

# TurboLinux Integration Guide for IBM server xSeries and Netfinity

The complete guide to running TurboLinux on  
xSeries and Netfinity

Netfinity server-specific coverage you  
can't find anywhere else, including  
ServeRAID configuration

Plan, configure, and install  
key services, step-by-step:  
Samba, Apache, sendmail,  
DNS, DHCP, LDAP, and more



Jakob Carstensen  
Rufus Credle  
Justin Davies  
Ivo Gomilsek  
Jay Haskins  
Georg Holzknrecht  
Ted McDaniel

[ibm.com/redbooks](http://ibm.com/redbooks)

**Redbooks**





International Technical Support Organization

**TurboLinux Integration Guide  
for IBM @server xSeries and Netfinity**

December 2000

**Take Note!**

Before using this information and the product it supports, be sure to read the general information in Appendix D, "Special notices" on page 355.

**Second Edition (December 2000)**

This edition applies to IBM xSeries and Netfinity systems preparing for the installation of TurboLinux.

Comments may be addressed to:  
IBM Corporation, International Technical Support Organization  
Dept. HQ7 Building 678  
P.O. Box 12195  
Research Triangle Park, NC 27709-2195

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 2000. All rights reserved.  
Note to U.S Government Users – Documentation related to restricted rights – Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

---

# Contents

<b>Preface</b> .....	ix
The team that wrote this redbook .....	x
Comments welcome .....	xii
<b>Chapter 1. Introduction</b> .....	1
1.1 The IBM commitment to Linux .....	1
1.2 TurboLinux .....	2
1.3 Introducing the xSeries family of servers .....	2
<b>Chapter 2. Linux installation</b> .....	3
2.1 What you need to know about your hardware .....	3
2.2 Installing on a ServerRAID adapter .....	4
2.3 Operating system installation .....	5
2.4 Miscellaneous .....	12
2.5 Disk partitioning .....	15
2.6 Configure filesystems .....	22
2.7 Configure the primary network interface card .....	25
2.8 Software package installation .....	27
2.9 Selecting a kernel .....	29
2.10 LILO .....	30
2.11 X Server setup .....	35
<b>Chapter 3. Basic system administration</b> .....	39
3.1 Configuring X with most Netfinity and xSeries servers .....	39
3.1.1 X Windows configuration and startup .....	39
3.1.2 Installing the VESA frame buffer server .....	43
3.2 Turbonetcfg .....	45
3.3 Turboprintcfg .....	48
3.3.1 Configuring locally attached printers .....	49
3.3.2 Configuring remote printers over TCP/IP .....	52
3.3.3 Adding NetBIOS based remote printers .....	54
3.4 Adding and removing software packages .....	56
3.4.1 Adding additional packages from the CD-ROM with Turbopkg ..	56
3.4.2 Adding packages via FTP with Turbopkg .....	59
3.4.3 Removing packages using Turbopkg .....	60
3.4.4 Package management using the RPM command .....	61
3.5 User and group administration .....	62
3.5.1 Adding new groups .....	62
3.5.2 Adding new users .....	65
3.6 Administering file systems and the boot record .....	68
3.6.1 Managing file systems .....	69

3.6.2	The Boot Record . . . . .	76
3.7	Determining your hardware . . . . .	79
3.8	Server Services . . . . .	80
3.9	Time zone and time server configuration . . . . .	83
3.10	Enabling remote services to your server . . . . .	84
3.11	File system permissions . . . . .	87
<b>Chapter 4. The ServeRAID controller and TurboLinux . . . . .</b>		<b>91</b>
4.1	Updating the ServeRaid device driver for Linux . . . . .	91
4.1.1	Obtaining the required source code . . . . .	92
4.1.2	Compiling the ServeRaid driver in order to install TurboLinux 6 . . . . .	93
4.1.3	Compiling the ServeRAID driver after TurboLinux is installed . . . . .	94
4.2	Installing the ServeRAID command line tools for Linux . . . . .	97
4.2.1	getconfig . . . . .	100
4.2.2	The getstatus command . . . . .	104
4.2.3	The devinfo command . . . . .	105
4.2.4	The hsrebuild command . . . . .	106
4.2.5	The setstate command . . . . .	108
4.2.6	The synch command . . . . .	110
4.2.7	The unattended command . . . . .	110
4.2.8	The rebuild command . . . . .	111
4.3	Replacing a defunct drive . . . . .	112
4.3.1	Replacing a defunct drive with disabled Hot Spare Rebuild . . . . .	113
4.3.2	Replacing a defunct drive with a hot spare drive installed . . . . .	114
4.4	Using the ServeRAID Manager utility . . . . .	120
4.5	Remote management of the ServeRAID adapter . . . . .	123
<b>Chapter 5. DNS - Domain Name System . . . . .</b>		<b>129</b>
5.1	Installation of software . . . . .	131
5.2	DNS sample configuration . . . . .	131
5.3	Configuration tips . . . . .	137
<b>Chapter 6. Samba . . . . .</b>		<b>139</b>
6.1	What can you do with Samba? . . . . .	139
6.2	Setting up the Samba server . . . . .	139
6.2.1	Configuring the Samba server . . . . .	140
6.2.2	Starting and stopping the Samba server . . . . .	148
6.2.3	Using SWAT . . . . .	148
6.3	Sources and additional information . . . . .	165
<b>Chapter 7. Apache and IBM HTTP Servers . . . . .</b>		<b>167</b>
7.1	The IBM HTTP Server . . . . .	168
7.2	Apache HTTP Server installation . . . . .	169
7.3	IBM HTTP Server installation . . . . .	170

7.3.1	Setting up the administration server . . . . .	172
7.4	General performance tips . . . . .	176
<b>Chapter 8.</b>	<b>Packet filtering with IP Chains . . . . .</b>	<b>179</b>
8.1	What is packet filtering? . . . . .	179
8.2	What can you do with Linux packet filtering? . . . . .	179
8.3	What do you need to run packet filtering? . . . . .	180
8.4	Network configuration for a packet filtering implementation . . . . .	180
8.5	How to permanently enable IP Forwarding . . . . .	182
8.6	Your first IP Chains rule . . . . .	183
8.7	How packets travel through a gateway . . . . .	184
8.8	Using IP Chains . . . . .	186
8.8.1	How to create a rule . . . . .	187
8.8.2	Making the rules permanent . . . . .	188
8.9	Sources of additional information . . . . .	188
<b>Chapter 9.</b>	<b>sendmail . . . . .</b>	<b>189</b>
9.1	What is sendmail? . . . . .	189
9.2	What you can do with sendmail . . . . .	189
9.3	Before you begin . . . . .	189
9.4	Network configuration. . . . .	192
9.4.1	Setting up the master DNS . . . . .	193
9.4.2	Setting up the DNS for the first subdomain . . . . .	196
9.4.3	Setting up the DNS for the second subdomain . . . . .	199
9.4.4	Setting up sendmail . . . . .	202
9.4.5	Setting up the mail client . . . . .	205
9.5	Sources of additional information . . . . .	208
<b>Chapter 10.</b>	<b>DHCP - Dynamic Host Configuration Protocol . . . . .</b>	<b>209</b>
10.1	What is DHCP? . . . . .	209
10.2	Why should I use DHCP? . . . . .	209
10.3	Implementation on TurboLinux 6 . . . . .	209
<b>Chapter 11.</b>	<b>NFS - Network File System . . . . .</b>	<b>213</b>
11.1	The NFS process . . . . .	213
11.2	Using turbonetcfg to share data with NFS . . . . .	215
11.3	Sharing data with NFS: command-line process . . . . .	217
11.4	Accessing data remotely with NFS - the command line view. . . . .	219
<b>Chapter 12.</b>	<b>NIS - Network Information System . . . . .</b>	<b>221</b>
12.1	What is NIS? . . . . .	221
12.2	How can I use NIS? . . . . .	221
12.3	Implementation on TurboLinux . . . . .	222
12.3.1	Using the nsswitch file for lookups . . . . .	222

12.3.2 NIS server . . . . .	223
12.3.3 NIS Client . . . . .	228
12.4 Sources of additional information . . . . .	230
<b>Chapter 13. LDAP - Lightweight Directory Access Protocol . . . . .</b>	<b>231</b>
13.1 What is LDAP? . . . . .	231
13.1.1 Directory Services . . . . .	231
13.1.2 X.500 . . . . .	232
13.2 How can I use LDAP? . . . . .	232
13.3 LDAP basics. . . . .	232
13.4 Implementation on TurboLinux . . . . .	233
13.4.1 slapd.conf . . . . .	234
13.4.2 ldap.conf . . . . .	235
13.4.3 nsswitch.conf . . . . .	237
13.4.4 /etc/pam.d/login . . . . .	237
13.4.5 Starting OpenLDAP . . . . .	238
13.4.6 Testing authentication . . . . .	238
13.4.7 Migrating /etc/passwd . . . . .	239
<b>Chapter 14. General performance tools in Linux . . . . .</b>	<b>241</b>
14.1 General configuration hints. . . . .	241
14.1.1 Powertweak . . . . .	242
14.1.2 Services . . . . .	245
14.1.3 Kernel recompilation. . . . .	246
14.2 System monitoring and performance test tools . . . . .	247
<b>Chapter 15. Backup and recovery . . . . .</b>	<b>257</b>
15.1 BRU . . . . .	257
15.1.1 Installing BRU . . . . .	257
15.1.2 Basic commands . . . . .	259
15.1.3 Basic backup . . . . .	259
15.1.4 Basic restore . . . . .	259
15.1.5 Basic verification and listing commands . . . . .	260
15.1.6 X Interface . . . . .	261
15.1.7 The big buttons in BRU. . . . .	261
15.1.8 Creating archives . . . . .	262
15.1.9 Scheduling . . . . .	264
15.1.10 Restoring files. . . . .	265
15.1.11 Listing and verifying archives . . . . .	265
15.1.12 Summary . . . . .	266
15.2 Microlite BackupEDGE . . . . .	266
15.2.1 Installing Microlite BackupEDGE . . . . .	267
15.2.2 Initializing the tape . . . . .	268
15.2.3 Your first backup . . . . .	270



15.2.4 Restoring single files or directories . . . . .	274
15.2.5 Master and incremental backups . . . . .	276
15.2.6 Restoring master and incremental backups . . . . .	279
15.2.7 Performing scheduled backups . . . . .	280
15.2.8 Configuring the tape devices . . . . .	283
15.2.9 Defining the devices for making backups . . . . .	289
15.2.10 Microlite RecoverEDGE . . . . .	292
15.2.11 More information on Microlite . . . . .	302
15.3 Arkeia . . . . .	302
15.3.1 Installing Arkeia . . . . .	303
15.3.2 Configuring Arkeia . . . . .	303
15.3.3 Interactive backup . . . . .	318
15.3.4 Periodic Backup . . . . .	322
15.3.5 Restoration . . . . .	323
15.3.6 Advanced features of Arkeia . . . . .	327
<b>Appendix A. RAID levels . . . . .</b>	<b>329</b>
A.1 What is RAID? . . . . .	329
A.1.1 RAID-0 . . . . .	330
A.1.2 RAID-1 and RAID-1E . . . . .	331
A.1.3 RAID-10 . . . . .	332
A.1.4 RAID-5 . . . . .	333
A.1.5 RAID-5 enhanced . . . . .	337
A.1.6 Orthogonal RAID-5 . . . . .	339
A.1.7 Performance . . . . .	340
A.1.8 Recommendations . . . . .	342
A.1.9 Summary . . . . .	343
<b>Appendix B. Working video modes for IBM Netfinity servers . . . . .</b>	<b>345</b>
<b>Appendix C. Sample smb.conf Samba configuration file . . . . .</b>	<b>347</b>
<b>Appendix D. Special notices . . . . .</b>	<b>355</b>
<b>Appendix E. Related publications . . . . .</b>	<b>359</b>
E.1 IBM Redbooks . . . . .	359
E.2 IBM Redbooks collections . . . . .	359
E.3 Other publications . . . . .	359
E.4 Referenced Web sites . . . . .	360
<b>How to get IBM Redbooks . . . . .</b>	<b>363</b>
IBM Redbooks fax order form . . . . .	364

<b>Index</b> .....	365
<b>IBM Redbooks review</b> .....	369

---

## Preface

Here's all the information you need to maximize TurboLinux performance and reliability on state-of-the-art IBM @server xSeries and Netfinity server platforms. In this book, a team of IBM's top Linux experts presents start-to-finish, Netfinity server-specific coverage of TurboLinux 6.0 deployment and system administration throughout the entire system life cycle.

This redbook is aimed at beginners and intermediate Linux users and for all Windows users who are used to the safe and convenient graphical user interface.

The book covers the installation of TurboLinux 6.0. Once the installation has been completed, the book discusses some basic system administration tools that can help you manage your Linux system. Furthermore, this book provides an introduction to a wide range of services, such as Samba, NFS, and Apache among others. You will learn what each service is, what it is capable of, and how to install it. The services are not covered in detail, since they are very comprehensive. We recommend that you consult additional sources if you need more detailed information. These sources are mentioned during the chapters or at the end of each chapter.

---

## The team that wrote this redbook



Figure 1. The team (left to right) Credle, Holzknrecht, Carstensen, Haskins, Gomilsek, Davies, (lower) McDaniel

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

**Jakob Carstensen** is a Technical Support Marketing Specialist in the IBM Software Group. His most recent publication was *Small Business Suite for Linux Reviewer's Guide*. Before joining the IBM Software Group, he worked at the International Technical Support Organization center in Raleigh, where he worked as a Project Manager managing residencies and producing redbooks. Before joining the ITSO, he worked in Denmark both for the IBM PC Institute teaching TechConnect and Service Training courses, and for IBM PSS performing level-2 support of Netfinity products. He has a Bachelor of Electronics Engineering degree and has worked for IBM for the past ten years.

**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, IBM and OEM business applications, all running on IBM Netfinity and xSeries servers. Rufus's various positions during his IBM career have included assignments in administration and asset management, systems engineering, marketing and services. He holds a BS degree in Business Management from Saint Augustine's College. Rufus has been employed at IBM for 20 years.

**Justin Davies** is a systems administrator and product manager at SuSE UK. He has 5 years of Linux experience, and his expertise are in embedded Linux systems, systems administration and network intergration. He joined SuSE in May of 2000 after graduating from the University of Derby, with a diploma in computer science.

**Ivo Gomilsek** is an IT Specialist for Storage Area Networks and Storage in IBM Global Services - Slovenia for the CEE region. His areas of expertise include Storage Area Networks (SAN), Storage, IBM Netfinity servers, network operating systems (OS/2, Linux, Windows NT), Lotus Domino Servers. He is an IBM Certified Professional Server Specialist, Red Hat Certified Engineer, OS/2 Warp Certified Engineer and Certified Vinca Co-StandbyServer for Windows NT Engineer. Ivo was a member of the team that wrote the redbook Designing an IBM Storage Area Network, Implementing Vinca Solutions on IBM Netfinity Servers, and first edition of Netfinity and Linux Integration Guide. He also provide Level 2 support for IBM Netfinity servers, high availability solutions for IBM Netfinity servers and Linux. Ivo has been employed at IBM for 4 years.

**Jay Haskins** is a Systems Architect for IBM Global Services Enterprise Architecture and Design in Seattle, Washington. He has been a Linux and Open Source advocate for more than five years and currently spends most of his time developing dynamic monitoring tools using Perl and the Apache Web server. Before joining IBM, Jay worked in several different areas of the information technology field including UNIX system administration, database design and development, Windows application development, and network administration.

**Georg Holzknecht** is a Senior System Consultant at DeTeCSM, Darmstadt/Germany. He has 30 years of experience in different areas of the information technology field. He holds a diploma degree in electrical engineering from Technische Hochschule Darmstadt. His areas of expertise include system programming for mainframes, network operating systems (NetWare, Linux), database administration and design, application and driver development, systems management solutions with Tivoli.

**Ted McDaniel** is a Senior Support Specialist at the IBM PC HelpCenter in Research Triangle Park, NC. He is the World Wide Level 2 Linux support leader for IBM x-Series and Netfinity servers. Ted has six years of experience with Level 2 support.

Thanks to original authors, Lenz Grimmer and Joe Kaplenk, for their contribution to the first edition of this redbook, which was titled *TurboLinux and Netfinity Server Integration Guide*, SG24-5862-00.

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

Diane O'Shea, Gail Christensen, Linda Robinson, Margaret Ticknor, and Tamikia Barrow  
International Technical Support Organization, Raleigh Center

---

## Comments welcome

### Your comments are important to us!

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

- Fax the evaluation form found in "IBM Redbooks review" on page 369 to the fax number shown on the form.
- Use the online evaluation form found at [ibm.com/redbooks](http://ibm.com/redbooks)
- Send your comments in an Internet note to [redbook@us.ibm.com](mailto:redbook@us.ibm.com)

---

## Chapter 1. Introduction

Linux is a UNIX-like open-source operating system and was the original creation of Linus Torvalds from Helsinki, Finland in 1991. He wrote the first kernel, the underlying program interfacing and running the computer hardware. Torvalds invited programmers from around the world to comment on and to improve his code. This is one of the key ideas behind the success of Linux. With the world as your laboratory, the number of testers and developers is nearly endless. It is because of this resource that Linux is constantly evolving and improving.

With the Linux source code being freely available, several companies have developed different distributions of Linux. A distribution is a complete system. The key component is the Linux kernel. Other utilities, services, and various applications can be included as well, depending on the distribution and the intended use. There is no standard distribution. Each of the many distributions available has unique advantages.

IBM was early to recognize the value of Linux, investing in Linux-related product development, forming alliances with key Linux distributors, contributing to the open-source community, and aggressively supporting the platform. IBM believes this investment will benefit its customers as they continue to exploit Linux for their IT infrastructures and e-business applications.

---

### 1.1 The IBM commitment to Linux

IBM is fully committed to the open source movement and believes Linux will emerge as a key platform for e-business. IBM will work with the open source community, bringing relevant technologies and experience to the table to help enhance Linux, to define the standards and to extend Linux to the enterprise level. IBM provides continued support and participation in three main locations:

- The Open Source Development Lab
- IBM Development and Competency Centers for Linux
- IBM Technology Center

As part of this continuing commitment, IBM has teamed with leading commercial Linux distributors, Caldera Systems, Red Hat, SuSE, and TurboLinux to port, test, and certify the performance of IBM offerings running on various Linux distributions, enabling you to exploit the full potential of Linux.

---

## 1.2 TurboLinux



TurboLinux Inc. is a leader in Linux-based software solutions for Internet and enterprise computing infrastructure, including network operating systems and value-added clustering software to manage network traffic and provide peer-to-peer distributing computing capabilities. TurboLinux solutions run on a wide range of computing platforms with global support from major manufacturers including IBM. TurboLinux embraces the best of open source, and develops open systems and commercial software solutions that integrate into existing IT environments to leverage and protect an organization's computing investments.

Founded in 1992, TurboLinux is based near San Francisco with offices around the world and is a chosen IBM distribution partner for Linux.

The IBM @server xSeries Brand team works closely with TurboLinux and other distribution partners to fully test and certify xSeries and Netfinity servers are ready to perform with Linux.

---

## 1.3 Introducing the xSeries family of servers

IBM @server xSeries is the new IBM Intel server brand. xSeries are Intel processor-based servers with X-architecture technology enhancements, for a level of reliability, performance and manageability previously out of reach for industry-standard servers. This represents a full circle of technology evolution for Netfinity heritage in X-architecture, which is based on technologies derived from the IBM ES, RS and AS series servers, bringing mainframe category technology to the industry-standard architecture. Also, NUMA-Q will be aligned with xSeries to ensure IBM resources are focused most effectively on the Intel marketplace.

xSeries servers are available in the following four categories:

- Point Solution Servers
- Universal Servers
- Rack Optimized Servers
- Extremely Scalable Servers

For more information on the xSeries, visit the Web site at:

<http://www.pc.ibm.com/us/eserver/xseries/>



---

## Chapter 2. Linux installation

This chapter describes the basic installation steps needed to install and run TurboLinux on most xSeries and Netfinity servers. Although these steps have been tested on a number of different models, you may find subtle differences in the installation.

---

### 2.1 What you need to know about your hardware

Because Linux is very flexible and allows you a lot of control over your hardware, you will need to know some basic information about your computer hardware. This can be helpful in diagnosing and solving problems.

This information includes:

- Hard drives - interface (SCSI or IDE) and size
- CD-ROM - interface (SCSI or IDE) and make
- SCSI adapter - make and model number
- Display adapter - make and model number
- Mouse - mouse type and connector type
- Network card - make and model number
- RAM - amount of RAM in your system
- Monitor - make, model, maximum resolution, horizontal and vertical scan rates, sync rates and other information

Other information you may want to note, because it may be needed sometimes, includes:

- Interrupt level. The computer hardware uses interrupts to get the attention of the CPU. You need to see the documentation from the manufacturer of your hardware to figure out how to set it. Older ISA and some PCI hardware may have unique interrupts. Newer PCI hardware can share interrupts in some circumstances.
- I/O address. This is the address that is used to pass I/O (input/output) information back and forth from the I/O devices to the CPU. This is almost always unique to the hardware device.
- Base address. This is a memory location that is used by the I/O device as work space. This is in addition to any memory that may be on board the devices.

With the newer PCI devices and controllers, this information is often set automatically.

For IBM @server xSeries and Netfinity servers and other IBM products including monitors and SCSI adapters, the ultimate source for all information is:

`ftp://ftp.pc.ibm.com/pcicrse/psref`

Here you will find PSREF (Personal Systems Reference) sheets for all IBM PC products both current and withdrawn. You can also find a lot of useful information at the following Web sites:

- <http://www.pc.ibm.com/support>
- [http://www.pc.ibm.com/us/netfinity/tech\\_library.html](http://www.pc.ibm.com/us/netfinity/tech_library.html).

---

## 2.2 Installing on a ServeRAID adapter

Before you start the installation it is important to have the latest level of microcode for all your hardware. You can download the latest BIOS and diagnostic updates and ServeRAID adapter firmware for your IBM Netfinity server from:

`http://www.pc.ibm.com/support`

If you are installing to a RAID array, please read Chapter 4, “The ServeRAID controller and TurboLinux” on page 91 before continuing. You may have to modify the boot diskettes in order to complete the installation.

For the IBM networking products, you will find all the latest available code at:

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

For other manufacturers' products, you will need to consult their Web site or contact their technical support.

### **Stop**

Always update the BIOS of your IBM Netfinity server to the latest level and firmware of all adapters before installing the operating system.



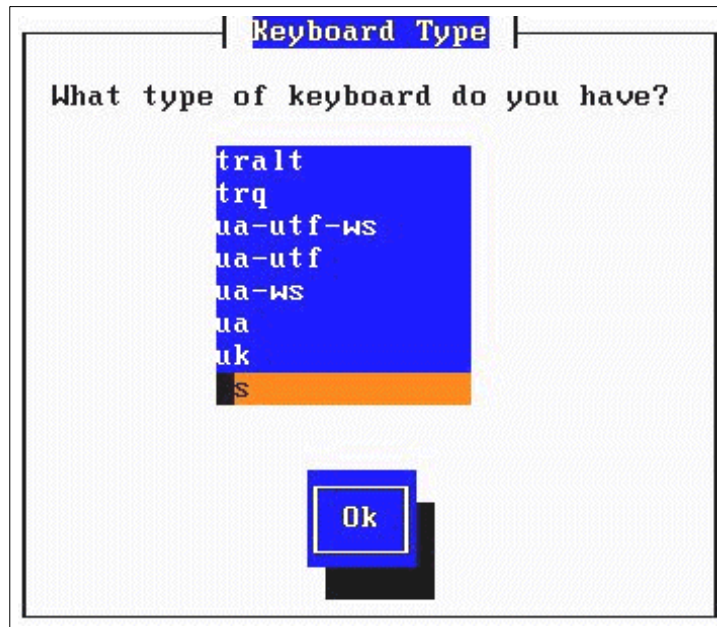


Figure 3. Keyboard type window

We can now choose from a number of keyboard types. They are listed in Figure 3. If you are installing or using Linux in a language other than English, you will want to choose the proper keyboard language.

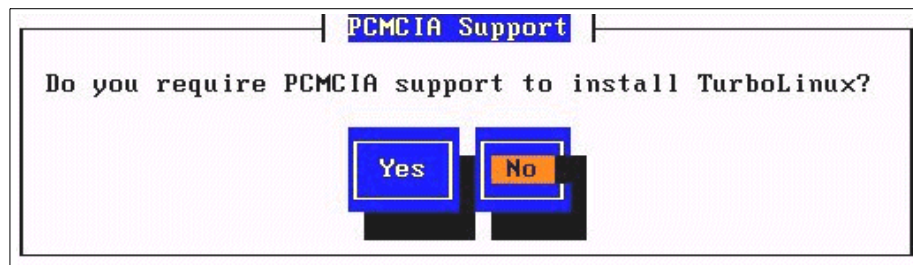


Figure 4. Is PCMCIA needed?

The next window (Figure 4) asks if PCMCIA support is needed for the install to continue (for example, if you were installing with a PCMCIA CD-ROM). This support is not needed on any IBM server.

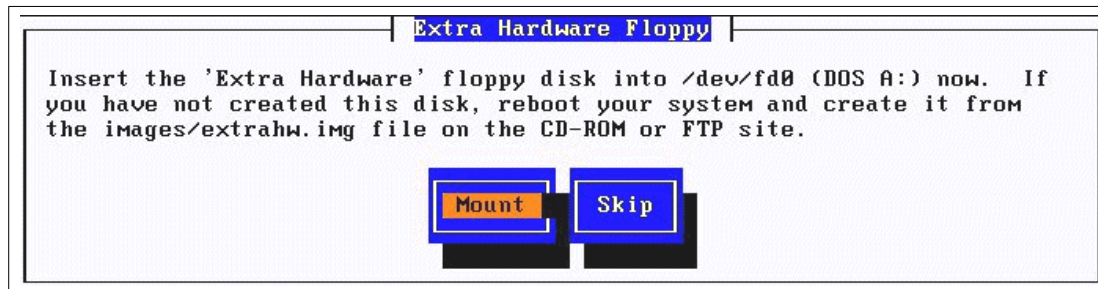


Figure 5. Prompt for Extra Hardware diskette

A new feature in TurboLinux 6.0 is the ability to add new drivers that are not on the distribution CD. If you are installing TurboLinux on a RAID array, you will need to insert the diskette and choose **Mount** so TurboLinux can load the proper drivers.

The “Extra Hardware” prompt only appears if you boot from diskette. It does not appear if you boot from the CD. Therefore, if you are installing on a RAID array or any other DASD device that requires an updated driver, you must insert the diskette. If you are installing on a ServeRaid 4 card, you should have already created a modified “Extra Hardware with IPS 4.4” diskette. That process is outlined in Chapter 4, “The ServeRAID controller and TurboLinux” on page 91.

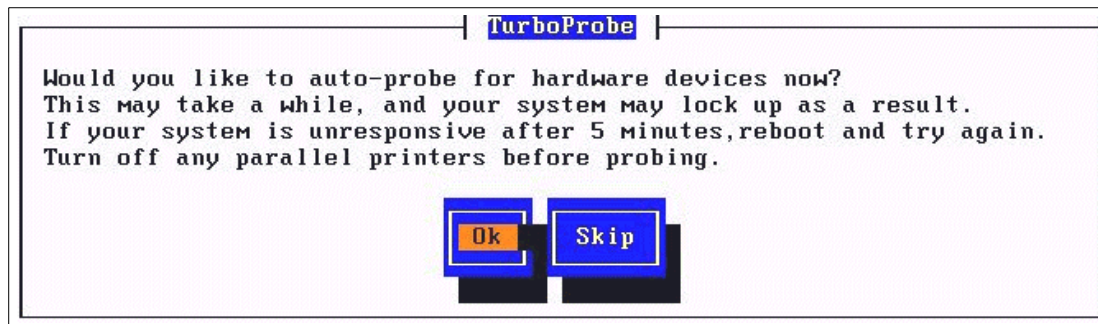


Figure 6. Auto-probing for hardware

Auto-probing for hardware is next. Linux will collect configuration information about on-board adapters and those you have installed into ISA and PCI slots. This procedure generally works very well. In certain circumstances, though, probing for information may cause a hardware device to hang. If this occurs, you would need to turn your machine off and on again and try to figure out the device that is causing the hang and either manually enter the information by

skipping the auto-probe or removing the offending card and restarting the autoprobe. Once your system is configured you can usually install the card, configure it manually, and reboot Linux without a problem.

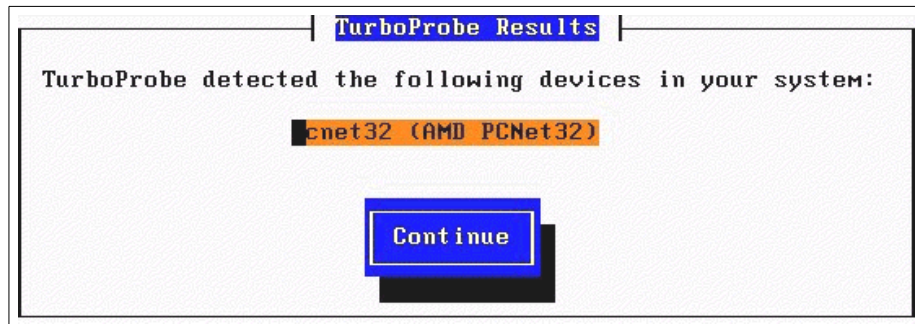


Figure 7. Hardware auto-probe results

After the system probes your hardware you will see the results of the probe in Figure 7. You see in the results that an Ethernet card was found, the AMD PCNet32. You will find additional or different devices on your system, depending on the configuration of your server.

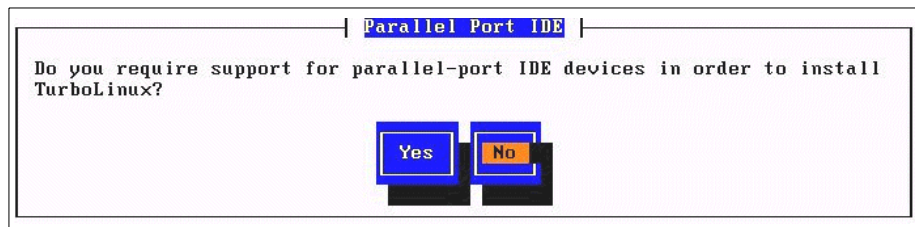


Figure 8. Is parallel-port IDE support needed?

You will next be given an opportunity to probe for parallel-port IDE devices, as shown in Figure 8. If you do not have any such devices you can skip the probe. The parallel-port devices include such things as parallel-port attached CD-ROMs, but do not include printers, which are installed later in the installation.

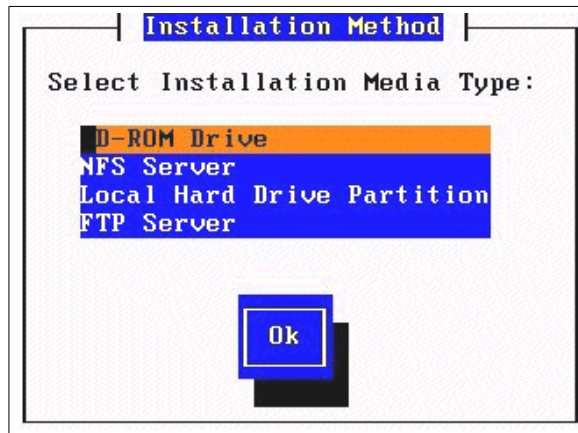


Figure 9. Installation method

There are several ways that you can install Linux. In Figure 9 you see four methods:

1. **CD-ROM.** This is the most common installation method. If your CD-ROM is locally attached to your system, choose this method. This might be an IDE, SCSI, or parallel-port device. If you are using this method, the installation description continues in 2.5, “Disk partitioning” on page 15.
2. **NFS Server.** The CD-ROM can be copied or mounted on another server that is already installed and running. This can be done with any network operating system that supports the network file system (NFS).

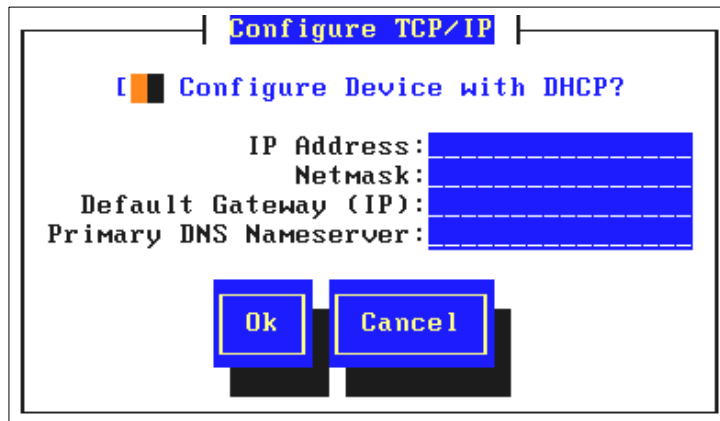


Figure 10. Network configuration window

Choosing to install via NFS generates the network configuration window. All of these fields must be completed for the install to proceed.

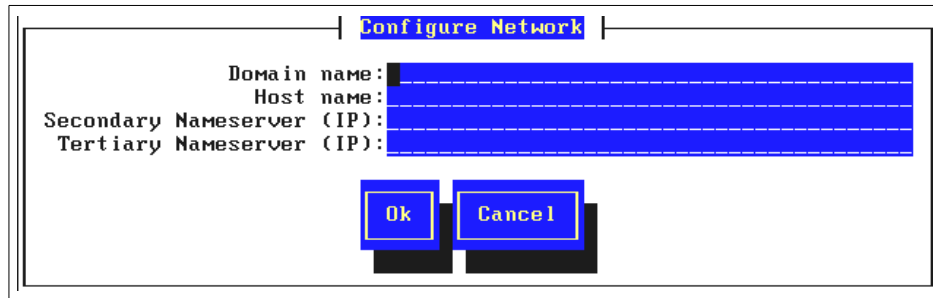


Figure 11. DNS Client configuration window

The domain and host name fields must also be completed, even if the information is not correct. The nameservers do not have to be filled in, unless you need host name resolution to connect to the NFS server that is hosting the install images.

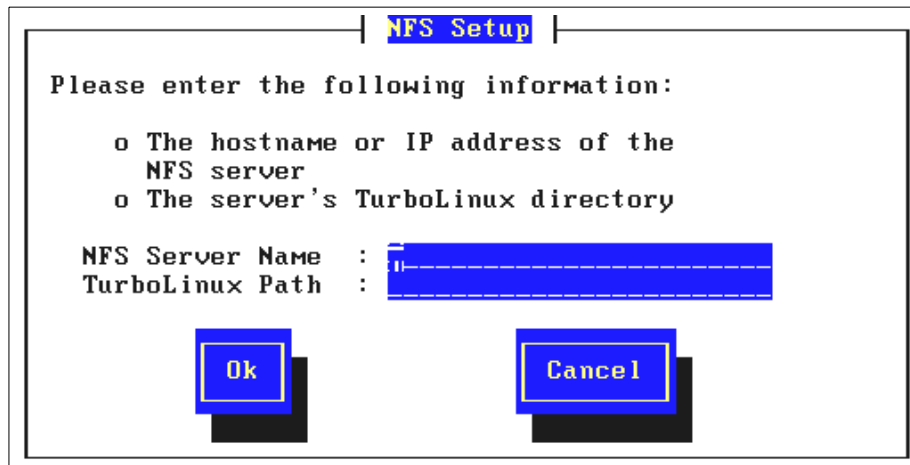


Figure 12. NFS Installation configuration window

Here you give the IP address or host name of the NFS server hosting the install images. You must also define the NFS mount point being used on the server. The install now proceeds to 2.5, "Disk partitioning" on page 15.

3. **Local Hard Drive Partition.** You can either create a dedicated partition with the install image or you can store the install image as a file and mount



it that way for an install. If you are using this method, move on to 2.5, “Disk partitioning” on page 15.

4. **FTP Server.** You can install by connecting to a properly configured FTP server. TurboLinux has several publicly available FTP servers connected to the Internet, and you can also configure an FTP server in your intranet. Selecting the FTP method gives you the following prompt:

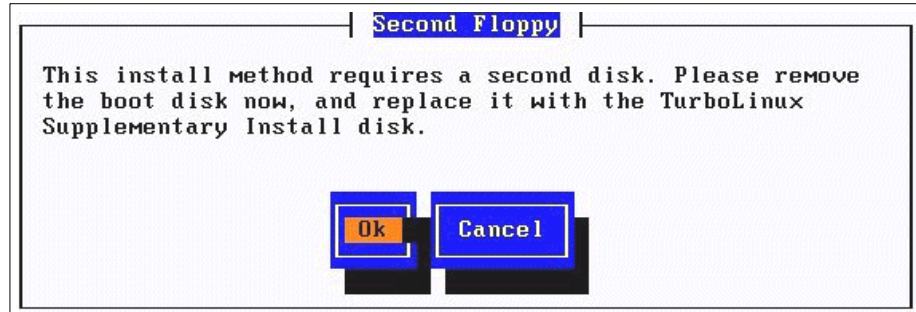


Figure 13. FTP server supplementary install disk window

If you have not created the supplementary install disk, you must do it now.

After that has completed, the FTP install takes you through the steps in 2.5, “Disk partitioning” on page 15 and 2.6, “Configure filesystems” on page 22 in this book, then returns to the following window.

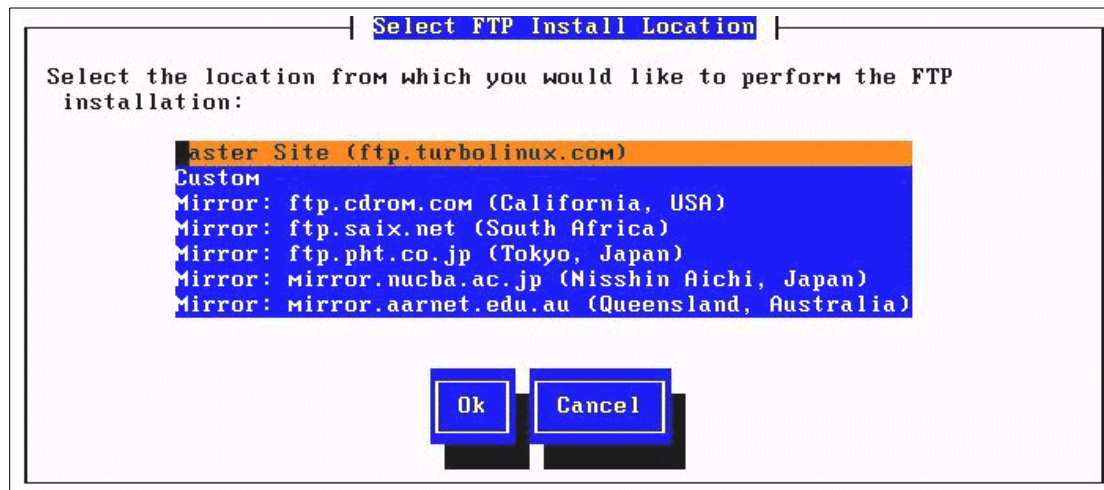


Figure 14. FTP site selection window

After the installer has read the diskette, you will see a list of publicly available FTP servers you can install from, and the option "Custom." If you choose one of the predefined FTP servers, you will then be prompted whether there is an FTP proxy between this server and the Internet. You will need to specify the IP address for that proxy in order for the installer to connect to one of them.

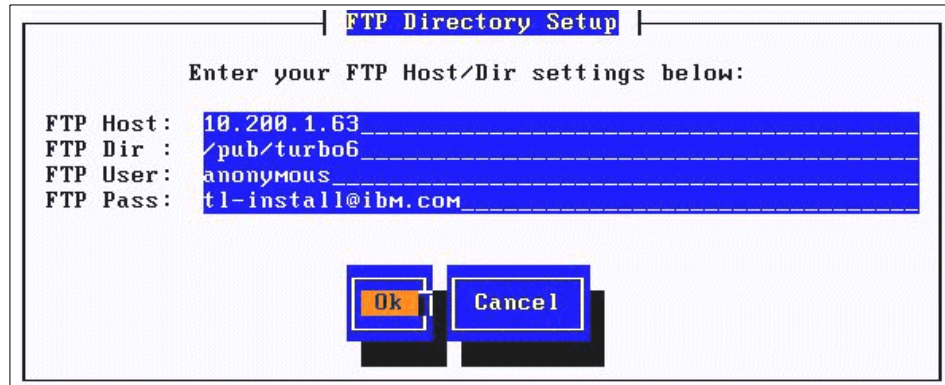


Figure 15. FTP Directory setup window

If you choose **Custom**, the installation process will look for a server which you have configured to give anonymous access to the TurboLinux install media.

---

## 2.4 Miscellaneous

Next you will be asked about how much detail you want your messages to have. Normal Verbosity is the default and should be used unless you are diagnosing an install problem.

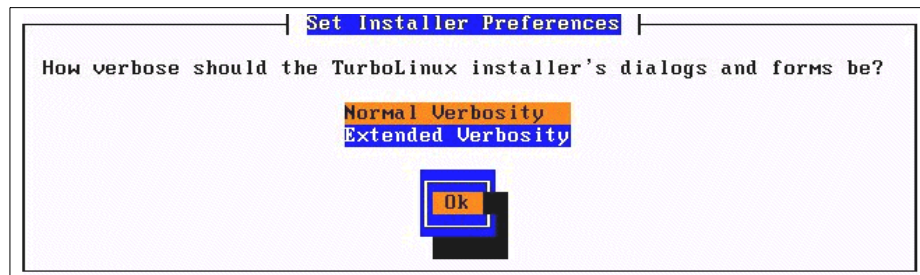


Figure 16. Selection of verbosity window

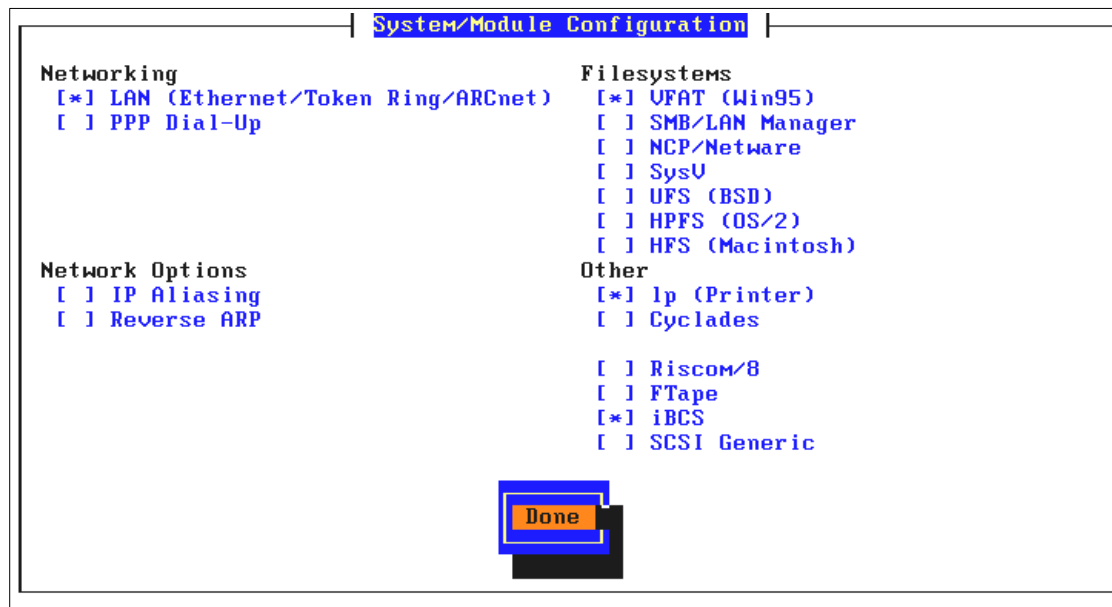


Figure 17. System/Module Configuration window

In Figure 17 you see the full menu that will be displayed if you choose **Extended Verbosity**. If you only see the Networking section on this page, you have selected **Normal Verbosity**. This window does not affect your ability to enable these options after the install has completed.

These choices include:

1. **Networking.** The item that is selected here is LAN (Local Area Networking). If you are going to be doing any networking using a Network Interface Card (NIC) you will want to leave this selected. If you are going to be doing dial-up networking using the Point-to-Point Protocol (PPP) to get e-mail and access the Internet, you will want to choose PPP. Most dial-up Internet service providers (ISPs) use PPP to allow customers to connect.
2. **Network Options.** These are generally items that you may use in particular situations. For an initial install you will probably want to ignore them until you see how the installed system operates on your network.
3. **Filesystems.** Linux supports many file system types. Because Microsoft Windows products are so common you will probably want to enable VFAT(Win95). You might also consider enabling SMB/LAN Manager support if you are going to be accessing Windows file systems and

printers across the network, or you are allowing them to access your system.

4. **Other.** The other choices are generally more specific applications that are hardware devices you may or may not have on your system. These include:

- **Ip (Printer)** support is obviously very useful for printing.
- **Cyclades** support is for the Cyclades multi-serial adapter.
- **Riscom/8** is for the Riscom/8 multi-serial adapter
- **FTape** support is for floppy tape drives, which are not typically used on the Netfinity servers.
- **iBCS** is a compatibility option to allow you to run binaries that are compiled for the iBCS standard and that can be run on several versions of UNIX as well as Linux on PCs. This is short for Intel Binary Compatibility Standard.
- **SCSI Generic** support is not necessary if your system is working okay with your current SCSI hardware, if you have any. It will determine your SCSI hardware and load the appropriate basic kernel. You might find this option useful to add later if you need more SCSI support.

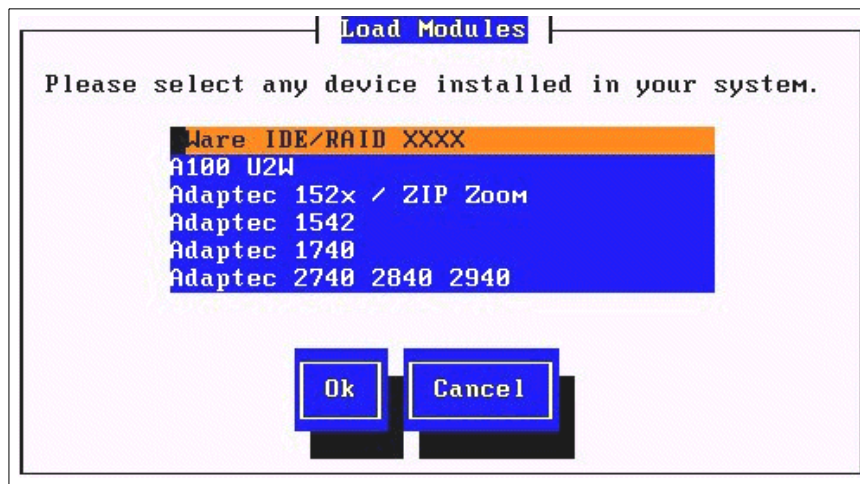


Figure 18. SCSI configuration window

If you answer “yes” to the “Do you have any SCSI devices,” prompt, you are presented the window shown in Figure 18. Here you can load modules for additional SCSI devices which have not yet been detected.

You have now completed the basic hardware setup for TurboLinux. Next you will need to lay out your hard disks.

---

## 2.5 Disk partitioning

There are as many ways to partition your disk as there are uses for a computer. Although, the use of your server will dictate the optimal partitioning structure, it is usually a good idea to do several test installs of the operating system before you actually commit to a firm layout. This way you can adjust the partition sizes with each install in order to be sure that the proper amount of space is available to each partition.

For additional information about how to partition Linux servers, an excellent discussion can be found at:

<http://www.linuxdoc.org/HOWTO/mini/Partition/index.html>

*Table 1. Starting disk space layout and usage for TurboLinux 4.0*

Partition Name	Mount Point		Partition Size Created Here( in MB)	Comment
hda1	/boot		20	This is not necessary but is advisable
hda5	/var		200	Logging
hda6	/tmp		250	Space for temp files
hda7	/home		250	The user's login area
hda8	SWAP		300	Swap is used if RAM is full
hda9	/		2000	Everything else

In Table 1 you see our suggested layout for a general-purpose Linux server. It should be read as a guideline only. Depending on the applications you are installing, you may want to have a separate file system for other mount points. Used in this example, we had a server with 256 MB of RAM with a 3 GB drive available for Linux. We used the following considerations to generate this table:

- **/boot** stores kernels (Vmlinuz.x) and initial RAM disk images (initrd.img)  
We have left space for multiple kernels and RAM disk images.

