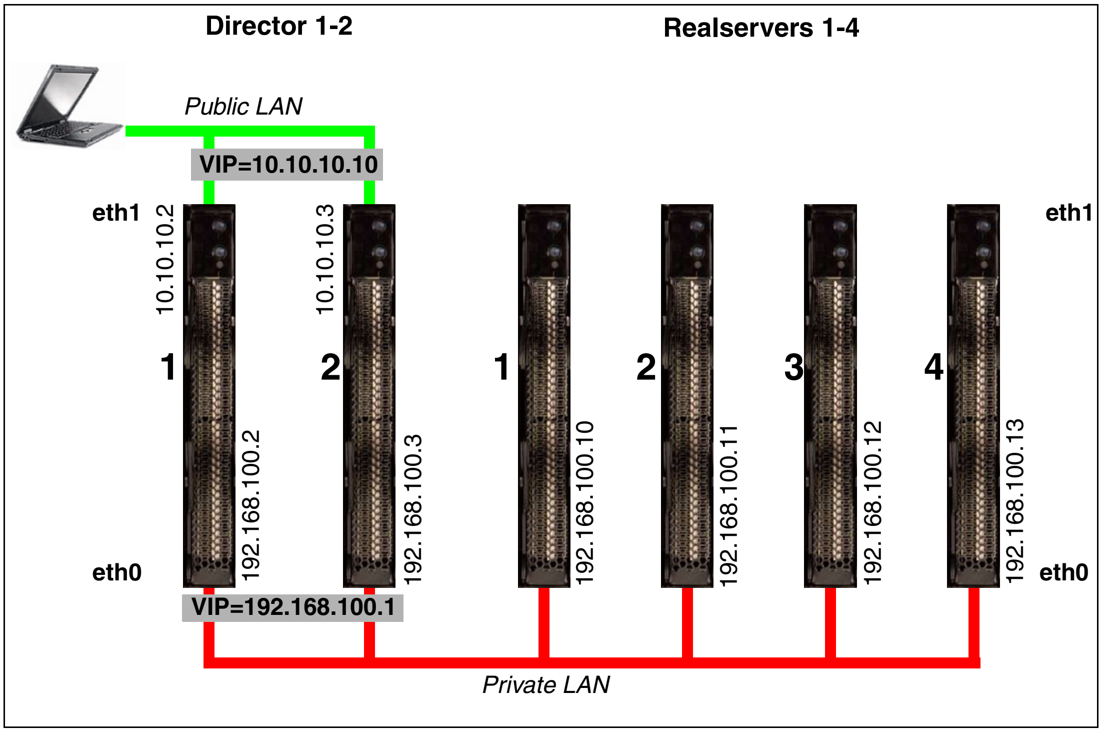


Figure 6-1 Overview of the Web cluster

The four blade servers on the right will be called real server 1 to real server 4. They will each contain a Web server. The Directors will be the default gateway for the real servers and will be connected via a private LAN. The private LAN will be handled by switch 1.

We will be using the Linux Virtual Server project (LVS) on the Directors to provide for the load balancing. We will use LVS to create 2 Virtual IP addresses (VIP). The VIP 10.10.10.10 will be created on the public LAN and will provide access to the Web cluster for the clients. The other VIP 192.168.100.1 will provide access for the real servers to the outside world. Therefore, it will be created on the private LAN.

We will also use the Keepalived project. This project is a daemon that will provide three functions:

- It will set up LVS to configure the VIPs and the load balancing.
- It will monitor the real servers and reconfigure LVS is any of them fail or are added.
- It will provide failover for the Directors.

The combination of these two projects allows you to easily set up and manage any network cluster. Since they are both generic, they can cluster not only Web servers but, in theory, any network service.

For more information about these projects, we refer you to their Web sites at:

```
http://www.linuxvirtualserver.org/
http://www.keepalived.org/
```

6.1.2 The hardware

In our example, we use an IBM @server BladeCenter with six blade servers installed. For the Directors, the smallest blade server is sufficient (1 x Intel XEON 2 GHz, 512 MB RAM). For the real servers, the choice is yours. The more power you give each blade server, the faster the total cluster will be.

In the BladeCenter, we have installed one management module, two power supply modules and two Ethernet switch modules. If you install more than six blade servers, you will need two extra power supply modules.

For storage, we used an internal IDE disk in each blade server. In your case, you may choose SCSI storage of Fibre Channel access to external storage.