- **/var** contains all the server logs. While 200 MB might seem to be quite large for log files, services can be configured to be quite verbose.
- **/tmp** is where temporary files are stored by the system.
- **/home** is where users store all their files. Users can also install applications in this filesystem, so 250 MB is quite minimal. Most likely you will want to move this file system to a larger drive if one is available.
- The **Swap** file system functions the same way as “virtual memory” in Microsoft Windows. As a rule of thumb, you should have as much swap as RAM.
- **/** is the root filesystem. It contains all the files necessary to run Linux. In this configuration, most of the space will be taken by **/usr**, which is where our user level programs reside.

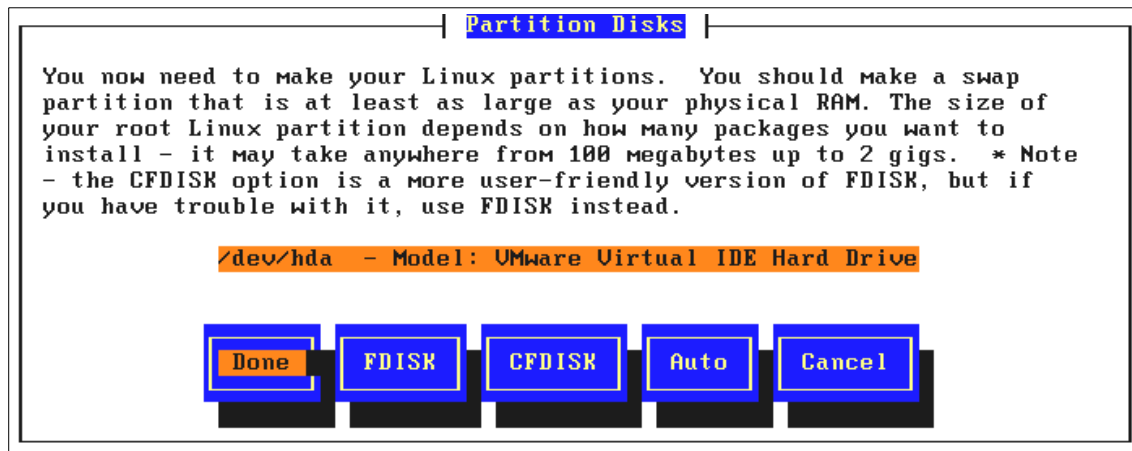


Figure 19. Disk partitioning method window

In Figure 19 you are presented with several methods to partition your disks.

- **FDISK**. Choose this if you are familiar with FDISK, or have an unusual configuration in FDISK. However, for most users, CFDISK is a better option.
- **CFDISK**. This is a more user-friendly version of FDISK. This is recommended if you choose to partition your disks manually.
- **AUTO**. The TurboLinux system will determine best how to lay out the partitions based on the amount of space available.

**Note**

CFDISK is not supported on FTP-based installs of TurboLinux 6. If you are installing via FTP, you will have to create your partitions with FDISK, or use the auto-partition option.

**Stop**

If you repartition or format your disk or partitions, you will lose any data that is on those partitions. If there is data that you want to save, be sure to back it up and do not format or create any new partitions.

```

                                cfdisk 0.81
                                Disk Drive: /tmp/hda
                                Heads: 128 Sectors per Track: 63 Cylinders: 519
-----
Name      Flags      Part Type  FS Type      [Label]      Size (MB)
-----
                                Pri/Log     Free Space   2043.57
-----

[ Help ] [ New ] [ Print ] [ Quit ] [ Units ]
[ Write ]

                                Print help screen

```

Figure 20. CFDISK introductory window

Clicking CFDISK brings up Figure 20, where you see the version of CFDISK to be configured. This is followed by information about the generic information about the drive. Then you will see a breakdown of the drive layout as it currently exists. The program will scan your drives to determine how it is laid out. The columns refer to the following information:

- **Name** is the name of the partition. This would be in the format /dev/hda1. The whole drive is called /dev/hda and the number 1 refers to a partition on /dev/hda.
- **Flags** would include information such as bootable, which we will see in Figure 18.
- **Part Type** refers to whether the partition is a primary or logical partition. A primary partition can be bootable, whereas a logical partition cannot.
- **FS Type** is the file system type. CFDISK labels all the partitions it creates as "Linux." If you would like to create partitions that are tagged for a different file system, you must first create them, then change their type. That process is described below.
- **Label** should be thought of as a nickname for a partition. It has no bearing on the function of the partition, and is there to make it clearer what the partition is being used for. This cannot be edited during the install of TurboLinux.
- **Size** is the size of the partition or free space, by default in MB.

At the bottom of the window in you will see the following entries:

- **Help** will give you help for items on the menu.
- **New** allows you to take the free space and to create a partition out of it.
- **Print** will allow you to print or display more detailed information about the drive to the window or to a file. You should note that print in UNIX and Linux does not necessarily mean sending the information to a printer. It can also mean saving it to a file or displaying on the window.
- **Quit** is used to quit this menu and return to a previous menu.
- **Units** allows you to specify how your drive information is displayed. This includes MB, sectors or cylinders. Choosing **Units** will toggle the size column. The available units are MB, Sectors, and Cylinders.
- **Write** will write the results to the hard disk. Until this point nothing you have done to the disk is permanent. Once you write the partition information to the disk it becomes permanent and whatever layout or information you had on the disk is overwritten. However, if you are working with a disk that has data on it and if you are dividing a current partition and the rest of the information has not changed, then you will affect only the areas on the disk that are different. The rest of your disk will still be the same.



**Note**

If you are going to be running multiple operating systems on your computer, including Microsoft Windows versions, it is preferable to install Linux last. LILO, the Linux bootloader, is a much more flexible bootloader than other operating systems provide, which makes the process of configuring a server to boot multiple operating systems much easier.

When you have create your primary and logical partition, you are given the following options (see Figure 21):

1. **Bootable** allows you to specify that the partition you created is bootable. Once you choose this you will see that the value in the Flags column for this partition will change to Bootable.
2. **Delete** enables you to delete the partition you have highlighted.
3. **Help** will give you help on your selection.
4. **Maximize** says to use the rest of the free space for this partition.
5. **Print** allows you to display a more detailed layout of the drives or to save it as a file.
6. **Quit** to quit.
7. **Type** will let you set the partition type that you have created. You will see the various partition types in Figure 21.
8. **Units** will allow you to toggle through various units that can be used including Megabytes, Sectors, or Cylinders.
9. **Write** writes your changes to the disk. Whatever changes you have made since the last write will need to be saved again.

```

                                cfdisk 0.81
                                Disk Drive: /tmp/hda
                                Heads: 128 Sectors per Track: 63 Cylinders: 519

```

Name	Flags	Part Type	FS Type	[Label]	Size (MB)
hda1	Boot	Primary	Linux ext2		19.69
hda5		Logical	Linux		200.82
hda6		Logical	Linux		248.07
hda7		Logical	Linux		248.07
<b>hda8</b>		Logical	Linux Swap		<b>299.25</b>
hda9		Logical	Linux		1027.69

```


```

**[Bootable]** [ Delete ] [ Help ] [Maximize] [ Print ]  
 [ Quit ] [ Type ] [ Units ] [ Write ]  
 Wrote partition table to disk  
 Toggle bootable flag of the current partition

Figure 21. Fdisk final disk layout window

In Figure 21, we have created a Primary partition and marked it bootable, created a Linux Swap partition, and created a few other partitions to mount other file systems. You should now choose the **Write** option and answer Yes when prompted as to whether you really want to write these changes.

When you are done making any changes to the disk partitions you can **Quit** the CFDISK program. This places you back in the Partition Disks window (Figure 19), where choosing **Done** moves you to the next step in the installation.

### Stop

You should never use Linux to create file systems for an operating system type other than Linux. Because each operating system manufacturer has its own peculiarities, you should use the software provided by the manufacturer. Use the operating system that supports that file format natively. In this case use Windows NT to create an NT file system, OS/2 to create an OS/2 filesystem and so forth. Failure to do this may result in your partition or drive becoming unusable. You should also create and format any non-Linux partitions first for this reason.

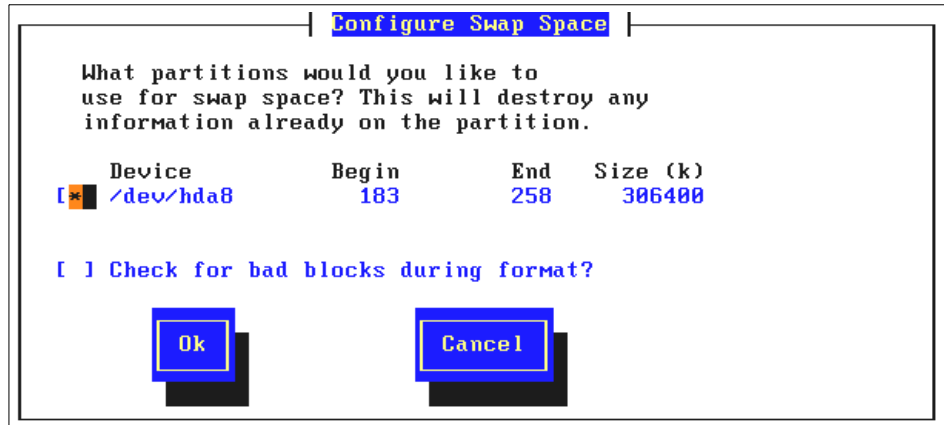


Figure 22. Activate swap space

The next step is to activate the swap space as shown in Figure 22.

If you have already set up the swap space TurboLinux will find it and prompt you to activate it. This means that the system will use this device as the swap space for the TurboLinux system. It will tell you what the size of the device is that was found.

You are also asked whether you want to check for bad blocks during formatting. Because of the high quality of today's drives, this is almost never needed. On a RAID system, this is never needed, as the RAID controller handles such issues itself. Also many newer drives will automatically redirect bad disk blocks to good disk blocks, so this becomes unnecessary. If you have a questionable drive attached to a non-RAID controller, you might want to check this option.

## 2.6 Configure filesystems

In this section, you will configure your file systems.

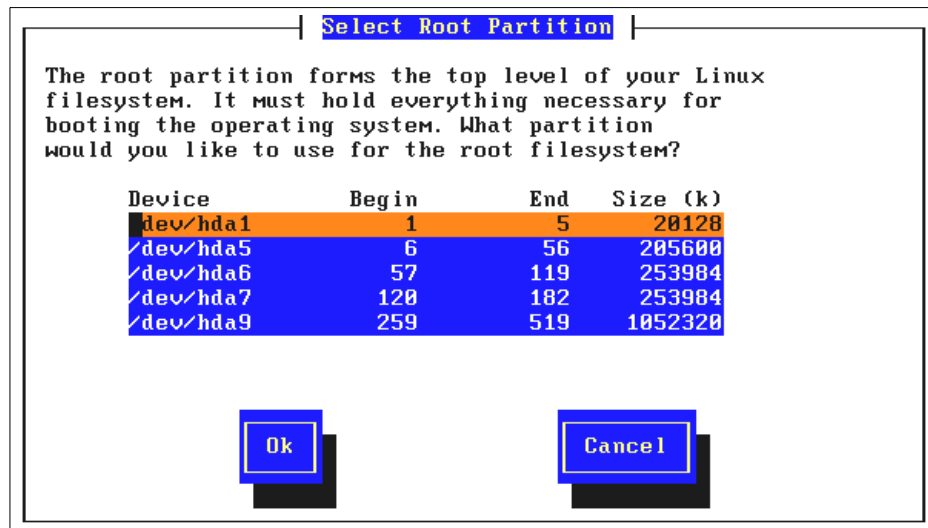


Figure 23. Define the root filesystem

The next step is to confirm which partition you want to use as the root partition (/). When you created your partitions you have already made that decision in your own mind. Now you must tell Linux which partition will be the root (/).

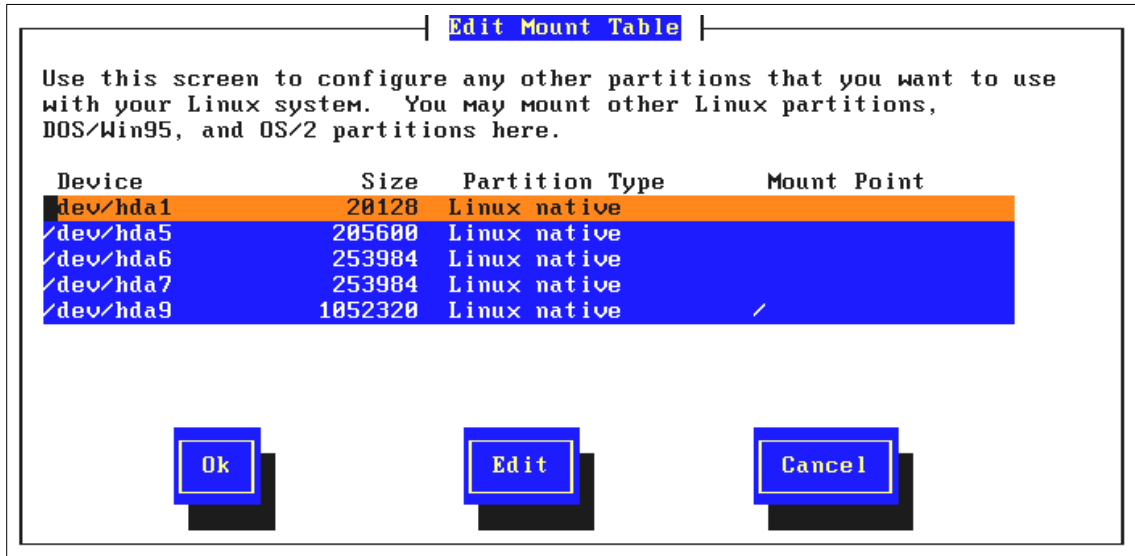


Figure 24. The Initial Mount Table window

After that, you define mount points for any other partitions you have created. Figure 24 is the initial mount table.

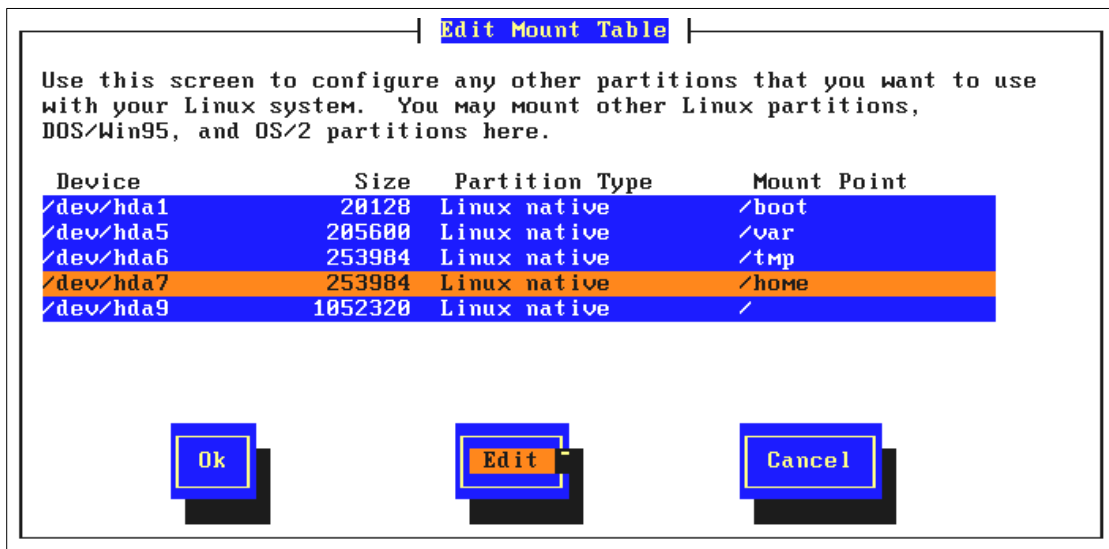


Figure 25. The completed Mount Table window

The completed mount table is shown Figure 25. You will note that this is similar to the display that you saw with CFDISK, except that the swap space is not shown. This is because the swap space is not a mounted file system, but is accessed at the device level instead of the file system level.

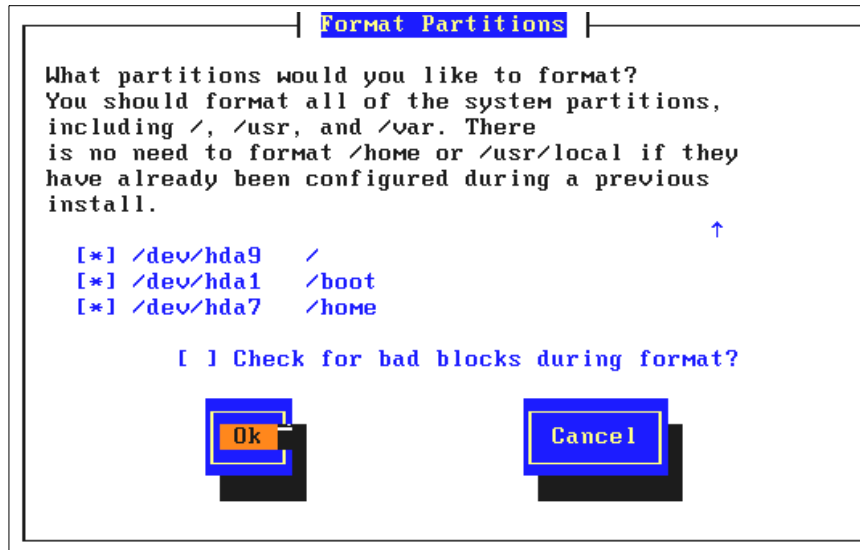


Figure 26. Format partitions

You can now format the partitions. Since this is a new install, we have chosen to format all partitions. If this had been a reinstall, you would probably not want to format all of them, since they would contain data that you would not want deleted.

If the partitions already existed with that size and layout and if you have data in any of the partitions that you want to save, you should deselect the format. In any case, any data in these areas should have been backed up before this point. If this is not the case, you may want to cancel out of this process and back up the data before continuing.

## 2.7 Configure the primary network interface card

The next step will be to configure your network interface card (NIC). If TurboLinux automatically detects the NIC, you will not have the option to manually input any information. TurboLinux 6.05 autodetects all NICs that IBM supports in xSeries and Netfinity servers. Therefore, since the NIC is automatically detected you will begin to configure the TCP/IP protocol.

This protocol is used to communicate on the Internet and in the vast majority of local area networks. Many local area networks allow you to get IP addresses automatically from a pool of addresses. TCP/IP uses the process BOOTP or DHCP to get the address.

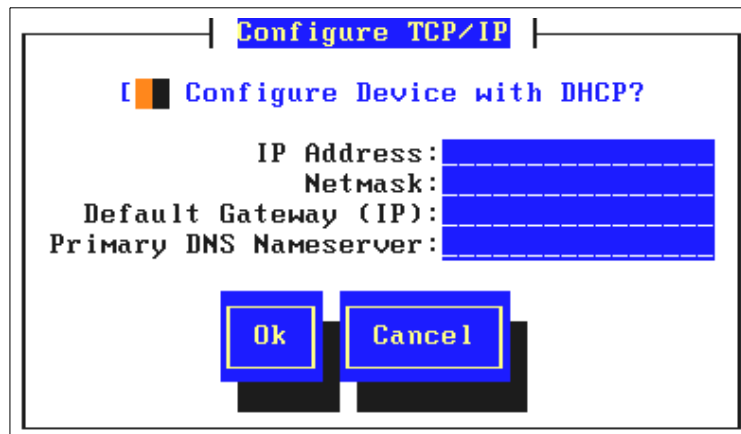


Figure 27. Configure TCP/IP window

In Figure 23, you should check **Configure Device with BOOTP/DHCP** if your network uses this.

If you are using fixed IP addresses instead, you will need to fill in the following information. This information is assigned by the network administrator.

- **IP Address:** every host on the network has a unique IP address.
- **Netmask** will determine the range of addresses that are considered part of your local network.
- **Default Gateway (IP)** is the address of a router that connects to other networks.

**Primary DNS Nameserver** is the address of the nameserver that will take a host name and return an actual IP address.

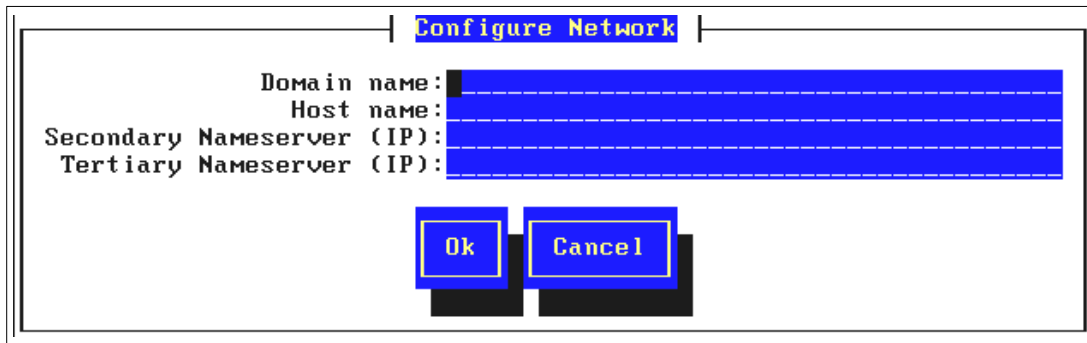


Figure 28. DNS client configuration window

If you gave the address for a DNS name server in the previous window, you will now be able to edit or complete the domain and host name information on the window shown in Figure 28.



## 2.8 Software package installation

When you get to the step of installing the actual packages you will need to decide which packages you want to install. Remember that you can always add or remove packages after the install has finished and the server is up and running. Those steps are covered more completely in Chapter 3, “Basic system administration” on page 39. For now, we have the options listed in Figure 29:

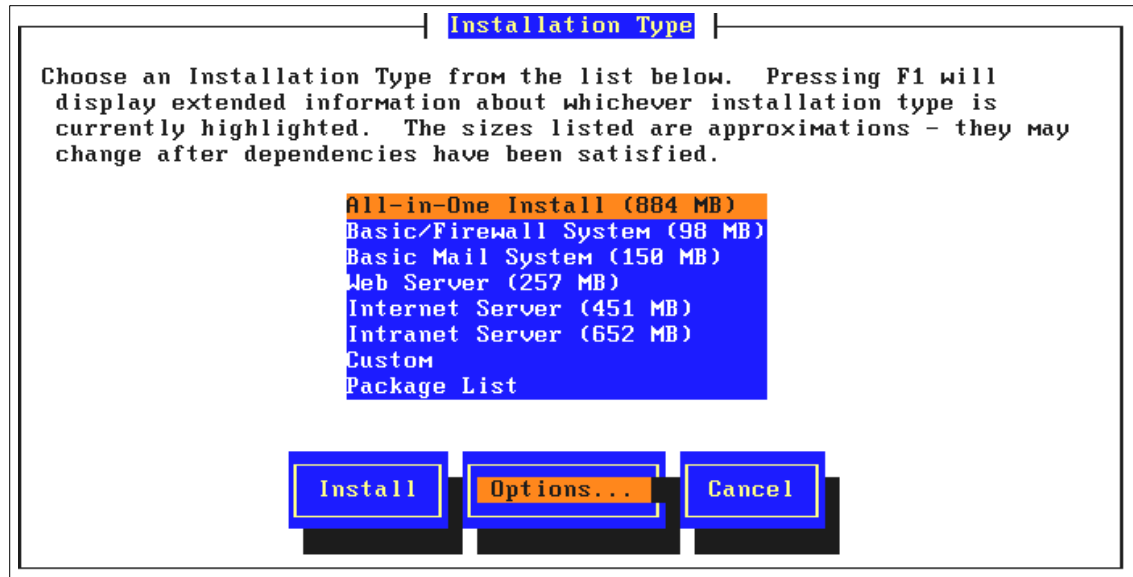


Figure 29. Installation Type window

There are several self-explanatory preconfigured sets of packages, as well as Custom and Package List. Selecting **Custom** allows you to select individual packages, while choosing **Package List** allows you to insert a diskette that contains a list of packages created on another Linux machine.

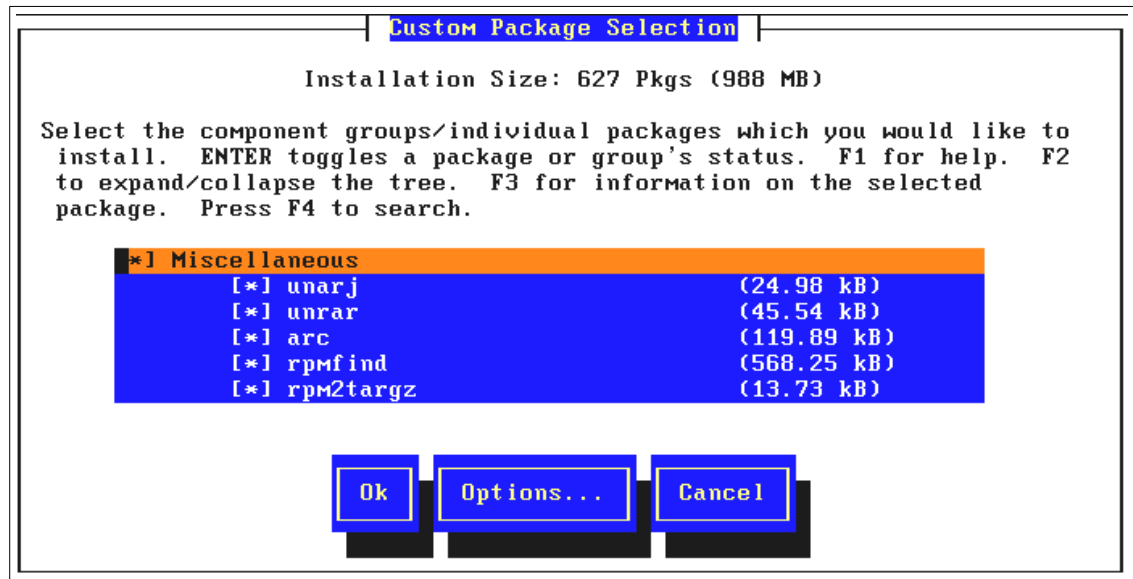


Figure 30. Custom Package Selection window

If you choose **Custom**, you can pick individual packages to install. You can also get to this window by choosing **Customize** after selecting one of the predefined package sets.

## 2.9 Selecting a kernel

In this section, you will select a kernel to use.

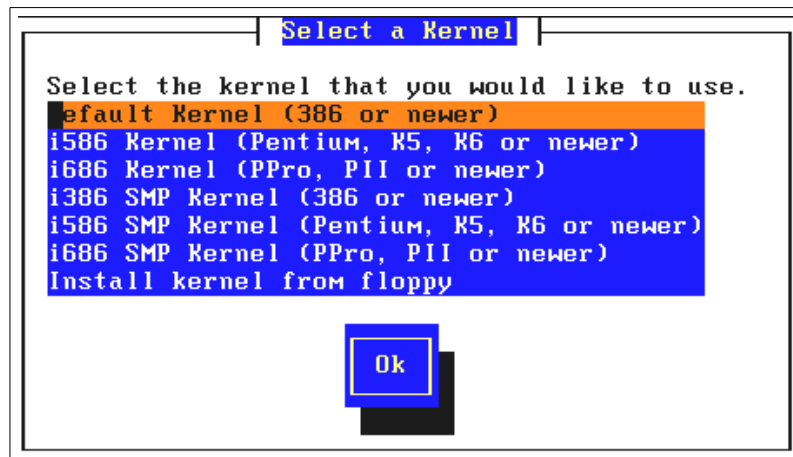


Figure 31. Kernel selection window

As you can see in Figure 31, TurboLinux comes with six different precompiled kernels. All currently marketed xSeries and Netfinity servers can use the i686 kernel. There is also an option to install a kernel from an image you have placed on a floppy disk.

If you are installing on a ServeRaid 4, you should have created a diskette labeled “ServeRaid 4.4 driver” at the end of 4.1.3, “Compiling the ServeRAID driver after TurboLinux is installed” on page 94. You now need to insert that floppy in the diskette drive and type the following commands:

```
Press ALT-F2. That should take you to a command prompt]
mount /dev/fd0 /mnt/mnt/floppy
rm /mnt/lib/modules/current/scsi/ips.o
cp /mnt/mnt/floppy/ips.o /mnt/lib/modules/current/scsi/ips.o
rm /tmp/extramodules/ips.o
cp /mnt/mnt/floppy/ips.o /tmp/extramodules/ips.o
umount /mnt/mnt/floppy
Press ALT-F1. You should be back to the kernel selection screen]]
```

Now you can select your kernel. Remember that it must be the same one selected when compiling the driver in 4.1.3, “Compiling the ServeRAID driver after TurboLinux is installed” on page 94.

## 2.10 LILO

In Figure 32 you must provide a choice of where to place the LILO, which is used to manage the boot process.

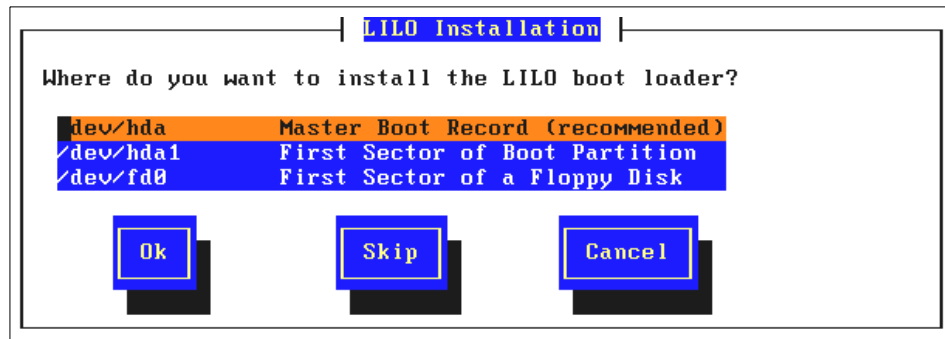


Figure 32. LILO Installation window

LILO is the most commonly used bootloader for Linux. You have a few choices of where LILO resides. They are as follows:

1. **Master Boot Record.** This is the master area of a disk that determines which partition to boot from. You can either install LILO here or you can install a commercial software package such as System Commander, Partition-IT or any of a number of other packages here. They can then point to the various partitions on your system in case you are running multiple operating systems.
2. **First Sector of Boot Partition.** You can install the LILO at the start of a partition. This is pointed to by the master boot record. The contents of the lilo.conf file are identical in either case. You will want to put the LILO in the boot partition if you are running multiple operating systems.
3. **First Sector of a Floppy Disk.** You may want to boot off a floppy if you do not want to modify your master boot record to point to your Linux install. You will also want to create a backup disk with the LILO installed on it as a protection in case the boot information for your LILO gets corrupted. This would allow you to access your system and then to rebuild LILO.

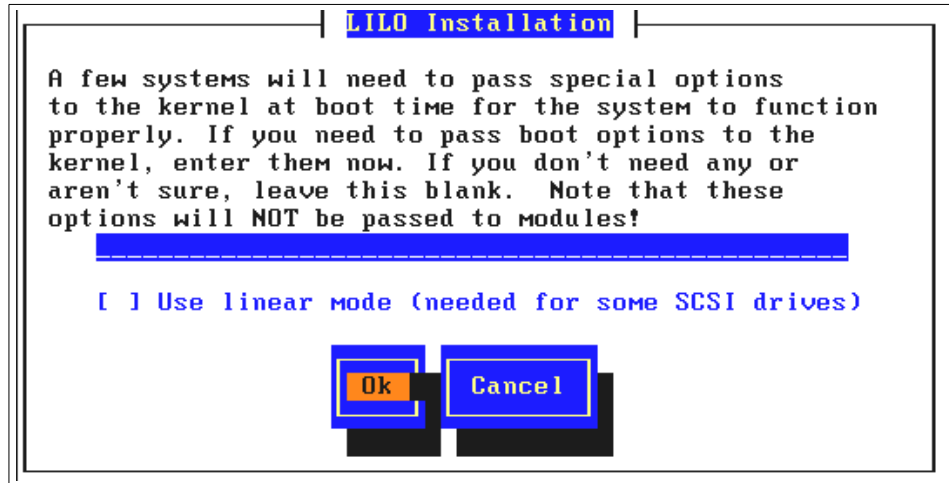


Figure 33. LILO installation parameters window

There may be additional parameters that you want to specify to the LILO process. You can specify these parameters in the window in Figure 34. This is necessary in some cases where:

- The hardware is not recognized by the auto-probe process.
- The hardware is recognized, but requires additional parameters to work or be fully functional.
- You want to add tuning parameters such as memory used, RAM disk sizes, etc.
- Other configuration or boot issues that need to be addressed.

You can get more information about LILO in the online documentation, as well as in many books on Linux. There is also much documentation on the Internet about LILO configuration. Two useful discussions are:

<http://www.linuxdoc.org/HOWTO/mini/LILO.html>

<http://www.linuxdoc.org/HOWTO/LILO-crash-rescue-HOWTO.html>

After LILO has written itself to disk, Some of the information that is saved in the file allows you to set up an initial RAM disk (stored in the file system as the file `/boot/initrd.img`), space for the kernel to be loaded, and hardware-specific information LILO that you specified above.

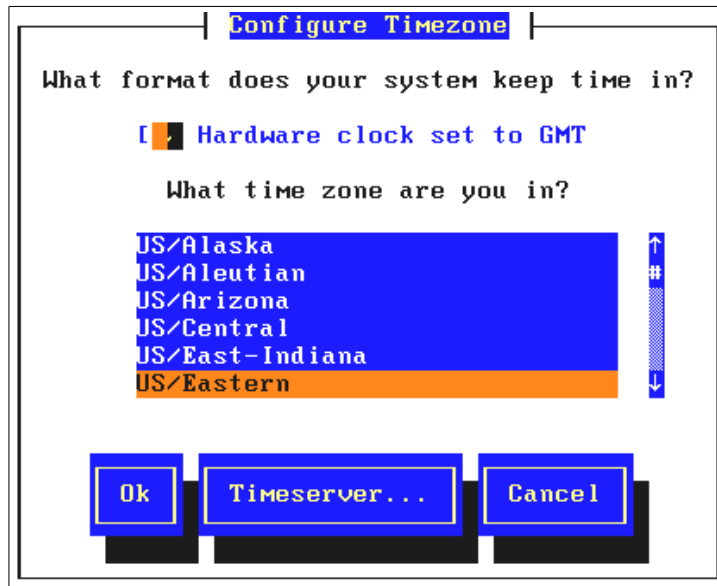


Figure 34. Time zone configuration window

Time zone configuration is next, including the ability to have the server get its time from a timeserver with either the NTP or RDATE protocols.

You also have the option of setting the hardware clock to GMT, then selecting the **Hardware clock set to GMT** here. The advantage of choosing GMT for your clock is that you can access your server across various time zones. Each user can be assigned his or her own time zone in their .profile file and the timestamps on the files will always agree. Otherwise, the timestamps on files will not be consistent if different users access files from different time zones and have customized their time zone entry.

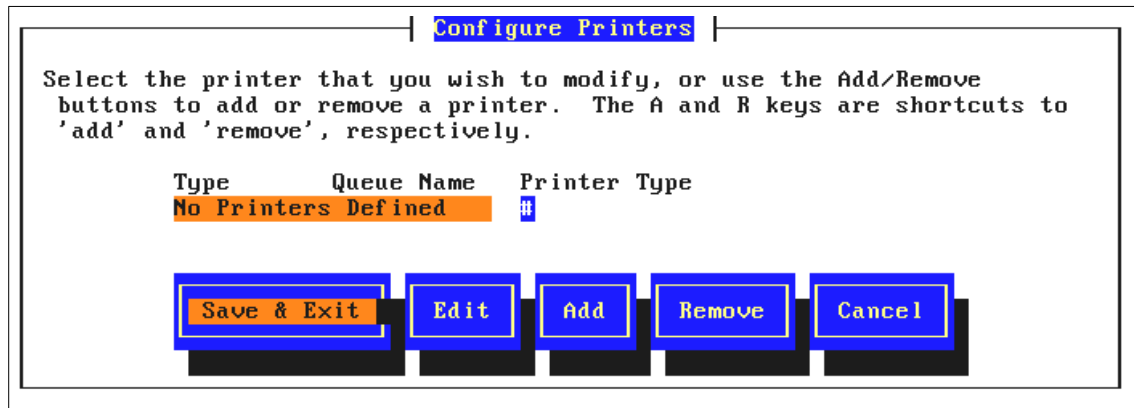


Figure 35. Configure printers

You will eventually want to access a printer from Linux. You can set up your printers during the initial installation as shown in Figure 35.

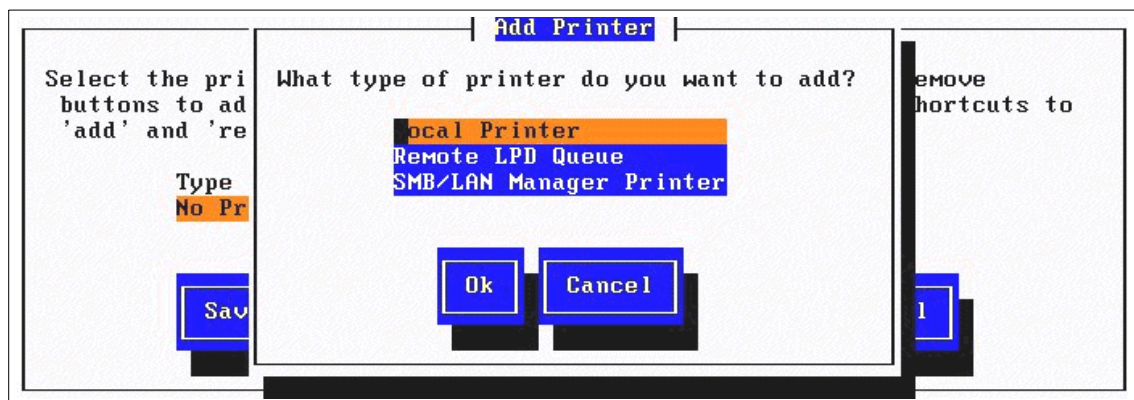


Figure 36. Add print type window

When choose the **Add** option you will see the options listed below:

- **Local Printer.** Choose this option to configure a printer attached locally to a parallel or serial port.
- **Remote LPD Queue.** Choose this option to configure a printer that is connected to a server on the network that you will access with TCP/IP printing.
- **SMB/LAN Manager Printer.** Choose this option to configure a printer that is connected to a server on the network that you will access with NetBIOS. Most Microsoft Windows and IBM OS/2 printers are NetBIOS printers.

If you are not sure what your printer configuration will be, you can skip this step and create print queues after the installation is complete.

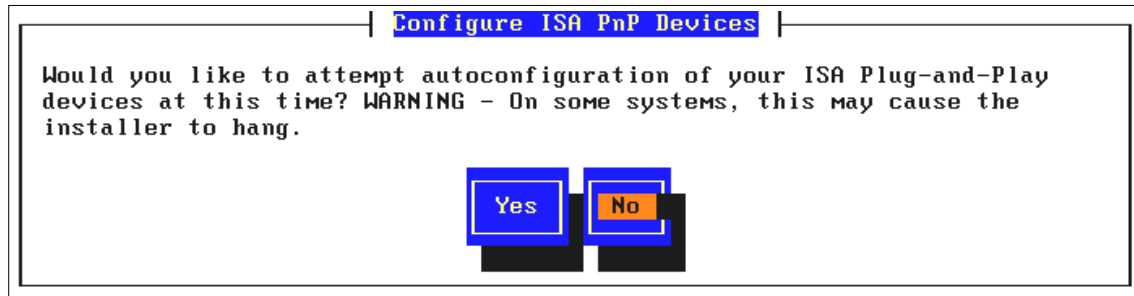


Figure 37. Configure ISA PnP Devices window

If you have any ISA PnP cards, TurboLinux can probe for them here.

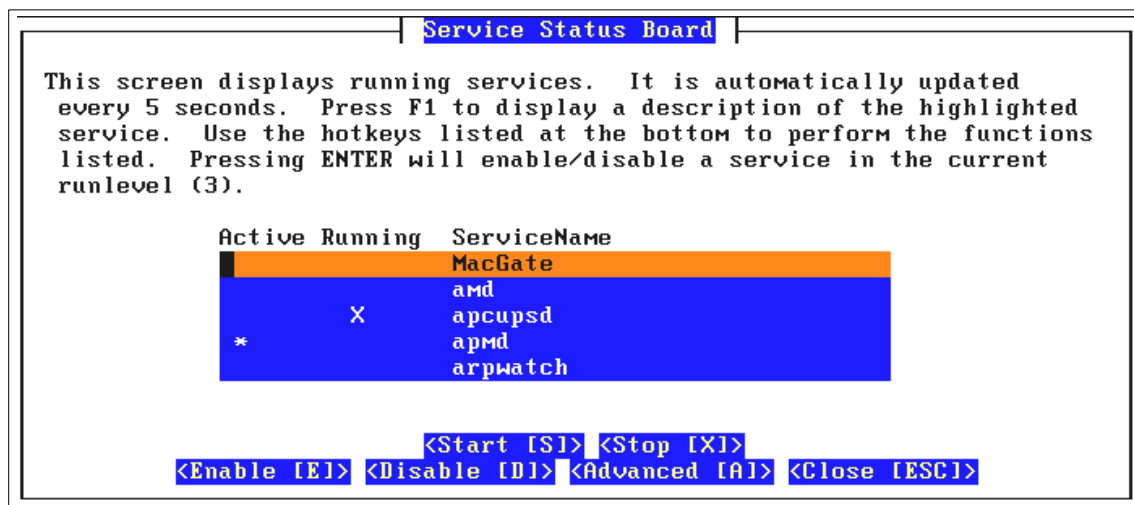


Figure 38. Services Status Board window

This next window ( Figure 38) allows you to modify the services that are running currently. There is not really any reason to configure services now, and we recommend that you skip this window and deal services after the installation is complete.

Exiting the services window brings you to a prompt to set the root password (Figure 39).



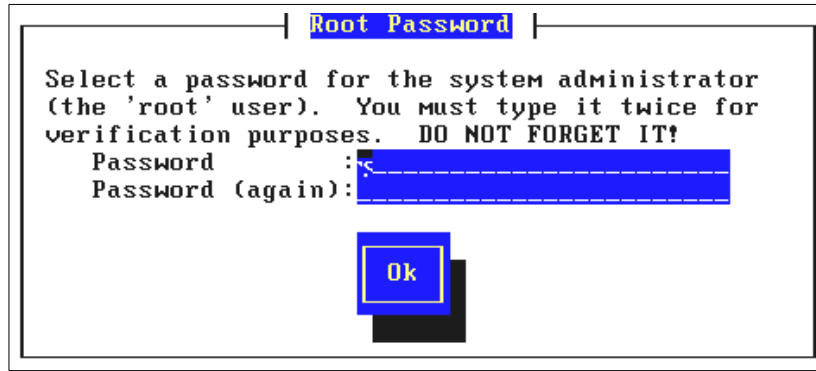


Figure 39. Root Password window

## 2.11 X Server setup

One of the most common problems when installing Linux on xSeries and Netfinity servers is the configuration of the X Server, which is the part of Linux used to provide the GUI. In the steps that follow, a minimal install of X is done, but most of the configuration of X will be deferred. Configuring X will be addressed in Chapter 3, “Basic system administration” on page 39.

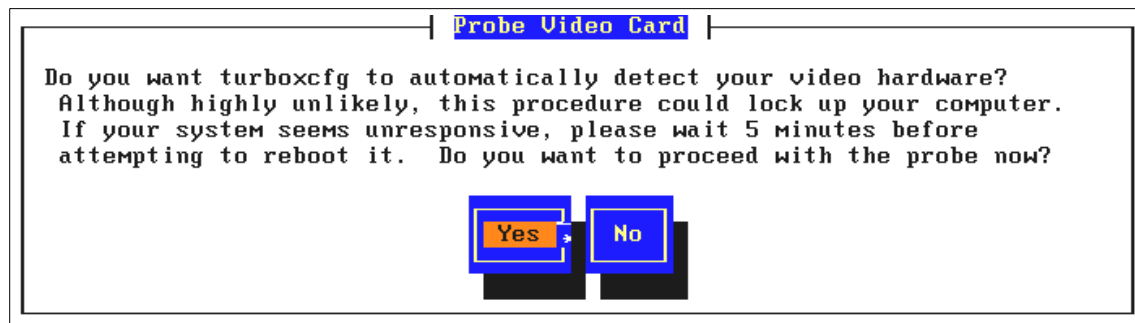


Figure 40. Probe Video Card window

Say **No** at the above prompt.

The Select Video Card window allows you to select the model of the video card in the server. Select **Unlisted Card** on the Figure 41.

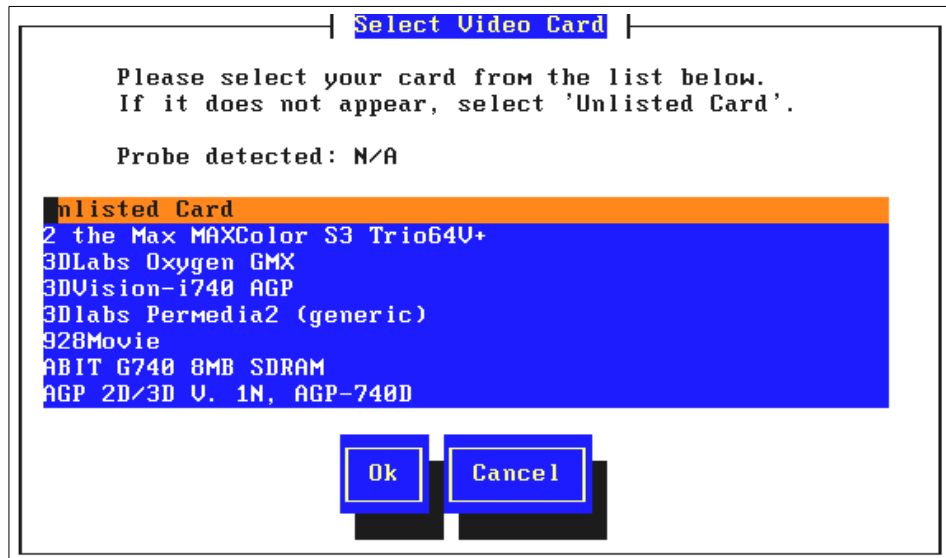


Figure 41. Video card probe results

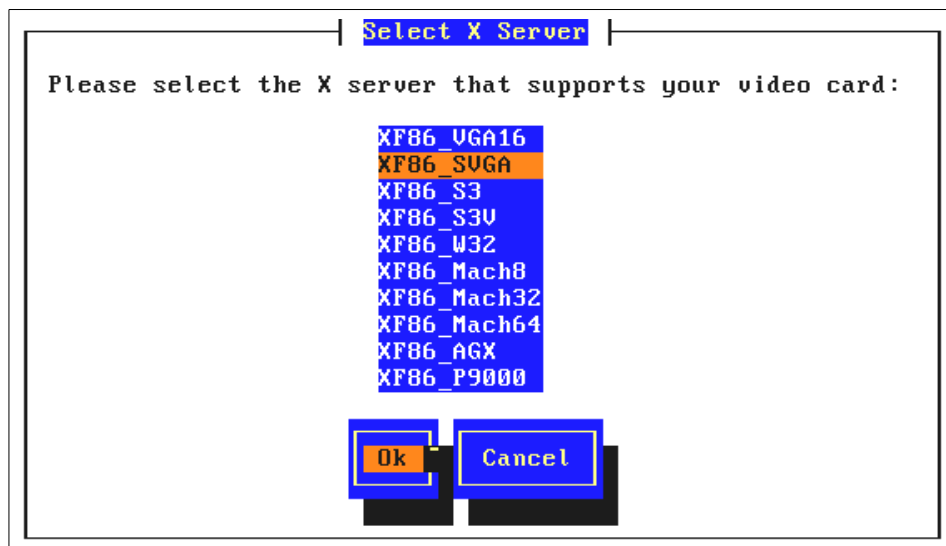


Figure 42. Select X Server window

Selecting the server XF86\_SVGA here is safe for all xSeries and Netfinity servers.

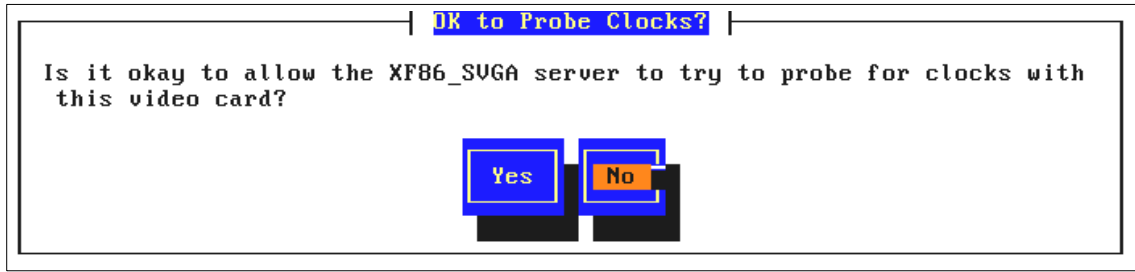


Figure 43. Probe Clocks window

Select **No** on this window to avoid unnecessary complications.

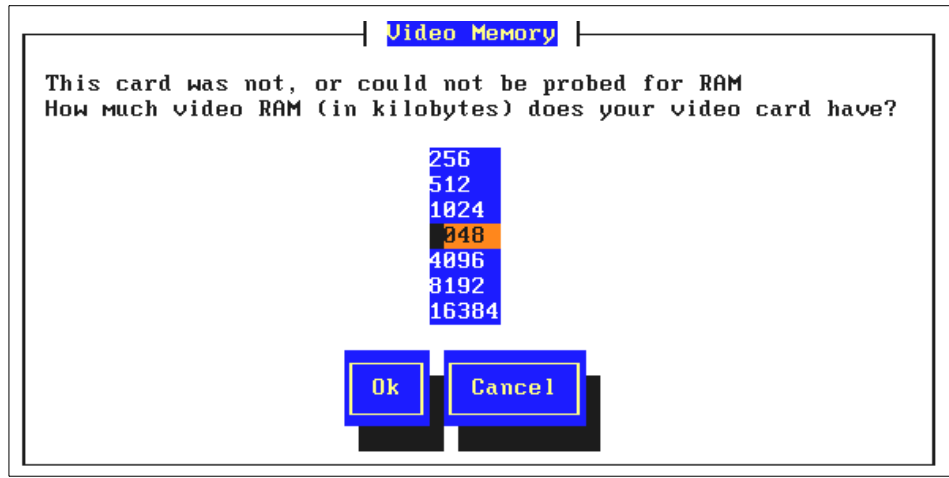


Figure 44. Select Video Memory window

Selecting **512** is a safe number to choose for video memory. If you are sure of the amount of video memory in your server, you can specify it here. However, remember that we will revisit X server configuration after installation.

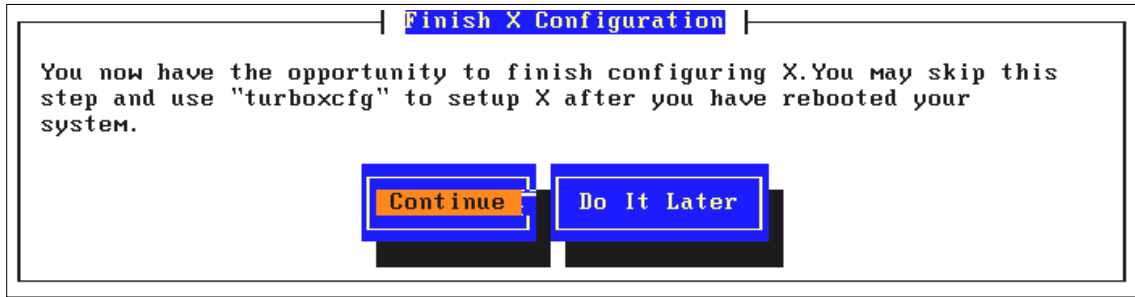


Figure 45. Finish X Configuration window

This dialog box defaults to Continue. You should choose **Do It Later**, which will complete the install and display the following window.

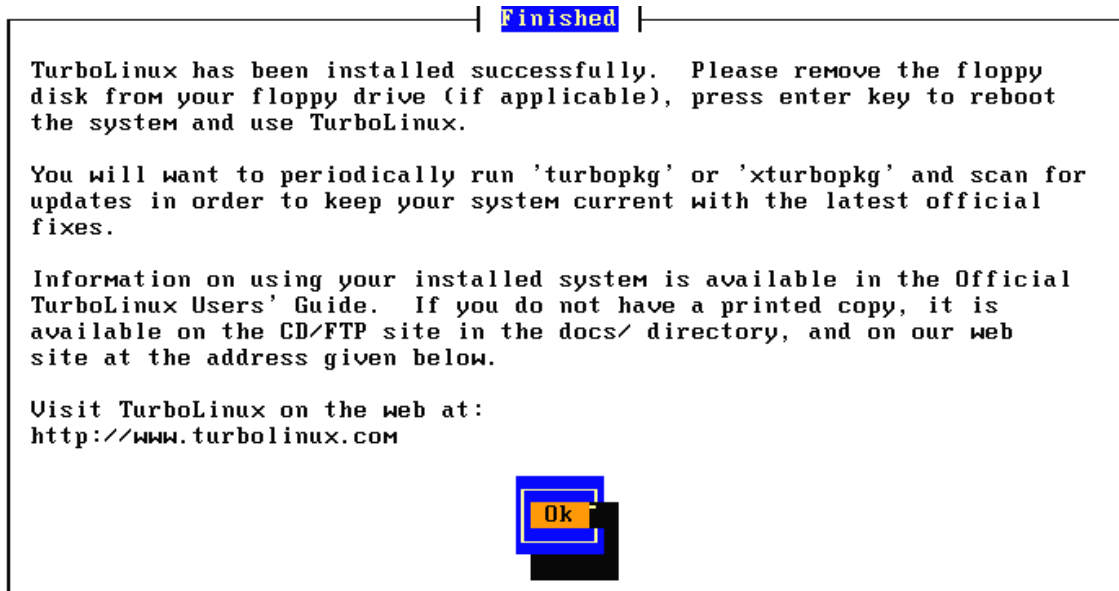


Figure 46. Installation completed

You have now completed the basic install of TurboLinux 6. Chapter 3, “Basic system administration” on page 39 will address configuration and administration tools.

---

## Chapter 3. Basic system administration

Linux follows the conventional UNIX model of storing configuration information in plain text files under the directory /etc. Many of these files are human readable, and many others can be understood with a little experience with the system. However, it is quite time consuming and confusing to try administering a Linux server by directly editing files in /etc, at least until you are more experienced. For this reason we will emphasize the tools TurboLinux provides to make administration more convenient and understandable to novice and intermediate Linux users, while at the same time pointing to the actual files in /etc being modified.

---

### 3.1 Configuring X with most Netfinity and xSeries servers

Current Netfinity and xSeries servers have several versions of S3 video cards that are not fully supported by the version of XFree86 (the X server) that ships with TurboLinux. Therefore, you may encounter an issue of configuring X to work properly. In this section we will start with instructions to configure an X server with the generic SVGA server. We will then give instructions for using the VESA frame buffer server (see 3.1.2, “Installing the VESA frame buffer server” on page 43), a generic driver that will give basic support to any video card.

#### 3.1.1 X Windows configuration and startup

X Windows configuration is a process that is still a work in progress. In Appendix B, “Working video modes for IBM Netfinity servers” on page 345 are some of the monitor and video adapter settings you need to be concerned about when setting up your X Windows system. In order to configure your X Windows you need to run a tool that will probe the system for information and build or modify a file to include the appropriate information. Two tools available to perform the X Windows configuration that can run from your Linux command prompt are:

- `turboxcfg`. This is also called Xconfigurator.
- `XF86Setup`. This program is a standard X Windows tool and can sometimes provide information that is different from `turboxcfg`.

In Figure 47 is an example of executing `turboxcfg`.

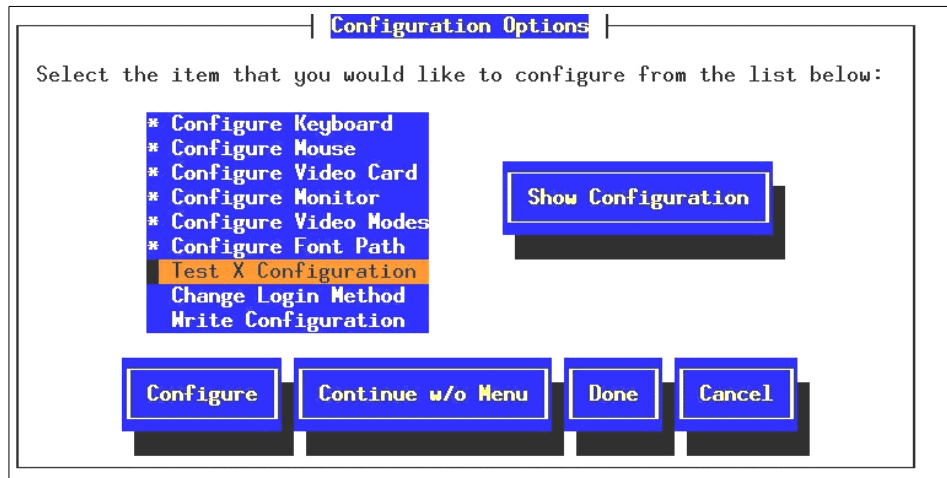


Figure 47. Configuration Options window

It is advisable to start out by selecting **Show Configuration** to see how the system is currently configured.

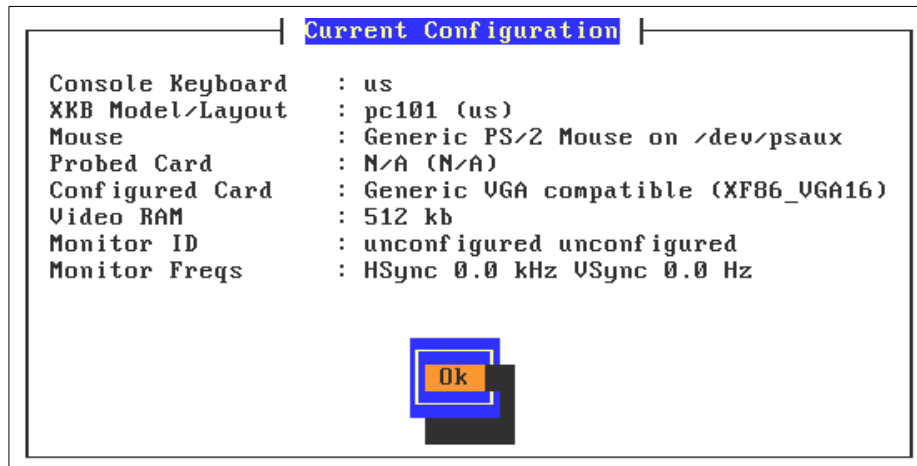


Figure 48. Current Configuration window

Above you see our incomplete configuration left from the install. To properly configure X, follow the following steps:

1. First we must confirm that the correct X server has been installed. Highlight **Configure Video Card** (Figure 48) and press Enter. You will see a window similar to Figure 49.

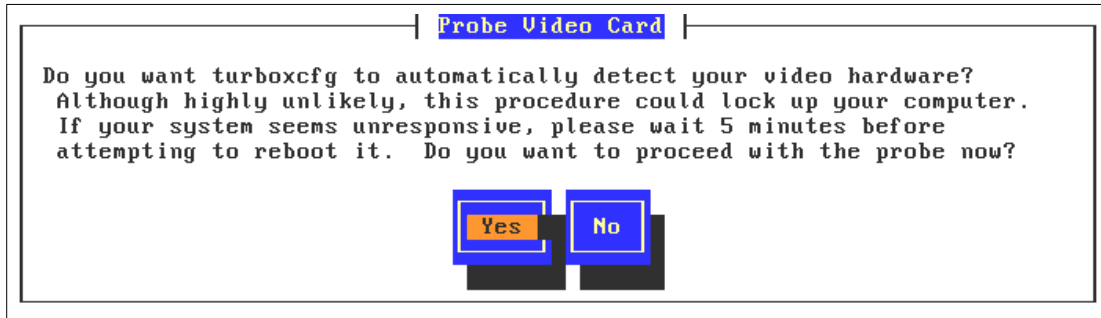


Figure 49. Probe Video Card window

2. Select **Yes** in Figure 49.

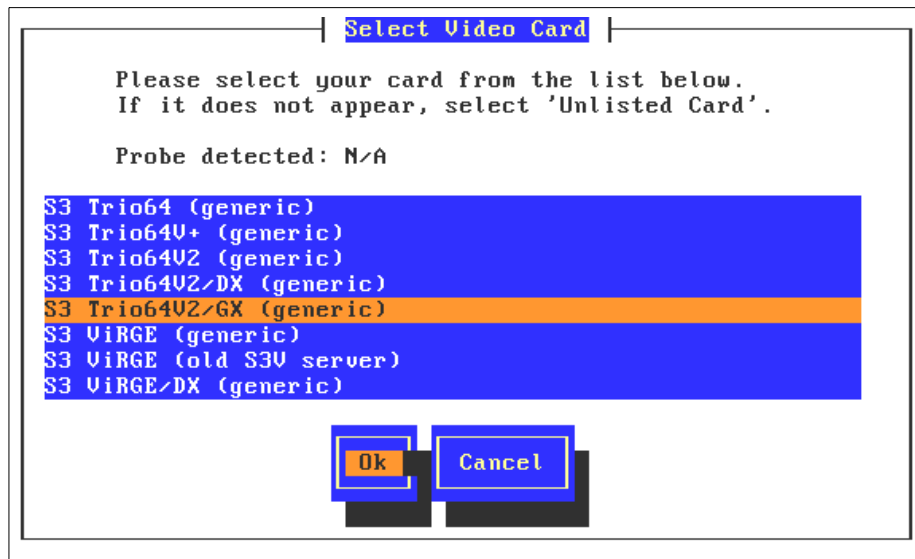


Figure 50. Select Video Card window

3. In Figure 50, note that the card has not been detected but we know that the Netfinity 5000 we are using has a S3 TrioV2/GX video card installed. Therefore, we select it and press **OK**. You will see a window similar to Figure 51.

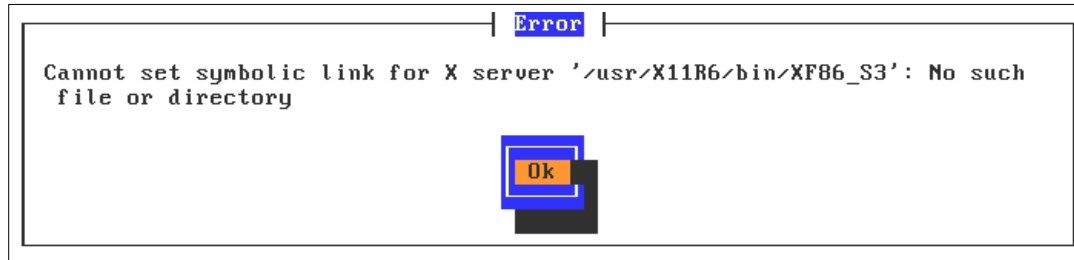


Figure 51. Error window

4. Here the correct X server was not installed, and turboxcfg complains about this fact. The error above indicates that the X server XF86\_S3 has not been installed, so we now install it by mounting the TurboLinux 6 CD and installing the file. The commands to do this are:

```
mount /mnt/cdrom
cd /mnt/cdrom/TurboLinux/RPMS
rpm -Uhv XFree86-S3-3.3.6-6.i386.rpm
```

After the XFree86-S3 package is installed, you will see a window similar to Figure 52.

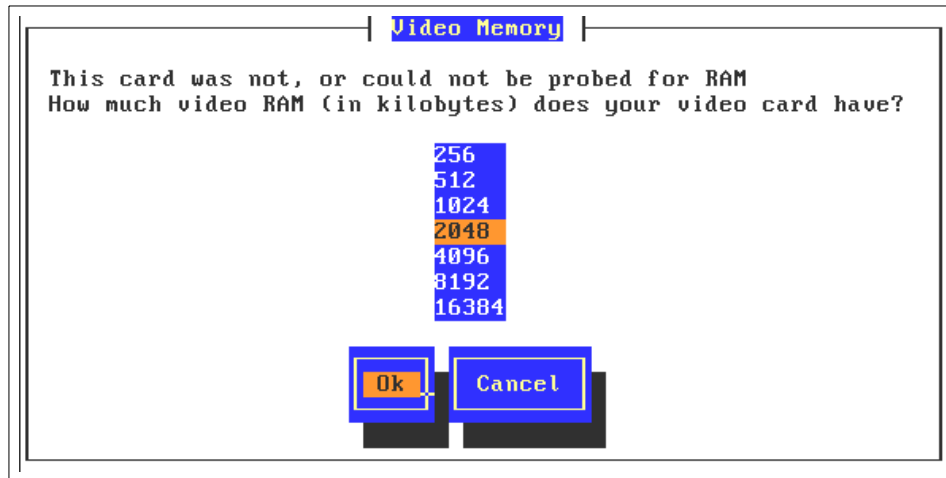


Figure 52. Video Memory window

5. Turboxcfg finds the server and proceeds to ask for the amount of video RAM on the card.



6. Next you should select **Configure Keyboard**. The questions in this section ask which keymap code to use, how many keys are on your keyboard, and the country code to use.
7. Configure Mouse asks for you to specify the type of mouse and number of buttons.
8. Configure Monitor requires you to choose the manufacturer and model of your monitor.
9. Configure Video Modes allows you to set the maximum video resolution and color depth.
10. Configure Font Path allows you to use either 75dpi or 100dpi fonts.
11. Test X Configuration allows you to see if your configuration works properly.
12. If the X configuration works, you can choose **Change Login Method** to change to graphics, or choose to leave it in text mode.
13. After everything has completed successfully, you can select **Write Configuration**, which saves the information to the file `/etc/X11/XF86Config`.

### 3.1.2 Installing the VESA frame buffer server

If you have problems configuring the X server, or would like to create an image or process that runs on any video card, you should install the VESA frame buffer driver. The frame buffer is designed to give limited functionality to any VESA compliant video card. The following is a set of instructions needed to configure the frame buffer.

1. With the system started in command mode, log in as "root".
2. Mount the TurboLinux Companion CD and install the frame buffer server package with the following commands:

```
mount /dev/cdrom /mnt/cdrom
cd /mnt/cdrom/TurboContrib/RPMS
rpm -ivh XFree86-FBDev-3.3.6-6.i386.rpm
```

3. Save the current symbolic link and create a new symbolic link by running the following commands:

```
mv /etc/X11/X /etc/X11/X.old
ln -s /usr/X11R6/bin/XF86_FBDev /etc/X11/X
```

4. Open `/etc/lilo.conf` to add a new entry for the frame buffer server.

```
pico /etc/lilo.conf
```

The file should look something like this:

```

boot=/dev/sda
map=/boot/map
install=/boot/boot.b
prompt
timeout=50
default=linux
image=/boot/vmlinuz
    label=linux
    root/dev/sda1
    initrd=/boot/initrd
    read-only

```

5. Make a copy of the existing entry. Change the label on the new entry to linux-fb or something else intuitive to you. In the new entry, and the line vga=xxx (where "xxx" is defined in Table 2) after the image line and change the label so that it is unique.

Table 2. Screen resolution table

Screen Resolution 640x480	Screen Resolution 800x800	Screen Resolution 1024x768	Screen Resolution 1280x1024	Bits/Pixel
769	771	773	775	256
784	787	790	793	32K
785	788	791	794	64K
786	789	792	795	16M

The following example is an entry in /etc/lilo.conf with the frame buffer server installed at a resolution of 800 x 600 and 64K colors:

```

image=/boot/vmlinuz
    label=linux-fb
    root/dev/sda1
    vga=788
    initrd=/boot/initrd
    read-only

```

6. Do not change the default image until you have verified that the new image works correctly. Update the master boot record and the LILO boot loader by running the command:

```
lilo
```

7. Edit /etc/X11/XF86Config to create a new section screen entry for the frame buffer server. Copy the following example of a screen entry and

make the necessary changes. The corresponding depth value defined by `zzz` is from the table above. `xxxx` and `yyyy` depend on predefined strings in `XF86Config`. Replace `xxxx` with the string following the `Identifier` under the `Device` section. Replace `yyyy` with the string following the `Identifier` under the `Monitor` section:

```
Section Screen
Driver fbdev
Device xxxx
Monitor yyyy
Subsection Display
Depth = zzz
Modes default
EndSubsection
EndSection
```

8. Reboot the system and remove all the media.
9. You may receive a virus warning after you restart the server; this warning is normal. Select **Change is expected**.
10. At the LILO boot prompt, press the Tab key for kernel options. You should now have two options: `linux` and `linux-fb`. Select **linux-fb**.
11. After the system has restarted in command mode, you can start the X server by issuing command:  

```
startx
```
12. If this works, you may want to edit `/etc/lilo.conf` again and make `linux-fb` the default. However, that is not necessary.

---

## 3.2 Turbonetcfg

`Turbonetcfg` is TurboLinux's multi-purpose network configuration tool. With it you can configure everything from the IP address of your NIC to the Apache Web server. To invoke this tool, type `turbonetcfg`. Although it is a text-mode utility, we recommend that you run it from inside X, since some of the windows require a larger console than is normally available without X.

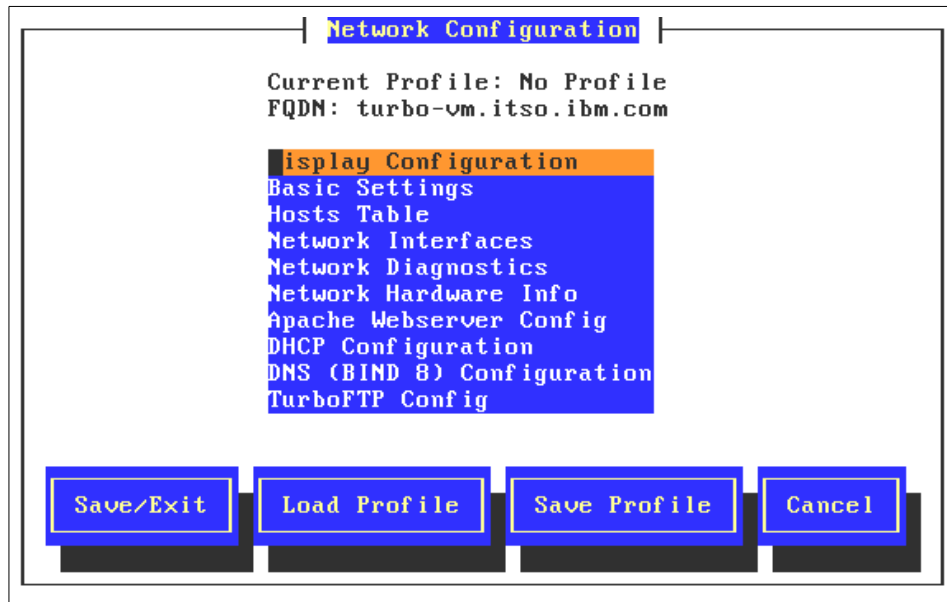


Figure 53. Main window of turbonetcfg

Figure 53 is the main window for turbonetcfg. What follows are brief explanations of all the options seen on this window.

- **Display Configuration.** Displays the current configuration of host and domain name, as well as the NICs in the machine and default router.
- **Basic Settings.** Allows you to set the host and domain name, and add search domains, secondary nameservers, and the default gateway and gateway device.
- **Hosts Table.** Used if you do not have a nameserver, or would like to specify the IP address of frequently used machines. It should also have 127.0.0.1 as localhost. It is a good idea to add your own IP and host name to this table, as it causes GNOME to start much more quickly.
- **Network Interfaces.** The network interfaces dialogue allows you to manage all the NICs in the server, including the ability to add, remove, or change interfaces without rebooting.
- **Network Diagnostics.** This is a very interesting feature of turbonetcfg. Selecting this option will cause a TurboLinux to run a series of network tests. Note that if the nameserver cannot be contacted during the “Testing Name Lookup (getbyhostname)” query. The query stays on the window for several minutes as if it has hung. However, it will eventually time out and show you a windows with the results similar to Figure 54.

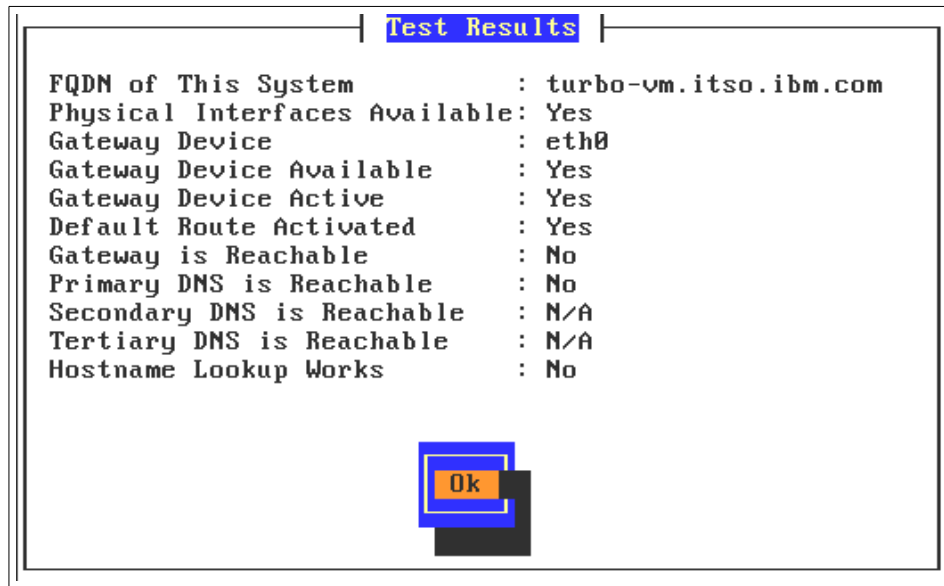


Figure 54. Test results window

- **Network Hardware Info.** This choice lists all the active interfaces and the corresponding kernel modules that support them.
- **Apache Webserver Config.** Apache is the default Web server for TurboLinux. The diagnostics allow the administrator to confirm that the server is running, as well as the current number of connections and the amount of disk space being used by `/var/log/httpd`.
- **DHCP Configuration.** This allows configuration of the DHCP Server.
- Note that this should be run from within X, as the configuration window requires 30 lines.
- **DNS (BIND 8) Configuration.** The TurboLinux nameserver is configured with this choice. The server can also be stopped and started here.
- **TurboFTP Config.** Configuration for the FTP server included (PROFTPD is the default FTP server). WU-FTPD is also included with TurboLinux 6, but configuration for it must be done manually.
- **IPX Config.** Allows you to activate IPX and set the IPX internal network and node number.
- **Appletalk Exports Config.** If you have Macintosh clients in your environment, the Appletalk exports config allows you to create an Appletalk share for the network.

- **NFS Exports Config.** NFS is the traditional protocol used in UNIX environments to share files.
- **PPP Config.** This configuration tool is for the client side dialup only. It does not set up a PPP server on this machine.
- **TCP/IP Routing Config.** Allows you to set the default route and net routes. It also has an option to enable IP forwarding.

### 3.3 Turboprintcfg

This configuration window is identical to the print configuration window you saw during the installation of TurboLinux. We will now discuss printer configuration in more detail.

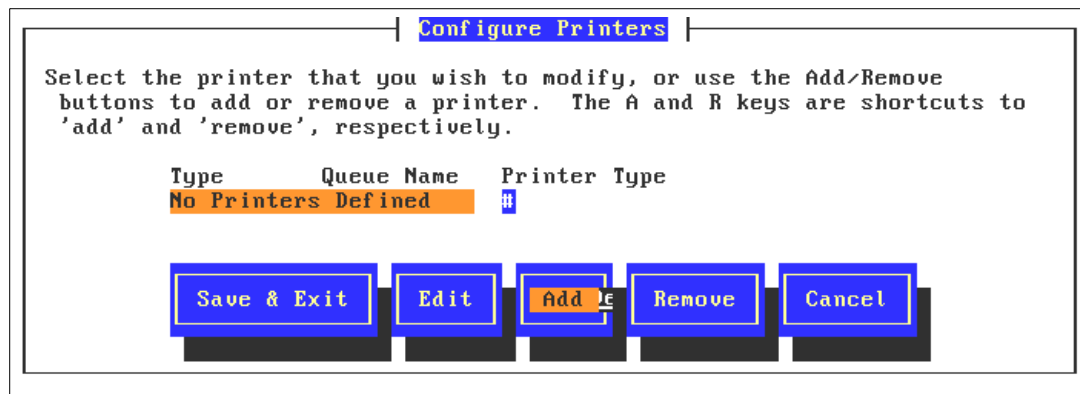


Figure 55. Configure Printers window

By selecting **Add** on the main window, you will be able to configure printers that are either local to this server, attached to another server on the network, or attached to the network directly.

### 3.3.1 Configuring locally attached printers

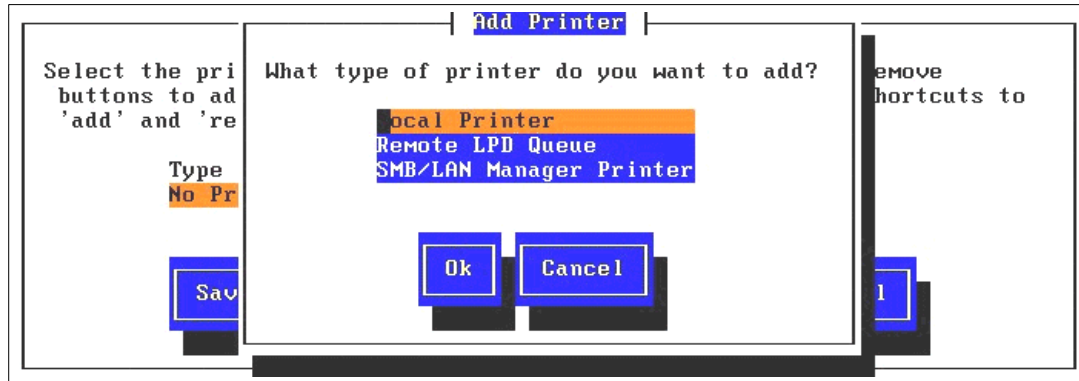


Figure 56. Add Printer window

The simplest configuration to be added is a locally attached printer. In the next figure you will see the menu displayed when **Local Printer** is selected.

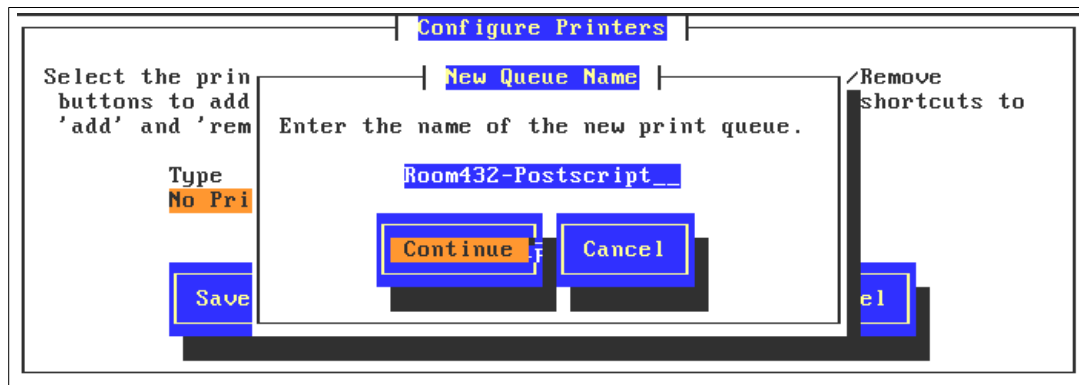


Figure 57. New Queue Name window

Above we have named the printer Room432-Postscript and selected **Continue**.

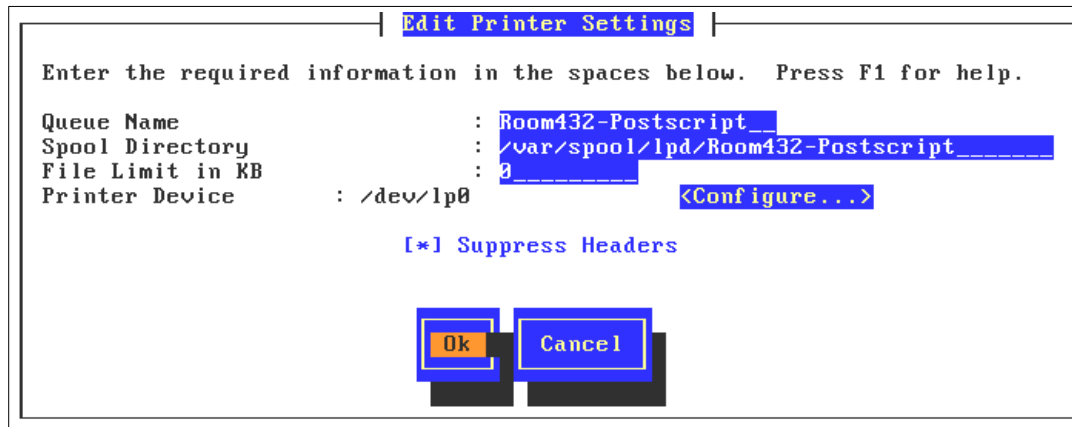


Figure 58. Edit Printer Settings window

The following information is available here:

- **Queue Name:** This will be the name users will reference when sending print jobs to this printer, whether the user is local or remote.
- **Spool Directory:** By default, Linux spools to /var/spool/lpd/[queue name]/. This can be changed if the print jobs being sent to this printer are larger than is available in the /var filesystem.
- **File Limit in KB:** This can be used to prevent large jobs from being sent to a particular printer.
- **Printer Device:** This is the output device used to access the printer.
- **Configure:** selecting this option leads to Figure 60.



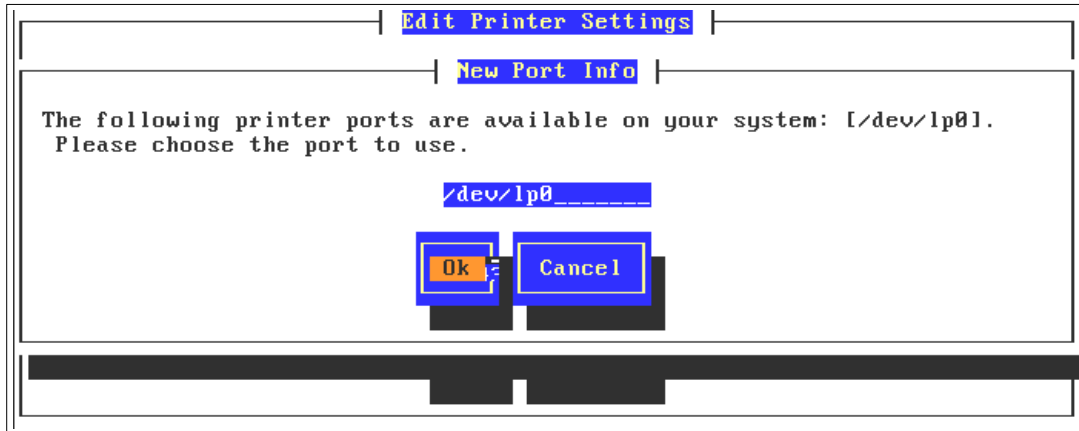


Figure 59. New Port Info window

Selecting Configure on the previous window brings up a dialog that allows you to select the output device being used to access the printer. Linux uses the convention /dev/lp0, /dev/lp1, etc. to signify what is called LPT1, LPT2, etc in Microsoft Windows. Output to serial printers is /dev/cua0, /dev/cua1, etc., to signify COM1, COM2, etc. Support for USB printers will be available in the near future when the 2.4 kernel is released. After selecting **OK**, you will return to the main window.

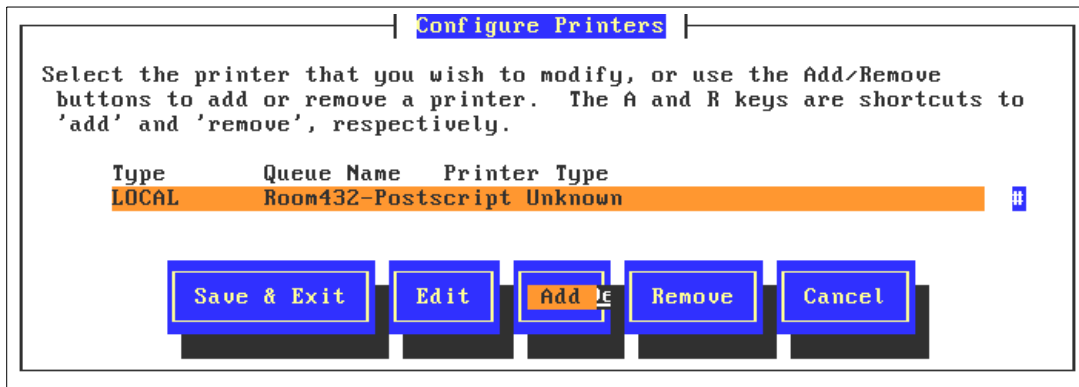


Figure 60. Configure Printers window

### 3.3.2 Configuring remote printers over TCP/IP

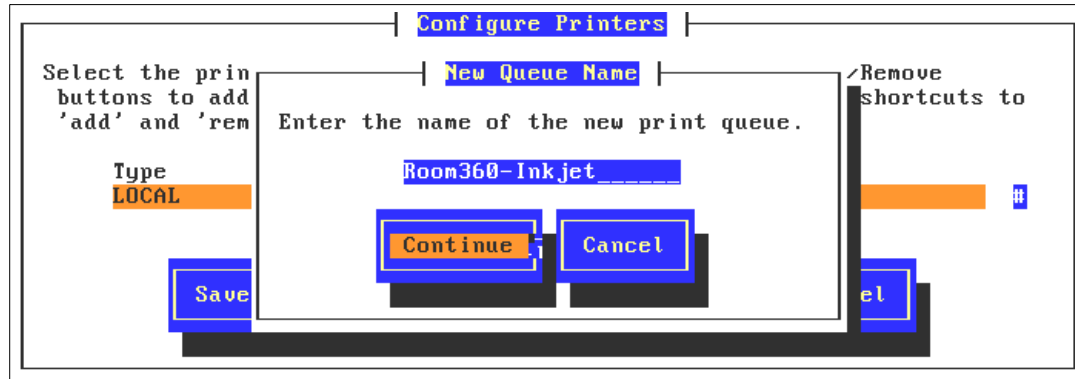


Figure 61. New Queue Name window

The process of naming a remote queue is identical to the process used for local printers. Above we have selected **Add**, then **Remote LPD Printer** and named this print queue Room360-Inkjet.

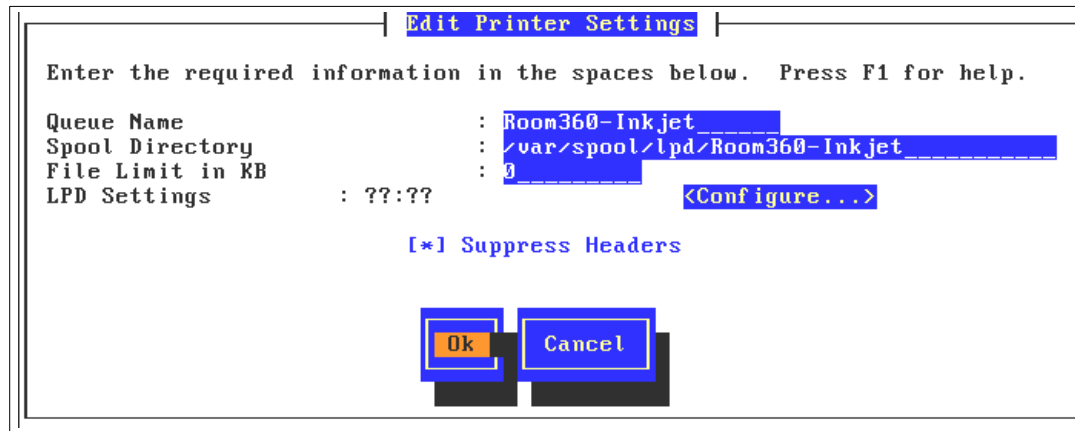


Figure 62. Edit Printer Settings window

The **Edit Printer Settings** page is also the same. Notice that **LPD Settings** is currently in the format `???:??`. It is actually `HOSTNAME:/QUEUE`. That information is added by selecting **Configure** here.

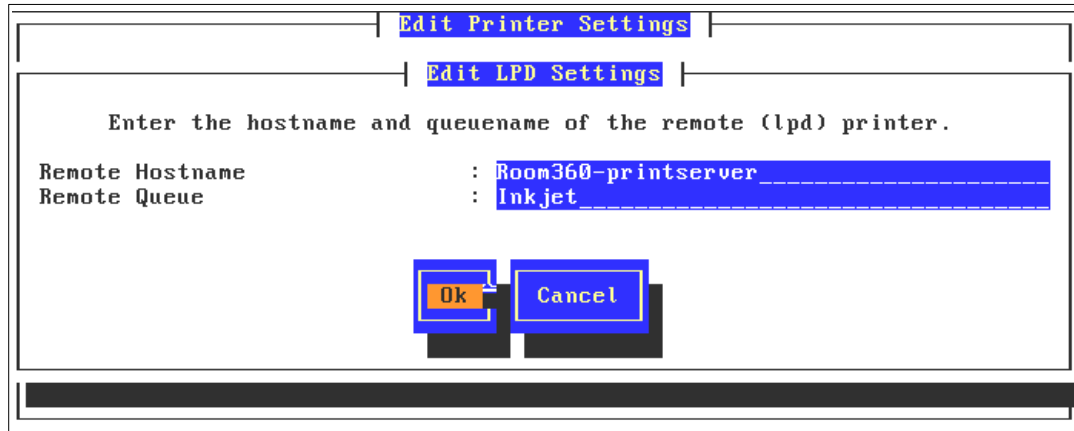


Figure 63. Edit LPD Settings window

Selecting Configure brings us to the Edit LPD Settings window (Figure 63), on which you must specify the host name and queue of the printserver you will be accessing. For LAN-attached printers that support LPD, this window will have the host name and queue defined by the printer.

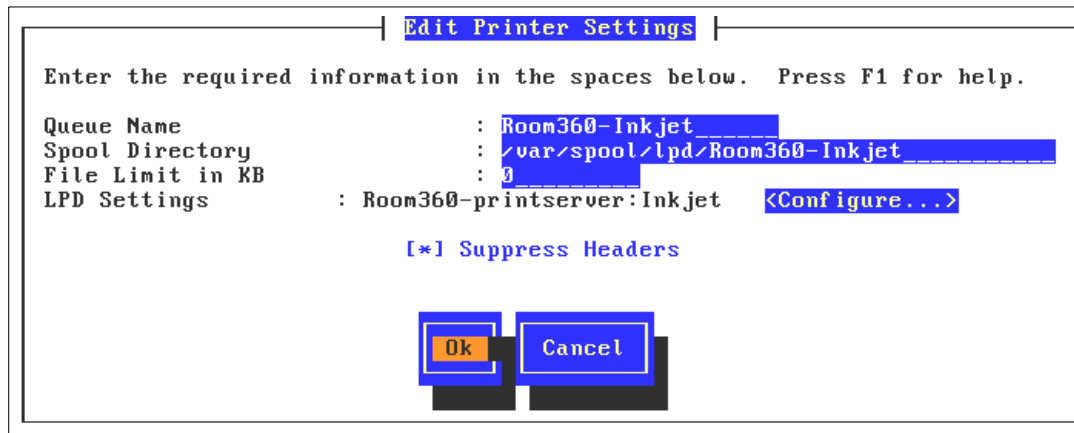


Figure 64. Edit Printer Settings window

Here you can see that LPD Settings fields now fits the pattern HOSTNAME:/QUEUE.

### 3.3.3 Adding NetBIOS based remote printers

Microsoft Windows and IBM OS/2 default to sharing printers over a an SMB Server Messaging Block (SMB) protocol commonly referred to as NetBIOS. If you have print shares that are accessed by SMB, you can configure Linux to act as a gateway to those print shares.

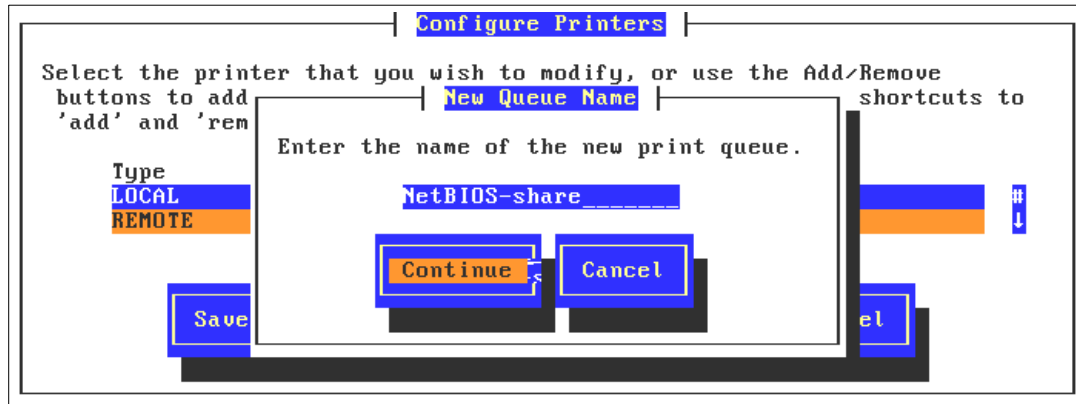


Figure 65. New Queue Name window

Above we have selected **Add** and then **SMB/LAN Manager Printer**. We have named this printer NetBIOS-share for clarity.

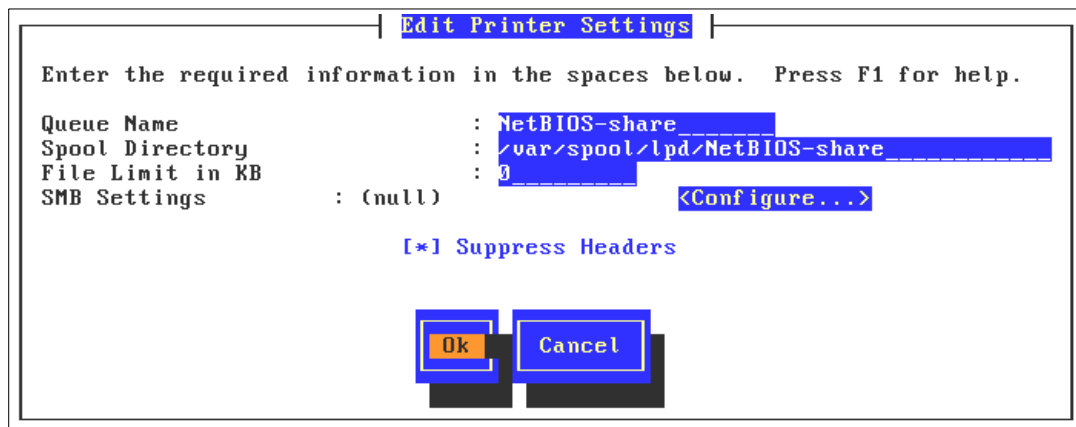


Figure 66. Edit Printer Settings window

The window in Figure 66 is identical to the window we saw in the LPD printer configuration, with the exception of the SMB Settings field. That information can be completed by selecting **Configure**.

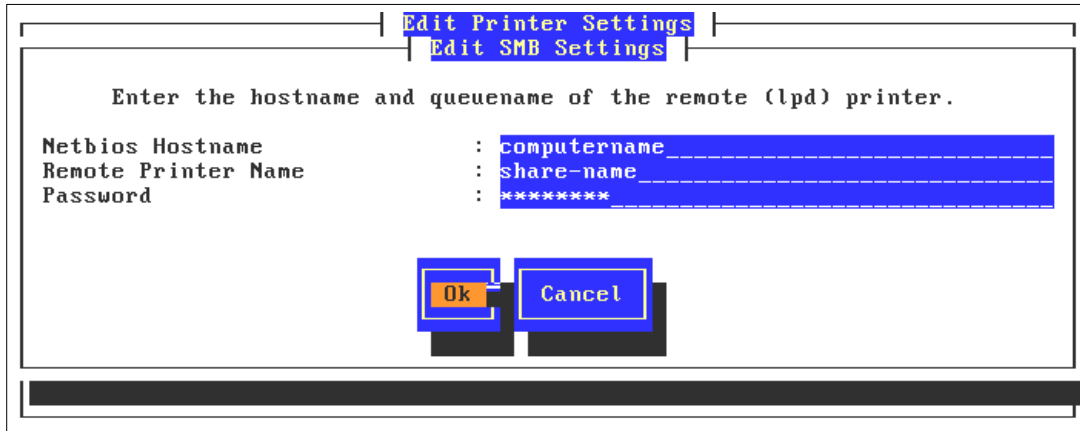


Figure 67. Edit SMB Settings window

In Figure 67 we have specified the printer settings as if it were connected to a Microsoft Windows server. NetBIOS Hostname is the Windows Computername, the Remote Printer Name is the share name in Windows, and the Password applies if the printer is not open to everyone. Selecting **OK** then **OK** again adds the SMB printer to Linux.