6.1.3 The real servers

Real server is the name given to computers in an LVS cluster that actually perform the service(s) that will be virtualized. The real servers are grouped per service that is virtualized to a Virtual Server. A real server can be part of different groups at the same and therefore can have different services running at the same time.

There is no need to install any software on the real server specific to LVS. The LVS code is only needed on the Director. Still, you made need to install cluster-related software on the real servers. For example, on an LVS for Web servers, you might install/configure network filesystems like NFS, Coda, Intermezzo, and so forth to share the document root for the Web sites.

Be careful with the real servers when you are clustering services that use sessions. When a client connects for a second time to the cluster, it may be directed to another real server other than the first real server. You will have to make sure that any information the first real server has is also known by the second real server. You can, for example, use a common database that will contain this information.

LVS itself also has provisions for this. You can set a persistence time for each virtual server. If you have this enabled, then LVS will connect the same client to the same real server for as long as the persistence timer has not run out.

The real server can be any OS that has decent TCP/IP support, but this depends on the way the virtual server is created. See the LVS documentation to see which LVS method is the best for your application and which OS is supported by that method.

6.1.4 The Directors

The Directors with the LVS software will provide the routing, filtering and packet forwarding functions to create the virtual servers. It can be run on any machine that is at least a 486 and preferably with two Ethernet adapters. The Linux installed on it should be at least one with a kernel 2.2.14 or greater. But the latest kernel from the 2.4 series is preferred.

There are three ways to set up a virtual server:

- **Network Address Translation**: here, the public and private LANs are different network and IP packages that go through the Director and will be rewritten so that it seems they came from the Director itself.
- **IP Tunneling**: here, the Director will deliver IP packages from clients to the real servers using an IP Tunnel, and the real server will then answer directly to the client.
- **Direct Routing**: here, the Director and the real server are all on the same physical network. This network is used by the Director to forward packages from the clients to the real servers. The real server will then answer the clients using another network.

In our setup, we will be using NAT.

LVS has different algorithms to perform the load balancing; the best choice depends on what service the virtual server will deliver and how it is used. The complete list of algorithms and accompanying explanations can be found at:

`http://www.linuxvirtualserver.org/docs/scheduling.html`

We are also going to implement high availability into our virtual server. This means two things. real servers that are no longer responding will be removed from the virtual server, and we will provide failover between the two Directors if the active Director goes down.

This we will implement using Keepalived. It is a daemon that provides health checks for real servers and will remove a real server from a virtual server if it stops responding. It will also provide failover for the Director using VRRPv2. This is a protocol used for router failover.

6.2 Installing the Web cluster

In this section, we will install all the software needed for the Web cluster.

6.2.1 Installing the real servers

Since nothing special has to be done on the real servers, you can follow the instructions in Chapter 3, “Installation of Linux and IBM Director Agent” on page 29 and Chapter 4, “Installation of Apache” on page 57 for the installation of Linux and Apache. You can even use Windows on the real servers if you want.

For the network configuration, you have to provide an IP address as indicated in Figure 6-1 on page 86 and use the VIP 192.168.100.1 as the default gateway.

6.2.2 Installing the Directors

In our Web cluster, we will use Red Hat Linux 7.3 for the Directors. You can follow the instructions in Chapter 3, “Installation of Linux and IBM Director Agent” on page 29 to install this on the Directors.

When configuring the networking on the Directors, please use the IP addresses indicated in Figure 6-1 on page 86 for the external/public side. This means the eth1 interfaces. You may use a setup that is compatible with your own networks, but make sure you reserve three IP addresses, one for each Director and one for the public VIP.

After the installation of Red Hat Linux on the Directors, make sure you also apply all updates for Red Hat Linux as indicated in 3.2.3, “Applying updates and installing the latest kernel” on page 35.

When you have installed all the updates, also install the kernel-source package for the kernel that you will be running. We will need this source later.

Installing LVS

The kernels from Red Hat already include the LVS code. But this is not the latest version of the LVS code. If you so desire, you may build a new kernel with the latest LVS code. But we will not do that in our setup.

What you do need to install is the ipvsadm code. This is a small tool that allows you to configure LVS and view its status. In our setup, we worked with kernel V2.4.18-18.7.x. This kernel includes V1.0.4 of LVS.