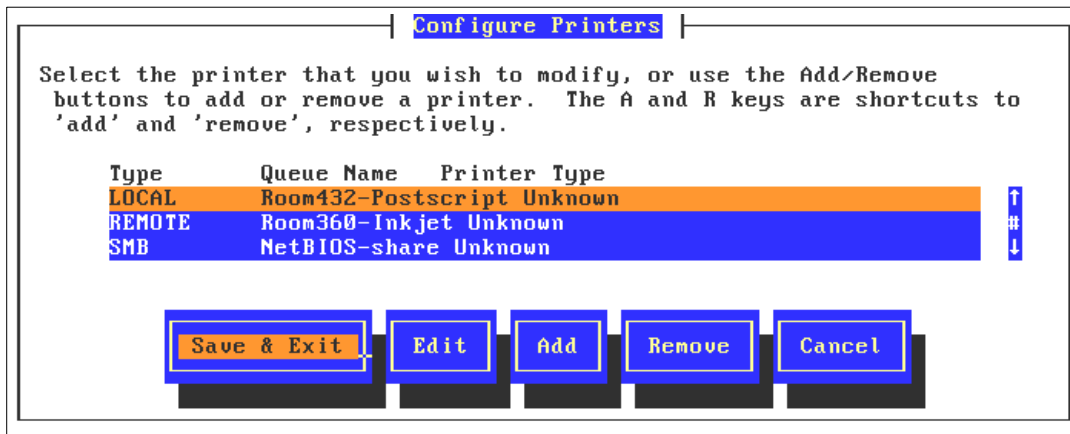


Figure 68. Configure Printers window

Selecting **Save & Exit** saves the configuration information to /etc/printcap.

---

### 3.4 Adding and removing software packages

TurboLinux uses the RPM (RedHat Package Manager) system to manage software packages. RPM uses a database to store information about the packages installed, the files that a package installs, and other relevant information needed for package management. Although several books have been written to explain all the complexity and flexibility available with RPM, we will discuss it simply as a means to easily install and remove programs.

#### 3.4.1 Adding additional packages from the CD-ROM with Turbopkg

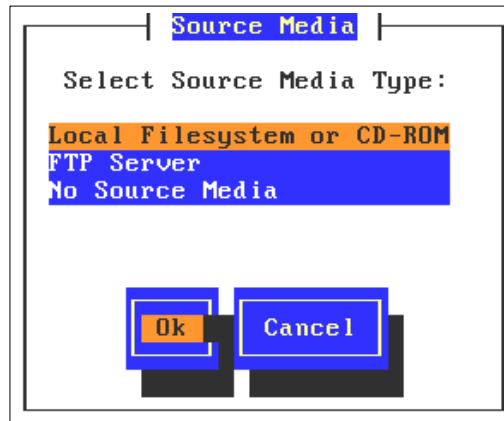


Figure 69. Source Media window

TurboLinux provides a configuration tool for the RPM system called `turbopkg`. Typing the command `turbopkg` opens the window you see in Figure 69. Here you have three options for the source of the RPMs you would like to upgrade or install. We will start by choosing **Local Filesystem or CD-ROM**.

After making the Local Filesystem or CD-ROM selection, you are presented the options of selecting User Base Path or Select Individually. Note the following information:

- **User Base Path.** This is the option you will almost always use.
- **Select Individually.** This option allows you to point to stored comp files, RPMS, and RPM header lists that do not reside in the same directory. If you choose this option, you should be aware that during a local install of packages, Turbopkg looks for three different pieces:
  - a. The comps file defines the categories (for example, “Editors,” “Basic Mail Services,” etc.) Turbopkg uses to organize all the available

packages. If the comps file does not exist, TurboLinux presents all the packages available in one long list.

- b. RPMS are files that end with the extension .RPM. They contain the files to be copied, scripts to be run during or after the install, and a list other packages that are prerequisites (called dependencies).
- c. RPM header files (hdlist) contain more detailed information about the RPM being installed.

In this case, we will choose **User Base Path** and proceed. This displays the window shown in Figure 70.

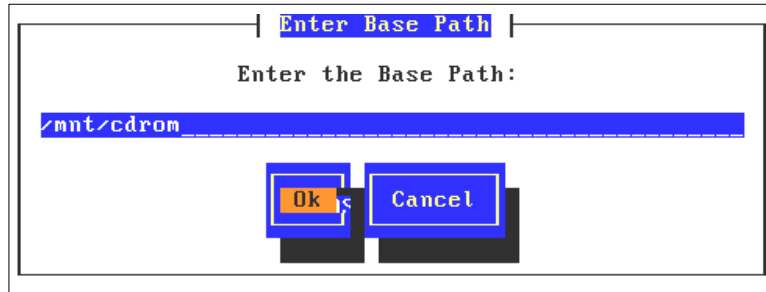


Figure 70. Enter Base Path window

Turbopkg defaults to look at your CD-ROM, but this can be changed to point anywhere. In this case we will insert the Companion CD and proceed.

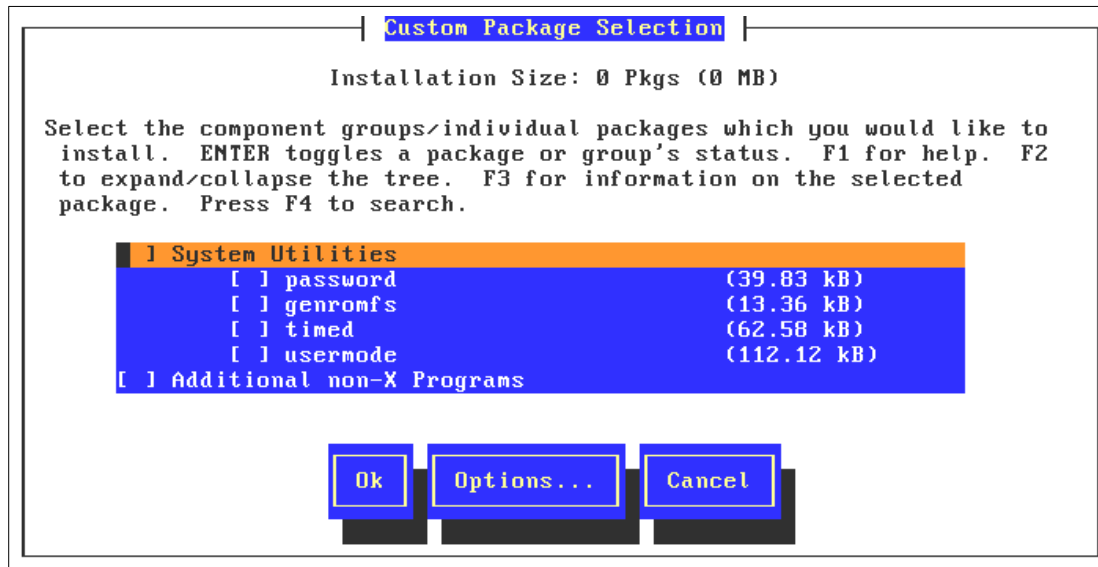


Figure 71. Custom Package Selection window

Because we selected **OK** on the previous window, turbopkg now reads the Companion CD and presents us with a list of packages to install.

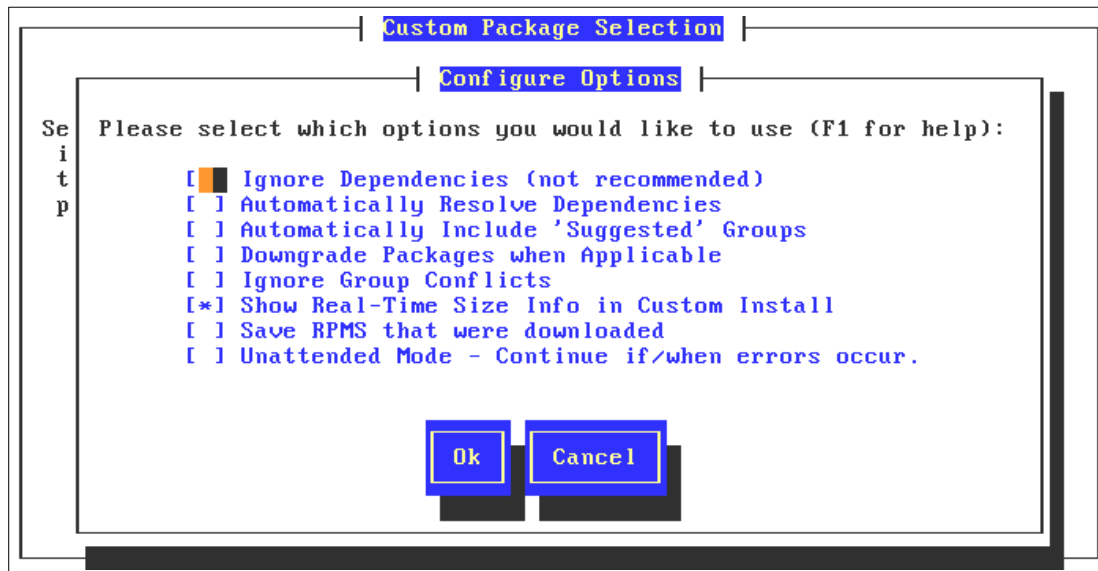


Figure 72. Configure Options window



Selecting **Options** on Figure 71 allows advanced Linux administrators to override the defaults of turbopkg. Novice and Intermediate Linux users should not change the defaults.

### 3.4.2 Adding packages via FTP with Turbopkg

Just as TurboLinux has the ability to install from an FTP server, Turbopkg has the ability to add new or updated packages from your own Intranet server, or publicly available Internet servers.

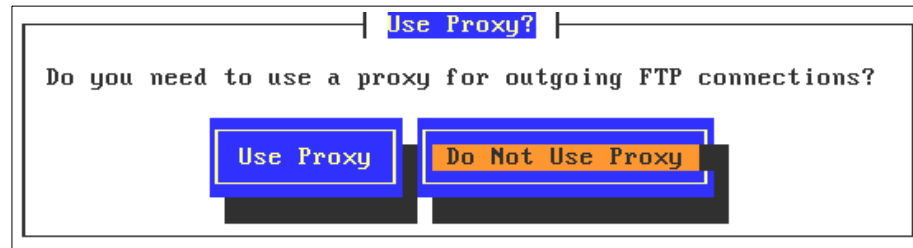


Figure 73. Use Proxy window

First we see the same dialog that appeared during the install. If your machine must pass through an FTP proxy in order to get to the server you are accessing, you must indicate that in Figure 73.

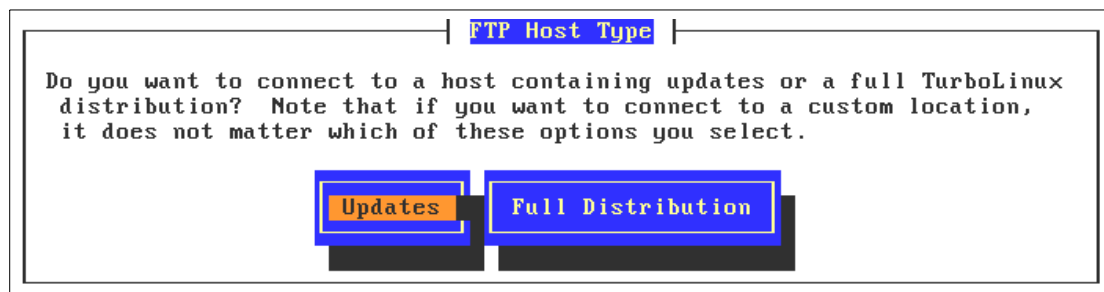


Figure 74. FTP Host Type window

We choose not to use a proxy, and since we have already completed our install, we will choose **Updates** on the window shown in Figure 74. As the text in the dialog box indicates, the option to define a custom server (for example, one within your own Intranet) will be present on the next window regardless of your choice here.

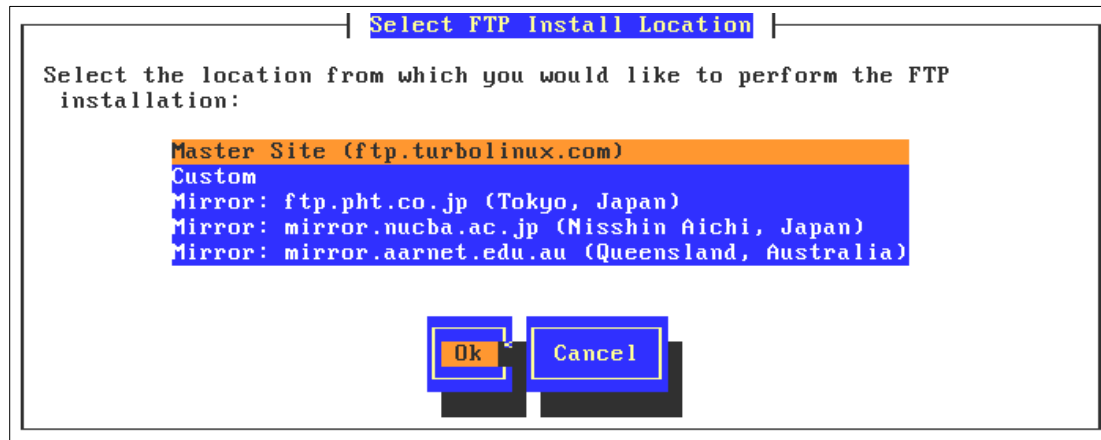


Figure 75. Select FTP Install Location window

This list of updates in Figure 75 includes both Internet sites and a Custom option, which allow you to create an update server inside your intranet. After selecting a source for the packages, the FTP method goes to the same windows we saw in the local installation.

### 3.4.3 Removing packages using Turbopkg

Choosing **No Source Media** in Figure 69 on page 56 takes you to the same Custom Package Selection window you have seen before. However, here your only option is to remove packages. To mark a package for removal, highlight the packages and press the key R. That will toggle the appearance of an R inside the brackets next to the package in question. Figure 76 demonstrates this.

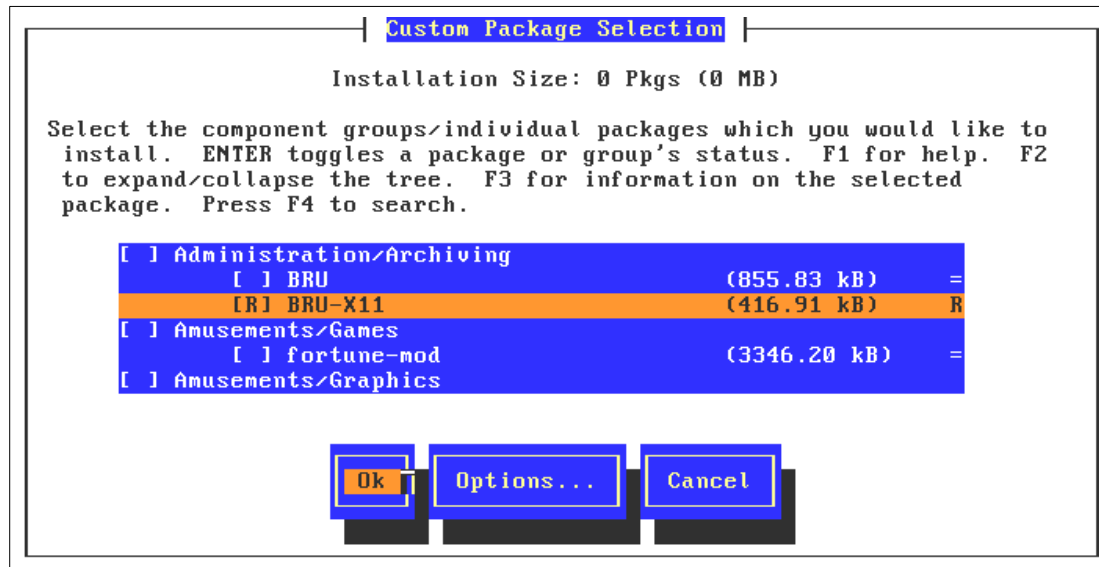


Figure 76. Custom Package Selection window

### 3.4.4 Package management using the RPM command

Package management can also be done directly from the command line. The command line is often used to build scripts to do package management. Table 3 table below shows some frequently used commands.

Table 3. Basic RPM commands

Command	Description
<code>rpm -q &lt;package&gt;</code>	Query RPM database. If package is installed, display version and build number of installed package.
<code>rpm -qi &lt;package&gt;</code>	Obtain some more information about an installed package.
<code>rpm -qa</code>	List all installed packages.
<code>rpm -qf &lt;filename&gt;</code>	Determine the (installed) package that <filename> belongs to.
<code>rpm -Uhv &lt;package.rpm&gt;</code>	Update/Install the file <package.rpm> showing a progress bar.
<code>rpm -F -v ./*.rpm</code>	Update (refresh) all currently installed packages using the RPM files in the current directory.

Command	Description
<code>rpm -e &lt;package&gt;</code>	Erase or remove a package

More information about RPM can be found in the manual page (`man rpm`), the RPM HOWTO or the RPM Web site at <http://www.rpm.org>. You can also display a short overview by running `rpm --help`.

## 3.5 User and group administration

Linux is a multi-user operating system. To differentiate between the various users, each user has to log in with a unique user name and password. Each user belongs to a primary user group, but he can also be a member of additional other groups as well (up to 16 groups). Each user name is assigned a numeric identifier called a UID (User Identifier) which is unique throughout the system. Groups also have a numeric identifier, called a GID (Group Identifier), that is unique to the system as well. For environments where security is handled by individual machines, this can be important, since some services rely on the UID and GID to determine permissions. Those issues can be resolved by using NIS (Network Information Service) or LDAP (Lightweight Directory Access Protocol), but for now we will put those questions aside and look at the menu system TurboLinux provides for managing groups and users on individual machines.

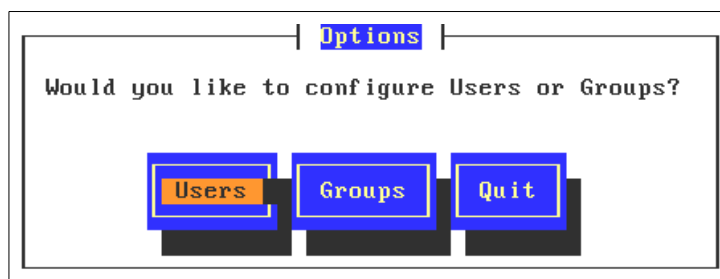


Figure 77. Options window

Issuing the command `turbousercfg` creates the window shown in Figure 77. On it you have the ability to manage both **Users** and **Groups**.

### 3.5.1 Adding new groups

You should consider adding groups before adding users. Sometimes there are concerns about restricting access to some parts of the user file system. You can do this by creating separate user groups to control access to various

files and file systems. Also if you are going to be creating a system with many users, you should consider creating separate groups divided by what they are doing on the system. You can create an admin group for admins, a db2user group for DB2 users, and so forth. Linux allows you to control access to both files and directories by users, groups, and everyone on the system.

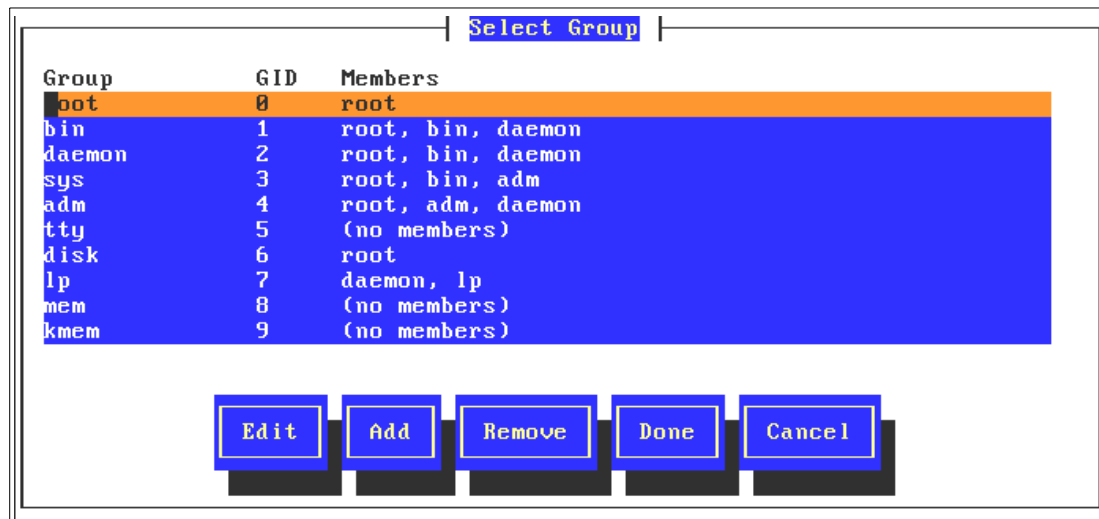


Figure 78. Select Group window

Selecting **Groups** creates the menu you see Figure 78, which is a partial listing of the default groups created by a complete install of TurboLinux. This is a very long list, and if you scroll through you will get some feeling for the different groups that can be created. The list of groups is stored in the file `/etc/group`.

The information that is displayed is:

- **Group.** This is the unique name of the group.
- **GID.** The system knows a user and group only by a number. In this case the group is known by the group ID. The group ID must be unique.
- **Members.** This is a list of the members of the group.

You should also notice that you have the option to select:

- **Edit** a current group listing. This allows you to change characteristics of groups that are on the list.
- **Add** a new group.
- **Remove** a current group.

- **Done.** This allows you to save any changes you have made.
- **Cancel.** This allows you to back out of any changes that have not been saved.

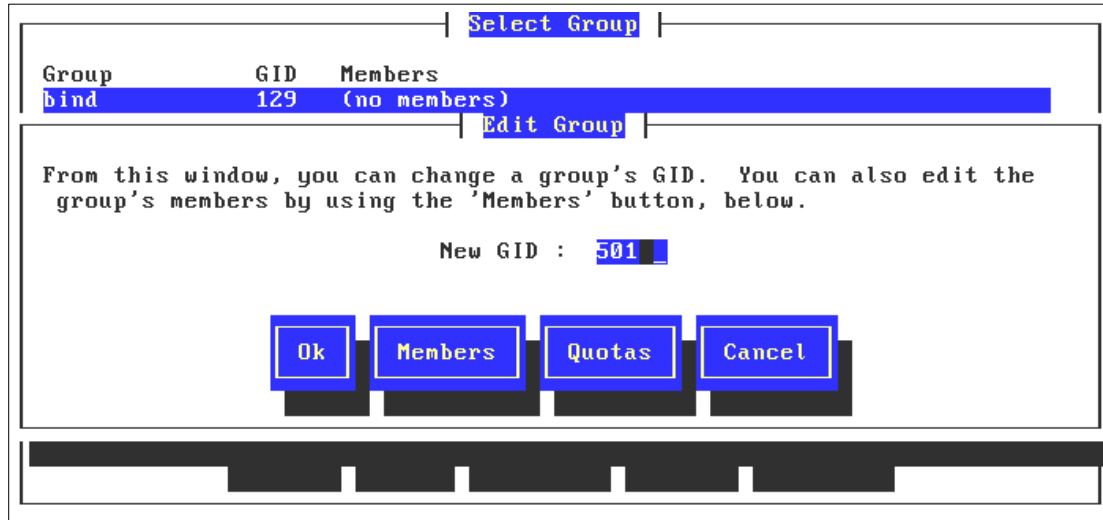


Figure 79. Select Group window

If you decide to Add a group you will first see a window that asks you to name the group, and will then see the window in Figure 79. Here you can change the GID for the group, as well as add and remove Members (users) from the group. If you have disk Quotas enabled, you can set quotas for the group as well.

### 3.5.2 Adding new users

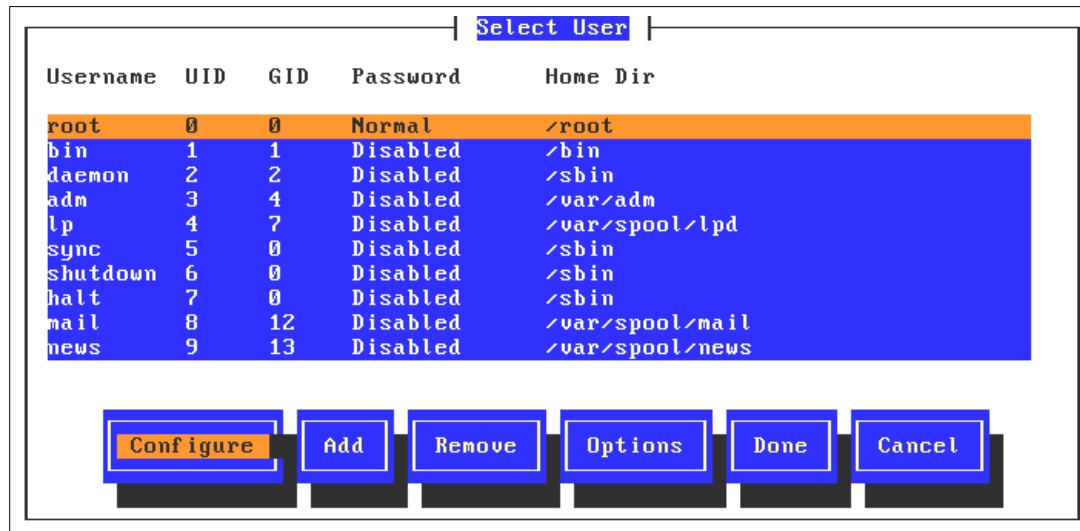


Figure 80. Select User window

Above you see a human readable form of the user information stored in the file `/etc/passwd`.

The information in Figure 80 is organized in columns by:

- **Username.** This is the unique name that a user types at the login prompt. It can also be called the login name, ID, user login, user, or user account.
- **UID.** This is the number that the system uses to identify each user. Each user on a particular system has a unique UID.
- **GID.** This is the unique number assigned to a group. Every user has a default group. In TurboLinux the default GID is 100.
- **Password.** This does not contain the password but tells you the information about its state. The column is either:
  - **Shadowed.** Which means it is using the shadow password file to store the password instead of `/etc/passwd`.
  - **Disabled.** The account is disabled, or is a service. Services are assigned a UID but no password, as they do not log in in the same way human users do.
  - **Home Dir.** This is the user's home directory. It is the first place a user goes when logging in. It contains files and programs that are owned and used by that user.

In addition there are several choices for adding or configuring users that are given to you in the boxes along the bottom. They are:

- **Configure.** This allows you to change the characteristics that were set up when the ID was created.
- **Add.** Allows you to add users, which will be discussed later.
- **Remove.** Allows you to remove the user. You can optionally also remove the home directory.
- **Options.** Allows you to configure options for the user such as file system space quotas if they have been enabled.
- **Done.** Will allow you to quit and will save any unsaved information.
- **Cancel.** Will allow you to quit and *not* save any information.

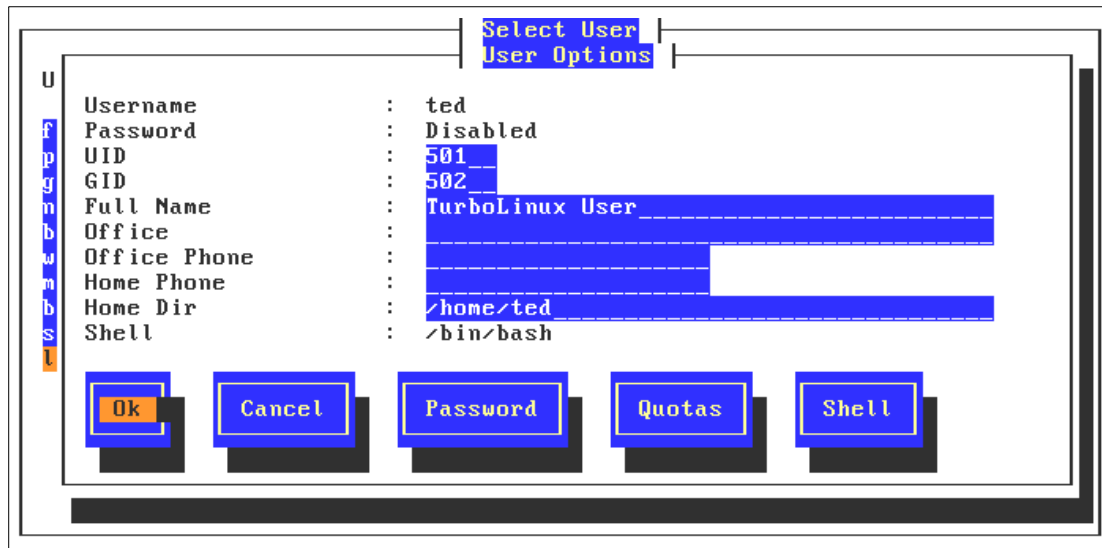


Figure 81. User Options window

To add a user select the **Add** option at the bottom of the window. You will then see a window asking you to give the name of the new user, and specify the home directory for this user. We recommend you accept the default location of `/home/<username>/`.

You will then come to the window Figure 81. The information that can be added for each user includes:

- UID
- GID



- Full Name
- Office
- Office Phone
- Home Phone
- Home Dir

The buttons at the bottom of the window allow you to modify the following:

- **Passwords** must be at least 4 characters long.
- Disk **Quotas** can be defined per users if you have file systems on this server that have quotas enabled.
- The default **Shell**, bash, will suffice for almost all users. On occasion, some users may prefer a different shell. You can set the user's default shell here.

Here is a brief description of the included shells:

- **/bin/bash**. This is the Bourne Again Shell, which is an extension to the Bourne Shell. This is the most popular shell for Linux.
- **/bin/sh**. This is the standard Bourne Shell that has been around since almost the beginning of UNIX.
- **/bin/ash**. This is another version of the Bourne Shell.
- **/bin/bsh**. This is the same as /bin/ash to which it is linked.
- **/bin/ksh**. This is the standard Korn Shell that is the most popular shell for UNIX administration.
- **/bin/tcsh**. This is a public domain extension of the C Shell.
- **/bin/csh**. This is the standard C Shell that was originated by the University of California at Berkeley.
- **/bin/zsh**. This is another extension of the Bourne Shell.

Your choice of shells is strictly a matter of preference, but generally UNIX admins prefer Bourne or Korn Shell programs, whereas programmers tend to prefer C Shell-based programs.

### 3.6 Administering file systems and the boot record

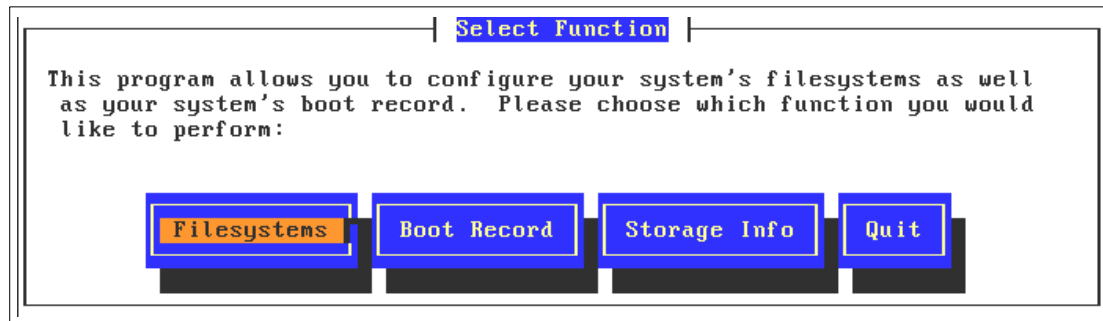


Figure 82. Select Function window

Running the command `turbofscfg` generates the menu you see Figure 82. From here the administrator can manage almost all issues that touch the DASD attached to this system. To view all the DASD Linux sees on your server, select the option **Storage Info**. That will display a window similar to Figure 83.

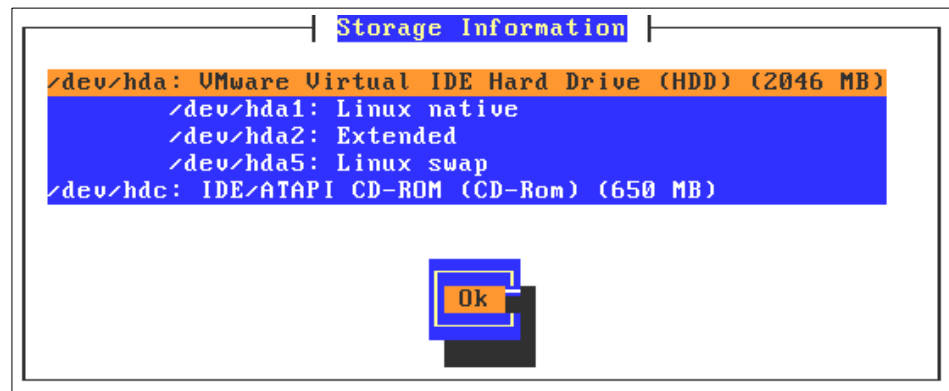


Figure 83. Storage Information window

Notice that in the example above, the mounted CD-ROM appeared as well.

### 3.6.1 Managing file systems

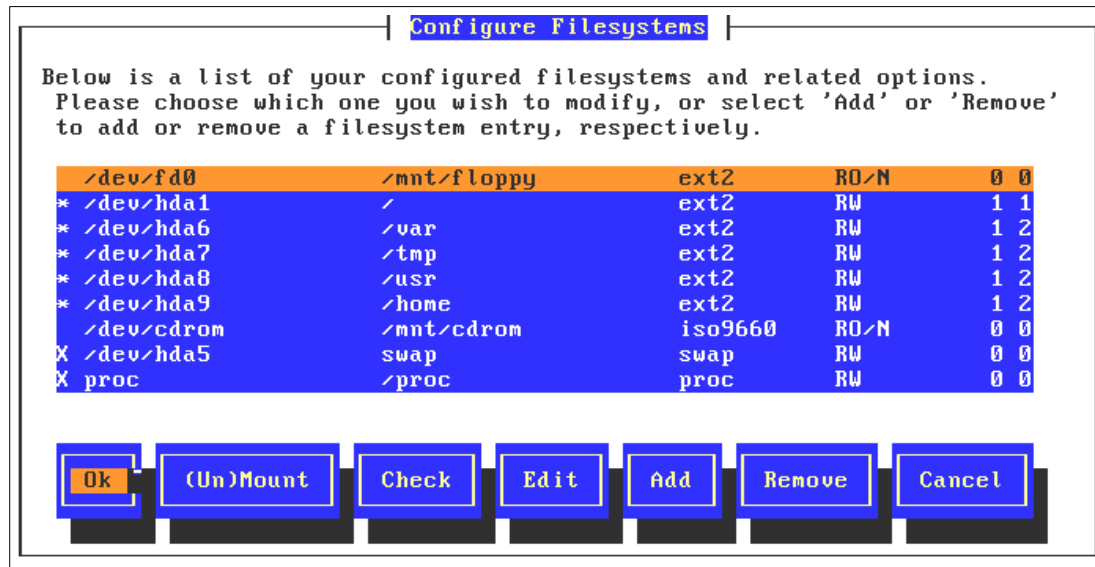


Figure 84. Configure Filesystems window

Above you see the main window created when the Filesystems option is selected from the main turboscfg window. The columns of information displayed (which is being read from /etc/fstab) are from left to right:

- **Mount status** is indicated by an \* for mounted file systems and a blank space for unmounted file systems. In the figure above the floppy and CD-ROM are not mounted.
- **Device**, which is where raw device exists in the file system. This is how the kernel sees the device.
- **Mount Point**. Linux uses a concept of mount points to give the user access to DASD devices. This allows you as the user to map an arbitrary name to a fixed name that is known to the kernel. So, for example, if you wanted the TurboLinux 6 CD-ROM to be accessible to FTP users, you could mount the device /dev/cdrom to /home/ftp/pub/TurboLinux6. This column lists the mount points currently assigned to the filesystem table.
- **Default filesystem Type**. When a device mounts, it mounts with the file-system in this column. Notice that the floppy drive defaults to ext2, which is actually rather rare. To mount FAT-formatted floppies, you will need to change the file system type to MSDOS. That example is given at step 3. on page 70 when we discuss the **Edit** option.

- **Permissions.** Whether the file system is readable and/or writable.
- **FSCCK options.** These two numbers determine whether the file system should be dumped if the system crashes (0 for no, 1 for yes), and the priority FSCCK should use when running (0 indicates the file system should not be checked, 1 indicates that FSCCK should check this file system first, 2 indicates that FSCCK should check after all the file systems numbered 1 have been completed).

In addition to the columns of information, there are several options at the bottom of the window. We now address them:

1. **(Un)Mount** toggles the mount status of file systems. With this option you can mount filesystems that are currently mounted, and unmount nonessential mounted filesystems.
2. **Check** allows you to force FSCCK to run now. You can only run FSCCK against unmounted filesystems.
3. **Edit** allows you to change various attributes of the mount point. For example, the window below was generated by highlighting the first line in the table (the floppy drive) and choosing **Edit**. In the next few windows, we show you how to change the default file system to MSDOS.

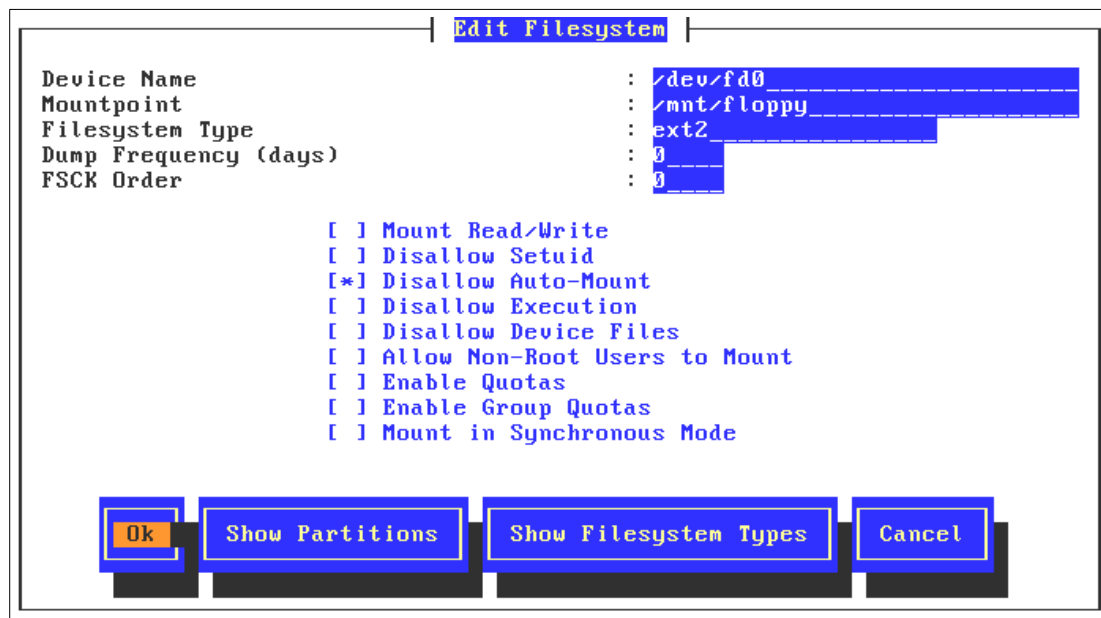


Figure 85. Edit Filesystem window

You can see in Figure 85 that the Filesystem Type is set to ext2. Selecting **Show Filesystem Types** allows you to view the other available options in the Figure 86.

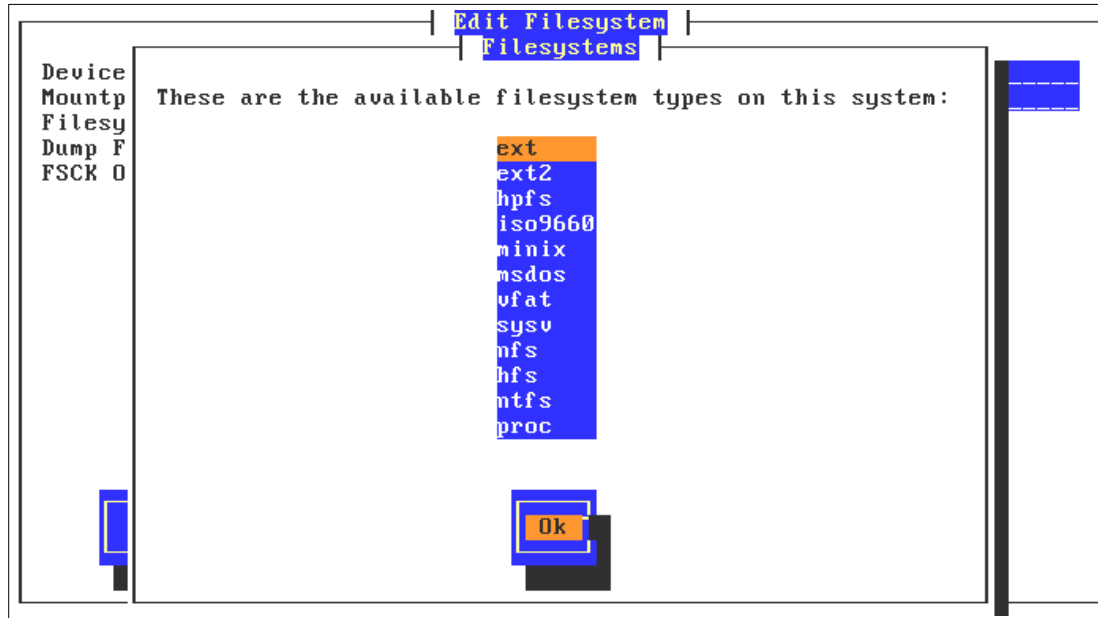


Figure 86. Filesystems window

The DOS filesystem FAT is referred to by Linux as MSDOS, and we can see from this list that it is available. Note that this list is a real-time listing of all kernel modules in the directory `/lib/modules/current/fs/`. If modules are removed or added to the directory, the list will change.

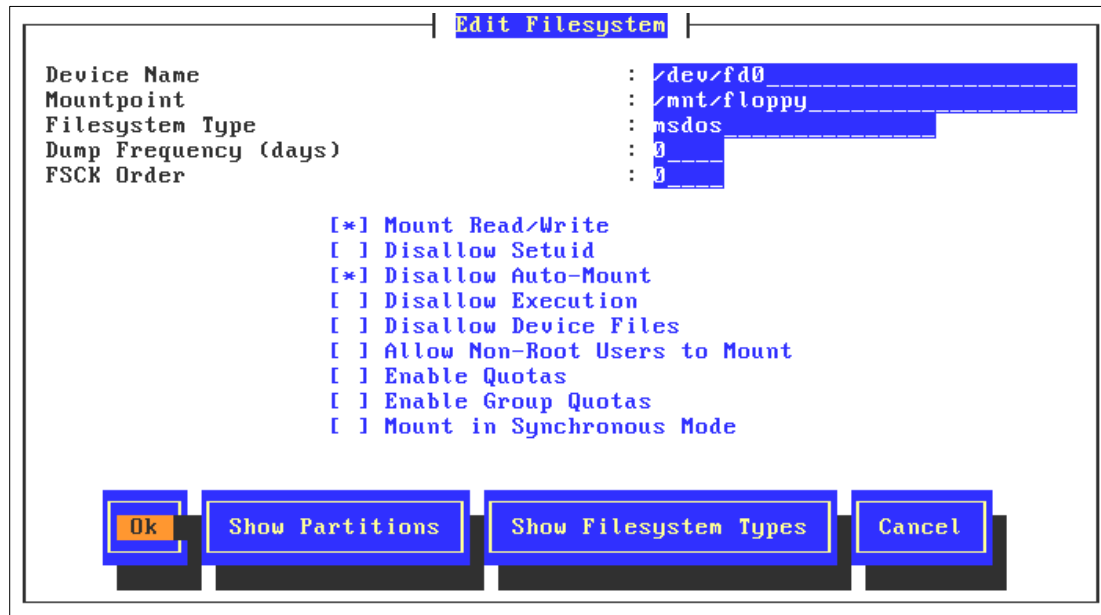


Figure 87. Edit Filesystem window

In Figure 87 we have changed the Filesystem Type to msdos, and selected the option to mount the floppy read-write, since a read-only floppy is not very useful. Clicking **OK** on this window saves our changes to /etc/fstab.

4. You are allowed to add local or remote file systems to this server. Selecting Add will give you the options shown in Figure 88:

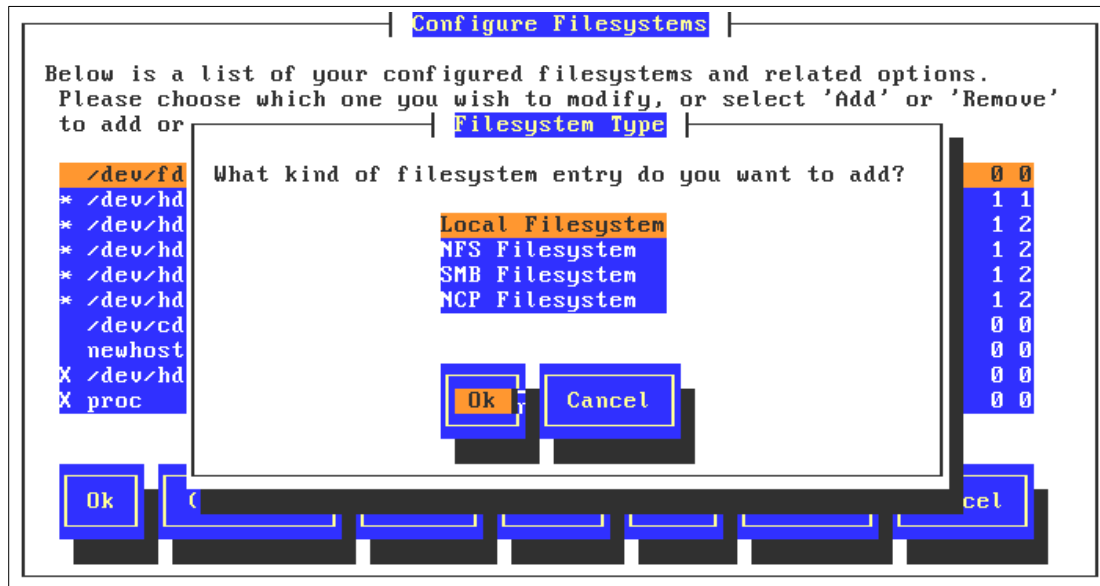


Figure 88. Configure Filesystems window

The four types of filesystems that can be added are:

- Local Filesystem.** This window is identical to the Edit Filesystem window shown in Figure 88. When adding a file system, the options Show Partitions, and Show Filesystem Types are quite helpful, as they allow you to view all the partitions and file systems known to the system.

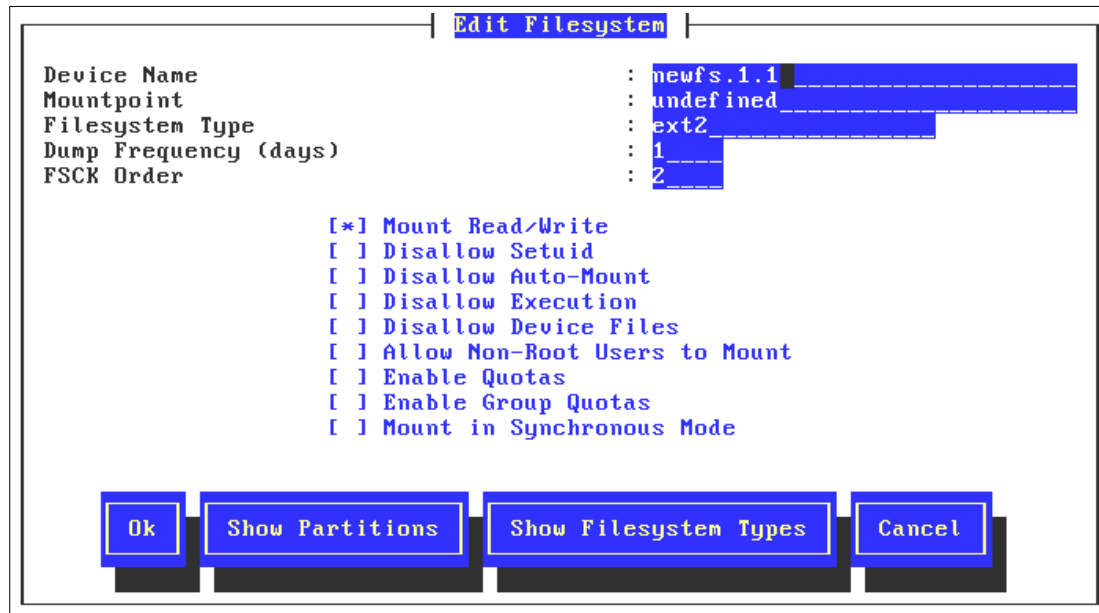


Figure 89. Edit Filesystem window

- b. **NFS Filesystem.** This options allows you to mount filesystems being exported by an NFS server. You can get a list of the directories being shared by a server by completing the first line, Hostname of IP Address and selecting the option **NFS Exports**. That will query the NFS server in question and send back a list of directories being exported (also called exports).



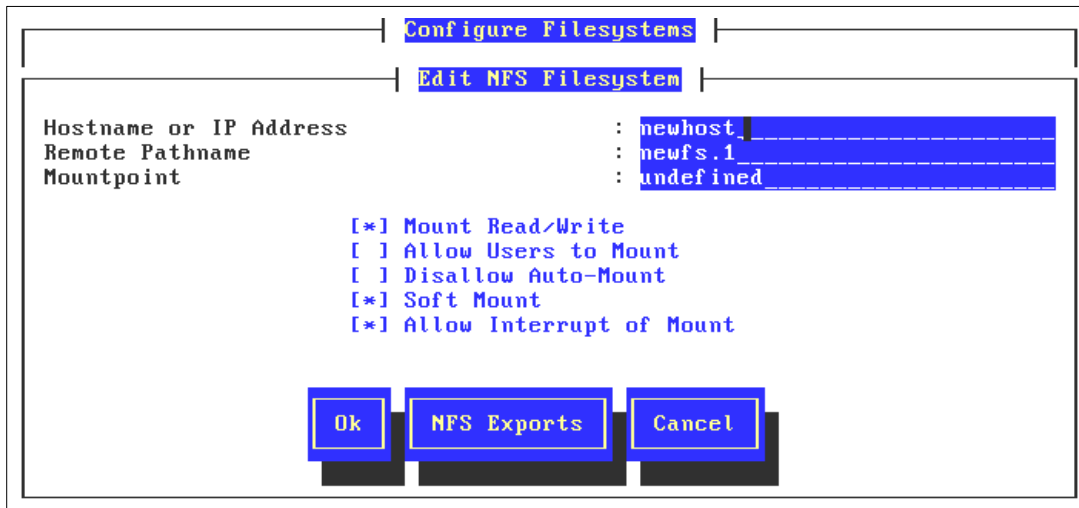


Figure 90. Edit NFS Filesystem window

- c. **SMB Filesystems** are the NetBIOS shares offered by Microsoft Windows and IBM OS/2 servers. Enter the server's NetBIOS Name and select the **SMB Shares** option shows the shares being offered by the SMB server.

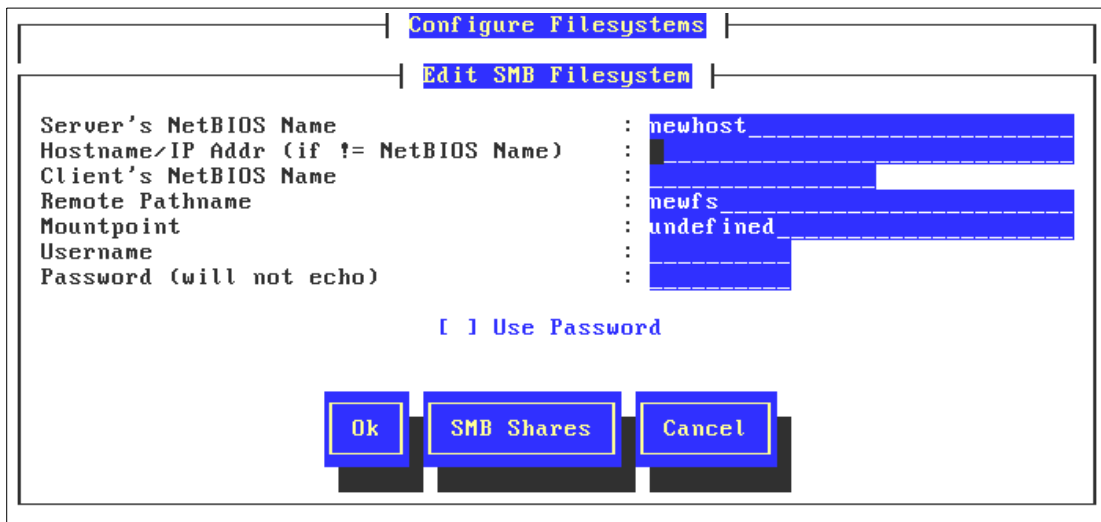


Figure 91. Edit SMB Filesystem window

- d. **NCP Filesystem** offered by Novell NetWare servers can be added by choosing the last option.

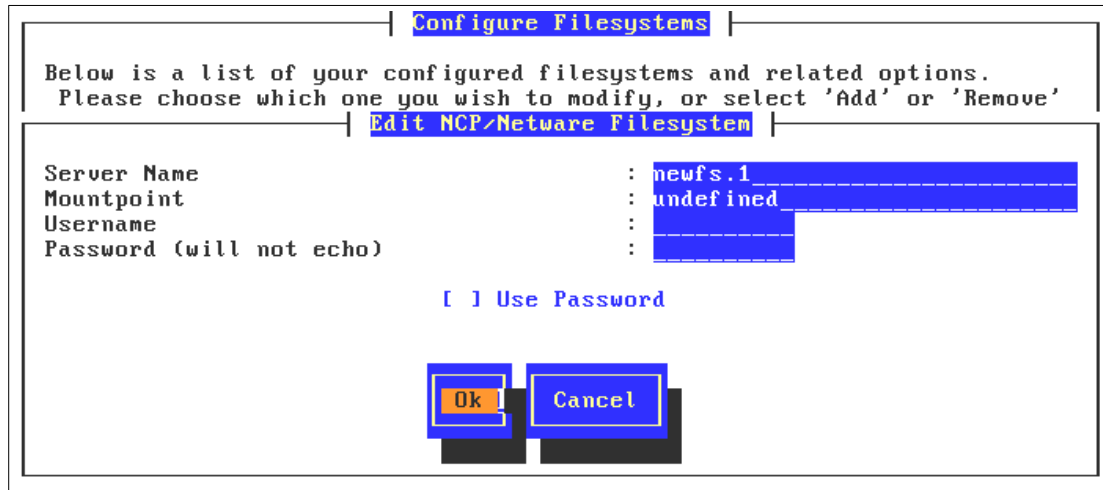


Figure 92. Edit NCP/NetWare Filesystem window

### 3.6.2 The Boot Record

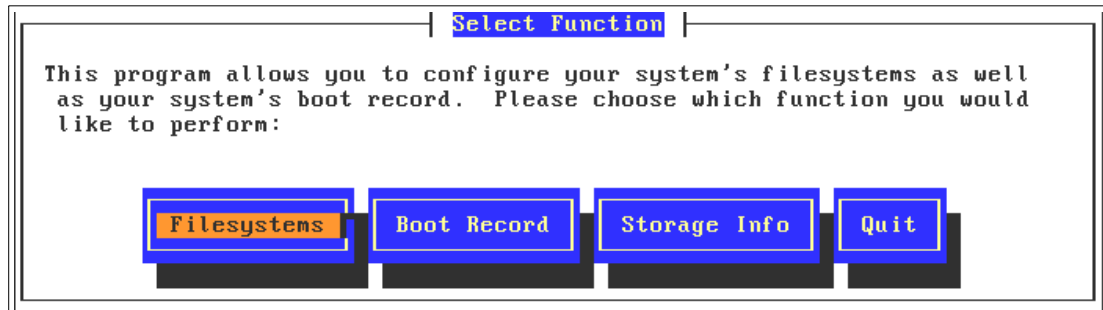


Figure 93. Select Function window

When Linux is installed, it is the responsibility of the program LILO to write the correct information to the Master Boot Record and Boot Record of the computer so Linux can boot. In Figure 93 you see the main window for `turbosfcfg` again. Choose **Boot Record** here and proceed.

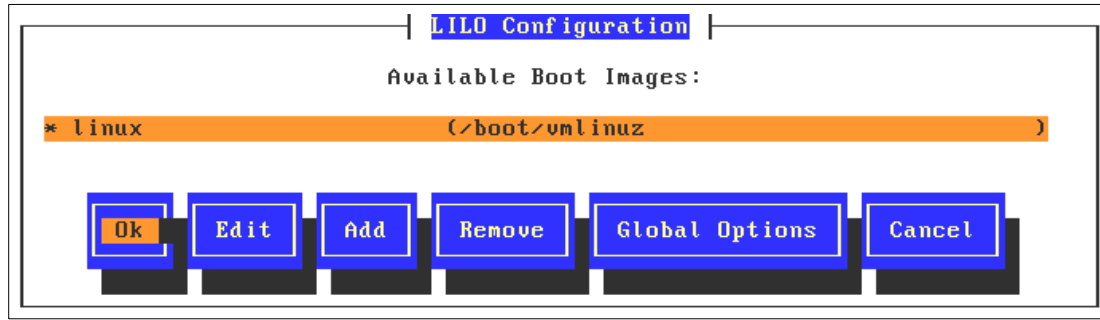


Figure 94. LILO Configuration window

Choosing Boot Record on the main turbofscfg window brings you to Figure 94. By default, TurboLinux creates a boot image labeled “linux” that boots the kernel /boot/vmlinuz. The \* on the left side denotes that this is the default image to boot if there are multiple images. The other options on this window are:

1. **Edit**, which allows you to change the configuration of an image. Below you see the details for the default “linux” configuration.

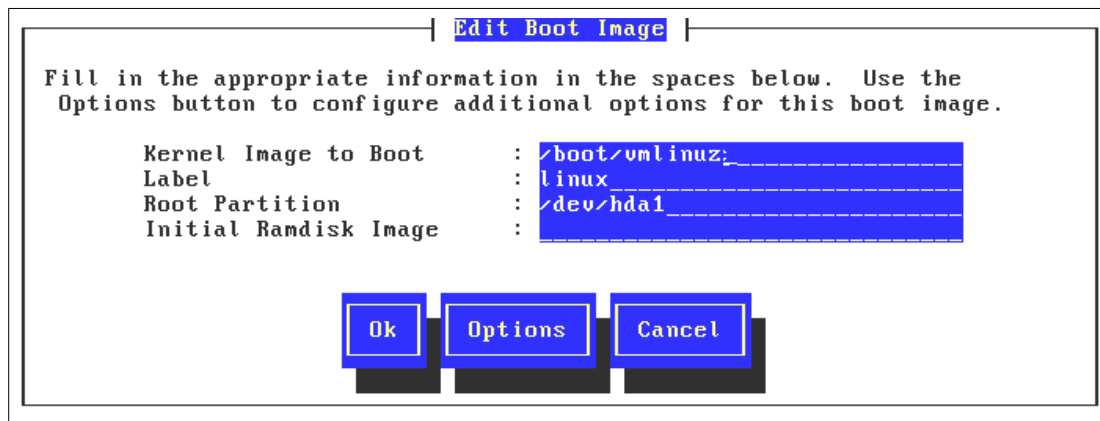


Figure 95. Edit Boot Image window

2. **Add**, which allows you to create a configuration for a different Linux kernel, or a different operating system on a different partition. When you choose Add, you will be asked if you would like to add a Linux or non-Linux boot image. Adding a Linux partition generates the same window you see in Figure 95. Choosing to create an image for a non-Linux operating system creates the following window:

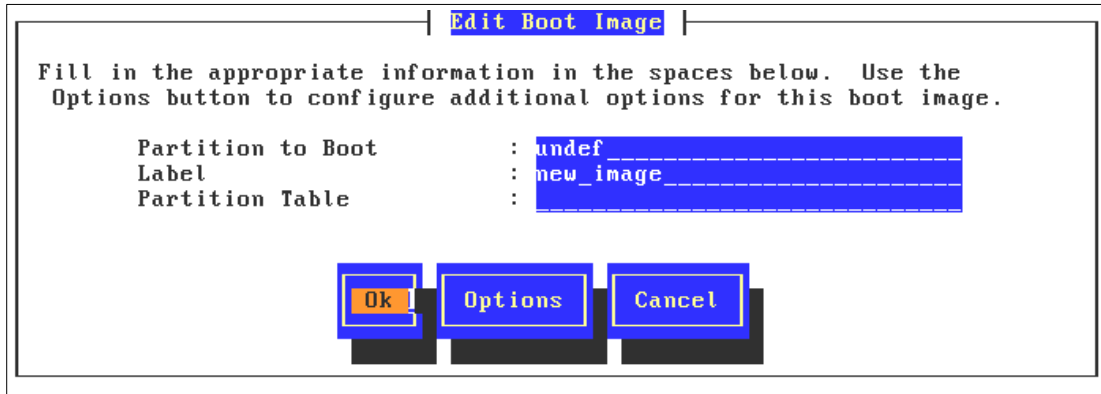


Figure 96. Edit Boot Image window

Here you define the partition you would like to boot. The partitions are defined as Linux sees them, so /dev/sda and /dev/sdb are the first and second SCSI drives, and /dev/hda and /dev/hdb are the first and second IDE drives.

3. **Remove** will erase the entry from /etc/fstab.
4. **Global Options** sets many other LILO options.

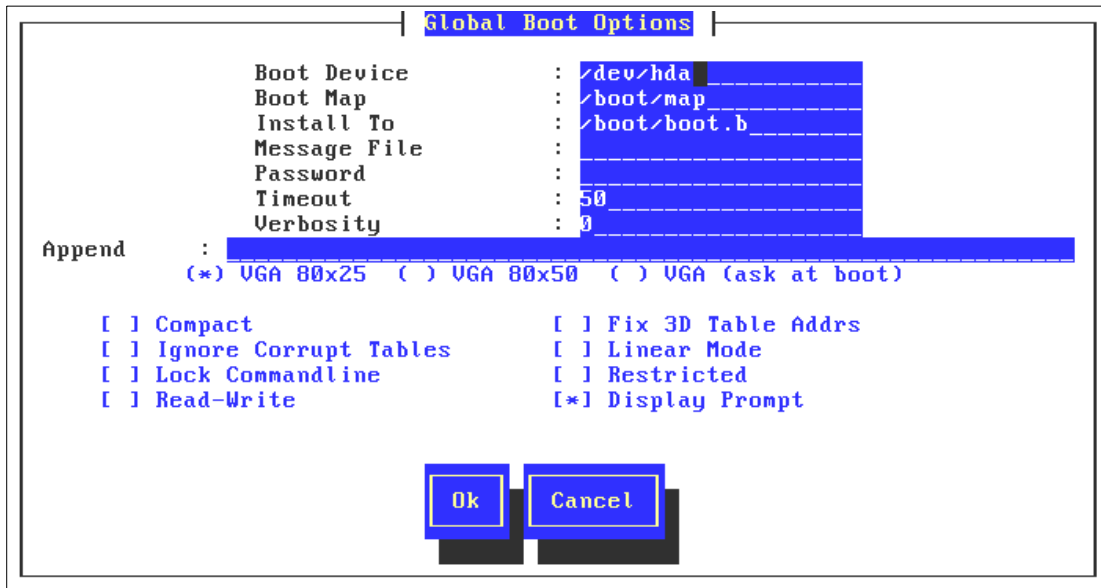


Figure 97. Global Boot Options window

Those options are:

- **Boot Device.** The hard drive on which LILO will write its information, and the choice of using the either Master Boot Record (in this case /dev/hda) or the boot record of a partition (for example, /dev/hda1). You can write LILO to a partition if you would like some other boot loader program to control the Master Boot Record.
- **Boot Map** and **Install To** are internal configurations of LILO, and should not be changed.
- **Message** allows you to insert a message that is seen when LILO starts.
- **Password** allows you to add a password to LILO.
- **Timeout** is time in seconds before the default boot image is executed.
- **Verbosity** defines the amount of information LILO gives to the administrator when it is writing to the drive. The scale is from a low of 0 to a high of 5, and the information is only shown if the LILO command is issued from a command line. You will not see any additional information if you use turbofscfg.
- **Append** allows you to add extra parameters to the kernel. Some configurations may need extra parameters to start certain devices.
- **VGA** resolution choices are 80x25 or 80x50, or you can require the choice to be made at boot time. Note that “ask at boot” offers a few other resolution choices.

The other options on this window are beyond the scope of this book. If you need more information on them you can read the LILO HOWTO at

<http://www.linuxdoc.org/HOWTO/mini/LILO.html>

---

### 3.7 Determining your hardware

There are several ways you can determine your hardware. These methods include:

- **Bootup messages.** The file /var/log/messages is a plain text file containing the bootup messages. The system will attempt to find hardware devices when you boot up. It may recognize the hardware devices and then attempt to use modules that are compiled in the kernel or modules that are loaded separately. Sometimes the system will recognize the hardware but will be unable to load the modules due to some hardware or setup inconsistencies or version dependencies.

- **dmesg**. This is a command that you can run anytime and will display many of the messages that you see on bootup.
- **Mail**. TurboLinux will mail you a copy of the configuration and bootup messages for every reboot. This can be more extensive than the messages from `dmesg`. To get access to these messages type `mail`.
- **turbohw**. The `turbohw` command will probe your TurboLinux system and will give you a listing of hardware that it finds. It also allows you to generate a text file with detailed information on the hardware installed in your machine, including currently used resources (for example, IRQ, IO ports).

---

### 3.8 Server Services

Linux uses a concept of runlevels (0-6) to help manage the operation of the system. On boot, the system reads the file `/etc/inittab` to determine which runlevel it should enter. It then reads the subdirectories under `/etc/rc.d` that conform to that runlevel. The runlevel directories are:

- `/etc/rc.d/rc0.d`
- `/etc/rc.d/rc1.d`
- `/etc/rc.d/rc2.d`
- `/etc/rc.d/rc3.d`
- `/etc/rc.d/rc4.d`
- `/etc/rc.d/rc5.d`
- `/etc/rc.d/rc6.d`

For example, if the runlevel is set to 3, which is the level TurboLinux sets if you ask for a text-mode login during the install, the system reads the directory `/etc/rc.d/rc3.d/` and starts or stops services based on the scripts in that directory.

The directory contains a number of symbolic links that point to scripts created to start, stop, and restart all the services provided by Linux. Looking at the directory, you will notice the files look similar. Below are a few examples from `/etc/rc.d/rc3.d`:

```
S10network
S11portmap
S14nfslock
S15nfsfs
```

You can see that they conform to the format `[K or S][number less than 100][filename]`. Each part may be explained as follows:

- The letter S denotes that this service should be started when entering the runlevel. The letter K indicates the services that will be killed when entering a runlevel from another runlevel.
- Services are started in order of the number here, with the lowest numbers being started first.
- the name of the script to run. The scripts reside in /etc/rc.d/init.d.

Runlevels are user configurable, but the conventional assignments in Linux (which TurboLinux follows) are:

- 0 -- Shut down the machine
- 1 -- Single user mode
- 2 -- Multiple user mode, but no networking
- 3 -- Full networking, text mode login
- 4 -- User configurable
- 5 -- Full networking, graphical login
- 6 -- Initiate a warm reboot

As we pointed out earlier, runlevels are user configurable, and TurboLinux provides the tool turboservice to edit the services run at each runlevel. When you start turboservice, it presents you with the following window:

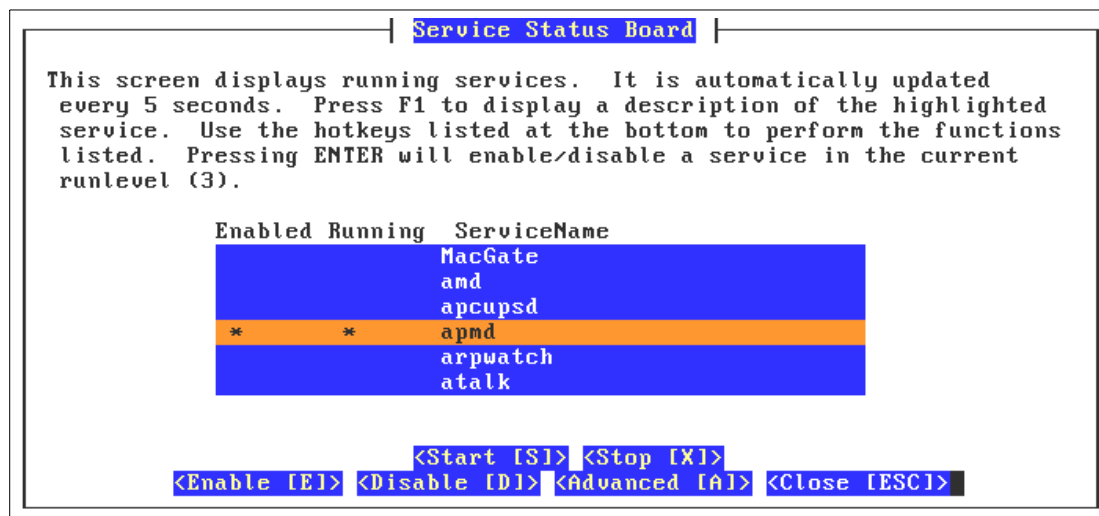


Figure 98. Service Status Board window

Notice that on this window, runlevels are not mentioned. That is because the main screen of turboservice runs in the current runlevel. In this case we are running at runlevel 3, so all the changes we make will be written to

/etc/rc.d/rc3.d/. Once turboservice is running, the columns are laid out in an easy-to-understand format:

- An \* in the **Enabled** column means that TurboLinux will try to start this service when the runlevel is started.
- An \* in the **Running** column means that the service is running at the moment.

The options at the bottom of the window are also simple: You can **Start** or **Stop** a service, as well as choose to **Enable** or **Disable** a service.

Selecting the **Advanced** option takes you the window shown in Figure 99. Here you can choose to enable services to start in runlevels 1-5.

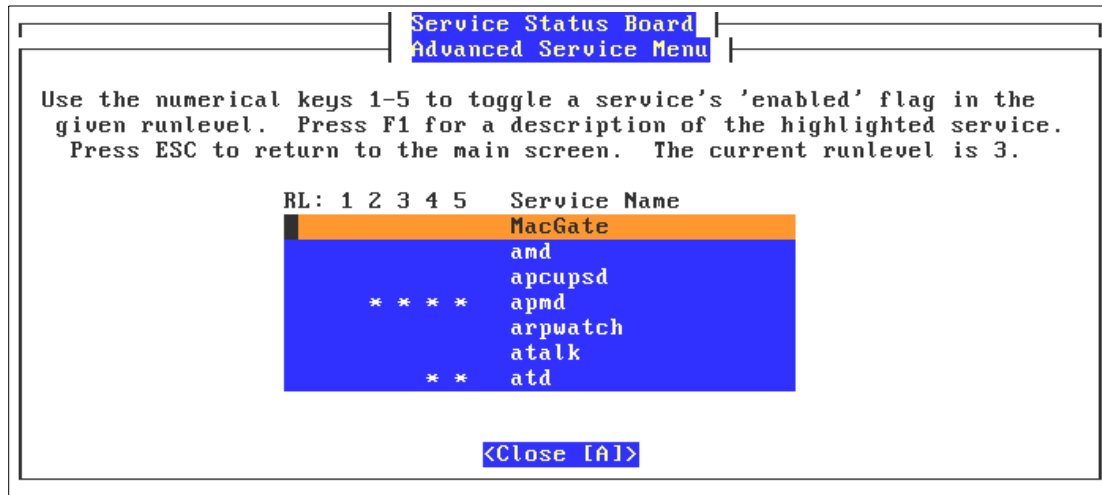


Figure 99. Advanced Service Menu window

Turboservice does not edit the services to stop. If you would like to do that, you must create symbolic links from /etc/rc.d/init.d/ to the runlevel you would like to edit. For example, if you wanted your Web server to stop if you entered runlevel 2, you would type the following command:

```
ln -s /etc/rc.d/initd/httpd /etc/rc.d/rc2.d/K15httpd
```

Of course, you could use a number other than 15. We use it here because that is the default for the HTTPD service.



### 3.9 Time zone and time server configuration

Setting the time zone on a server can be a non-trivial consideration in large environments. TurboLinux provides a single interface that can be used to configure time zone and time server properties.

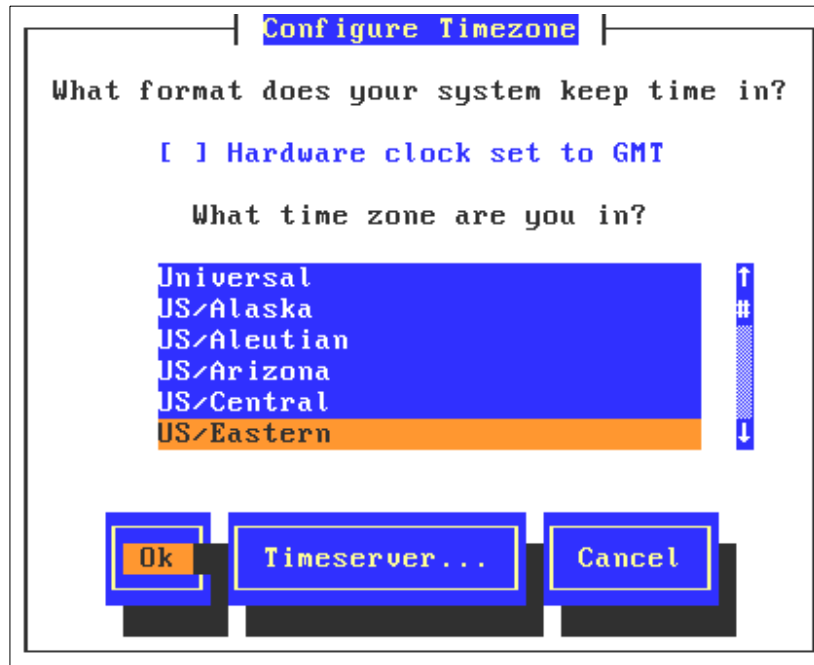


Figure 100. Configure Timezone window

By default, Linux uses the hardware clock in your server to set time. Setting the hardware clock to GMT is a good idea if clients will be logging into your server from other time zones, since the profile for each user can contain their time zone adjustment. If the clock on your server is set to local time, leave the option **Hardware clock set to GMT** unselected.

Choosing the correct time zone for this server is required whether or not the hardware clock is set to GMT.

On the bottom of the window you have an option to have the time for Linux set by a remote Timeserver instead of the local clock.

Selecting the **Timeserver** option generates Figure 101. You have options to connect to an NTP or RDATE based timeserver, and an option of resyncing the clock with varying frequencies.

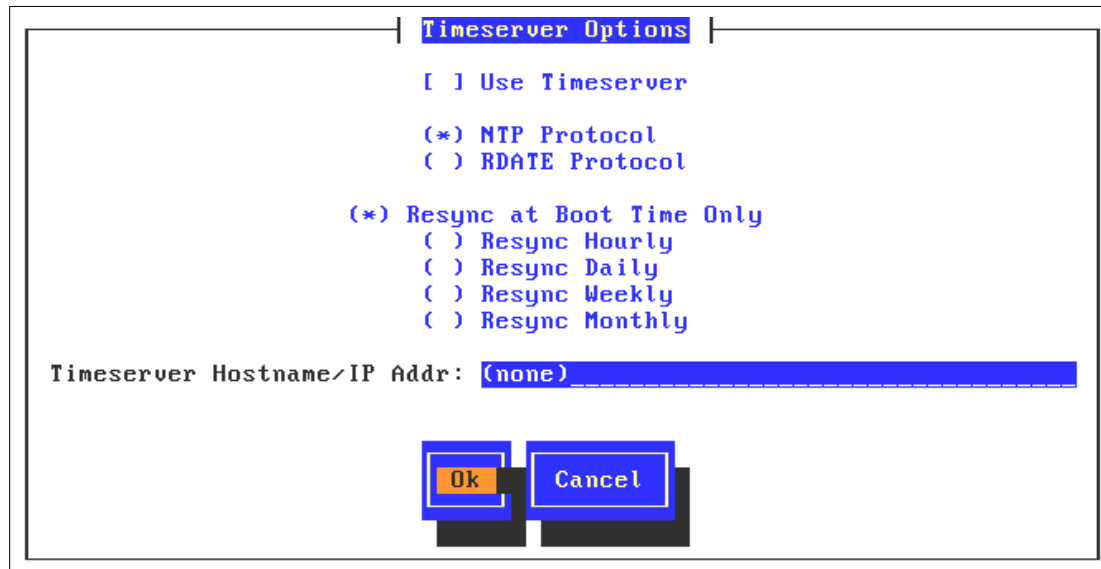


Figure 101. Timeserver Options window

### 3.10 Enabling remote services to your server

Linux provides two different ways to start server services such as FTP or a Web server. You can either start them separately in “stand-alone mode” through the runlevel structure (`/etc/rc.d/rcX.d`), or you can have them wait until a client machine requests the service. This second method is done through the program `inetd`, often called the “super server” because of its role. TurboLinux also comes with `xinetd`, a newer “super server” that offers more flexibility and features than `inetd`. However, `inetd` is the default in TurboLinux, so we will address it in this chapter.

By default, `inetd` is started in all runlevels that have networking support. `inetd` monitors all TCP/IP ports, and starts programs when a request comes to one of the well-known ports on any of the server’s interfaces. The well-known TCP/IP ports are defined in the flat text file `/etc/services`. As usual, `inetd` is configured in a plain-text file, this time the file `/etc/inetd.conf`. Figure 102 is the top part of the file. The rest of the file of organized in much the same way:

```

#inetd.conf This file describes the services that will be available
# through the INETD TCP/IP super server. To re-configure
# the running INETD process, edit this file, then send the
# INETD process a SIGHUP signal:'killall -HUP inetd'
#
# Version:/etc/inetd.conf6.0 Mar 12 2000
#
# Format:
# <service_name> <sock_type> <proto> <flags> <user> <server_path> <args>
#
# For security reasons, all services are turned off by default. Uncomment (or
# add lines) to have services started by inetd (see inetd.conf(8)for details).
#
# Don't forget to also edit /etc/hosts.allow for services which are started
# through tcp_wrappers (/usr/sbin/tcpd in the configuration lines below).
#
# Note: Some servers (typical examples: Web servers like Apache and MTAs like
#       Sendmail) run usually in stand alone mode, i.e. they are _not_ started
#       by the inetd. They are started at boot time (or manually) and keep
#       running.
#####
# ProFTP (standard TurboLinux ftp server)
# Warning: the authentication information for ftp goes as clear text over
# the net. This is especially dangerous if the same login/password combination
# can be used for any shell logins (telnet, ssh). Make sure remote ftp users
# have either /usr/bin/ftponly or /usr/bin/passwd as their login "shells".
# If you choose passwd, they can change their ftp password using telnet
# without having a real shell account on your system.
ftp stream tcpnowaitroot/usr/sbin/tcpdin.proftpd
#####
# WU ftpd (an alternative ftp server)
#ftp stream tcpnowaitroot/usr/sbin/tcpdin.ftpd -l -a
#####
# Telnet
# Warning: telnet is inherently insecure as a protocol. All network traffic,
# including authentication information (login and password) are transmitted
# as clear text. Look for secure alternatives (e.g. ssh).
# The -h option prevents your telnetd from giving away information which
# may be useful for potential system crackers. See telnetd(8) for details.
#telnet stream tcp nowaitroot/usr/sbin/tcpdin.telnetd -h
#####
# POP3 mail server
#pop-3 stream tcp nowait root /usr/sbin/tcpdipop3d
#####

```

Figure 102. inetd.conf file

As you can see from the selection of /etc/inetd.conf, the comments explain what services can be run, and give some warnings as well. In the example above we have enabled the ProFTP server by removing the # at the beginning of the line shown in bold.

The networking subsystem maintains two more files for security purposes that will have to be edited in order for remote users to access an FTP server

on this system. The files are /etc/hosts.allow and /etc/hosts.deny. We will discuss them now.

The default /etc/hosts.allow is listed below:

```
hosts.allow  This file describes the names of the hosts which are
#           allowed to use the local INET services, as decided
#           by the '/usr/sbin/tcpd' server.
#
# See man hosts_access(5) for more information

ALL : 127.0.0.1
```

Figure 103. hosts.allow file

In order to allow other computers on the network to access this server, you will have to add the line

```
ALL : ALL
```

This will enable your remote access.

You also need to edit the /etc/hosts.deny file. In Figure 104, you will notice that access is denied to all systems, including the localhost. This will also prevent access to any systems, even though it is specified in the /etc/hosts.allow file.

```
#
# hosts.deny  This file describes the names of the hosts which are
#            *not* allowed to use the local INET services, as decided
#            by the '/usr/sbin/tcpd' server.
#
# See man hosts_access(5) for more information.

ALL: ALL
```

Figure 104. Hosts.deny file

In order to allow access from all remote hosts you need to change the last line in Figure 104 to the following:

```
# ALL: ALL
```

Adding a # at the front of the line disables the exclusion, thus allowing the hosts.allow file to give access to your system. The next time inetd is started, your FTP server will be available to hosts on your network. To restart inetd immediately, type the command

```
killall -HUP inetd
```

### 3.11 File system permissions

Linux has inherent security features, the most noticeable being file system permissions. Setting permissions on files allows the system administrator to restrict access to parts of the file system.

File permissions can be set on files and directories. The easiest way to see an example of this is looking in the /home directory:

```
mail:/home # ls -l
total 1
drwxr-xr-x 19 root    root    396 Nov 15 21:06 .
drwxr-xr-x 22 root    root    467 Nov 13 16:28 ..
drwx----- 6 davej   users  912 Nov 15 21:05 davej
drwx----- 6 george  users  912 Nov 15 21:03 george
drwx----- 6 ivo    users  912 Nov 15 21:02 ivo
drwx----- 6 jakob  users  912 Nov 15 21:03 jakob
drwx----- 6 jasmin  users  912 Nov 15 21:04 jasmin
drwx----- 6 jens    users  912 Nov 15 21:04 jens
drwx----- 6 jhaskins users  912 Nov 15 21:02 jhaskins
drwx----- 6 justin  users  912 Nov 15 21:06 justin
drwx----- 6 lenz   users  912 Nov 15 21:03 lenz
drwx----- 6 linux  users  912 Nov 15 21:03 linux
drwx----- 6 malcom  users  912 Nov 15 21:04 malcom
drwx----- 6 rachael users  912 Nov 15 21:03 rachael
drwx----- 6 rafiu  users  912 Nov 15 21:04 rafiu
drwx----- 6 ruediger users  912 Nov 15 21:04 ruediger
drwx----- 6 rufus  users  912 Nov 15 21:02 rufus
drwx----- 6 ted    users  912 Nov 15 21:03 ted
drwx----- 6 uzi    users  912 Nov 15 21:04 uzi
mail:/home #
```

Figure 105. Viewing file permissions

Taking the user “linux” as an example:

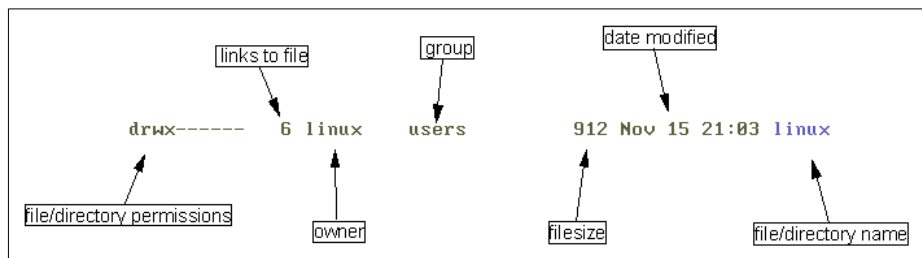


Figure 106. Explanation of **ls** output

What we are most interested in is the file/directory permissions. This signifies a lot of information in a short amount of space:

**d** - The first character in the permissions signifies that this is a **directory**. Other files are represented by:

- - a normal file.

**l** - a symbolic link to another file.

**c** - refers to files in the /dev directory. This signifies the file represents a character device.

**b** - refers to files in the /dev directory. This signifies the file represents a block device.

**rwX** - In this case it allows only the owner of the file (in this case “linux”) to read, write and execute this file.

Type	Owner	Group	World
d	rwX	---	---

The owner of the file is the user that created the file. The group part is the group that owns the file (for example, the group **users**). The world part means everyone else; setting a permission in the world part sets the permission for every user, irrelevant of their group membership and so on.

Here is another example:

-rwxr-xr--

This means that this is a normal file, the owner can read, write and execute the file, the group can read and execute the file, and everyone else can read the file, but not modify or execute it.

If you set a directory as:

drwxrw-rw-

you are saying that only the directory owner is allowed to execute something “inside” the directory. So if another user tries to change directory into this directory, they will get a “permission denied” error message. This is exactly what happens with regards to users’ home directories.

To change the permissions on a file, you use the **chmod** command. Only *root* can modify files that do not belong to them. You must own the file to be able to change its permissions.

The easiest way to change permissions is to use symbolic representations of what you want permissions to be.

**Note**

The other way to represent file permissions is to use octals. For more information about this and the `chmod` command see the `chmod` man page.

```
chmod g+rw myfile
```

The command above is one of the simplest ways of changing a permission. You are saying that you want the file `myfile` to allow all members of the group to be able to read and write to it.

If you used a - (minus sign) instead of a plus, you would be taking away those permissions. This would mean that members of the group would not be allowed to read or write to the file.

You can mix adding and removing permissions in the same command:

```
chmod u+x-rw myfile
```

This will allow executing the file, but will not allow reading or writing the file for the file owner.

Here is a summary of the symbolic representations available in `chmod`:

**r** - read

**w** - write

**x** - execute

- - take away the permissions

+ - add the permissions

**s** - set the SUID bit. This says that if the file is executable, it will be run as the owner of the file, not as the user that is running the file.





---

## Chapter 4. The ServeRAID controller and TurboLinux

In this section we will describe how to install TurboLinux on the xSeries and Netfinity servers with an IBM ServeRAID controller and how to use the features of the IBM ServeRAID controller. If you have a ServeRaid controller on your server, please read this entire chapter before installing TurboLinux.

Before you start the installation, you need to define the RAID arrays and the logical drives. The logical drives are represented to the operating system as if they were physical disk drives. For more information on RAID levels and performance issues, see Appendix A, "RAID levels" on page 329.

### Stop

Before installing TurboLinux on the xSeries and Netfinity server with an IBM ServeRAID controller, you need to define RAID arrays and logical drives. You can do this with ServerGuide, which comes with all IBM Netfinity servers, or with the ServeRAID Configuration CD, which is available at <http://www.pc.ibm.com/support>.

We strongly recommend that you use hot spare hard disks in your system to secure your data the best possible way.

---

### 4.1 Updating the ServeRaid device driver for Linux

TurboLinux supports the ServeRAID controller. The driver version included in TurboLinux 6 is 3.60. This driver release can be only used with the same or lower firmware release. If you are using a ServeRaid 4 card, you will not be able to install until you have followed the instructions outlined in this section.

If you have a ServeRaid 2 or 3 card, we suggest you do the following:

1. Install TurboLinux 6 and reboot.
2. Update the ServeRaid driver to the current level. Reboot the server and confirm that it will boot with the new ServeRaid driver installed.
3. Update the firmware and BIOS of the ServeRaid to the current level.

In our case, the current firmware and driver are 4.4, so we must compile the new driver and update the "Extra Hardware" disk provided with TurboLinux. However, the process we will outline should be the same when newer versions of the ServeRaid driver and firmware are available.

**Note**

If your ServeRAID BIOS/Firmware is higher than 3.50C you can force downgrade in the BIOS/Firmware utility by pressing CTRL+F.

### 4.1.1 Obtaining the required source code

If the kernel source tree was not installed during the installation, you will need to install it. To determine if the package is installed, type the command:

```
rpm -q kernel-source
```

If it is not installed, insert the main TurboLinux CD and type the commands:

```
mount /mnt/cdrom  
cd /mnt/cdrom/TurboLinux/RPMS  
rpm -Uhv kernel-source-2.2.14-8.i386.rpm
```

You can get the latest drivers and utilities for ServeRAID administration from a few different locations. Primarily, you can download the “ServeRaid Support CD” from either of these two sites:

```
http://www.pc.ibm.com/support  
http://www.developer.ibm.com/welcome/netfinity/serveraid.html
```

You can also download the individual files from the same site. The files are:

- `ips-440.tgz`: contains the kernel patch for the 2.2.x kernel, which enables the support for IBM ServeRAID adapter in those kernels, Alternatively, you can download the file. This file is also on the ServeRaid CD in the directory `/programs/linux/driver`.
- `ipsutils.rpm`: this file contains the Linux utilities for IBM ServeRAID SCSI adapter.
- `RaidMan.rpm`; this file contains the Linux ServeRAID Manager, which can be used to locally and remotely configure and monitor the ServeRAID controller used in Linux installation through the graphical user interface.

In our case, we have through downloaded the file `ips-440.tgz` to `/home/ftp/pub`. Next we will unpack the driver and copy it into the kernel source tree.

```
tar -zxvf ips-440.tgz  
cp ips.h /usr/src/linux/drivers/scsi  
cp ips.c /usr/src/linux/drivers/scsi
```

## 4.1.2 Compiling the ServeRaid driver in order to install TurboLinux 6

### Note

The ServeRaid BIOS 4.x is only supported with the ServeRaid driver 4.x. For ServeRaid adapters 4L/M you must use at least ServeRaid driver 4.20 and for ServeRaid adapter 4H you must use at least ServeRaid driver 4.00.

If you intend to install TurboLinux 6 on a ServeRAID controller, you must install TurboLinux on a machine that does not have an IBM ServeRAID controller and compile the driver on that machine. If you are compiling this driver to use during the install, follow these steps:

1. Copy the boot kernel config to the current config file with the commands:

```
cd /usr/src/linux/  
cp ./configs/kernel-2.2.14-i386-BOOT.config ./config
```

2. Using the text editor of your choice, edit the file /usr/src/linux/Makefile and make sure the fourth line reads:

```
EXTRAVERSION = -8BOOT
```

3. You now compile the kernel modules by typing the command:

```
make dep modules
```

This will take from 10 to 30 minutes, depending on the speed of the server's CPU. After the compile has completed, the new driver is placed on the disk as:

```
/usr/src/linux/drivers/scsi/ips.o
```

4. Next, the older driver on the "Extra Hardware" diskette must be replaced with the new one you have just compiled. To do that, do the following:
  - a. Create a temporary directory to do our work, and `cd` into that directory:

```
mkdir /tmp/ips-4.4  
cd /tmp/ips4.4
```

- b. Insert the main TurboLinux CD and copy the "Extra Hardware" diskette image to your working directory:

```
mount /mnt/cdrom  
cp /mnt/cdrom/images/extrahw.img /tmp/ips-4.4
```

- c. The image file is actually a zipped file, so we first rename it to a file with the `.gz` extension (in Linux, the `mv` command functions as both "move" and "ren" in DOS/Windows) and unzip it. Notice that the resulting file is considerable larger than the original.

```
mv extrahw.img extrahw.img.gz
gunzip -d extrahw.img
```

- d. The image is an entire file system, so we create a directory and mount the file system. The parameter “-o loop” is used to tell Linux you are mounting a file, not a piece of hardware.

```
mkdir hw
mount -t ext2 -o loop extrahw.img hw
```

- e. Copy the new driver into the mounted image (when asked to overwrite the file, type *y*). After the new driver has been copied, unmount the image.

```
cp /usr/src/linux/drivers/scsi/ips.o hw
umount hw
```

- f. Zip the file back up with maximum compression [-9], then change the name back to the original

```
gzip -9 extrahw.img
mv extrahw.img.gz extrahw.img
```

- g. Label a blank floppy diskette “Extra Hardware with IPS 4.4,” and insert it into the first floppy drive and issue this command to create your modified “Extra Hardware” diskette:

```
dd if=extrahw.img of=/dev/fd0
```

Linux uses a different kernel during the install than when it is up and running. Therefore you must now proceed to the next section and create another version of the ServeRAID driver that will be installed at the end of the install, but before the server is rebooted.

### 4.1.3 Compiling the ServeRAID driver after TurboLinux is installed

Before we build a new driver, you should make a boot disk so the system can still be booted if there is a problem. To do this, label a disk “Emergency boot disk - TurboLinux 6” and type the following command:

```
mkbootdisk --device /dev/fd0 2.2.14-8
```

With this diskette still in the drive, reboot the server and confirm that you can log in to the server as root. If you can, then this is a good boot disk for this server. You should now reboot the server without the diskette in and proceed.

It is also recommended that you make a backup copy of the kernel, initial RAM disk, and current ServeRAID driver with the following commands:

```
cp /boot/vmlinuz /boot/vmlinuz-safe
cp /boot/initrd-safe /boot/initrd-safe
```

```
cp /lib/modules/current/scsi/ips.o /boot/ips.o-safe
```

Now you can modify `/etc/lilo.conf` to add an entry for your “safe” kernel. Run the program `turboscfg` (explained in detail in 3.6, “Administering file systems and the boot record” on page 68), choose **Bootrecord-->Add-->Linux Image** and create an entry like the one pictured below.

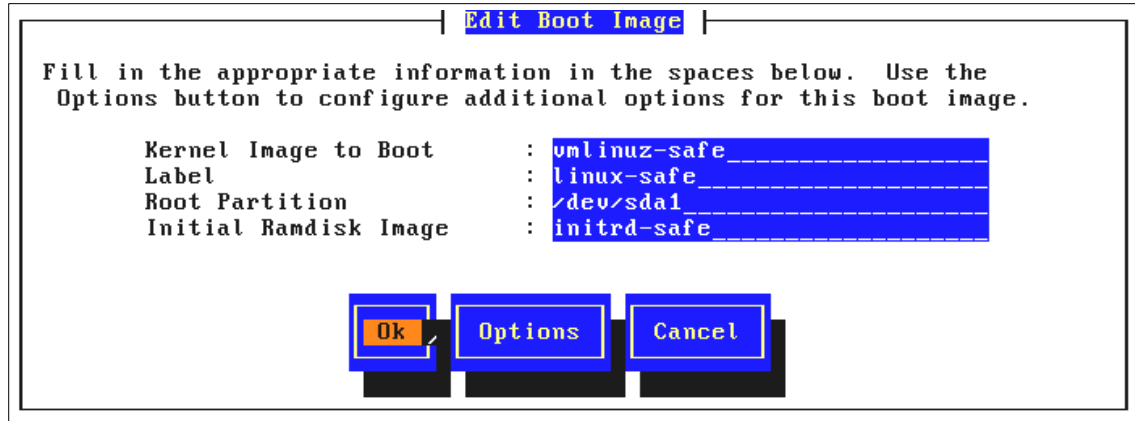


Figure 107. Edit Boot Image window

Select **OK** to update LILO with the new entry.

The driver we will build in this section is for use after the machine is up and running, or to be inserted during the end of the install on a ServeRAID 4 card. The steps to create this driver are as follows.

1. Copy the proper kernel config file with the commands:

```
cd /usr/src/linux/  
cp ./configs/kernel-2.2.14-i686.config ./config
```

The exact kernel config file depends on which kernel you are using. For example, we would use `kernel-2.2.14-i686-SMP.config` for a server with multiple processors. Remember that you must use the same kernel here that you intend to choose during the install at Figure 31 on page 29.

2. Using the text editor of your choice, edit the file `/usr/src/linux/Makefile` and make sure the fourth line reads:

```
EXTRAVERSION = -8
```

We do not want this driver to have the word “BOOT” in as the driver in the previous section did.

3. Next, you create dependencies, compile the kernel, compile the modules, install the kernel, and install the modules:

```
make depmake bzImage modules install modules_install
```

These parameters to the `make` command can be run separately. We have chosen to run them at once for simplicity.

If you have just finished creating a modified “Extra Hardware” diskette, you must now copy this newly compiled drive to a diskette. You will need this diskette near the end of the install. Label a diskette “ServeRAID 4.4 driver,” insert it into the diskette drive and do the following:

1. Format a floppy with the ext2 filesystem and mount it:

```
mke2fs /dev/fd0
mount /mnt/floppy
```

2. Copy the newly compiled ServeRAID driver to the diskette and unmount it:

```
cp /lib/modules/current/scsi/ips.o /mnt/floppy
umount /mnt/floppy
```

3. You can now start the installation of TurboLinux 6.

Restart the server now to use the new kernel. You can check if the correct driver is loaded by executing the command:

```
cat /proc/scsi/ips/2
```

The last number depends on your configuration and it could be 1 or something else. If the correct driver is loaded you should see a window similar to Figure 108.

```
[root@x230 col]# cat /proc/scsi/ips/2
IBM ServeRAID General Information:

Controller Type           : ServeRAID 4M
Memory region            : 0xf7ffc000 (8192 bytes)
Shared memory address    : 0x8803e000
IRQ number               : 20
BIOS Version             : 4.40.03
Firmware Version        : 4.40.03
Boot Block Version      : 4.40.03
Driver Version          : 4.40.03
Max Physical Devices    : 30
Max Active Commands     : 128
Current Queued Commands : 0
Current Active Commands : 0
Current Queued PT Commands : 0
Current Active PT Commands : 0
```

Figure 108. ServeRAID 4 status report

## 4.2 Installing the ServeRAID command line tools for Linux

To successfully install the ipsutils package you have to be logged in as “root”. After you have downloaded the ipsutil.rpm package you need to install it. The ipsutil package is a standard Red Hat Package Manager (RPM) package. TurboLinux uses RPM for installing packages, so the RPM utility is already installed on your system. To check if your RPM utility is working, open a terminal window and execute this command:

More GUI-oriented users will probably want to use the kpackage tool, which is part of the KDE graphic environment and is used for installing packages. But if you try to install the ipsutils.rpm with this tool you will see a window similar to Figure 109.