1. Download the ipvsadm source RPM from:

   `http://www.linuxvirtualserver.org/software/index.html`

   For our kernel and version of LVS, we downloaded the package ipvsadm-1.21-3.src.rpm.
2. Compile this source RPM using **`rpm --rebuild ipvsadm-1.21-3.src.rpm`**.
3. Install the compiled RPM using **`rpm -ivh /usr/src/redhat/RPMS/i386/ipvsadm-1.21-3.i386.rpm`**.

Installing Keepalived

Now that LVS is installed, we continue with Keepalived.

1. Download the latest Keepalived source RPM from `http://www.keepalived.org/download.html`. We downloaded keepalived-0.7.6-1.src.rpm.
2. Create a symbolic links called /usr/src/linux to the kernel source code. We did **`cd /usr/src; ln -s linux-2.4.18-18.7.x/ linux; cd`**.
   This link is needed for Keepalived to find the source code of LVS.
3. Compile the Keepalived source RPM with **`rpm --rebuild keepalived-0.7.6-1.src.rpm`**.
4. Install the compiled RPM with **`rpm -ivh /usr/src/redhat/RPMS/i386/keepalived-0.7.6-1.i386.rpm`**.

Our Directors are now ready.

Note: The compilation of the source RPM of `ipvsadm` and `keepalived` has to be done only once. You can use the compiled packages directly on the other Director.

6.3 Configuring the Web cluster

We will set up the cluster in two steps. In the first step, we will use only one Director, this means no Director failover, and two real servers. With this, you can test the basic functioning of the cluster.

Then we will set up the cluster using both Directors and all the real servers. We will then also add support for both the HTTP and HTTPS protocols.

6.3.1 Configuring the basic Web cluster

We will configure Director 1 to create the virtual server and use real server 1 and 2 as the servers part of the virtual server.

1. Make sure that you have set the network configuration of all the blade servers as indicated in the previous sections.

> **Note:** All the next steps have to be performed on Director 1.

2. For LVS to work, we need to enable the forwarding of IP packets by the kernel. You can do this with **`echo "1" > /proc/sys/net/ipv4/ip_forward`**.
3. To enable IP forwarding by default, edit the file /etc/sysctl.conf and set the line `net.ipv4.ip_forward = 0` to `= 1`.
4. Open the file /etc/keepalived/keepalived.conf in an editor.
5. Edit it to look like the file shown in Example 6-1.

Example 6-1 Configuration of Keepalived for the basic Web cluster

```
! Configuration File for keepalived

global_defs {
   ! who will get alert emails
   notification_email {
   root@localhost
   }
   ! the alert emails will come from
   notification_email_from keepalived@localhost
   ! use this server to send the alert mails
   smtp_server 127.0.0.1
   smtp_connect_timeout 30
   ! the name/id of this load balancer
   ! (it should be unique)
   lvs_id LVS_DIRECTOR_1
}

! this defines the VIP on the public LAN, the virtual server
vrrp_instance VI_1 {
    ! this director will start as master
    state MASTER
    ! create the VIP on interface eth1
    interface eth1
    ! each virtual server needs a unique id
    virtual_router_id 51
    ! the priority determines who is master in failover setups,
    ! the highest priority becomes master
    priority 200
    ! how often check for failover between master and backup
    advert_int 1
    ! authentication for the syncronisation between master and backup
    authentication {
        auth_type PASS
        auth_pass 1111
    }
    ! the IP adresses that are part of this virtual server
    virtual_ipaddress {
        10.10.10.10
    }
}

! this defines the gateway on the private LAN
! setup is the same as above, only different interface (eth0)
! different id (52) and different IP address
vrrp_instance VI_GATEWAY {
    state MASTER
    interface eth0
    virtual_router_id 52
    priority 200
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass 1111
    }
    virtual_ipaddress {
        192.168.100.1
    }
}

! this links the VIPs on the public and private LAN so that they both are
! taken over by the backup in case of failover
vrrp_sync_group VSG_1 {
    group {
        VI_1
        VI_GATEWAY
    }
}

! this defines the IP address 10.10.10.10 if our virtual server
! for port 80, the http port
virtual_server 10.10.10.10 80 {
    ! time between the health checks of the realservers
    delay_loop 6
    ! load balancing algorithm (rr = round robin)
    lb_algo rr
    ! type of LVS (NAT = network address translation)
    lb_kind NAT
    ! protocol of the virtual IP
    protocol TCP

    ! realserver 1 on port 80
    real_server 192.168.100.10 80 {
        ! weight is used in weighted algorithms
        weight 1
        ! type of health check, get a html page
        HTTP_GET {
            ! which url to test
            url {
              path /
              ! checksum of the recieved page to test with
              ! you need to set this for your own environment
              digest ff20ad2481f97b1754ef3e12ecd3a9cc
            }
            ! on what port must we test
            connect_port 80
            ! timeout value for the test
            connect_timeout 3
            ! how many retries before marking server dead
            nb_get_retry 3
            ! how long to wait between retries
            delay_before_retry 3
        }
    }

    ! realserver 2, simular setup as realserver 1
    real_server 192.168.100.11 80 {
        weight 1
        HTTP_GET {
            url {
              path /
              digest ff20ad2481f97b1754ef3e12ecd3a9cc
            }
            connect_port 80
            connect_timeout 3
            nb_get_retry 3
            delay_before_retry 3
        }
    }
}
```