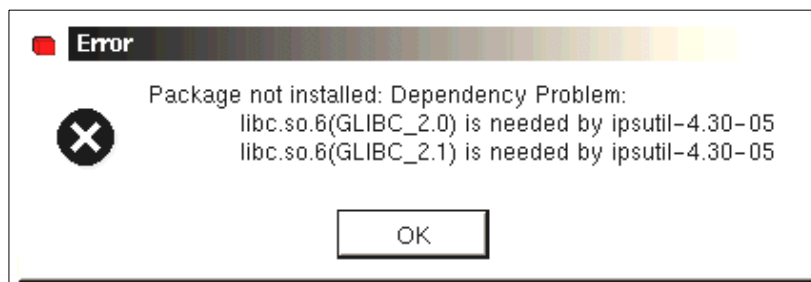


Figure 109. Error installing ipsutils.rpm with kpackage

Experienced users will recognize that this is a script, but even if you try to install without enabling the **Check Dependencies** option, as you can see in Figure 110, you will still get the same error.

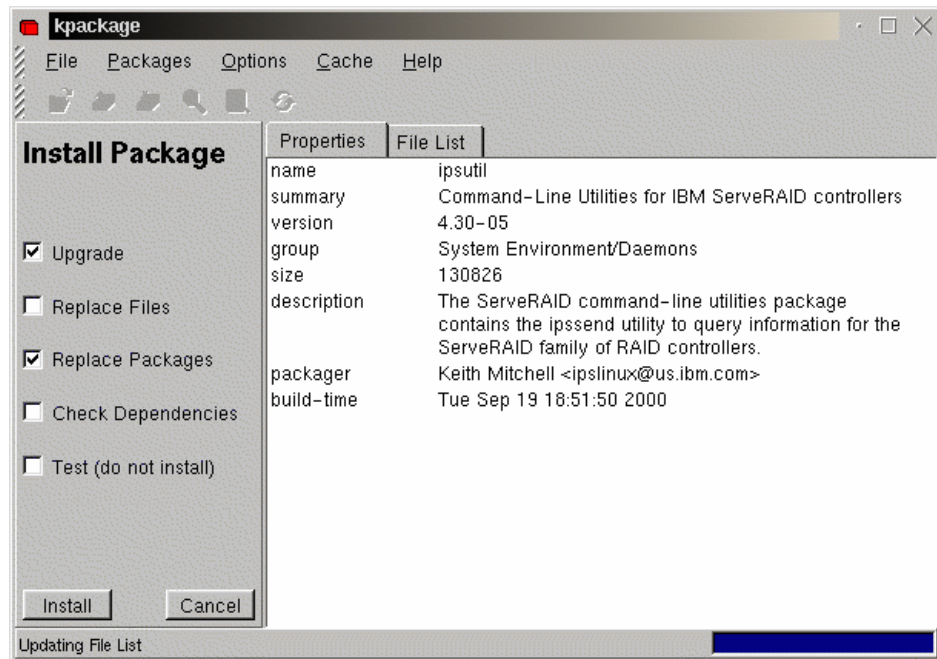


Figure 110. Installing without dependencies in kpackage

This is because ipsutils.rpm displays the copyright message before installing. So the only way to install the package is to execute the following command from a terminal:

```
rpm -Uhv --nodeps ipsutil.rpm
```



**Note**

You can also copy the `ipssend` program from your ServeRAID CD-ROM to the `/usr/bin` directory with the commands:

```
mount /mnt/cdrom
```

```
cp /mnt/cdrom/programs/linux/cmdline/ipssend /usr/bin/
```

Then you need to change the permissions so that you can execute the command with:

```
chmod 755 /usr/bin/ipssend
```

This assumes that your current directory is where the `ipsutil.prm` file resides. The necessary files will be installed in the `/usr/bin` directory. To see if the utilities are working, type the following command:

```
ipssend
```

You will see output similar to Figure 111.

```
Licensed Material - Property of IBM Corporation
IBM ServeRAID Command Line Interface v4.40.03
(C) Copyright IBM Corp. 1994, 2000. All Rights Reserved.
US Government Restricted Rights - Use, Duplication, or Disclosure
Restricted by GSA ADP Schedule Contract with IBM Corporation

Usage: IPSEND <Command> <Param 1> ... <Param N>
Help : IPSEND <Command> for specific help on any command.

  Command | Param 1 | Param 2 | Param 3 | Param 4 | Param 5
  -----|-----|-----|-----|-----|-----
AUTOSYNC | Controller | Logical Drive | NOPROMPT |
BACKUP   | Controller | Filename     | NOPROMPT |
DEVINFO  | Controller | Channel      | SCSI ID  |
DRIVEVER | Controller | Channel      | SCSI ID  |
ERASEEVENT | Controller | Options      |
GETCONFIG | Controller | Options      |
GETEVENT  | Controller | Options      |
GETSTATUS | Controller |
HSREBUILD | Controller | Options      |
INIT     | Controller | Logical Drive | NOPROMPT |
REBUILD  | Controller | Channel      | SCSI ID  | New Channel | New SCSI ID
RESTORE  | Controller | Filename     | NOPROMPT |
SETSTATE | Controller | Channel      | SCSI ID  | New State  |
SYNCH    | Controller | Scope        | Scope ID |
UNATTENDED | Controller | Options      |
UNBLOCK  | Controller | Logical Drive |
```

Figure 111. `Ipssend` command output

As you can see, `ipssend` supports quite a lot of commands for dealing with the IBM ServeRAID controller. In this section we will cover the ones that are necessary in order to use the ServeRAID controller efficiently.

#### 4.2.1 `getconfig`

This command is used to get the configuration information of the IBM ServeRAID controller, the logical drives and the physical drives. The `getconfig` command has the following syntax:

```
ipssend getconfig <Controller> <Options>
```

The parameters are explained in Table 4.

Table 4. `getconfig` command parameters

Parameter	Description
Controller	Number of controller (1 to 12)
Options	AD for Controller Information
	LD for Logical Drive Information
	PD for Physical Device Information
	AL (default) for All Information

To get all information about the first ServeRAID controller, execute the following command:

```
ipssend getconfig 1
```

You will see a window similar to Figure 112.

```

Found 1 IBM ServeRAID Controller(s).
Read Configuration has been initiated for Controller 1...
-----
Controller Information
-----
Firmware Version      : 3.73.00
Boot Block Version    : 3.00.16
BIOS Version          : 4.40.03
Controller Type       : ServeRAID-3L
Controller Slot Information : 1
Controller Configuration ID : Null Config
SCSI Channel Description : 1 parallel SCSI wide
Initiator IDs (Channel/SCSI ID) : 1/7
Maximum Physical Devices : 15
Defunct Disk Drive Count : 0
Logical Drives/Offline/Critical : 1/0/0
Rebuild Rate (Low/Medium/High) : High
Read Ahead           : Adaptive
Unattended Mode (Yes/No) : No
Part of Cluster (Yes/No) : No
Concurrent Commands Supported : 32
Configuration Update Count : 1
-----
Logical Drive Information
-----
Logical Drive Number 1
Status of Logical Drive : Okay (OKY)
Raid Level              : 5
Size (in MB)           : 52068
Write Cache Status     : Write Back (WB)
Number of Chunks       : 7
Stripe Unit Size      : 8K
Access Blocked         : No
Part of Array          : A
Part of Merge Group    : 207

Array A Stripe Order (Channel/SCSI ID) : 1,1 1,2 1,3 1,4 1,8 1,9 1,10
-----
Physical Device Information
-----
Channel #1:
Initiator at SCSI ID 7
Target on SCSI ID 0
Device is a 16 bit, Fast SCSI, tag queuing Hard Disk
SCSI ID : 0
PFA (Yes/No) : No
State : Hot Spare (HSP)
Size (in MB)/(in Sectors) : 8678/17773888
Device ID : IBM-PSG ST39175L04303AL0A27C

```

Figure 112. Executing `ipssend getconfig 1`

In this output you can see all information about the ServeRAID configuration. If you want information only about the controller itself, execute this command:

```
ipssend getconfig 1 ad
```

You will see output similar to Figure 113.

```
[root@nf3500a /root]# ipssend getconfig 1 ad
Found 1 IBM ServeRAID Controller(s).
Read Configuration has been initiated for Controller 1...
-----
Controller Information
-----
Firmware Version      : 3.73.00
Boot Block Version    : 3.00.16
BIOS Version          : 4.40.03
Controller Type       : ServeRAID-3L
Controller Slot Information : 1
Controller Configuration ID : Null Config
SCSI Channel Description : 1 parallel SCSI wide
Initiator IDs (Channel/SCSI ID) : 1/7
Maximum Physical Devices : 15
Defunct Disk Drive Count : 0
Logical Drives/Offline/Critical : 2/0/0
Rebuild Rate (Low/Medium/High) : High
Read Ahead            : Adaptive
Unattended Mode (Yes/No) : No
Part of Cluster (Yes/No) : No
Concurrent Commands Supported : 32
Configuration Update Count : 24
Command Completed Successfully.
```

Figure 113. Executing ipssend getconfig 1 ad

To get information about logical drives execute this command:

```
ipssend getconfig 1 ld
```

You will get output similar to Figure 114.

```
[root@nf3500a /root]# ipssend getconfig 1 ld
Found 1 IBM ServeRAID Controller(s).
Read Configuration has been initiated for Controller 1...
-----
Logical Drive Information
-----
Logical Drive Number 1
  Status of Logical Drive      : Okay (OKY)
  Raid Level                   : 5
  Size (in MB)                 : 2000
  Write Cache Status           : Write Through (WT)
  Number of Chunks             : 3
  Stripe Unit Size             : 8K
  Access Blocked               : No
  Part of Array                : A
  Part of Merge Group          : 207
Logical Drive Number 2
  Status of Logical Drive      : Okay (OKY)
  Raid Level                   : 5
  Size (in MB)                 : 2000
  Write Cache Status           : Write Through (WT)
  Number of Chunks             : 3
  Stripe Unit Size             : 8K
  Access Blocked               : No
  Part of Array                : A
  Part of Merge Group          : 207

  Array A Stripe Order (Channel/SCSI ID) : 1,1 1,2 1,3
Command Completed Successfully.
```

Figure 114. Executing ipssend getconfig 1 ld

From this output you can get all information about the logical drives:

- Drive status
- RAID Level
- Size
- Write Cache Status
- Number of Chunks
- Stripe Unit Size
- Access
- Array

To get detailed information about a physical drive, execute this command:

```
ipssend getconfig 1 pd
```

You will see output similar to Figure 115.

```

[root@nf3500a /root]# ipssend getconfig 1 pd
Found 1 IBM ServeRAID Controller(s).
Read Configuration has been initiated for Controller 1...
-----
Physical Device Information
-----
Channel #1:
Initiator at SCSI ID 7
Target on SCSI ID 0
Device is a 16 bit, Fast SCSI, tag queuing Hard Disk
SCSI ID : 0
PFA (Yes/No) : No
State : Ready (RDY)
Size (in MB)/(in Sectors): 8678/17773888
Device ID : IBM-PSG ST39175L04303AL0A27C
Target on SCSI ID 1
Device is a 16 bit, Fast SCSI, tag queuing Hard Disk
SCSI ID : 1
PFA (Yes/No) : No
State : Online (ONL)
Size (in MB)/(in Sectors): 8678/17773888
Device ID : IBM-PSG ST39175L04303AL09YSS
Target on SCSI ID 2
Device is a 16 bit, Fast SCSI, tag queuing Hard Disk
SCSI ID : 2
PFA (Yes/No) : No
State : Online (ONL)
Size (in MB)/(in Sectors): 8678/17773888
Device ID : IBM-PSG ST39175L04303AL0A2QK
Target on SCSI ID 3
Device is a 16 bit, Fast SCSI, tag queuing Hard Disk
SCSI ID : 3
PFA (Yes/No) : No
State : Online (ONL)
Size (in MB)/(in Sectors): 8678/17773888
Device ID : IBM-PSG DMVSO9D 01B0F802F9F4

```

Figure 115. Executing ipssend getconfig 1 pd

#### 4.2.2 The getstatus command

This command is used to retrieve the current status of the IBM ServeRAID controller. The getstatus command has the following syntax:

```
ipssend getstatus <Controller>
```

The parameters are explained in Table 5.

Table 5. getstatus command parameters

Parameter	Description
Controller	Number of controller (1 to 12)

To get the status of first ServeRAID controller in your IBM Netfinity server, execute this command:

```
ipssend getstatus 1
```

You will see output similar to Figure 116.

```
[root@nf3500a /root]# ipssend getstatus 1

Found 1 IBM ServeRAID Controller(s).
Background Command Progress Status for controller 1...
  Current/Most Recent Operation : Rebuild
  Logical Drive in Progress     : 2
  Rebuild Rate                  : High
  Status                        : Successfully Completed
  Logical Drive Size (in Stripes): 128000
  Number of Remaining Stripes   : 0
  Percentage Complete           : 100.00%
Command Completed Successfully.
```

Figure 116. Executing `ipssend getstatus 1`

If the ServeRAID controller is in the middle of rebuilding a drive, you will see output similar to Figure 117.

```
[root@nf3500a /root]# ipssend getstatus 1

Found 1 IBM ServeRAID Controller(s).
Background Command Progress Status for controller 1...
  Current/Most Recent Operation : Rebuild
  Logical Drive in Progress     : 1
  Rebuild Rate                  : High
  Status                        : In Progress
  Logical Drive Size (in Stripes): 128000
  Number of Remaining Stripes   : 126473
  Percentage Complete           : 1.19%
Command Completed Successfully.
```

Figure 117. Executing `ipssend getstatus 1` during rebuilding of a drive

### 4.2.3 The `devinfo` command

This command is used to retrieve the current status of the devices connected to the IBM ServeRAID controller. The `devinfo` command has the following syntax:

```
ipssend devinfo <Controller> <Channel> <SCSI ID>
```

The parameters are explained in Table 6.

Table 6. `devinfo` command parameters

Parameter	Description
Controller	Number of controller (1 to 12)
Channel	Channel of Device (1 to 3)

Parameter	Description
SCSI ID	SCSI ID of Device (0 to 15)

To get the status of a device with SCSI ID 1 on channel 1 on the first ServeRAID controller in your IBM Netfinity server, execute the command:

```
ipssend devinfo 1 1 1
```

You will see output similar to Figure 118.

```
[root@nf3500a /root]# ipssend devinfo 1 1 1
Found 1 IBM ServeRAID Controller(s).
Device Information has been initiated for controller 1...
Device is a 16 bit, Fast SCSI, tag queuing Hard Disk
Channel          : 1
SCSI ID          : 1
PFA (Yes/No)     : No
State            : Online (ONL)
Size (in MB)/(in Sectors): 8678/17773888
Device ID        : IBM-PSG ST39175L04303AL09YSS
Command Completed Successfully.
```

Figure 118. Executing `ipssend devinfo 1 1 1`

If the ServeRAID controller is in the middle of rebuilding a drive, you will see output similar to Figure 119.

```
[root@nf3500a /root]# ipssend devinfo 1 1 2
Found 1 IBM ServeRAID Controller(s).
Device Information has been initiated for controller 1...
Device is a 16 bit, Fast SCSI, tag queuing Hard Disk
Channel          : 1
SCSI ID          : 2
PFA (Yes/No)     : No
State            : Rebuild (RBL)
Size (in MB)/(in Sectors): 8678/17773888
Device ID        : IBM-PSG ST39175L04303AL0A2QK
Command Completed Successfully.
```

Figure 119. Executing `ipssend devinfo 1 1 2` during rebuilding of a drive

#### 4.2.4 The `hsrebuild` command

This command is used for setting the state of the Hot Swap Rebuild option. The `hsrebuild` command has the following syntax:

```
ipssend hsrebuild <Controller> <Options>
```



The parameters are explained in Table 7.

Table 7. *hsrebuild* command parameters

Parameter	Description
Controller	Number of controller (1 to 12)
Options	ON: enable Hot Swap Rebuild
	?: Display status of Hot Swap Rebuild feature

With this command you can retrieve or set the Hot Swap Rebuild feature. If the Hot Swap Rebuild feature is ON, it means that if one drive in the RAID array fails, rebuilding of this drive will start automatically when you replace the failed drive with a new one. This can improve the safety of your data.

**Note**

The Hot Swap Rebuild feature should not be confused with a hot spare drive. A hot spare drive means that a drive is in a waiting state as long as the RAID array is in an Okay state. Once the RAID array becomes in a Critical state, the hot spare drive is enabled and the data from the defunct drive automatically gets rebuilt onto the hot spare drive, disregarding the Hot Swap Rebuild setting.

To retrieve the information about the Hot Swap Rebuild status on the first ServeRAID controller, execute this command:

```
ipssend hsrebuild 1 ?
```

You will see output similar to Figure 120.

```
[root@nf3500a /root]# ipssend hsrebuild 1 ?
Found 1 IBM ServeRAID Controller(s).
Set Hot Swap Rebuild has been initiated for controller 1...
Hot Swap Rebuild is On for controller 1.
```

Figure 120. Executing `ipssend hsrebuild 1 ?`

To enable the Hot Swap Rebuild option, execute this command:

```
ipssend hsrebuild 1 on
```

You will see output similar to Figure 121.

```
[root@nf3500a /root]# ipssend hsrebuild 1 on
Found 1 IBM ServeRAID Controller(s).
Set Hot Swap Rebuild has been initiated for controller 1...
Hot Swap Rebuild is already On for controller 1.
```

Figure 121. Executing `ipssend hsrebuild 1 on`

#### 4.2.5 The `setstate` command

With the `setstate` command you redefine the state of a physical device from the current state to the designated state. The `setstate` command has the following syntax:

```
ipssend setstate <Controller> <Channel> <SCSI ID> <New State>
```

The parameters are explained in Table 8.

Table 8. *setstate* command parameters

Parameter	Description
Controller	Number of controller (1 to 12)
Channel	Channel of device (1 to 3)
SCSI ID	SCSI ID of device (0 to 15)
New State	EMP (Empty) RDY (Ready) HSP (Hot Spare) SHS (Standby Hot Spare) DDD (Defunct Disk Drive) DHS (Defunct Hot Spare) RBL (Rebuild) SBY (Standby) ONL (Online)

**Stop**

Extreme caution must be taken when executing this command! For example, redefining a defunct (DDD) device to online (ONL) without going through a rebuild is extremely dangerous.

Before changing the state of a physical device, you can check the current status with this command:

```
ipssend getconfig 1 pd
```

With this command you will see all physical devices, except empty ones, on the first IBM ServeRAID controller. For example if you want to set the state of the device on the first ServeRAID controller, channel 1 and SCSI ID 0 to RDY - Ready, execute this command:

```
ipssend setstate 1 1 0 rdy
```

You will see output similar to Figure 122.

```
[root@nf3500a /root]# ipssend setstate 1 1 0 rdy
Found 1 IBM ServeRAID Controller(s).
Set Device State has been initiated for Controller 1...
Command Completed Successfully.
```

Figure 122. Executing *ipssend setstate 1 1 0 rdy*

You can verify the change of the device state by executing this command:

```
ipssend getconfig 1 pd
```

#### 4.2.6 The `synch` command

This command is used to synchronize the parity information on redundant logical drives. If the parity information is inconsistent, it will automatically be repaired. The `synch` command has the following syntax:

```
ipssend synch <Controller> <Scope> <Scope ID>
```

The parameters are explained in Table 9.

Table 9. `synch` command parameters

Parameter	Description
Controller	Number of controller (1 to 12)
Scope	Drive for a single logical drive
Scope ID	Number of logical drive (1 to 8)

**Note**

We recommend that you use this command on a weekly basis.

#### 4.2.7 The `unattended` command

This command is used to alter the unattended mode of the ServeRAID controller. The `unattended` command has the following syntax:

```
ipssend unattended <Controller> <Options>
```

The parameters are explained in Table 10.

Table 10. `unattended` command parameters

Parameter	Description
Controller	Number of controller (1 to 12)
Options	ON: enable unattended mode
	OFF: disable unattended mode
	?: display status of unattended mode feature

If you want to see the current status of the first ServeRAID controller, execute this command:

```
ipssend unattended 1 ?
```

You will see output similar to Figure 123.

```
[root@nf3500a /root]# ipssend unattended 1 ?
Found 1 IBM ServeRAID Controller(s).
Set Unattended Mode has been initiated for controller 1...
Unattended Mode is set Off.
```

Figure 123. Executing `ipssend unattended 1 ?`

If you want to set the unattended mode to ON, execute this command:

```
ipssend unattended 1 on
```

You will see output similar to Figure 124.

```
[root@nf3500a /root]# ipssend unattended 1 on
Found 1 IBM ServeRAID Controller(s).
Set Unattended Mode has been initiated for controller 1...
Command Completed Successfully.
```

Figure 124. Executing `ipssend unattended 1 on`

#### 4.2.8 The rebuild command

This command starts a rebuild to the designated drive. The `rebuild` command has the following syntax:

```
ipssend rebuild <Controller> <Channel> <SCSI ID> <New Channel> <New SCSI ID>
```

The parameters are explained in Table 11.

Table 11. Rebuild command parameters

Parameter	Description
Controller	Number of controller (1 to 12)
Channel	Channel of defunct drive (1 to 3)
SCSI ID	SCSI ID of defunct drive (0 to 15)
New Channel	Channel of new drive (1 to 3)
New SCSI ID	SCSI ID of new drive (0 to 15)

This operation is valid for disk arrays containing one or more logical drives in a Critical (CRT) state. For example, if you want to rebuild a defunct drive on SCSI ID 2 on channel 1 on the first ServeRAID controller to a new drive on SCSI ID 0 on the same channel, you will execute this command:

```
ipssend rebuild 1 1 2 1 0
```

You will see output similar to Figure 125.

```
[root@nf3500a /root]# ipssend rebuild 1 1 2 1 0

Found 1 IBM ServeRAID Controller(s).
Rebuild Drive has been initiated for controller 1...
Rebuilding Logical Drive #1:
.....10% Done
.....20% Done
.....30% Done
.....40% Done
.....50% Done
.....60% Done
.....70% Done
.....80% Done
.....90% Done
.....Done Logical Drive #1
Rebuilding Logical Drive #2:
.....10% Done
.....20% Done
```

Figure 125. Executing `ipssend rebuild 1 1 2 1 0`

---

### 4.3 Replacing a defunct drive

When a physical drive in a RAID array becomes defunct you will see a light signal on the drive. You can simulate a defunct drive by executing the following command:

```
ipssend setstate 1 1 3 ddd
```

In this case we are simulating that the drive with SCSI ID 3 on channel 1 on the first ServeRAID controller is defunct. The following steps should be taken to replace the defunct drive:

1. Physically replace the defunct drive with a good drive.
2. The IBM ServeRAID controller will start rebuilding the drive automatically.

**Note**

Automatically rebuilding will work only on ServeRAID II and III. And Enable Hot Spare Rebuild must be set to Enabled!

You can check the progress of rebuilding the logical drives on the first IBM ServeRAID controller with this command:

```
ipssend getstatus 1
```

You will see output similar to Figure 117 on page 105.

If the rebuild is not completed successfully, you will see output similar to Figure 126.

```
[root@nf3500a /root]# ipssend getstatus 1
Found 1 IBM ServeRAID Controller(s).
Background Command Progress Status for controller 1...
  Current/Most Recent Operation   : Rebuild
  Logical Drive in Progress       : 1
  Rebuild Rate                    : High
  Status                          : Drive Failed
    Channel Number is             : 1
    SCSI ID Number is             : 0
  Logical Drive Size (in Stripes) : 128000
  Number of Remaining Stripes     : 89562
  Percentage Complete              : 30.03%
Command Completed Successfully.
```

Figure 126. Failed rebuild

### 4.3.1 Replacing a defunct drive with disabled Hot Spare Rebuild

When you have disabled the Hot Spare Rebuild function in the IBM ServeRAID controller configuration, the following steps should be taken to replace the defunct drive. In our example, the drive with SCSI ID 1 on channel 1 on the first ServeRAID controller is defunct.

1. Physically replace the defunct drive with a working one.
2. Execute the following command to start rebuilding the drive:

```
ipssend setstate 1 1 3 rbl
```

You will see output similar to this:

```
[root@nf3500a /root]# ipssend setstate 1 1 3 rbl
Found 1 IBM ServeRAID Controller(s).
Set Device State has been initiated for Controller 1...
Command Completed Successfully.
```

Figure 127. Forced rebuild of the defunct drive

You can check the progress of rebuilding the logical drives on the first IBM ServeRAID controller with this command:

```
ipssend getstatus 1
```

You will see output similar to Figure 117 on page 105.

### 4.3.2 Replacing a defunct drive with a hot spare drive installed

When you have configured the hot spare drive in your IBM ServeRAID configuration, the defunct physical drive is automatically rebuilt to the hot spare drive. Follow these steps to replace the defunct physical drive and set it as a hot spare drive:

1. You find out that there is a defunct physical drive in your RAID array on the first ServeRAID controller. In our example the physical drive on SCSI ID 2 on channel 1 was defined as a hot spare drive. You can check this by executing the command:

```
ipssend getconfig 1 pd
```

You will see output similar to Figure 128.



```

[root@nf3500a /root]# ipssend getconfig 1 pd
Found 1 IBM ServeRAID Controller(s).
Read Configuration has been initiated for Controller 1...
-----
Physical Device Information
-----
Channel #1:
Initiator at SCSI ID 7
Target on SCSI ID 0
Device is a 16 bit, Fast SCSI, tag queuing Hard Disk
SCSI ID : 0
PFA (Yes/No) : No
State : Online (ONL)
Size (in MB)/(in Sectors): 8678/17773888
Device ID : IBM-PSG ST39175L04303AL0A27C
Target on SCSI ID 1
Device is a 16 bit, Fast SCSI, tag queuing Hard Disk
SCSI ID : 1
PFA (Yes/No) : No
State : Online (ONL)
Size (in MB)/(in Sectors): 8678/17773888
Device ID : IBM-PSG ST39175L04303AL09YSS
Target on SCSI ID 2
Device is a 16 bit, Fast SCSI, tag queuing Hard Disk
SCSI ID : 2
PFA (Yes/No) : No
State : Rebuild (RBL)
Size (in MB)/(in Sectors): 8678/17773888
Device ID : IBM-PSG ST39175L04303AL0A2QK
Target on SCSI ID 3
Device is a 16 bit, Fast SCSI, tag queuing Hard Disk
SCSI ID : 3
PFA (Yes/No) : No
State : Defunct Hot Spare (DHS)
Size (in MB)/(in Sectors): 8678/17773888
Device ID : IBM-PSG DNES-309SAHRAJLJ6230
Target on SCSI ID 15
Device is a 16 bit, Fast SCSI, tag queuing Processor Device
SCSI ID : 15
PFA (Yes/No) : No
State : Standby (SBY)
Size (in MB)/(in Sectors): 0/0
Device ID : IBM EXP200 10D792063452
Command Completed Successfully.

```

Figure 128. After failing the drive in RAID array

As you can see, the hot spare drive is already rebuilding and the defunct drive is in Defunct Hot Spare (DHS) state.

2. Remove the defunct drive from the server. In our example this is the drive with SCSI ID 3 on channel 1.
3. Set the state of the drive to Empty (EMP) with the command:

```
ipssend setstate 1 1 3 emp
```

You will see output similar to Figure 129.

```
[root@nf3500a /root]# ipssend setstate 1 1 3 emp
Found 1 IBM ServeRAID Controller(s).
Set Device State has been initiated for Controller 1...
Command Completed Successfully.
```

Figure 129. Setting the DHS to EMP

You can check the result of this operation by executing this command:

```
ipssend getconfig 1 pd
```

You will see output similar to Figure 130.

```
[root@nf3500a /root]# ipssend getconfig 1 pd
Found 1 IBM ServeRAID Controller(s).
Read Configuration has been initiated for Controller 1...
-----
Physical Device Information
-----
Channel #1:
  Initiator at SCSI ID 7
  Target on SCSI ID 0
    Device is a 16 bit, Fast SCSI, tag queuing Hard Disk
    SCSI ID      : 0
    PFA (Yes/No) : No
    State       : Online (ONL)
    Size (in MB)/(in Sectors): 8678/17773888
    Device ID   : IBM-PSG ST39175L04303AL0A27C
  Target on SCSI ID 1
    Device is a 16 bit, Fast SCSI, tag queuing Hard Disk
    SCSI ID      : 1
    PFA (Yes/No) : No
    State       : Online (ONL)
    Size (in MB)/(in Sectors): 8678/17773888
    Device ID   : IBM-PSG ST39175L04303AL09YSS
  Target on SCSI ID 2
    Device is a 16 bit, Fast SCSI, tag queuing Hard Disk
    SCSI ID      : 2
    PFA (Yes/No) : No
    State       : Rebuild (RBL)
    Size (in MB)/(in Sectors): 8678/17773888
    Device ID   : IBM-PSG ST39175L04303AL0A2QK
  Target on SCSI ID 15
    Device is a 16 bit, Fast SCSI, tag queuing Processor Device
    SCSI ID      : 15
    PFA (Yes/No) : No
    State       : Standby (SBY)
    Size (in MB)/(in Sectors): 0/0
    Device ID   : IBM EXP200 10D792063452
Command Completed Successfully.
```

Figure 130. After removing defunct drive

As you can see, there is no entry for the defunct drive anymore.

4. Insert a new drive into the server. In our example this will be the same location as the defunct drive.
5. Set the state of that drive to Ready (RDY) with this command:

```
ipssend setstate 1 1 3 rdy
```

You will see output similar to Figure 131.

```
[root@nf3500a /root]# ipssend setstate 1 1 3 rdy
Found 1 IBM ServeRAID Controller(s).
Set Device State has been initiated for Controller 1...
Command Completed Successfully.
```

*Figure 131. Setting the new drive state to RDY*

With setting the state to Ready (RDY) the drive is started.

**Note**

All new drives must first be set to ready (RDY).

You can check the result of this operation by executing this command:

```
ipssend getconfig 1 pd
```

You will see output similar to Figure 132.

```
[root@nf3500a /root]# ipssend getconfig 1 pd
Found 1 IBM ServeRAID Controller(s).
Read Configuration has been initiated for Controller 1...
-----
Physical Device Information
-----
Channel #1:
Initiator at SCSI ID 7
Target on SCSI ID 0
Device is a 16 bit, Fast SCSI, tag queuing Hard Disk
SCSI ID : 0
PFA (Yes/No) : No
State : Online (ONL)
Size (in MB)/(in Sectors): 8678/17773888
Device ID : IBM-PSG ST39175L04303AL0A27C
Target on SCSI ID 1
Device is a 16 bit, Fast SCSI, tag queuing Hard Disk
SCSI ID : 1
PFA (Yes/No) : No
State : Online (ONL)
Size (in MB)/(in Sectors): 8678/17773888
Device ID : IBM-PSG ST39175L04303AL09YSS
Target on SCSI ID 2
Device is a 16 bit, Fast SCSI, tag queuing Hard Disk
SCSI ID : 2
PFA (Yes/No) : No
State : Rebuild (RBL)
Size (in MB)/(in Sectors): 8678/17773888
Device ID : IBM-PSG ST39175L04303AL0A2QK
Target on SCSI ID 3
Device is a 16 bit, Fast SCSI, tag queuing Hard Disk
SCSI ID : 3
PFA (Yes/No) : No
State : Ready (RDY)
Size (in MB)/(in Sectors): 8678/17773888
Device ID : IBM-PSG DNES-309SAHRAJLJ6230
Target on SCSI ID 15
Device is a 16 bit, Fast SCSI, tag queuing Processor Device
SCSI ID : 15
PFA (Yes/No) : No
State : Standby (SBY)
Size (in MB)/(in Sectors): 0/0
Device ID : IBM EXP200 10D792063452
Command Completed Successfully.
```

Figure 132. After setting the state to RDY

As you can see, the new drive appears as a Ready (RDY) device, in our example under SCSI ID 3 on channel 1.

- 6. Change the state of the new drive to Hot Spare (HSP) with this command:

```
ipssend setstate 1 1 3 hsp
```

You will see output similar to Figure 133.

```
[root@nf3500a /root]# ipssend setstate 1 1 3 hsp
Found 1 IBM ServeRAID Controller(s).
Set Device State has been initiated for Controller 1...
Command Completed Successfully.
```

Figure 133. Changing the state to HSP

You can check the result of this operation by executing this command:

```
ipssend getconfig 1 pd
```

You will see output similar to Figure 134.

```
[root@nf3500a /root]# ipssend getconfig 1 pd
Found 1 IBM ServeRAID Controller(s).
Read Configuration has been initiated for Controller 1...
-----
Physical Device Information
-----
Channel #1:
Initiator at SCSI ID 7
Target on SCSI ID 0
  Device is a 16 bit, Fast SCSI, tag queuing Hard Disk
  SCSI ID      : 0
  PFA (Yes/No) : No
  State        : Online (ONL)
  Size (in MB)/(in Sectors): 8678/17773888
  Device ID    : IBM-PSG ST39175L04303AL0A27C
Target on SCSI ID 1
  Device is a 16 bit, Fast SCSI, tag queuing Hard Disk
  SCSI ID      : 1
  PFA (Yes/No) : No
  State        : Online (ONL)
  Size (in MB)/(in Sectors): 8678/17773888
  Device ID    : IBM-PSG ST39175L04303AL09YSS
Target on SCSI ID 2
  Device is a 16 bit, Fast SCSI, tag queuing Hard Disk
  SCSI ID      : 2
  PFA (Yes/No) : No
  State        : Online (ONL)
  Size (in MB)/(in Sectors): 8678/17773888
  Device ID    : IBM-PSG ST39175L04303AL0A20K
Target on SCSI ID 3
  Device is a 16 bit, Fast SCSI, tag queuing Hard Disk
  SCSI ID      : 3
  PFA (Yes/No) : No
  State        : Hot Spare (HSP)
  Size (in MB)/(in Sectors): 8678/17773888
  Device ID    : IBM-PSG DNES-309SAHRAJLJ6230
Target on SCSI ID 15
  Device is a 16 bit, Fast SCSI, tag queuing Processor Device
  SCSI ID      : 15
  PFA (Yes/No) : No
  State        : Standby (SBY)
  Size (in MB)/(in Sectors): 0/0
  Device ID    : IBM      EXP200  10D792063452
Command Completed Successfully.
```

Figure 134. After setting the state to HSP

Congratulations! You have just installed a brand new the new hot spare drive and it is ready to use.

---

#### 4.4 Using the ServeRAID Manager utility

The ServeRAID manager for Linux offers you to manage your ServeRAID controller from Linux, without the need for the not-so-stable Windows control workstation. It is a Java-based tool and it has the same functionality across all supported platforms. With the ServeRAID manager for Linux you can manage the ServeRAID controller locally or remotely. That means that you can install it on the server with the ServeRAID controller and manage the controller in the server, or you can install it on a separate Linux box and manage the ServeRAID remotely.

**Note**

For remote management you also need to install the ServeRAID manager on the server with the ServeRAID controller, because the agent needed for remote management is included in the package. Also the server and management station have to be connected with TCP/IP.

After you get the file `RaidMan-4.40-03.i386.rpm` from the Web or from the CD, install it with the command:

```
rpm -ihv RaidMan-4.40-03.i386.rpm
```

**Note**

Before using this version of the ServeRAID manager software you must have BIOS/firmware and the driver for the controller on the same level.

During the installation you have the option to enable the background agent, which is then used for remote management. If you plan to manage the ServeRAID adapter remotely you should answer yes. If you answer yes the installation program will add the following line into the `/etc/inittab` file:

```
nfra:123456:once:/usr/RaidMan/RaidAgnt.sh #RaidMan
```

This will start the agent in every runlevel. The installation program also starts the agent right after installation, so you do not need to reboot to start using remote management, as in some other operating systems.

The ServeRAID manager is installed in the following directory:

```
/usr/RaidMan
```

The installation program also installs the necessary Java runtime. This Java runtime will not interfere with an already installed Java environment.

To start ServeRAID manager simply execute the following command in the X windows environment:

```
/usr/RaidMan/RaidMan.sh
```

**Note**

To use the ServeRAID manager, you have to have a working X Windows

During the program startup you will see the window similar to Figure 135.

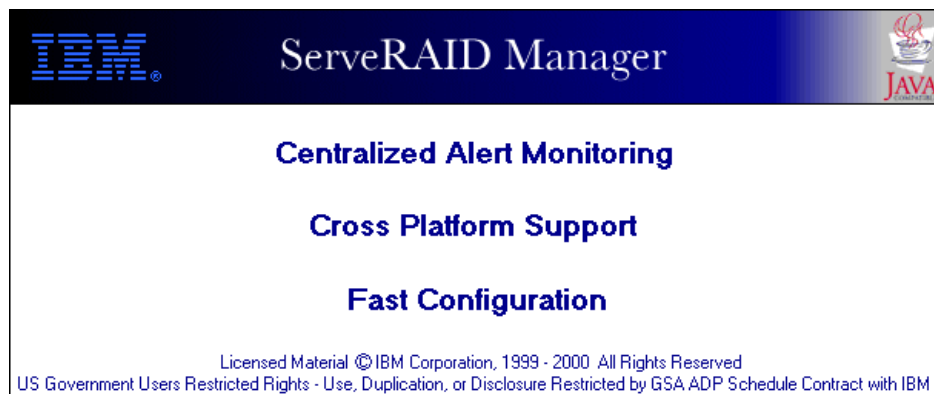


Figure 135. ServeRAID Manager startup

After the program is started you will see the window similar to Figure 136.

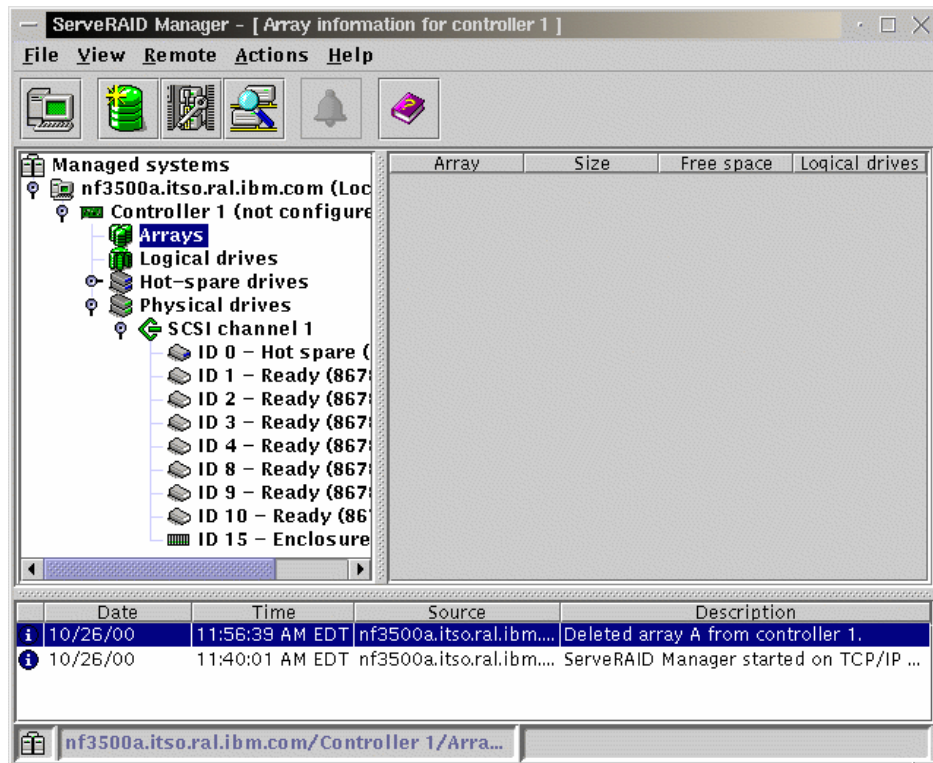


Figure 136. ServeRAID Manager

As you can see the window is divided into several areas:

- Menus - in the menus you can access all the functions available
- Icons - icons offers you shortcuts to the most often used functions
- Tree window - here you can see all the systems with the ServeRAID controller
- Info window - here you can see the information about arrays and logical drives
- Event log - all the events are displayed here.

**Note**

Instructions on how to use the ServeRAID manager functions can be found in the online help.



---

## 4.5 Remote management of the ServeRAID adapter

Your server with an installed ServeRAID controller can also be managed remotely. For this you need to do the following:

1. Install the ServeRAID manager on the server where the ServeRAID adapter is installed as we described in 4.4, “Using the ServeRAID Manager utility” on page 120. Do not forget to enable the ServeRAID manager agent running as a service at boot time.
2. Install the ServeRAID manager on the Linux workstation with the TCP/IP connection to the server you would like to manage.

Before you can start using the remote management function of the ServeRAID manager for Linux, you need to change the security settings on the server you would like to manage. If the server has a properly configured X Windows environment you can locally start the ServeRAID manager and update the security information. If you do not use X Windows on your server you need to follow these steps to properly enable access to your server. By default the security is enabled after the installation of the ServeRAID manager.

1. On the Linux workstation with the ServeRAID manager installed start the manager with the command:

```
/usr/RaidMan/RaidMan.sh
```

2. From the Actions menu select **Configure ServeRAID agent->Security** as you can see in Figure 137.

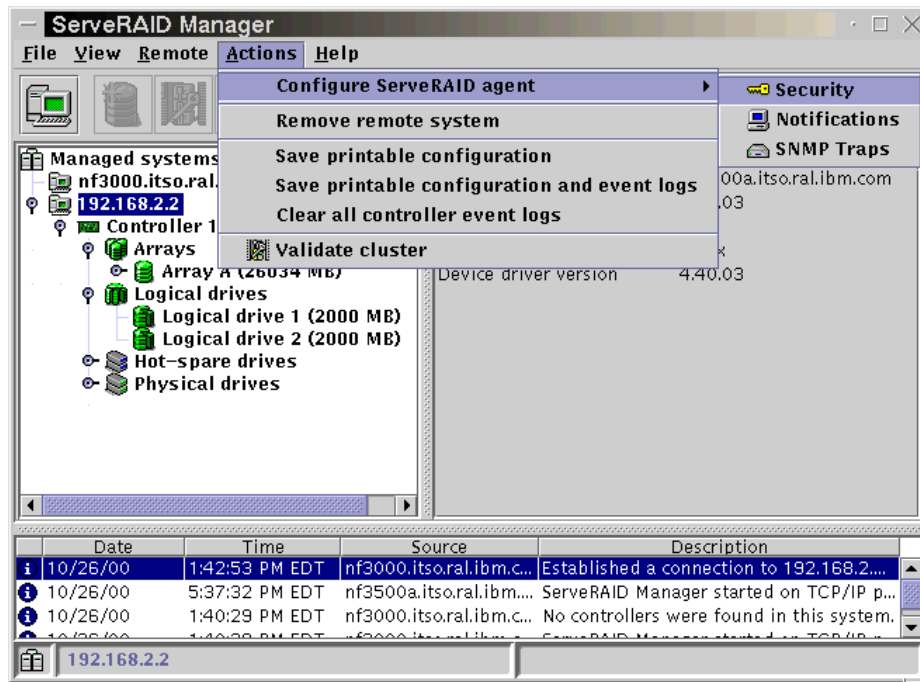


Figure 137. Starting security configuration

You will see a window similar to Figure 138.

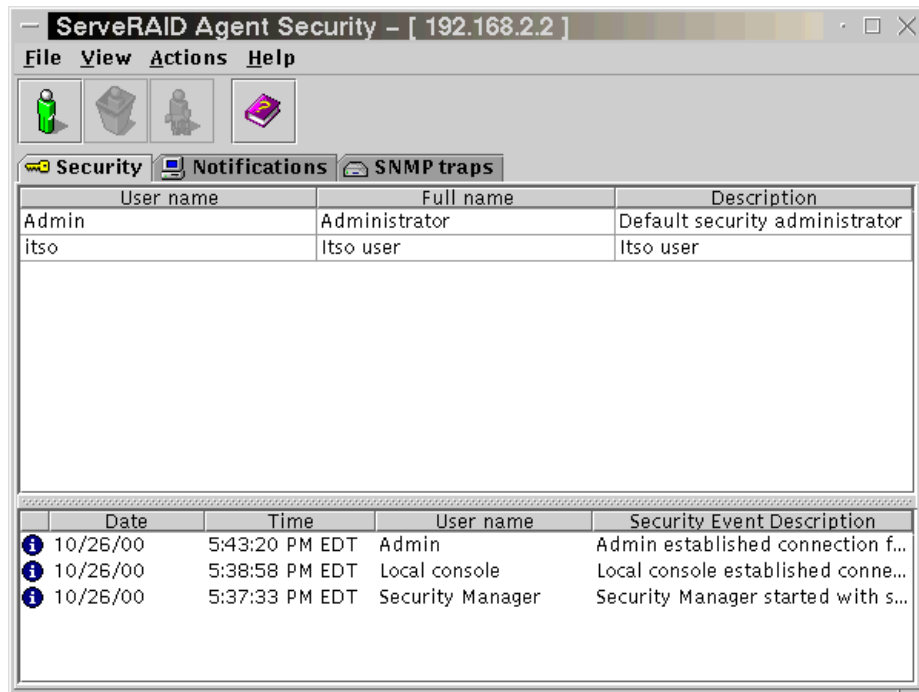


Figure 138. Security window

3. Double-click the username **Admin** (this is a built-in user, which cannot be removed) and you will see a window similar to Figure 139.



Figure 139. Changing the user

4. Type in the desired password and click **OK**.
5. Close the ServeRAID manager.

6. Install the ServeRAID manager on the server to be managed. Copy the file `/usr/RaidMan/RaidSLst.ser` from the workstation to the server.

**Note**

The directory on the server has to be the same as the ones on the workstation.

7. With the command:

```
ps ax | grep jre*
```

find all `jre` processes and kill them. You also need to kill the ServeRAID manager agent. You find the process ID with the command:

```
ps ax | grep RaidAgnt*
```

8. Use this command from the command prompt to restart the ServeRAID agent:

```
/usr/RaidMan/RaidMan.sh &
```

Congratulations! Your server is now ready for remote ServeRAID management.

You can connect to a remote server from a management workstation by selecting **Remote ->Add remote system** as you can see in Figure 140.

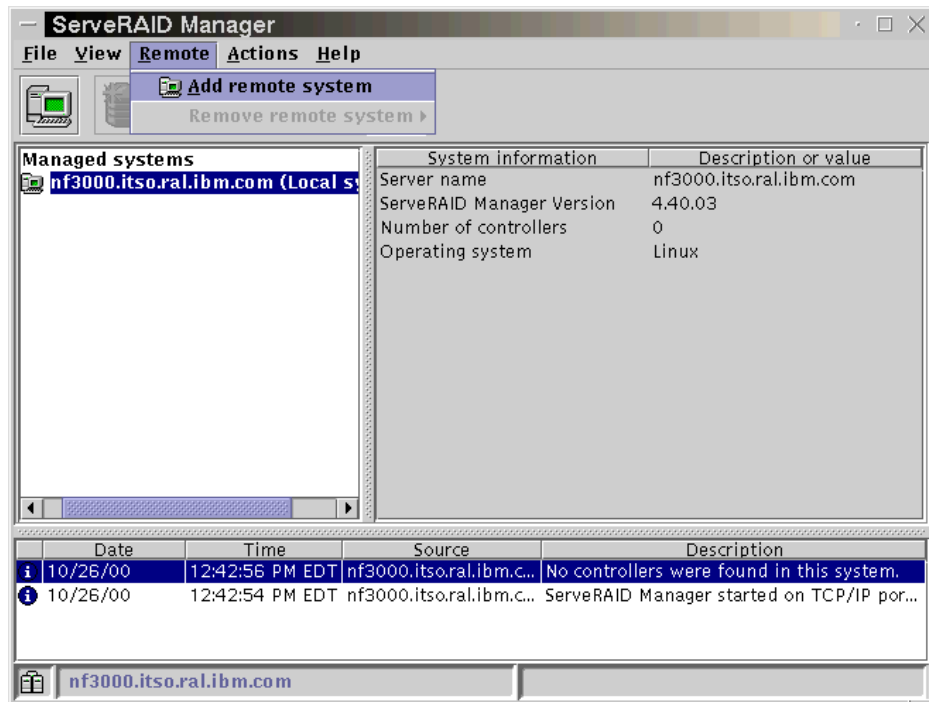


Figure 140. Accessing the remote server

You will see a window similar to Figure 141.

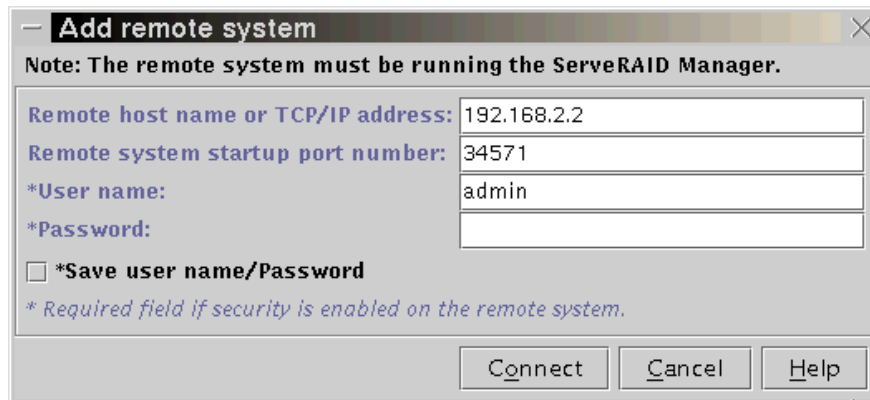


Figure 141. Remote system

Type in the necessary data and click **Connect**. After the system is connected you will a window similar to Figure 142.

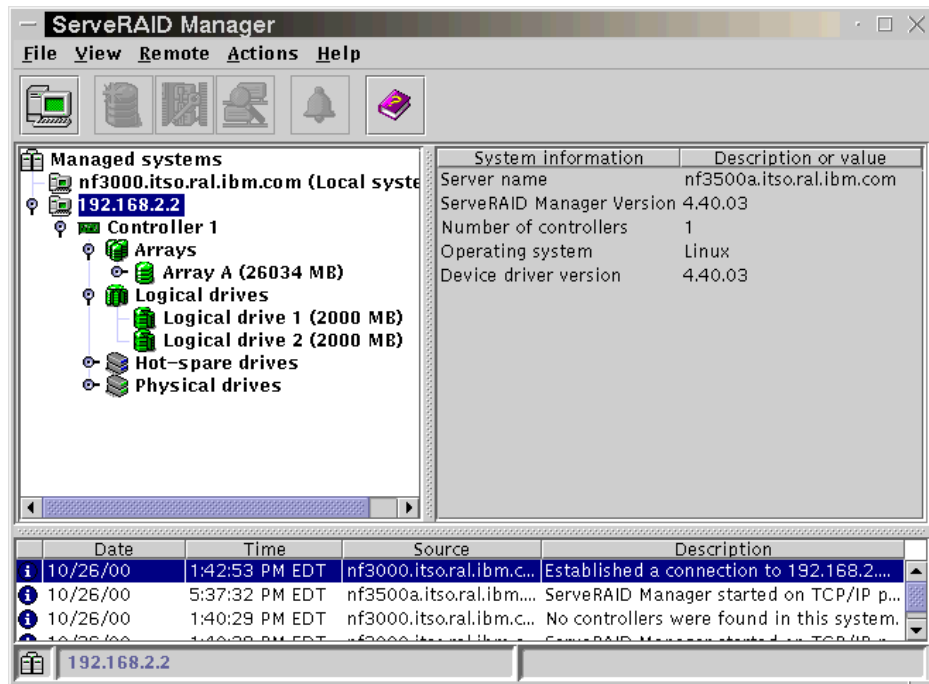


Figure 142. Remote system after connection

Now you can start managing the remote ServeRAID adapter.

---

## Chapter 5. DNS - Domain Name System

If you connect two or more computers to a network, they can share information and resources. However, these computers need to “talk in the same language” to be able to establish a connection. This “language” is called a network protocol. Today, the most popular communication protocol is TCP/IP. This is the protocol that is being used on the Internet and in many local area networks.

Hosts in a TCP/IP network communicate with each other by using unique IP addresses. These addresses consist of four 8-bit numbers (octets) that are divided by dots. For example, host A has the address 192.168.99.1, while host B uses 122.68.29.5.

However, this addressing scheme is not very comprehensible to human beings and it is almost impossible to memorize a number of hosts by their IP addresses. Therefore a naming scheme has been invented.

Each host has a host name (for example fred) and belongs to a certain domain (for example snake-oil.com). Domains can be organized in a hierarchical fashion and can consist of different subdomains (for example marketing.snake-oil.com). The combination of a host name and its domain name is called a fully qualified domain name (FQDN) (for example fred.marketing.snake-oil.com). Since domains are hierarchical, it is possible to have more hosts with the same host name in different subdomains. Therefore, fred.marketing.snake-oil.com can be a different host from fred.management.snake-oil.com. If you want these hosts to be addressable from the Internet, you need to register your domain name with a central registry. There are several top-level domains, such as .com, .org or .net. In addition to these generic top-level domains, each country in the world has its own country code as the top-level domain. For example, Germany has .de, Denmark has .dk, and Finland uses .fi.

Since the hosts internally still use their IP addresses to communicate, there needs to be a mapping between host names and the corresponding IP address. There are two ways this can be implemented.

All host names of a network, including their IP addresses, are put into a static text file. This file has to be copied on each host that wants to communicate with the others by name. As soon as a host has been added or removed from the network, or an IP address or host name has changed, and the host files on all computers have to be adjusted accordingly. This can get very tedious, if the number of hosts is large.

This is where the Domain Name System (DNS) comes in. The following description of DNS is very simplified, but it should give you a rough picture of what DNS is all about.

Instead of maintaining a separate host file on each machine, there is a central server that carries a list of all hosts and IP addresses of its domain. All clients now send their host name resolution request to this central server instead of looking in a local table. The name server will look up the requested host name and return the respective IP address. The opposite is also possible: the client can also ask for a host name that belongs to a certain IP address. If a client asks for an IP address of another domain, the local domain name server will forward the request to the next name server above in its hierarchy, if it cannot answer the request by itself. Therefore changes to the table of host names have to be made at one central point only rather than on all participants of the network.

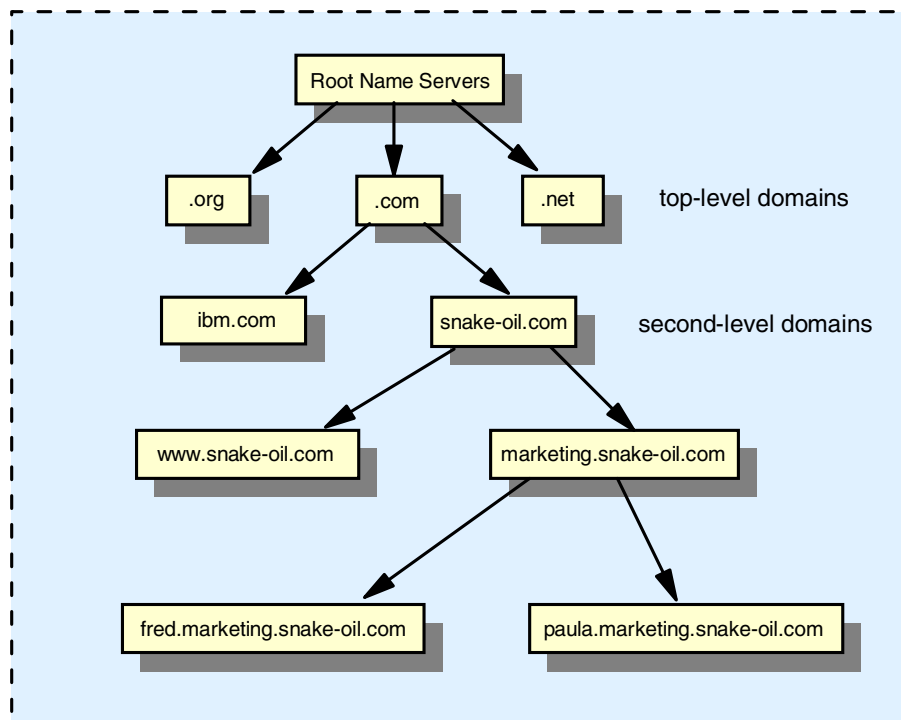


Figure 143. Internet domain hierarchy

This chapter will describe how to set up a name server for a local domain and how to maintain a host list for this domain.



---

## 5.1 Installation of software

The server that will be the DNS server needs to have a working TCP/IP network connection to the other hosts in its network before we start. The program that is responsible for this service is called *named* and belongs to the software package *bind*, which is maintained by Paul Vixie for The Internet Software Consortium. There are two major versions of *bind*: *bind4* and *bind8*. We will focus on the new version *bind8*, because it is more secure and is designed to replace *bind4* in the future. Most Linux distributions already contain a precompiled and preconfigured package for *bind8*.

### Note

The package *bind8* has been split up into two separate packages in TurboLinux 6.0: *bind*, which contains the actual server program, and *bindutil*, which contains the utilities such as *nslookup*, *dig* and *host*. We recommend that you install both on the server. A client machine only needs the *bindutil* package.

Make sure that the *bind* package is actually installed. In TurboLinux, you can use the RPM package manager to query the database of installed packages by entering the following command:

```
rpm -q bind
rpm -q bind-utils
```

If the packages are not installed, you can mount the main TurboLinux CD and install them with the following commands:

```
mount /nt/cdrom
cd /mnt/cdrom/TurboLinux/RPMS
rpm -Uhv bind-8.2.2P5-3.i386.rpm
rpm -Uhv bind-utils.8.2.2P5-3.i386.rpm
```

---

## 5.2 DNS sample configuration

Configuring DNS can be very complex, depending on the intended functionality. Covering this in depth is beyond the scope of this chapter. We will therefore focus on very a simplified example and recommend that you take a look at the very informative DNS how-to at:

<http://www.linuxdoc.org/HOWTO/DNS-HOWTO.html>

or at `/usr/share/doc/howto/en/DNS-HOWTO.gz` on your local file system for further information on DNS and bind.

Although TurboLinux has a graphical DNS configuration tool as part of `turbonetcfg`, we will first create a configuration from scratch as a means of better explaining how each file functions. After the configuration is complete, you can view it graphically with `turbonetcfg` to see how our example is represented in the TurboLinux DNS configuration tool.

We will construct a simple example: The company Snake Oil Ltd. wants to set up a local DNS server for their internal network (the internal IP address range is 192.168.99.xxx/24, a Class C network). They chose `snake-oil.com` as their local domain name. The network is also connected to the Internet. The name server will be configured to answer all requests about the local (internal) `snake-oil.com` domain and forward all other requests to the ISP's name server (`ns.bigisp.com`, fictional IP address 155.3.12.1) as a caching name server.

We begin with a simple example. At first the local DNS will be configured to act as a caching-only name server. This means that it forwards all requests to the ISP's name server(s) (forwarders) and caches all answers for further requests from its clients. This reduces the network traffic on the outside line.

Put the following lines in the `/etc/resolv.conf` file:

```
search snake-oil.com
nameserver 127.0.0.1
```

This will make sure that the server itself will use its local name server for host name resolution.

In SuSE Linux, you can use YaST to modify this entry. Choose **System administration -> Network configuration -> Configuration nameserver**. Enter the IP address 127.0.0.1 and your domain. To enter this dialog directly from the command line, enter the following command:

```
yast --mask nameserver --autoexit
```

The name server's main configuration file is `/etc/named.conf`. Most distributions ship with a very detailed example configuration file; you might want to save this for future reference. We will create a new file from scratch. Open up a text editor and create a new `/etc/named.conf` according to Figure 144:

```

options {
    directory "/var/named";
    pid-file "/var/named/slave/named.pid";
    listen-on { any; };
    forward only;
    forwarders { 155.3.12.1; };
    sortlist {
        { localhost; localnets; };
        { localnets; };
    };
};

logging {
    category lame-servers { null; };
    category cname { null; };
};

zone "localhost" IN {
    type master;
    file "localhost.zone";
    check-names fail;
    allow-update { none; };
};

zone "0.0.127.in-addr.arpa" IN {
    type master;
    file "127.0.0.zone";
    check-names fail;
    allow-update { none; };
};

```

Figure 144. *Named.conf* text file

Replace the IP address in the `forwarders` field with your ISP's name server IP address.

You also need to create the following `/var/named/localhost.zone` file:

```

$ORIGIN localhost.
@           1D IN SOA      @ root (
                    42           ; serial (d. adams)
                    3H           ; refresh
                    15M          ; retry
                    1W           ; expiry
                    1D )         ; minimum

                    1D IN NS      @
                    1D IN A      127.0.0.1

```

Figure 145. *localhost.zone* text file

Create the file `/var/named/127.0.0.zone` with the following content:

```

$ORIGIN 0.0.127.in-addr.arpa.

@                1D IN SOA      localhost. root.localhost. (
                    42          ; serial (d. adams)
                    3H          ; refresh
                    15M         ; retry
                    1W          ; expiry
                    1D )        ; minimum

1                1D IN NS      localhost.
1                1D IN PTR     localhost.

```

Figure 146. The 127.0.0. zone text file

Your network clients should all be configured to query the local DNS server's IP address instead of your ISP's name server.

You can now start the server with the command:

```
/etc/rc.d/initd/named start
```

Check `/var/log/messages` for the startup messages. The name server should now resolve DNS queries from its clients by forwarding them to the ISP's name server. You can verify this with the commands `host <somehostname>` and `nslookup`.

If you want the name server to be started at the next system reboot, run `turboservice`, choose **Advanced**, and marked the name as enabled for the runlevel. The server boots by default.

In the following step, we will configure the server to act as a primary name server for the local domain `snake-oil.com`. Stop the name server with the command `/etc/rc.d/init.d/named stop` and edit the file `/etc/named.conf` so that it looks like the following example:

```

options {
    directory "/var/named";
    pid-file "/var/named/slave/named.pid";
    listen-on { any; };
    forward only;
    forwarders {9.24.106.15;};
    sortlist {
        { localhost; localnets; };
        { localnets; };
    };
};

logging {
    category lame-servers { null; };
    category cname { null; };
};

zone "." {
    type hint;
    file "root.hint";
};

zone "localhost" IN {
    type master;
    file "localhost.zone";
    check-names fail;
    allow-update { none; };
};

zone "0.0.127.in-addr.arpa" IN {
    type master;
    file "127.0.0.zone";
    check-names fail;
    allow-update { none; };
};

zone "snake-oil.com" {
    type master;
    file "snake-oil.zone";
};

zone "99.168.192.IN-ADDR.APRA" {
    type master;
    file "snake-oil.rev";
};

```

Figure 147. Name.conf text file

We have now added the zone files (the databases) needed for our local domain "snake-oil.com". The file /var/named/snake-oil.zone (Figure 148) is responsible for the mapping of host names to IP addresses.

```

; Zone file for snake-oil.com
;
@      IN      SOA      ns.snake-oil.com. hostmaster.snake-oil.com. (
199910011      ; serial, todays date + todays serial #
8H          ; refresh, seconds
2H          ; retry, seconds
1W          ; expire, seconds
1D )        ; minimum, seconds

;
;          NS      ns          ; Inet Address of name server
;          MX      10 mail     ; Primary Mail Exchanger
;          MX      20 mail.bigisp.com. ; Secondary Mail Exchanger

;
localhost      A      127.0.0.1
gw              A      192.168.99.1
ns              A      192.168.99.2
fred           A      192.168.99.3
mail           A      192.168.99.4
ftp            A      192.168.99.5
www            A      192.168.99.6

```

Figure 148. Snake-oil.com text file

You should also create the zone file `/var/named/snake-oil.rev`. This is necessary for reverse name lookups, for example, if you need to resolve an IP address to its host name.

The MX record in the zone file tells other hosts on the Internet what mail server services this domain. In our case, mail for `fred@snake-oil.com` will be relayed through `mail.snake-oil.com`, and as a backup, through `mail.bigisp.com`. The 10 and 20 in the second column signifies the priority of the mail servers, effectively providing redundancy.

```

@      IN      SOA      ns.snake-oil.com. hostmaster.snake-oli.com. (
199910011 ; Serial, todays date + todays serial
8H      ; Refresh
2H      ; Retry
1W      ; Expire
1D)    ; Minimum TTL

;          NS      ns.snake-oil.com.

1          PTR     gw.snake-oil.com.
2          PTR     ns.snake-oil.com.
3          PTR     fred.snake-oil.com.
4          PTR     mail.snake-oil.com.
5          PTR     ftp.snake-oil.com.
6          PTR     www.snake-oil.com.
.

```

Figure 149. Snake-oil.rev zone file

Now let the name server reload its configuration again by running `rndc restart`. Have a look at the messages in `/var/log/messages`. If everything went well, you should see messages similar to the following:

```
Oct 26 18:03:20 ns named[14870]: starting
Oct 26 18:03:20 ns named[14870]: cache zone "" (IN) loaded (serial 0)
Oct 26 18:03:20 ns named[14870]: master zone "localhost" (IN) loaded (serial 42)
Oct 26 18:03:20 ns named[14870]: master zone "0.0.127.in-addr.arpa" (IN) loaded serial 42)
Oct 26 18:03:20 ns named[14870]: master zone "snake-oil.com" (IN) loaded (serial 199910011)
Oct 26 18:03:20 ns named[14870]: master zone "99.168.192.IN-ADDR.APRA" (IN) load ed (se ial 199910011)
Oct 26 18:03:20 ns named[14870]: listening on [127.0.0.1].53 (lo)
Oct 26 18:03:20 ns named[14870]: listening on [9.24.105.210].53 (eth0)
Oct 26 18:03:20 ns named[14870]: Forwarding source address is [0.0.0.0].1041
Oct 26 18:03:20 ns named[14871]: Ready to answer queries.
```

Figure 150. `/var/log/messages` file

Your name server should now correctly resolve host names for the snake-oil domain as well.

---

### 5.3 Configuration tips

Use the `listen-on` directive in the options section of the `named.conf` file. For each interface a name server listens on, a pair of filehandles is opened. On a busy name server, saving every filehandle is a big win.

Check the `/var/log/messages` file from time to time for errors. Named is pretty verbose in its error messages.

If you are constantly adding, removing or just making modifications to your zone records, you might want to have a look at the `nsupdate` tool, which also belongs to the `bind8` package.





---

## Chapter 6. Samba

If you look at any English dictionary, Samba is defined as a Brazilian dance, but Samba in Linux is something completely different. Samba is an implementation of a Server Message Block (SMB) protocol server that can be run on almost every variant of UNIX in existence. Samba is an open source project, just like Linux. The entire code is written in C so it is easily ported to all flavors of UNIX. Samba is a tool for the peaceful coexistence of UNIX and Windows on the same network on the level of file and print sharing over the NetBIOS protocol. It allows UNIX systems to move into a Windows "Network Neighborhood" without causing a mess. With Samba, UNIX servers are acting like any other Windows server, offering their resources to the SMB clients. Recently SMB was renamed by Microsoft to Common Internet File System (CIFS).

---

### 6.1 What can you do with Samba?

- With Samba, a Linux server can act as a file/print server for Windows networks. It can replace expensive Windows NT file/print server in this role, creating a less expensive solution.
- Samba can act as a NetBIOS name server (NBNS) in a Windows world, where it is referred to as WINS - Windows Internet Name Service.
- Samba can participate in NetBIOS browsing and master browser elections.
- Samba can provide a gateway for synchronizing UNIX and Windows NT passwords.
- With Samba client software, you can access any shared directory or printer on Windows NT servers or Samba servers and allow UNIX machines to access Windows NT files.
- With Samba File System (SMBFS) you can mount any share from a Windows NT server or Samba server in your directory structure (this is available only on Linux).

---

### 6.2 Setting up the Samba server

To ensure that Samba is already installed, query the RPM database with the commands:

```
rpm -q samba
rpm -q samba-nsswitch
rpm -q smbfs
```

This command will return either the version number of the installed package or a message indicating the package is not installed. If any of these packages are not installed, mount the TurboLinux 6 main CD-ROM and install the following packages:

```
mount /mnt/cdrom
cd /mnt/cdrom/TurboLinux/RPMS
rpm -Uvh samba-2.0.6-20000313.i386.rpm
rpm -Uvh samba-nsswitch-2.0.6-20000313.i386.rpm
rpm -Uvh smbfs-2.0.6-20000313.i386.rpm
```

You might also want to install the Samba debugtools. They reside on the Companion CD, so are not installed during the initial install process. The following commands will unmount the main CD-ROM, mount the Companion CD, and install the debug tools package:

```
umount /mnt/cdrom
mount /mnt/cdrom
cd /mnt/cdrom/TurboContrib/RPMS
rpm -Uvh samba-debugtools-2.0.6-20000313.i386.rpm
```

### 6.2.1 Configuring the Samba server

In this section we will explain how to configure Samba so it can participate as a file/print server in an existing Windows network or at a stand-alone file/print server for Windows and Linux clients.

TurboLinux 6 does not provide a menu-driven program to configure Samba, so you must edit the configuration file (/etc/smb.conf) directly. This file is the heart of the Samba server. The Samba configuration file is divided into two main sections:

1. Global Settings - here you set up parameters that affect the connection parameters.
2. Share Definitions - here you define shares. A share is a directory on the server that is accessible over the network and shared among users. This section has three subsections:
  - a. Homes - in this subsection you define the user's home directories.
  - b. Printers - in this subsection you define the available printers.
  - c. Shares - this subsection can have more entries, one for each share you want to define.

In the following sections we will describe how to modify the smb.conf file to efficiently and simply use Samba as a file/print server. We explain only the

most necessary parameters. If you need more information, see the manual entry for the smb.conf file or the Samba project Web site at:

<http://www.samba.org>

You can find our sample smb.conf configuration file in Appendix C, "Sample smb.conf Samba configuration file" on page 347.

### 6.2.1.1 Setting the NetBIOS parameters

The NetBIOS parameters are part of the Global Section. When you open your smb.conf file you will see something similar to this:

```
#===== Global Settings =====  
[global]  
    netbios name = NF5000  
    workgroup = LINUX  
    server string = Samba Server on TurboLinux 6
```

The parameters are described in Table 12.

Table 12. NetBIOS parameters

Parameter	Description
netbios name	The Samba server is known by this name on the network. This parameter has the same meaning as the Windows NT computer name. If you do not specify anything it defaults to the server's host name.
workgroup	This parameter specifies in which Window NT domain or workgroup the Samba server will participate. It is equivalent to Windows NT domain or workgroup name.
server string	This is the description string of the Samba server. It has the same role as the Windows NT description field.

### 6.2.1.2 Global printing settings

In the smb.conf file you will see something similar to this:

```
load printers = yes  
printcap name = /etc/printcap  
printing = lprng
```

The parameters are described in Table 13.

Table 13. Printing parameters

Parameter	Description
load printers	This parameter controls if Samba loads all printers in the printcap file for browsing.
printcap name	With this parameter you tell Samba the location of the printcap file. The default value is /etc/printcap
printing	This parameter tells Samba what printing style to use on your server. TurboLinux by default uses the LPRNG printing style.

### 6.2.1.3 Global security settings

In your smb.conf file you will see something similar to this:

```
security = user
; password server = <NT-Server-Name>
encrypt passwords = yes
smb passwd file = /etc/samba.d/smbpasswd
```

The parameters are described in Table 14.

Table 14. Security parameters

Parameter	Description
security	This parameter has four possible values: share, user, server, domain
password server	In the case of server or domain security level this server is used for authorization. For the parameter value you use the server NetBIOS name.
encrypt passwords	By setting this parameter to yes, you enable Samba to use the Encrypted Password Protocol, which began being used in Microsoft Windows products with Windows NT Service Pack 3 and Windows 98. This is needed to communicate with those clients.
smb passwd file	This parameter tells Samba where encrypted passwords are saved.

The security modes are as follows:

- Share - for this security mode, clients only need to supply the password for the resource. This mode of security is the default for Windows 9.x file/print

server. It is not recommended for use in UNIX environments, because it violates the UNIX security scheme.

- User - the user/password validation is done on the server that is offering the resource. This mode is most widely used.
- Server - the user/password validation is done on the specified authentication server. This server can be a Windows NT server or another Samba server.
- Domain - this security level is basically the same as the server security level, with the exception that the Samba server becomes a member of a Windows NT domain. In this case the Samba server can also participate in such things as trust relationships.

Because Windows NT 4.0 Service Pack 3 or later, Windows 95 with the latest patches, and Windows 98 use the encrypted passwords for accessing NetBIOS resources, you need to enable your Samba server to use the encrypted passwords. Before you start the Samba server for the first time, you need to create a Samba encrypted passwords file. This can be done with the `mksmbpasswd` utility. The recommended way is to first create the user accounts in Linux and then create the Samba password file with the command:

```
cat /etc/passwd | /usr/bin/mksmbpasswd.sh > /etc/samba/smbpasswd
```

This creates the Samba password file from the Linux password file.

**Note**

Use the same filename you specified for creating the Samba password file in the `smb.conf` configuration to tell the Samba server where the password file is.

By default the passwords for the Samba users are undefined. Before any connection is made to the Samba server, users need to create their passwords.

Now you need to specify the password for all users. If you are changing or specifying a password for a user, you can do this by executing the command:

```
/usr/bin/smbpasswd -U username
```

You will see a window similar to Figure 151.

```
# /usr/bin/smbpasswd -U user
New SMB password:
Retype new SMB password:
Password changed for user user.
#
```

Figure 151. Specifying the password for Samba user

**Note**

Anyone with write access to /usr/bin/smbpasswd can change passwords for the Samba users.

Another way is to have each Samba user change the password for himself, by remotely connecting to the Samba server and executing the command:

```
/usr/bin/smbpasswd
```

The output will be similar to Figure 151. If a Samba user already has defined a password he will need to type the old password before he can change to a new password.

If you want to add a Samba server user later, this can be done with the following command:

```
/usr/bin/smbpasswd -a username password
```

This will add a new user to the Samba password file.

**Note**

You have to be logged on as root if you want to manage other users. If you are logged on as a user, you can only change your own password. The smbpasswd utility uses the location of the password file from the smb.conf configuration file.

#### 6.2.1.4 Global name resolution settings

In your smb.conf file you will see something similar to this:

```
name resolve order = wins lmhosts bcast
wins support = yes
; wins server = w.x.y.z
```

The parameters are described in Table 15.

Table 15. Name resolution parameters

Parameter	Description
<code>name resolve order</code>	With this parameter you specify how the Samba server resolves NetBIOS names into IP addresses. The preferred value is <code>wins lmhosts broadcast</code> . Refer to the manual page of the <code>smb.conf</code> file for more information.
<code>wins support</code>	If this option is enabled the Samba server will also act as a WINS server.
<code>wins server</code>	With this parameter you tell Samba which WINS server to use.

**Note**

Samba can act as a WINS server or a WINS client, but not both. So only one of the parameters (`wins support` or `wins server`) can be set at the same time. If you specify the IP address of WINS server, then `wins support` must be set to `no`.

### 6.2.1.5 Creating shares

In the previous section we explained how to prepare general configuration parameters. But a Samba server can be useful when you offer resources to the users. In this section we will explain how to create a share. The simple share section in the `smb.conf` file looks similar to this:

```
[redbook]
comment = Redbook files
path = /redbook
browseable = yes
printable = no
writable = yes
write list = @users
```

Table 16 describes the most important parameters for creating a share.

Table 16. Share parameters

Parameter	Description
<code>comment</code>	This describes the function of the share.
<code>admin users</code>	This parameter is used to specify the users who have administrative privileges for the share. When they access the share they perform all operations as root.

Parameter	Description
path	Defines the full path to the directory you are sharing.
browseable	If this parameter is set to yes, you can see the share when you are browsing the resources on the Samba server. The value can be yes or no.
printable	This parameter specifies if the share is a print share. The value can be yes or no.
write list	Users specified in this list have write access to the share. If the name begins with @ it means a group name.
writable	This parameter specifies if the share is writable. The value can be yes or no.
read list	Users specified in this list have read access to the share. If the name begins with @ it means a group name.
read only	If this is set to yes, share is read only. The value can be yes or no.
valid users	This parameter specifies which users can access the share.

By using these parameters you can easily set up a new share. Each share definition starts with the share name in brackets “[ ]”. Below this name you can specify the values for the share parameters.

#### 6.2.1.6 Share permissions

Although you can control the share permissions with share parameters, UNIX permissions are applied before the user can access files on the share. So you need to take care of UNIX permissions, so the user also has access to the shared directory under UNIX.

When a user creates a new file on the shared directory, the default create mask used is 0744. For directory creation, the default create mask is 0755. If you want, you can force a different creation mask. The parameters for doing this are explained in Table 17.

Table 17. Create mask parameters

Parameter	Description
create mask	This is used for file creation to mask against UNIX mask calculated from the DOS mode requested.
directory mask	This is used for directory creation to mask against UNIX mask calculated from the DOS mode requested.



### 6.2.1.7 Creating shares for home directories

For handling home directories Samba has a special share section called `[homes]`. This share definition is used for all home directories, so you do not need to create separate shares for each user.

When a client requests a connection to a file share, existing file shares are scanned. If a match is found, that share is used. If no match is found, the requested share is treated as a user name and validated by security. If the name exists and the password is correct, a share with that name is created by cloning the `[homes]` section. The home share definition uses the same parameters as a normal share definition. The following is an example of a home share definition in the `smb.conf` configuration file:

```
[homes]
comment = Home Directories
path = %H
valid users = %S
browseable = no
writable = yes
create mode = 0700
directory mode = 0700
```

As you can see, we used some variables in this definition, which are explained in Table 18.

Table 18. Variable description

Parameter	Description
<code>%H</code>	This variable represents the home directory of the user.
<code>%S</code>	The name of the current service, which is, in the case of home share, equal to username.

As you can see in the example, we used creation masks for the files and the directories in such a way that we forced all new files or directories to be accessible only by the owner of the home directory.

### 6.2.1.8 Creating a printer share

A Samba server uses the same procedure for printer shares as for the home shares. If all share definitions and user names are tested against the requested share name and the matched definition is still not found, Samba searches for a printer with that name (if the `[printers]` section exists). If the match is found in the printer definitions that `[printers]` share section is cloned with the name of the requested service, which is really a printer name. The following is an example of the printers definition in the `smb.conf` configuration file:

```
[printers]
comment = All Printers
path = /var/spool/samba
browseable = no
# Set public = yes to allow user 'guest account' to print
guest ok = no
writable = no
printable = yes
create mask = 0700
```

As you can see, the `[printers]` section is just another share definition, because when a user prints they basically copy the data into a spool directory, after that the data is handled by the local printing system. The only big difference between a printer share and other share definitions is that the `printable` parameter is set to “yes”. This means that a user can write a spool file to the directory specified under the share definition. If the share is printable, then it is also writable by default.

## 6.2.2 Starting and stopping the Samba server

You can start the Samba server by executing the command:

```
/etc/rc.d/init.d/smb start
```

Two daemons have now been started: `smbd` and `nmbd`. `smbd` is the actual Samba server and `nmbd` is the WINS server.

The Samba server can be stopped by executing the command:

```
/etc/rc.d/init.d/smb stop
```

Whenever you make modifications to the `smb.conf` configuration file, you must restart the Samba server.

## 6.2.3 Using SWAT

The Samba Web Administration Tool (SWAT) allows the remote configuration of the `smb.conf` configuration file through a Web browser. That means you can configure Samba in a GUI-like environment. SWAT itself is a small Web server and CGI scripting application, designed to run from `inetd`, provides access to the `smb.conf` configuration file.

An authorized user with the root password can configure the `smb.conf` configuration file via Web pages. SWAT also places help links to all configurable options on every page, which lets an administrator easily understand the effect of the changes.

Before using SWAT you must check the following:

1. In the `/etc/services` file you must have the following line:

```
swat 901/tcp
```

2. In the file `/etc/inetd.conf` you must have enable SWAT entry. To do this, remove the leading `#` on the following line with `swat`:

```
# swat stream tcp nowait.400 root /usr/sbin/swat swat
```

You can control who can access the SWAT service with the `/etc/hosts.deny` file.

Now you are ready to use SWAT. To start SWAT point your favorite Web browser to the Internet address of your Samba server on port 901:

```
http://localhost:901
```

After you load the home page of SWAT, you will see a window similar to Figure 152.

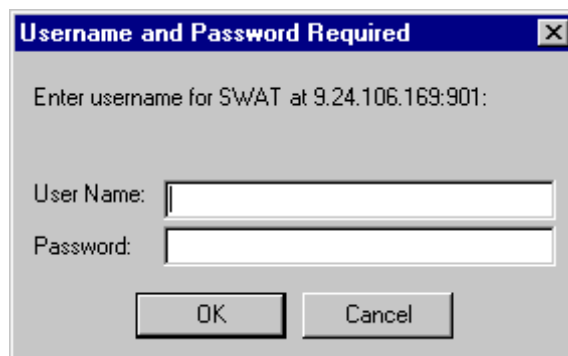


Figure 152. User authorization for SWAT

Type in the user name and password of the Linux user defined on your Linux server. Click **OK** to continue. You will see a window similar to Figure 153.

**Stop**

Any Linux user can access SWAT, but only a root user can make changes.

Remember, when you are logging on to SWAT from a remote machine, you are sending passwords in plain text. This can be a security issue, so we recommend that you do SWAT administration locally only.

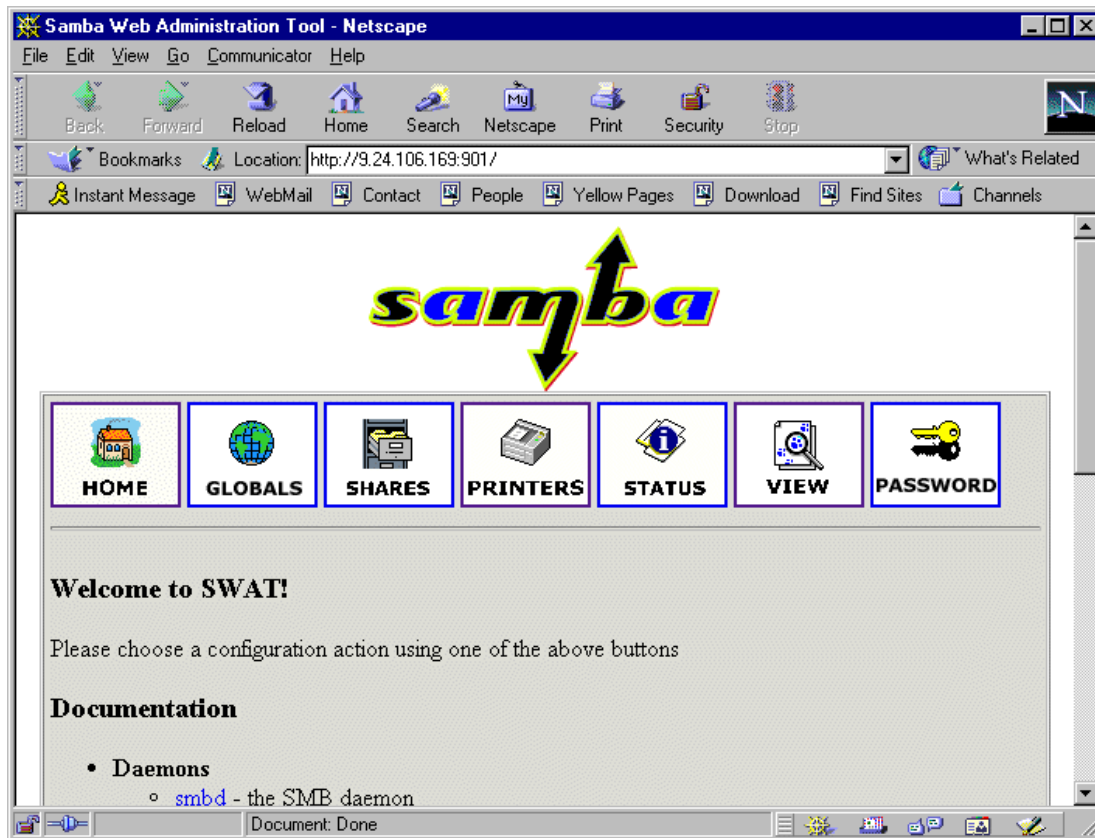


Figure 153. SWAT home page

As you can see in Figure 153, you have seven categories available:

1. Home - here you can find all the documentation you need about Samba.
2. Globals - here you can view and modify global parameters from the smb.conf configuration file.
3. Shares - here you can view, modify, and add shares.
4. Printers - here you can view, modify, and add printers.
5. Status - here you can check the current status of your Samba server.
6. View - here you can view current configuration of the smb.conf configuration file.
7. Passwords - here you can manage passwords for the Samba server.

Now we will briefly describe the functions available in SWAT.

**Note**

You can reach any of the seven functions on all SWAT Web pages. There are always icons for the functions on the top of each page.

After you make changes to smb.conf configuration file, the Samba server must be restarted.

**6.2.3.1 Globals**

When you click the **Globals** icon in the main SWAT window, and you will see a window similar to Figure 154.



Figure 154. Global section in SWAT

In this window you can modify the global parameters for the Samba server. By default you will see the Basic View; if you want to see the Advanced View click **Advanced View**. In the Advanced View you have all options available, while in the Basic View you can change only the basic options. To return from the Advanced View to the Basic View, click **Basic View**. After you have made your changes you can save them by clicking **Commit changes**. If you get a pop-up window similar to Figure 155, which warns you that you are sending non-secure information over the network, you can easily select **Continue** if you are working locally or if you know that your network is secure.

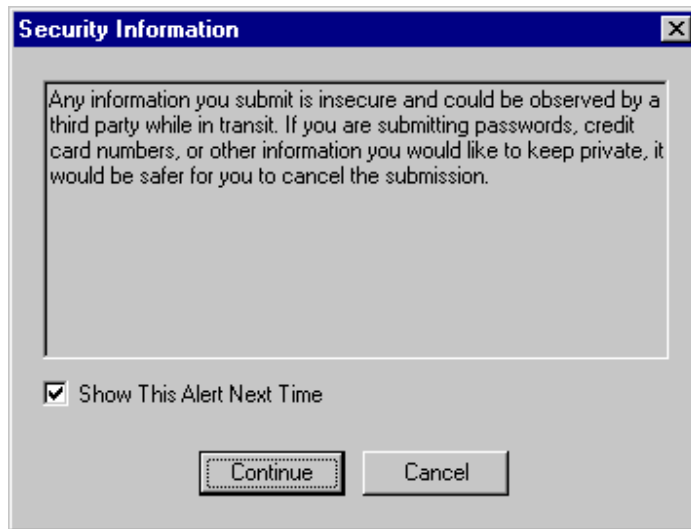


Figure 155. Security warning

### 6.2.3.2 Shares

When you click the **Shares** icon on any of the SWAT Web pages, you will see a window similar to Figure 156.

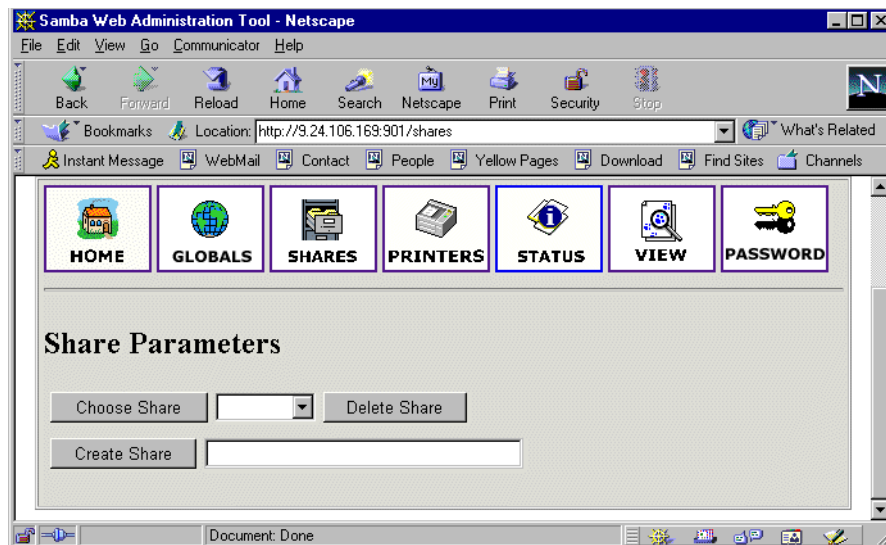


Figure 156. Shares section in SWAT

Here you can:

- View the defined share
- Delete share
- Create a new share

### 6.2.3.3 Viewing or modifying an existing share

To view an already defined share, select the share from the field to the right of the **Choose Share** button, similar to Figure 157.

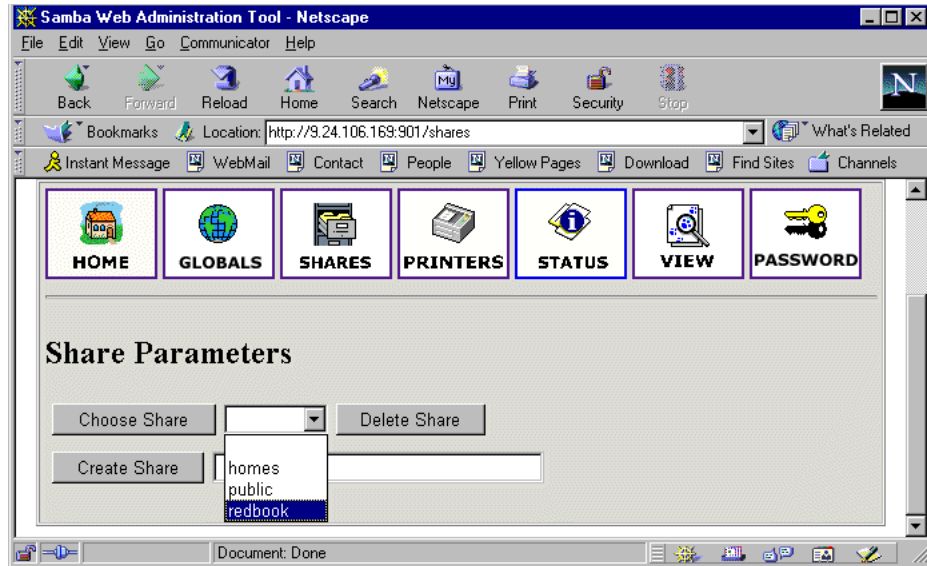


Figure 157. Choosing a share to view

After you have selected the share, click **Choose Share** to view the share properties. You will see a window similar to Figure 158.



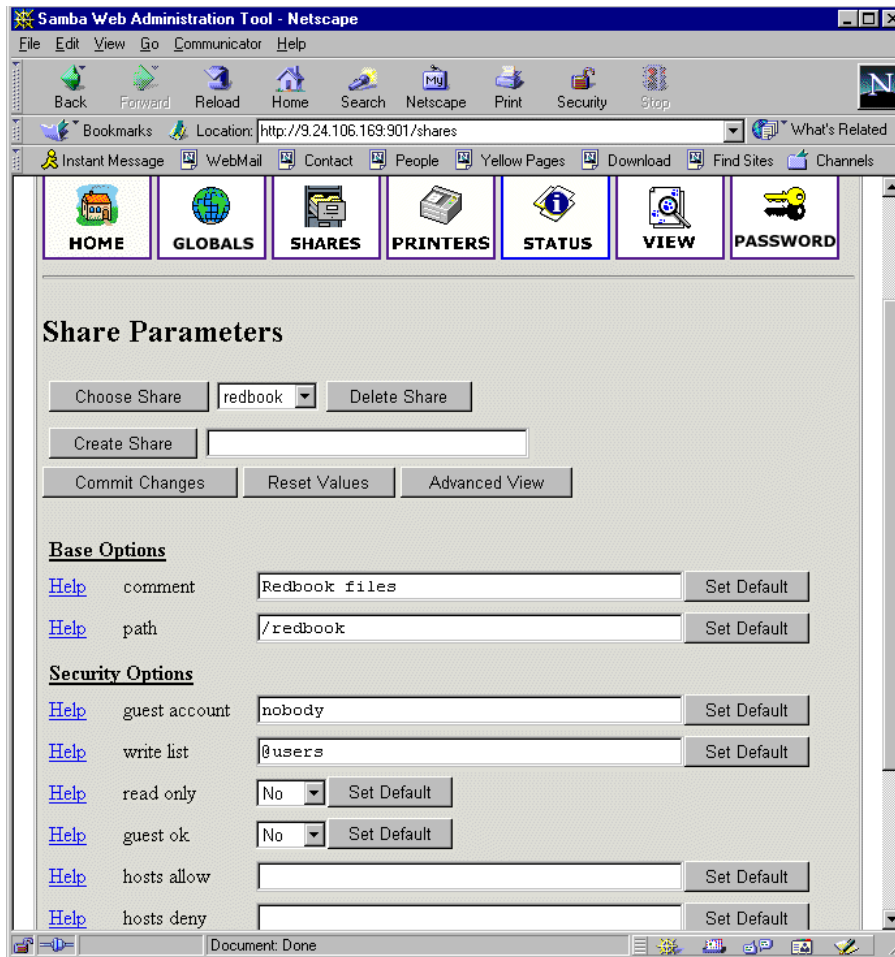


Figure 158. Share properties

If you want to view all available parameters, click **Advanced View**. In this view you can also make changes and save them by clicking **Commit Changes**.

#### 6.2.3.4 Deleting an existing share

To delete an existing share you must first select an already defined share similar to Figure 157. Then click **Delete Share**.

### Stop

Be aware that the share is deleted immediately and without warning.

After you have deleted the share, the Samba server must be restarted.

#### 6.2.3.5 Creating a new share

To create a simple share, follow these steps:

1. Create a directory that will be used for the share. You can do this by executing this command from the terminal:

```
mkdir /home/public
```

In our example we created a “public” directory in the “home” directory.

2. Make sure that the UNIX permissions are set correctly in that directory, so that only intended users have access rights to it.
3. In the shares view of the SWAT Web pages, type in the name of the share you are creating, similar to Figure 159.

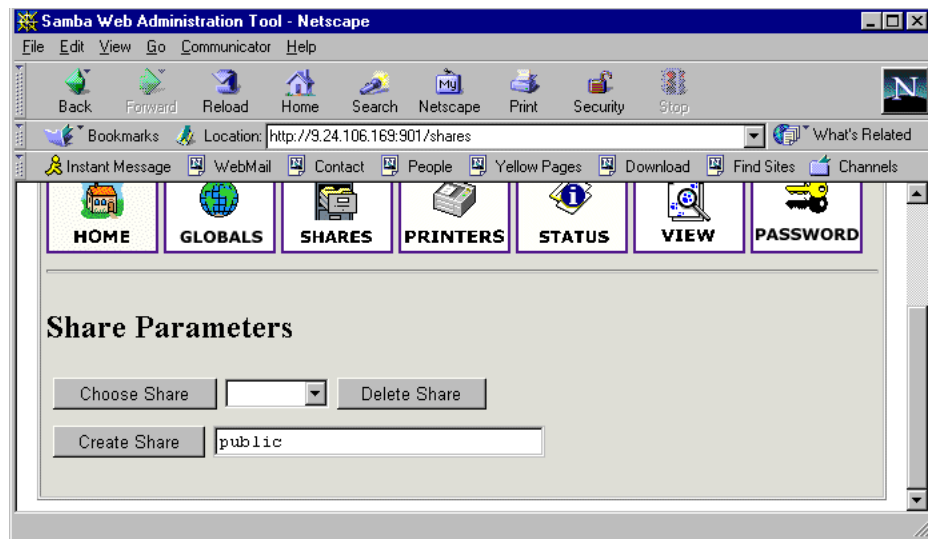


Figure 159. Entering the name for a new share

4. Click **Create Share** to continue, and you will see a window similar to Figure 160.

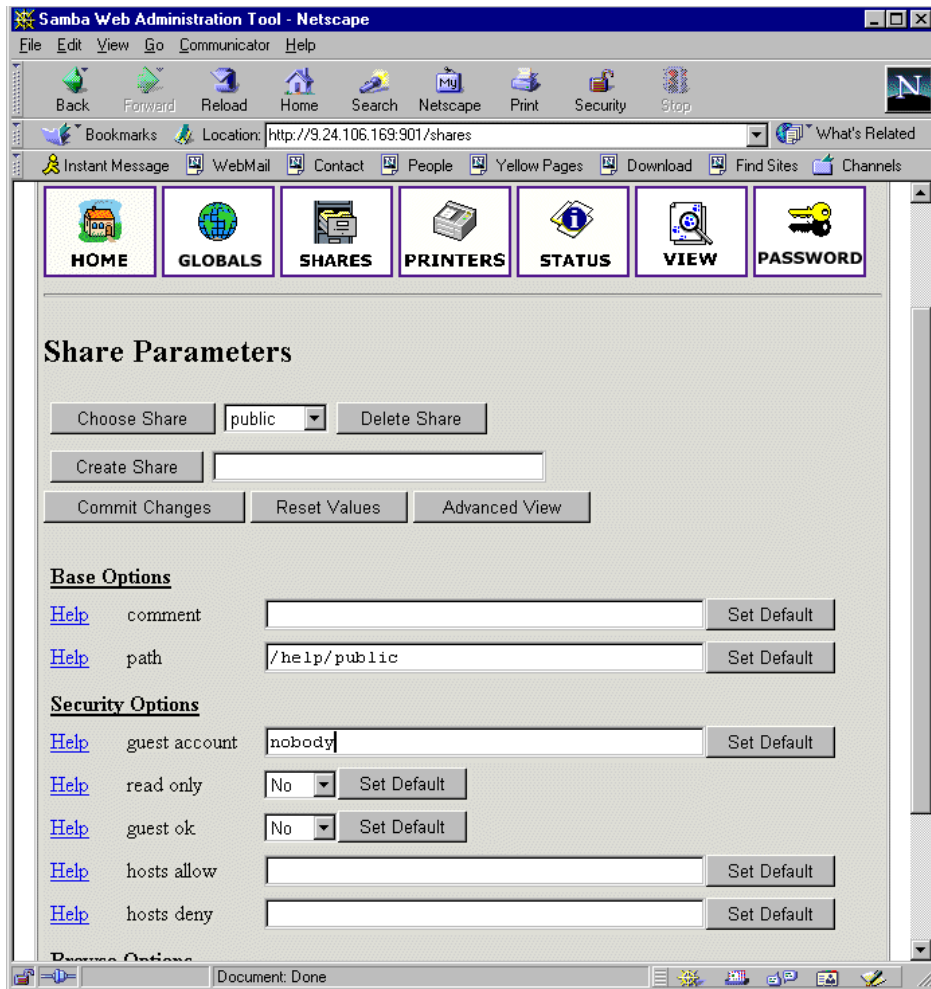


Figure 160. Entering the new share parameters

5. Fill in the needed parameters. If you need to set more advanced parameters, click **Advanced View** and you will see all available parameters. After you typed in all you want, click **Commit Changes** to save your new share.
6. You can see the changed smb.conf configuration file by selecting the **View** icon from the SWAT Web pages. You will see a window similar to Figure 161.