6. You need to modify the digest values for each real server. The digest value can be found using the **genhash** command. In our example, we created an index page that showed us the name of the real server with a different background each time. This helps us to see if everything is working correctly. We did the following to create the digest values for real server 1:

   a. Execute the command **genhash -s 192.168.100.10 -p 80 -u /**.

   b. At the end of the output from this command we saw the following output.

Example 6-2 Output of the genhash command

```
-----------------------[    HTML MD5 resulting    ]-----------------------
0000  ff 20 ad 24 81 f9 7b 17 - 54 ef 3e 12 ec d3 a9 cc   . .$..{.T.>.....
-----------------------[ HTML MD5 final resulting ]-----------------------
ff20ad2481f97b1754ef3e12ecd3a9cc
```

   c. The last line is the value to be filled in at the digest parameter.

   Do this for all the real servers you have defined. If you have the same index page on each real server then of course you only need to run **genhash** once and fill in the same value for each real server.

7. Save the Keepalived configuration file and close the editor.
8. Make sure all the Apache Web servers on the real servers are started.
9. To help in the initial debugging, we will let Keepalived output extra information into the system log. Edit the Keepalived start script /etc/init.d/keepalived.init and change `keepalived` to `keepalived -d` in the start and restart sections.
10. Start Keepalived with **/etc/init.d/keepalived.init start**.

11. Watch the system log file with **`less /var/log/messages`**. At the end, you should see a lot of information from Keepalived.

12. With **`ipvsadm`**, you can check if Keepalived has setup LVS properly.

13. With **`ip addr list`**, you should see the the VIP addresses added to the Director.

14. If all these commands give you the correct information, you can test the virtual server. On a client, browse to the VIP 10.10.10.10 and reload a couple of times. You should see all your Apache servers working.

> **Note:** The clients cannot be the Director or any of the real servers. It needs to be a separate machine connected to the public LAN.

Your basic cluster should work now. You can check everything by using the commands mentioned above and looking at the log files on the Director and real servers.

6.3.2 Configuring the complete Web cluster

Now that we have the basic clustering setup, we will expand the configuration to use all six of our blade servers.

This means adding a second Director that will provide failover for the first Director, adding two real servers to our virtual server and also creating a virtual server for HTTPS support.

The configuration of the three new blade servers is identical to that of the three we have already used. Only the network configuration, the index page on the Apache server and the keepalived.conf file on the second Director are different.

1. Make sure the real servers are running and have Apache set up for both HTTP and HTTPS support.
2. Edit the keepalived.conf file on Director 1; this will be the master. Make it look like the file shown in Example 6-3.

Example 6-3 Configuration of Keepalived for the complete cluster on Director 1

```
! Configuration File for keepalived

global_defs {
   ! who will get alert emails
   notification_email {
   root@localhost
   }
   ! the alert emails will come from
   notification_email_from keepalived@localhost
   ! use this server to send the alert mails
   smtp_server 127.0.0.1
   smtp_connect_timeout 30
   ! the name/id of this load balancer
   ! (it should be unique)
   lvs_id LVS_DIRECTOR_1
}

! this defines the VIP on the public LAN, the virtual server
vrrp_instance VI_1 {
    ! this director will start as master
    state MASTER
    ! create the VIP on interface eth1
    interface eth1
    ! this enabled a daemon, part of LVS, that syncronises both director
    lvs_sync_daemon_interface eth1
    ! each virtual server needs a unique id
    virtual_router_id 51
    ! the priority determines who is master in failover setups,
    ! the highest priority becomes master
    priority 200
    ! how often check for failover between master and backup
    advert_int 1
    ! authentication for the syncronisation between master and backup
    authentication {
        auth_type PASS
        auth_pass 1111
    }
    ! the IP adresses that are part of this virtual server
    virtual_ipaddress {
        10.10.10.10
    }
}

! this defines the gateway on the private LAN
! setup is the same as above, only different interface (eth1)
! different id (52) and different IP address
vrrp_instance VI_GATEWAY {
    state MASTER
    interface eth0
    lvs_sync_daemon_interface eth0
    virtual_router_id 52
    priority 200
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass 1111
    }
    virtual_ipaddress {
        192.168.100.1
    }
}

! this links the VIPs on the public and private LAN so that they both are
! taken over by the backup in case of failover
vrrp_sync_group VSG_1 {
    group {
        VI_1
        VI_GATEWAY
    }
}

! this defines the IP address 10.10.10.10 if our virtual server
! for port 80, the http port
virtual_server 10.10.10.10 80 {
    ! time between the health checks of the realservers
    delay_loop 6
    ! load balancing algorithm (rr = round robin)
    lb_algo rr
    ! type of LVS (NAT = network address translation)
    lb_kind NAT
    ! protocol of the virtual IP
    protocol TCP
```

```
! realserver 1 on port 80
real_server 192.168.100.10 80 {
    ! weight is used in weighted algorithms
    weight 1
    ! type of health check, get a html page
    HTTP_GET {
        ! which url to test
        url {
          path /
          ! checksum of the recieved page to test with
          ! you need to set this for your own environment
          digest ff20ad2481f97b1754ef3e12ecd3a9cc
        }
        ! on what port must we test
        connect_port 80
        ! timeout value for the test
        connect_timeout 3
        ! how many retries before marking server dead
        nb_get_retry 3
        ! how long to wait between retries
        delay_before_retry 3
    }
}

! realserver 2, simular setup as realserver 1
real_server 192.168.100.11 80 {
    weight 1
    HTTP_GET {
        url {
          path /
          digest ff20ad2481f97b1754ef3e12ecd3a9cc
        }
        connect_port 80
        connect_timeout 3
        nb_get_retry 3
        delay_before_retry 3
    }
}

real_server 192.168.100.12 80 {
    weight 1
    HTTP_GET {
        url {
          path /
          digest ff20ad2481f97b1754ef3e12ecd3a9cc
        }
        connect_port 80
        connect_timeout 3
        nb_get_retry 3
        delay_before_retry 3
    }
}

real_server 192.168.100.13 80 {
    weight 1
    HTTP_GET {
        url {
          path /
          digest ff20ad2481f97b1754ef3e12ecd3a9cc
        }
```

```
            connect_port 80
            connect_timeout 3
            nb_get_retry 3
            delay_before_retry 3
        }
    }
}

! same virtual server but now using port 443 = HTTPS
virtual_server 10.10.10.10 443 {
    delay_loop 6
    lb_algo rr
    lb_kind NAT
    protocol TCP
    ! this means that a client will be routed to the same virtual server
    ! for a period of 10 minutes. Needed for SSL connections, adjust for
    ! your own setup
    persistence_timeout 600

    real_server 192.168.100.10 443 {
        weight 1
        ! use HTTPS instead of HTTP
        SSL_GET {
            url {
              path /
              digest ff20ad2481f97b1754ef3e12ecd3a9cc
            }
        connect_port 443
        connect_timeout 3
        nb_get_retry 3
        delay_before_retry 3
        }
    }

    real_server 192.168.100.11 443 {
        weight 1
        SSL_GET {
            url {
              path /
              digest ff20ad2481f97b1754ef3e12ecd3a9cc
            }
        connect_port 443
        connect_timeout 3
        nb_get_retry 3
        delay_before_retry 3
        }
    }
    real_server 192.168.100.12 443 {
        weight 1
        SSL_GET {
            url {
              path /
              digest ff20ad2481f97b1754ef3e12ecd3a9cc
            }
        connect_port 443
        connect_timeout 3
        nb_get_retry 3
        delay_before_retry 3
        }
    }
    real_server 192.168.100.13 443 {
        weight 1
        SSL_GET {
            url {
              path /
              digest ff20ad2481f97b1754ef3e12ecd3a9cc
            }
        connect_port 443
        connect_timeout 3
        nb_get_retry 3
        delay_before_retry 3
        }
    }
}
```

Here again, adjust the digest value for each HTTP_GET or SSL_GET test to the value corresponding to your setup.

3. Copy the keepalived.conf file from Director 1 to Director 2.
4. Edit the keepalived.conf file on Director 2 and make the following changes.

Table 6-1 Changes needed on Director 2 to keepalived.conf

Change...	**to ...**
`lvs_id LVS_DIRECTOR_1`	`lvs_id LVS_DIRECTOR_2`
`state MASTER`	`state BACKUP`
`priority 200`	`priority 100`

You will have to change the state and priority statements twice, once for each VIP.

5. We also need to enable the forwarding of IP packets on Director 2. You can do this with **`echo "1" > /proc/sys/net/ipv4/ip_forward`**.

   To enable IP forwarding by default, edit the file /etc/sysctl.conf and set the line `net.ipv4.ip_forward = 0` to `= 1`.
6. Start Keepalived on Director 1 with **`/etc/init.d/keepalived.init start`**.
7. Confirm that is has started correctly by checking /var/log/messages, ipvsadm and ip addr list.
8. Test the cluster using a browser. Remember, the client must not be the real servers or the Director and must be on the public LAN. You should test both HTTP and HTTPS.
9. If the cluster works, start Keepalived on Director 2 with **`/etc/init.d/keepalived.init start`**.
10. Again, check that it has started correctly by looking at /var/log/messages, ipvsadm and the ip addr list. You should see in /var/log/messages that it is in backup state.
11. Test the cluster again from a client.
12. You can now stop Apache on one of the real servers with **`service httpd stop`**. In the logs, you should see that Keepalived has removed it from the virtual server. You should not be able to access it from the client.
13. Now shut down Director 1 to test failover of the Directors with **`shutdown -h now`**. Keep testing the cluster with the client(s). Director 2 should take over the load balancing.

If everything works, then you now have a high availability Web cluster.

7

A brief introduction to Layer 4-7 Switching

This chapter provides a brief overview of how Layer 4-7 Switching works. The goals are to present an alternative to traditional network design and to provide enough background to understand how a Layer 2-7 switch can be beneficial to a network. While this chapter gives a brief overview of the features and benefits of Layer 4-7 Switching, it also mentions the Layer 2-7 GbE Switch Module or GbESM. For a more in-depth look at Layer 4-7 Switching, refer to the *IBM @server BladeCenter Layer 2-7 Network Switching* REDP3755.

7.1 Layer 4-7 Switching

Most networks employ multiple servers without server load balancing. Each server usually specializes in providing one or two unique services. However, a server that provides applications or data in high demand can become overutilized. If this happens, it can strain network resources since when the server starts rejecting user requests, the users resubmit the requests for data.This often happens on networks where several other servers are sitting idle with resources available to service users.

Layer 4-7 Switching can harness these available idle servers without additional special equipment such as dedicated load-balancing servers. The term L4-7 is used because these switches are Layer 4 (TCP) aware. These switches use headers and data found at the OSI Layer 7 to deliver many of their key features. Layer 2 and Layer 3 switches operate on and are aware of Layers 2-3 only.

Layer 4-7 Switching can be used to improve the reliability, scalability, and performance of a wide variety of applications and services. Layer 4-7 isolates the delivery of a service, from a user's point of view, from the physical reality of how that service is delivered. A Layer 4-7 switch is aware of the services provided by each server. Based on several load-balancing algorithms, a switch can direct user session traffic to an appropriate server. This means that in addition to using the available resources on otherwise idle servers, resources can be provisioned "on the fly". This lets network administrators add or subtract to the service delivery capability to react to server failures, meet demands during peak periods and scale back resources once demand is over. All of this happens with no knowledge or participation from the users of that service.

To take advantage of L4-7 Switching, services or applications advertise themselves using normal hostnames such as www.ibm.com® to the outside world. However, DNS resolution will resolve that hostname back to a Virtual IP Address(V_IP) assigned to an L4-7 switch. A service advertised by a V_IP is a Virtual Service. When application requests are sent from users to a V_IP, the L4-7 switch forwards the request to one or more real servers with real IP addresses.

The collection of real servers to which a request for a particular Virtual Service can be sent is called the Virtual Service Pool. By adding real servers to a Virtual Service Pool, the total application delivery throughput can be increased and application delivery times reduced. By moving real servers between Virtual Service Pools, shifting loads and demands between services can be met without having to provision enough servers for each application to meet the peak demands (which would result in several unused servers in off-peak times).
Figure 7-1 on page 101 shows how the Layer 2-7 GbE Switch Module installed in BladeCenter could be used to redeploy blades to service a few different applications according to demand.

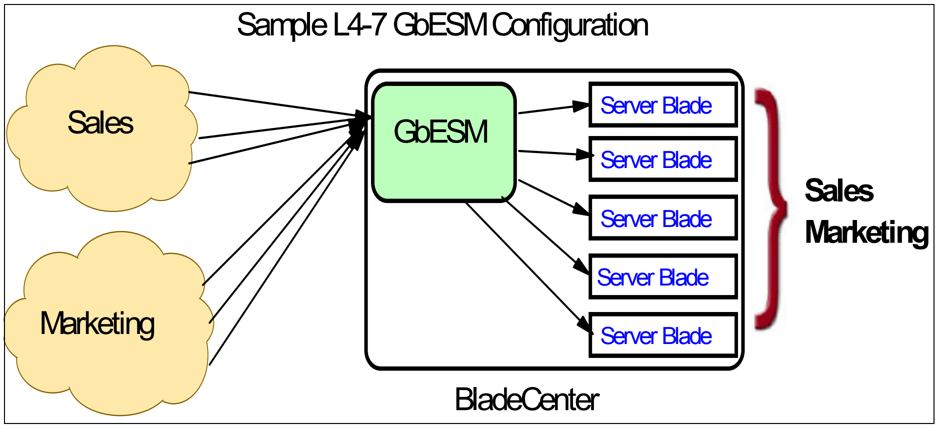


Figure 7-1 Sample L4-7 Configuration.

Figure 7-1 shows how services can be deployed using Layer 4-7 Switching. Using the GbESM in the BladeCenter, all server resources can be configured to support all applications, with the GbESM spreading the load among them at all times. As the loads on Sales and Marketing vary throughout the day or week or month, resources are automatically allocated in real time to meet the need.

In this scenario, to provide load balancing for any particular type of service, each server blade in the pool must have access to identical content, either directly (duplicated on each server) or through a back-end network (mounting the same file system or database server).

The GbESM acts as a front end to the servers, interpreting user session requests and distributing them among the available servers. Load balancing in the GbESM can be done in the following ways.

Virtual server-based load balancing

This is the traditional load-balancing method. The switch is configured to act as a virtual server and is given a V_IP for each service it will distribute. Each virtual server is assigned a list of the IP addresses (or range of addresses) of the real servers in the pool where its services reside. When the user stations request connections to a service, they will communicate with a virtual server on the switch. When the switch receives the request, it binds the session to the IP address of the best available real server and remaps the fields in each frame from virtual addresses to real addresses. HTTP, IP, FTP, RTSP, IDS, and static session WAP are examples of some of the services that use virtual servers for load balancing.

Filter-based load balancing

A filter allows you to control the types of traffic permitted through the switch. Filters are configured to allow, deny, or redirect traffic according to the IP address, protocol, or Layer 4 port criteria. In filter-based load balancing, a filter is used to redirect traffic to a real server group. If the group is configured with more than one real server entry, redirected traffic is load balanced among the available real servers in the group. Firewalls, WAP with RADIUS snooping, IDS, and WAN links use redirection filters to load balance traffic.

Content-based load balancing

Content-based load balancing uses Layer 7 application data (such as URL, cookies, and Host Headers) to make intelligent load balancing decisions. URL-based load balancing, browser-smart load balancing, and cookie-based preferential load balancing are a few examples of content-based load balancing.

In the example in Figure 7-1 on page 101, virtual server load balancing is used. The switch has two different V_IPs configured which correspond to each of the services (Sales and Marketing). DNS responds with these V_IPs when these services are requested. The GbESM is also configured for the real IP addresses (R_IPs) that are owned by blades that support requests coming into each V_IP. In this case, all server blades can respond to requests for all services, so each V_IP is assigned the entire set of R_IPs.

A TCP connection consists of a series of packets sent back and forth from two destinations. If packets are misrouted and sent to the wrong destination, the session will slow down due to retransmissions and possibly fail if misrouting continues. Therefore, a Layer 4-7 switch needs to be TCP-aware in the sense that it needs to be able to:

- Identify the beginning of a new TCP connection
- Assign that connection to a real server
- Make sure that all ensuing packets related to that TCP connection continue to be sent to the same real server

This requirement is what puts the “Layer 4” in L4-7. The GbESM provides this capability. All packets arriving that are destined to a V_IP are inspected to determine whether they are associated with an existing TCP connection (in which case they are re-directed to the real server already assigned) or whether they are a request to set up a new TCP connection. If the received packet is a request to establish a new TCP connection (a TCP SYN packet), the GbESM will determine the best available server to re-direct that request to and then do so. The “best available” consideration can be made based upon a number of factors that are beyond the scope of this chapter. It is enough to say that “best available” is based on a consideration of the present health and load of each of the candidate real servers.

This chapter introduces the concepts of Layer 4-7 Switching and how it benefits customers deploying applications on the IBM @server BladeCenter. For a more in-depth look at Layer 4-7 Switching, refer to *IBM @server BladeCenter Layer 2-7 Network Switching*, REDP3755. In addition to expanding on the concepts discussed here, this Redpaper is an excellent guide to deploying the GbESM in both Nortel and Cisco networking environments.

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this Redpaper.

IBM Redbooks

For information on ordering these publications, see “How to get IBM Redbooks” on page 104.

- *The Cutting Edge: IBM @server BladeCenter,* REDP3581
- *IBM @server BladeCenter Systems Management,* REDP3582
- *Deploying Citrix MetaFrame on IBM @server BladeCenter,* REDP3583
- *Deploying Lotus Domino on IBM @server BladeCenter,* REDP3584
- *Deploying Microsoft Exchange on IBM @server BladeCenter,* REDP3585
- *Deploying Samba on IBM @server BladeCenter*, REDP3595
- *Linux on IBM Netfinity Servers: A Collection of Papers*, SG24-5994-00
- *Red Hat Linux Integration Guide for IBM @server xSeries and Netfinity*, SG24-5853-01
- *SuSE Linux Integration Guide for IBM @server xSeries and Netfinity*, SG24-5863-01
- *IBM @server BladeCenter Layer 2-7 Network Switching*, REDP3755

Referenced Web sites

These Web sites are also relevant as further information sources:

- The Apache HTTP Server project Web site

  `http://httpd.apache.org/`
- Apache Module Registry

  `http://modules.apache.org/`
- Red Hat Web site

  `http://www.redhat.com/`
- SuSE Web site

  `http://www.suse.com/`
- IBM Support Web site

  `http://www.ibm.com/pc/support`
- Webmin Web site

  `http://www.Webmin.com/`
- Linux Virtual Server project Web site

  `http://www.linuxvirtualserver.org/`
- keepalived project Web site

  `http://www.keepalived.org/`

- IBM ServerProven

  http://www.ibm.com/pc/compat/
- IBM @server BladeCenter - Power Module Upgrade Guidelines

  http://www-1.ibm.com/support/docview.wss?uid=psg1MIGR-53353
- IBM @server BladeCenter - Power Module Upgrade Guidelines

  ftp://ftp.software.ibm.com/pc/pccbbs/pc_servers_pdf/13n0308.pdf

How to get IBM Redbooks

You can order hardcopy Redbooks, as well as view, download, or search for Redbooks at the following Web site:

ibm.com/redbooks

You can also download additional materials (code samples or diskette/CD-ROM images) from that site.

IBM Redbooks collections

Redbooks are also available on CD-ROMs. Click the **CD-ROMs** button on the Redbooks Web site for information about all the CD-ROMs offered, as well as updates and formats.

Deploying Apache on IBM @server BladeCenter

Redpaper

Installing Linux and IBM Director Agent

Installing Apache

Building a Web cluster

As data centers have grown with the proliferation of Intel-based servers over recent years, it is important to note that rack space and floor space can now be more efficiently used with the IBM @server BladeCenter HS20 servers. Attractive cost savings are also possible where a large number of rack installed servers are required since the equivalent number of 1U servers would be much more expensive.

This IBM Redpaper describes how to set up and configure Linux and Apache on the IBM @server BladeCenter and provides some pointers on what applications to use to manage the installation. We describe building a Web cluster using Linux, Apache, Linux Virtual Server and also discuss Keepalived and the functionality of the IBM @server BladeCenter in this type of environment.