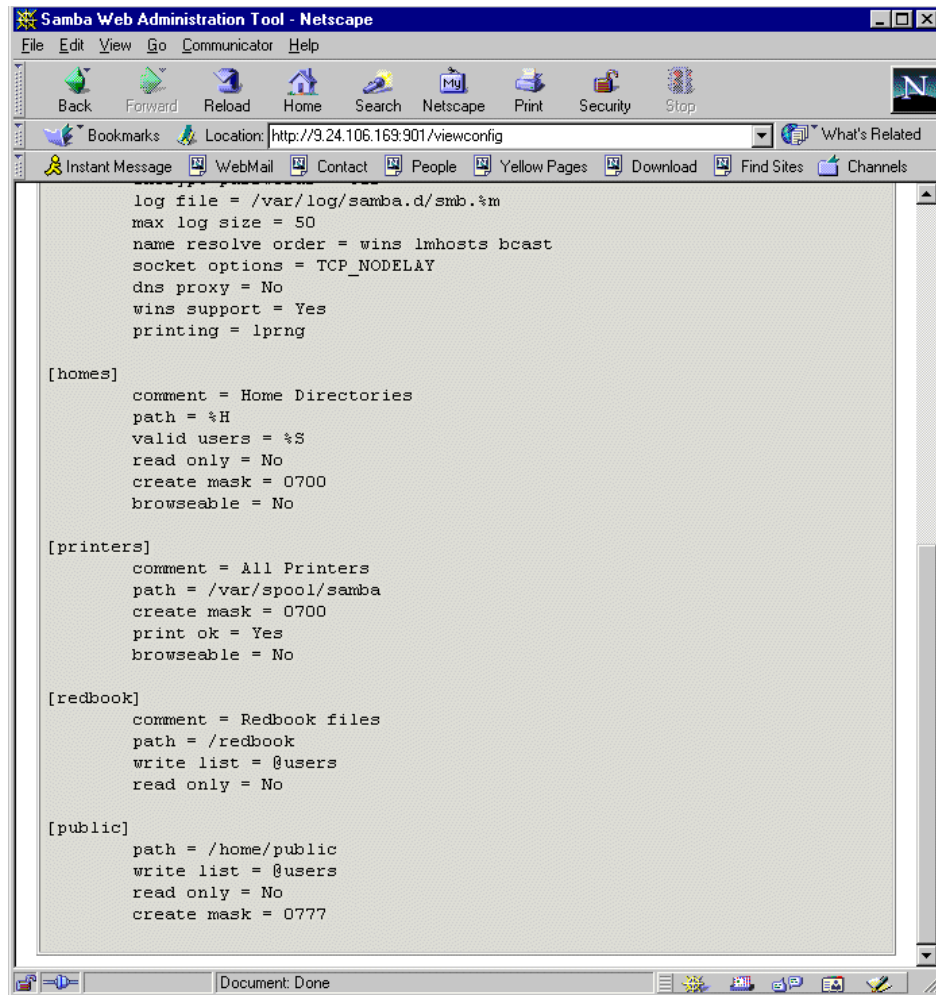


Figure 161. Viewing the smb.conf configuration file

7. Restart the Samba server.

Congratulations! You have just created your first usable share on the Samba server.

#### 6.2.3.6 Restarting the Samba server

The Samba server can be restarted by clicking the **Status** icon on any SWAT Web pages. You will see a window similar to Figure 162.

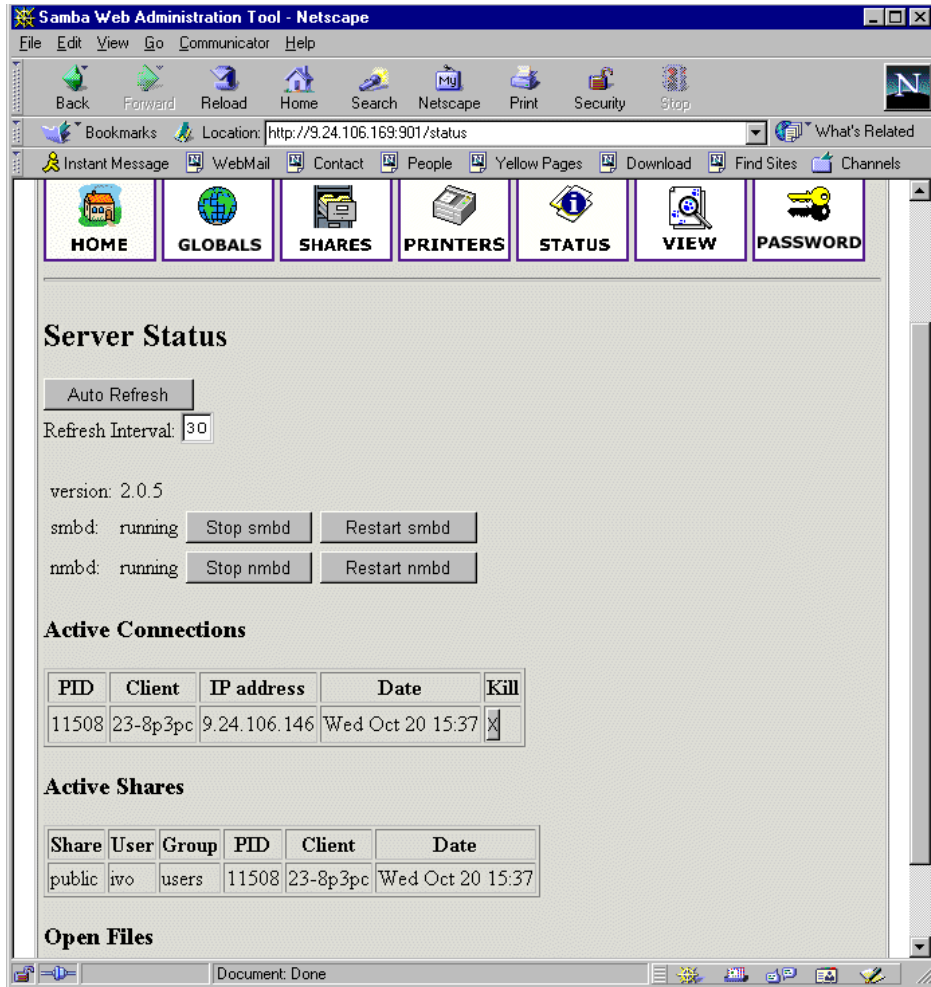


Figure 162. Restarting the Samba server

To restart the Samba server simply click **Restart smbd**. On this page you can also restart the WINS server by clicking **Restart nmbd**.

### 6.2.3.7 Printers

In the printer section you can view, modify, or add printers. The operations for handling printers are the same as for handling shares. You can access the printer settings by clicking the **Printers** icon on the SWAT Web page similar to Figure 163.

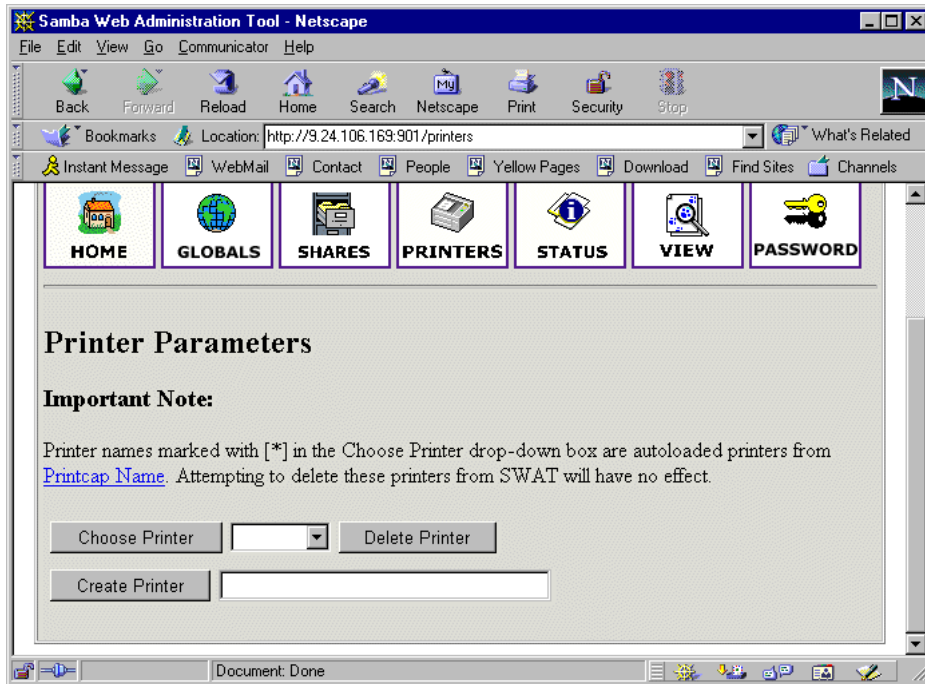


Figure 163. SWAT printers section

If you want to view the settings for a specific printer, select the printer from the list as you can see in Figure 164.

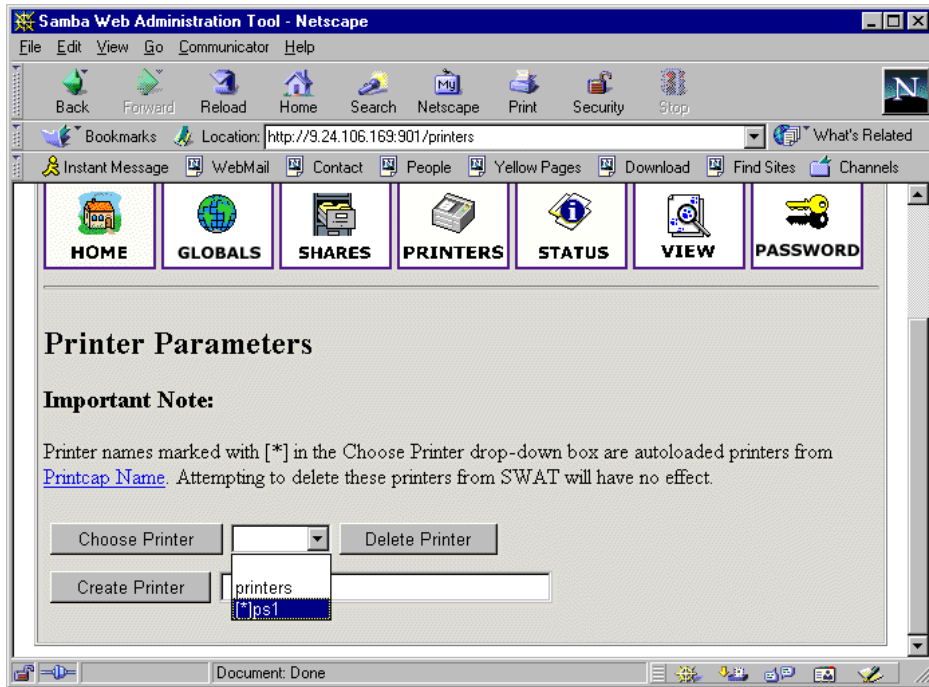


Figure 164. Selecting a printer

After you have selected the printer, click **Choose Printer** to view its properties. You will see a window similar to Figure 165.

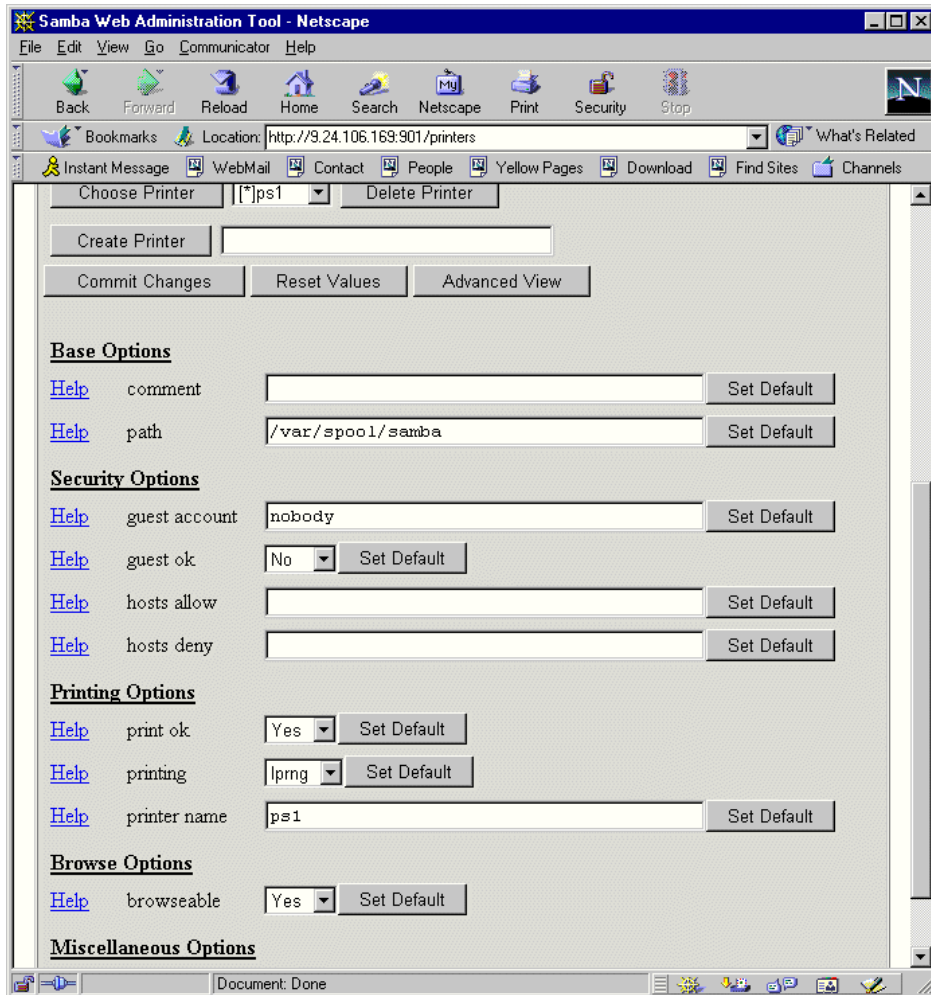


Figure 165. Printer properties

In this view you can also modify the printer properties. When you are done, save the settings by clicking **Commit Changes**.

#### 6.2.3.8 Status

In this section you can check the status of the Samba server. Here you can view all the connections and open files. You can also start or restart the Samba server or just its components. You can access the printer settings by clicking the **Status** icon on the SWAT Web pages, as you can see in Figure 166.



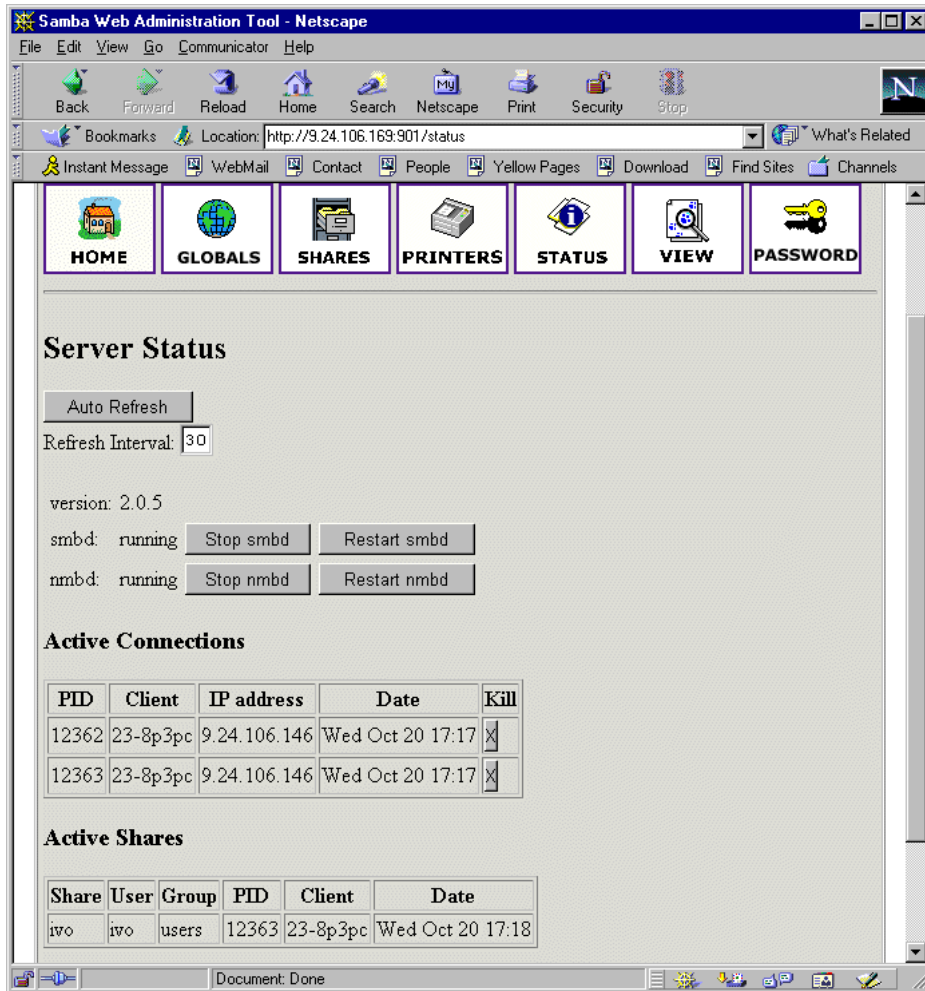


Figure 166. Status section

### 6.2.3.9 View

In this section you can view the current smb.conf configuration file. You can access printer settings by clicking the **View** icon on the SWAT Web pages similar to Figure 167.

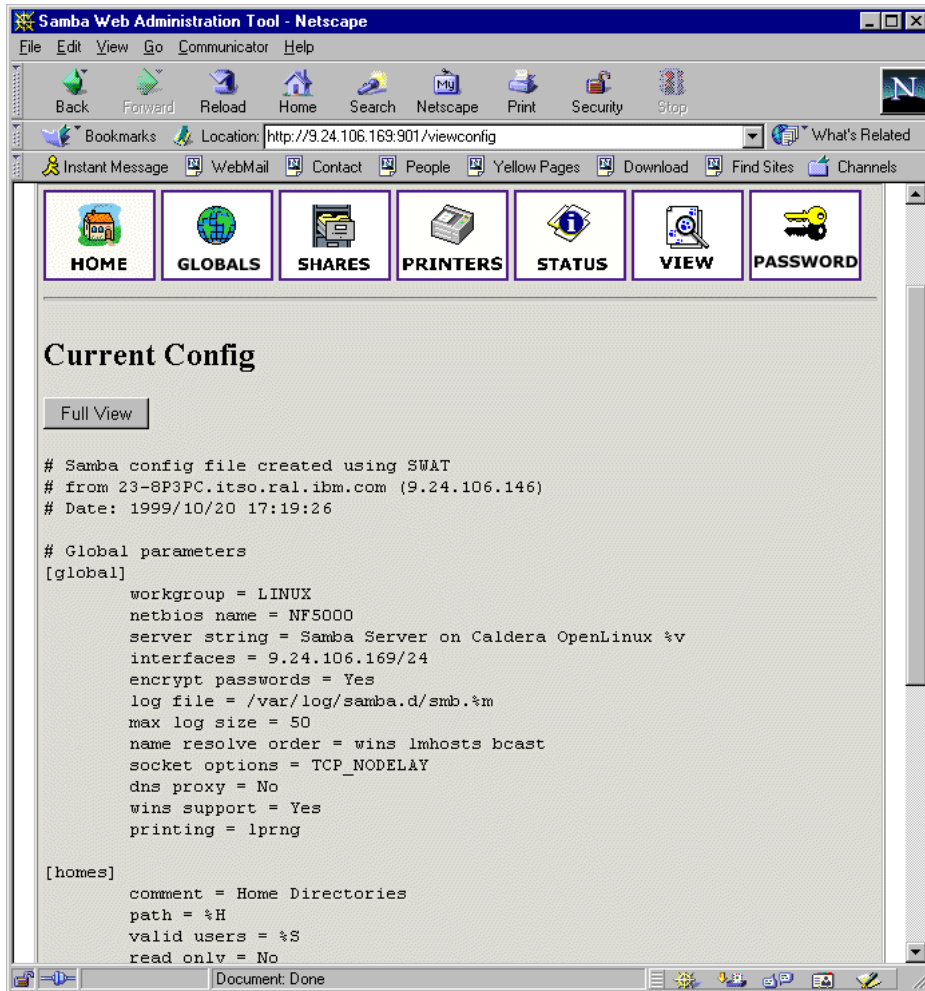


Figure 167. View section of SWAT

### 6.2.3.10 Password

In this section you can manage the passwords of all Samba users. You can access printer settings by clicking the **Password** icon on the SWAT Web pages similar to Figure 168.

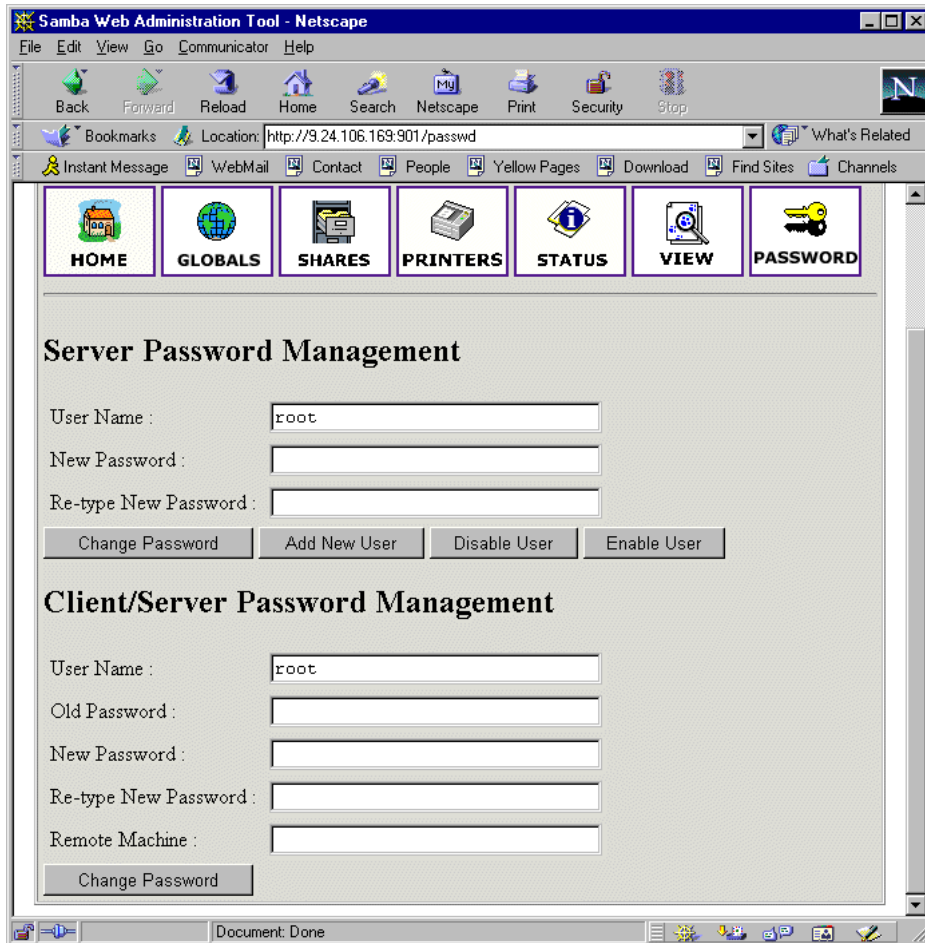


Figure 168. Managing passwords

---

### 6.3 Sources and additional information

You can find more information on the official Samba project Web site at:

<http://www.samba.org>

There are always good how-to documents on the Linux Documentation project home page:

<http://www.linuxdoc.org/>



---

## Chapter 7. Apache and IBM HTTP Servers

The Apache Web server is the most popular Web server software on today's Internet. According to the NetCraft Web server survey at <http://www.netcraft.com/survey/>, approximately 60% of all surveyed Web servers (more than 13 million) were running a version of Apache (as of the time of this writing). Apache is a very successful collaborative Open Source project. The Web site for Apache is <http://www.apache.org>. Because of the free availability of the full source code, it is a very flexible and powerful Web server solution. There are also a lot of additional modules, which can be used in combination with the Apache main program. Some popular examples are PHP (PHP: Hypertext Preprocessor, an embedded HTML scripting language), mod\_perl (an embedded perl interpreter) and mod\_ssl for secure transactions. More Apache modules can be downloaded from the Apache Module Registry at:

<http://modules.apache.org>.

Some of key features of Apache are:

- 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 DirectoryIndex 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).

---

## 7.1 The IBM HTTP Server

The IBM HTTP Server powered by Apache is based on the Apache HTTP Server. In addition to Linux, this HTTP Server also runs on AIX, Solaris and Windows NT. See the home page at:

<http://www-4.ibm.com/software/webservers/httpservers/>

IBM HTTP Server for Linux offers the following additional features:

- Remote Configuration: a browser-based configuration tool to allow manipulation of the server configuration via a GUI.
- SNMP Support: Simple Network Management Protocol (SNMP) is a well-established protocol for managing and gathering information about servers remotely. This new support allows IBM HTTP Server to be managed by the SNMP protocol.
- LDAP: The IBM HTTP Server Lightweight Directory Access Protocol (LDAP) plug-in allows authentication and authorization (which is required when accessing a protected resource) to be performed by an LDAP server, thereby greatly decreasing the administrative overhead for maintaining user and group information locally for each Web server.
- Machine Translation Support: This new function, when used with an available IBM Machine Translation Engine, enables the IBM HTTP Server to translate English Web pages into other languages without human intervention. This permits Web site visitors to read the page in their native language, effectively broadening the reach of your Web site. IBM Machine Translation Engines are included in the WebSphere Application Server 3.0 and include German, Simplified Chinese and Traditional Chinese. Additional languages will be available in the future.
- Support for SSL secure connections: The IBM HTTP Server powered by Apache supports both the SSL Version 2 and SSL Version 3 protocols. This protocol, implemented using IBM security libraries, ensures that data transferred between a client and a server remains private. Once your server has a digital certificate, SSL-enabled browsers such as Netscape Navigator and Microsoft Internet Explorer can communicate securely with your server using the SSL protocol. The IBM HTTP Server powered by

Apache supports client authentication, configurable cipher specifications, and session ID caching for improving SSL performance on the UNIX platforms.

- **Fast Response Cache Accelerator:** The Cache Accelerator can dramatically improve the performance of the IBM HTTP Server powered by Apache when serving static pages, for example, text and image files. Because the Cache Accelerator cache is automatically loaded during server operation, you are not required to list the files to be cached in your server configuration file. In addition, the server will automatically recache changed pages and remove outdated pages from the cache. The Cache Accelerator provides support for caching on Web servers with single and multiple TCP/IP adapters.

---

## 7.2 Apache HTTP Server installation

You can verify the installation by querying the RPM database to see if Apache and the Apache PHP module are installed by using the `rpm -q [package name]` to verify the following files:

```
apache-1.3.12-2.i386.rpm
mod_php3-3.0.13-1.i386.rpm
```

If it is not, mount the TurboLinux 6 CD and install it with the following commands:

```
mount /mnt/cdrom
cd /mnt/cdrom/TurboLinux/RPMS
rpm -Uhv [each file listed above]
```

In the TurboLinux default installation, Apache has the following default locations for configuration and error logs:

```
/home/httpd/html -- Default location of html files to be read.
/home/httpd/cgi-bin-- Default location of cgi files
/var/log/httpd/error.log -- Error log
/var/log/httpd/access.log -- Access log
/etc/httpd/conf/ -- Directory containing configuration files.
```

If you now point your browser to the server's IP address, you should see the start page (`/home//httpd/html/index.html`).

---

## 7.3 IBM HTTP Server installation

The IBM HTTP Server is not included with TurboLinux 6, so you first need to download the .tar file from the Web page (free but registration required):

```
http://www-4.ibm.com/software/websevers/httpsevers/download.html
```

The HTTPServer\_linux\_128\_tar.tar.gz (or HTTPServer\_linux\_56\_tar.tar.gz for 56-bit encryption) file contains the following packages:

- IBM\_HTTP\_Server-1.3.12-0.i386.rpm - IBM HTTP Server
- IBM\_Apache\_Source-1.3.12-0.i386.rpm - Apache 1.3.12 source
- IBM\_Admin\_Server-1.3.12-0.i386.rpm - Administration Server
- IBM\_Admin\_Server\_Forms-1.3.12-0.i386.rpm - Administration Server Web forms
- gsk4bas-4.0-3.57.i386.rpm - Security library
- IBM\_SSL\_128-1.3.12-0.i386.rpm - 128-bit SSL library
- or
- IBM\_SSL\_56-1.3.12-0.i386.rpm - 56-bit SSL library
- IBM\_SSL\_Base-1.3.12-0.i386.rpm - SSL module
- IBM\_Machine\_Translation-1.3.12-0.i386.rpm - Gateway to IBM MT engine)
- IBM\_SNMP-1.3.12-0.i386.rpm - SNMP client

We will not be discussing installation of the SSL and SNMP modules here. For more information about these, read the documentation included in the server by clicking **View Documentation** on the start page of the server site.

After you have downloaded the “gzipped tarball”, move it to the /tmp directory and extract it with the command:

```
tar -zxvf HTTPServer_linux_128_tar.tar.gz
```

This will extract the RPM files listed above from the tar archive into the subdirectory /tmp/IHS-1.3.12. You now need to become the root user (if you are not already). To avoid resource conflicts, you first have to shut down the currently running Apache Web server (if installed), by executing the following command:

```
/etc/rc.d/init.d/httpd stop
```



To prevent Apache from running automatically on boot, run `turboservice`, choose **Advanced**, and deselect `httpd` for the runlevel your server boots by default.

Before installing, you must create a symbolic link to one of the libraries in TurboLinux. Type the following commands to do this:

```
cd /usr/lib
ln -s libstdc++.so.2.9.0 libstdc++.so.2.9
```

You will also need to install the IBM JDK package located on the TurboLinux 6 Companion CD. Insert the CD and type the following commands:

```
mount /mnt/cdrom
cd /mnt/cdrom/TurboContrib/RPMS
rpm -Uvh ibm-jdk-1.1.8-1.i386.rpm
```

You now need to install the packages with the following commands (assuming the packages reside in the current directory):

```
rpm -Uvh --nodeps IBM_HTTP_Server-1.3.12-0.i386.rpm
rpm -Uvh --nodeps IBM_Admin_Server-1.3.12-0.i386.rpm
rpm -Uvh --nodeps IBM_Admin_Server_Forms-1.3.12-0.i386.rpm
```

The installation of the HTTP Server package will also attempt to start the server automatically. If this did not start, you might still have another HTTP Server running. Stop this one first, and try to restart the IBM HTTP Server with the following command:

```
/etc/rc.d/init.d/ibmhttpd start
```

If no errors are present on the command line or in the `/opt/IBMHTTPServer/logs/error_log` file, open the new HTTP Server's home page by pointing a browser to the server's IP address or host name. You should see the following page:

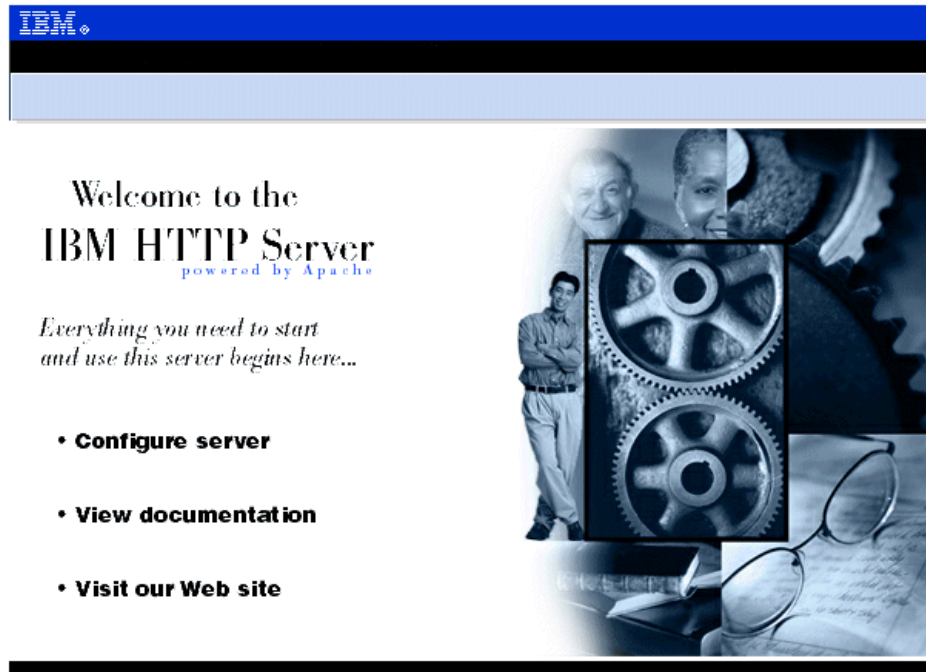


Figure 169. IBM HTTP Server startup page

If you still see the old Web server's startup page (see Figure 169), press Shift+Reload on the Netscape browser to force a reload of this page.

The basic installation of the IBM HTTP Server is now finished. In the default setup, it serves HTML pages from the directory `/opt/IBMHTTPD/htdocs` and CGI scripts from `/opt/IBMHTTPD/cgi-bin`. The log files reside in `/opt/IBMHTTPD/logs`.

### 7.3.1 Setting up the administration server

You have to perform some preliminary steps before you can start using the administration server to be able to modify the configuration files of your IBM HTTP Server remotely.

The administration server tasks allow the administration server read/write/execute access to the necessary configuration files and one executable file. The administration server should obtain read/write access through a unique user ID and group, which must be created. The User and Group directives of the administration server's configuration file should be changed to the unique user ID and group. The administration server's

configuration file's group access permissions should be changed to allow read/write group access. In addition there is a utility program that should have "Group execute permissions" and "Set User ID Root permissions". This executable must run as root in order to request restarts for the IBM HTTP Server and the Administration Server.

To properly set up these prerequisites, these tasks can be performed by executing the script `/opt/IBMHTTPserver/bin/setupadm`. After the invocation, it will ask you a few questions and will give detailed information about each step it is performing. Enter the keywords marked in boldface in the following windows:

```
bash-2.04# ./setupadm

*****
Please supply a User ID to run the Administration Server
We will create the USERID using System Administration tools
*****
[no default] -> wwwrun

*****
Please supply a GROUP NAME to run the Administration Server
We will create the Group using System Administration tools
*****
[no default] -> nogroup

*****
Please supply the Directory containing the files for
which a change in the permissions is necessary.
*****
[default: /opt/IBMHTTPserver/conf] -> [Enter]

These are the file(s) and directory for which we will be changing
Group permissions:

-rw-r--r--  1 root   root       4359 Jun  8 20:46 admin.conf
-rw-r--r--  1 root   root       4359 Jun  8 20:46 admin.conf.default
-rw-r--r--  1 root   root       7453 Jun  8 20:46 admin.msg
-rw-r--r--  1 root   root         1 Jun  8 20:45 admin.passwd
-rw-r--r--  1 root   root      31145 Nov  2 05:45 httpd.conf
-rw-r--r--  1 root   root      31145 Nov  2 05:45 httpd.conf.default
-rw-r--r--  1 root   root     48616 Nov  2 05:45 httpd.conf.sample
-rw-r--r--  1 root   root     12441 Jun  8 20:40 magic
```

Figure 170. Setupadm



To summarize the above steps: the administration server will be running under the user name “wwwrun” and the group “nobody.”

The administration server is protected with a user name and password. You can create an entry in the password file `/opt/IBMHTTPServer/bin/conf/admin.passwd` by issuing the following command from inside the directory `/opt/IBMHTTPServer/bin`:

```
./htpasswd -m ../conf/admin.passwd <user name>
```

Enter the password for the required user name twice. It is possible to have more than one user name in this password file, if you need to differentiate between multiple administrators.

Now you can start the administration server by running the following command:

```
/opt/IBMHTTPServer/bin/adminctl start
```

After clicking **Configure Server**, shown in Figure 169 on page 172, you need to enter the user name and password you defined for the administration server user. If entered correctly, you will see the welcome page of the administration server:

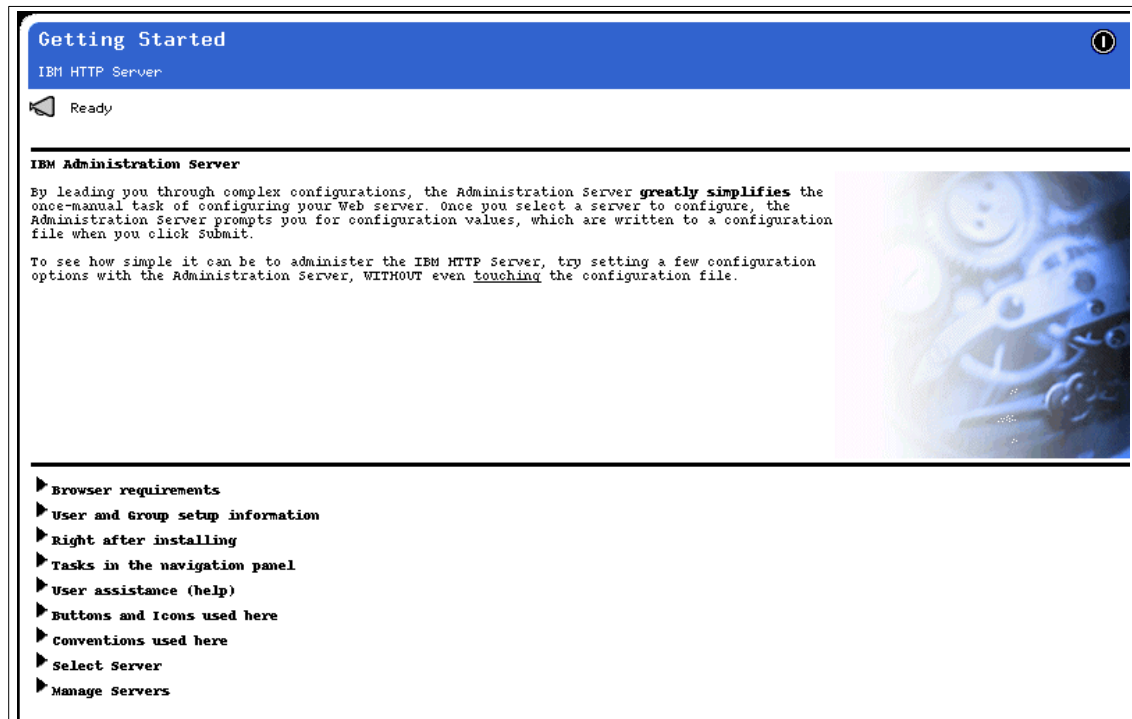


Figure 172. Administration Server startup window

You are now ready to start adjusting the configuration of your main Web server according to your needs. Please see the online documentation for help with the different configuration options.

---

## 7.4 General performance tips

Configuring Apache for maximum performance is dependent on many parameters. Apache is very flexible and gaining the best performance may require some research. A very informative document about Apache performance tuning can be found on the Apache Web site:

<http://www.apache.org/docs/misc/perf-tuning.html>

In short, experiment with the following options:

- Set the FollowSymLinks option unless you really don't want it.
- Set AllowOverride to None unless you really need it.

- Explicitly list all DirectoryIndex file options from most to least commonly used.
- Tune KeepAliveTimeout starting with 3 ranging to 30 per content and connection types.
- Apache (and the IBM HTTP Server as well) use multiple processes to handle individual requests. Tune StartServers starting with 64 increasing in steps of 32 until performance drops off. Tune MaxClients starting with the value of StartServers. **Note:** Scaling performance can fall off dramatically if Max Clients is too large!
- For SMP systems listening on a single socket, try recompiling after defining SINGLE\_LISTEN\_UNSERIALIZED\_ACCEPT.

A helpful utility to benchmark your Apache server is `ab`. In its simplest form, you can call it like this:

```
ab http://www.your-server.com/index.html
```

The following are `ab` options:

```
Usage: ab [options] [http://]hostname[:port]/path
Options are:
-n requests      Number of requests to perform
-c concurrency  Number of multiple requests to make
-t timelimit     Seconds to max. wait for responses
-p postfile     File containg data to POST
-T content-type Content-type header for POSTing
-v verbosity    How much troubleshooting info to print
-w             Print out results in HTML tables
-i            Use HEAD instead of GET
-x attributes   String to insert as table attributes
-y attributes   String to insert as tr attributes
-z attributes   String to insert as td or th attributes
-C attribute    Add cookie, eg. 'Apache=1234. (repeatable)
-H attribute    Add Arbitrary header line, eg. 'Accept-Encoding: zop'
                Inserted after all normal header lines. (repeatable)
-A attribute    Add Basic WWW Authentication, the attributes
                are a colon separated username and password.
-p attribute    Add Basic Proxy Authentication, the attributes
                are a colon separated username and password.
-V            Print version number and exit
-k            Use HTTP KeepAlive feature
-h            Display usage information (this message)
```

Figure 173. `ab` options





---

## Chapter 8. Packet filtering with IP Chains

Whenever you connect your computer to today's Internet world you are exposed to intruders from the outside. There are thousands of hackers just waiting to get into your computer to do damage or maybe to steal information. Therefore you need protection against them!

---

### 8.1 What is packet filtering?

As the name implies, packet filtering is a kind of a filter, filtering the data coming to your computer from a TCP/IP network. Packet filtering is one method commonly used in firewall implementations. With packet filtering you can implement a firewall that will protect your computer from the outside world.

Because everybody wants to communicate, sooner or later you need to connect your private network to the Internet. At that point it is time to think about security. You can also use a firewall on a single computer, which is for example connected to the Internet through a dial-up line. When you install a firewall to protect your internal network, every computer that wants to talk to a computer on the internal network must ask the firewall for permission. If the permission is not granted, access is denied.

---

### 8.2 What can you do with Linux packet filtering?

With Linux packet filtering you can do many things. Let us describe a few of them here:

- You can protect your internal network connected to the Internet from outside intruders.
- You can perform Network Address Translation (NAT), which allows internally connected computers without a registered Internet address to reach the Internet resources.
- You can filter the information going in or out of your internal network or just one computer.
- You can use your Linux server as a gateway between two different types of network, for example connecting token-ring and Ethernet worlds. This can be a cheap solution in comparison to buying an expensive router to this job.
- You can share your dial-up Internet connection with others.

---

### 8.3 What do you need to run packet filtering?

First, the IP Chains package must be installed. To confirm the package is installed, issue the command `rpm -q [package name]` to verify the package ipchains is installed. The package is:

```
ipchains-1.3.9-2.i386.rpm
```

If it is not installed, mount the TurboLinux 6 CD and install it with the following commands:

```
mount /mnt/cdrom
cd /mnt/cdrom/TurboLinux/RPMS
rpm -Uhv ipchains-1.3.9-2.i386.rpm
```

The default installation meets all these requirements except that the kernel is not optimized to be used as a router. If you want to increase the performance of the routing process, you should recompile the kernel and choose the **IP - optimize as router not host** option.

---

### 8.4 Network configuration for a packet filtering implementation

In this section we will describe our lab network setup for implementing a packet filtering solution.

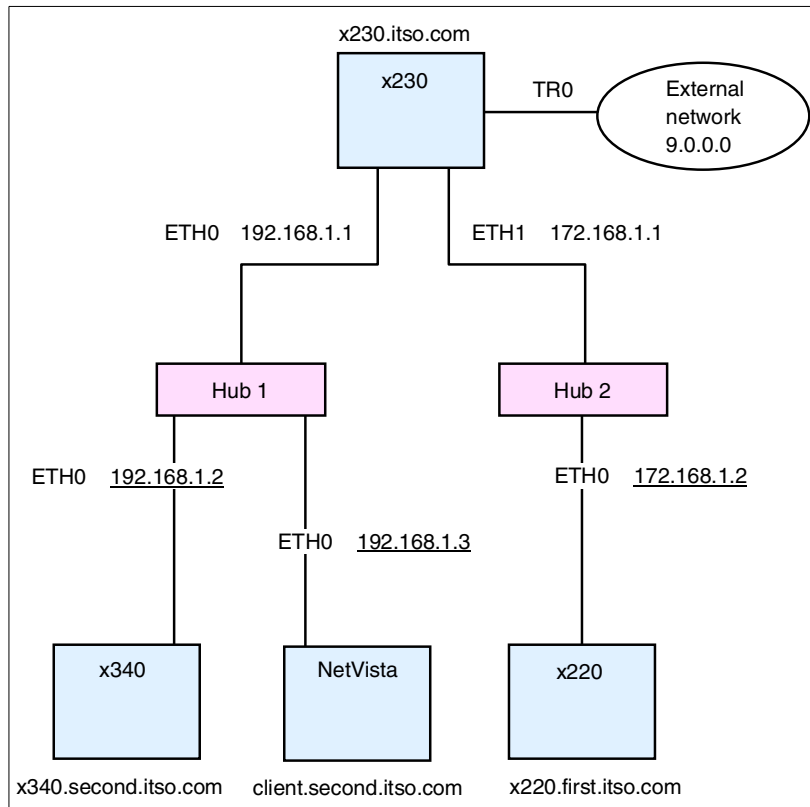


Figure 174. Lab network setup for firewall solution

Figure 174 shows our network setup:

- An x230 eServer with three Network Interface Cards (NIC) is acting as a gateway. The NICs have the following settings:
  - Eth0 - 192.168.1.1
  - Eth1 - 172.168.1.1
  - Tr0 - 9.24.104.28
- An x340 eServer with one NIC and the following settings:
  - Eth0 - 192.168.1.2, default gateway 192.168.1.1
- An x220 eServer with one NIC and the following settings:
  - Eth0 - 172.168.1.2, default gateway 172.168.1.1
- An NetVista all in one with one NIC and the following settings:

- Eth0 - 192.168.1.3, default gateway 192.168.1.1

You can see we have two separate networks, 192.168.1.0 and 172.168.1.0. These networks are connected to a Linux server that is acting as a gateway (router). You see that our gateway is connected to the Internet with a registered IP address. We enabled IP Forwarding on the server that was acting as a gateway.

---

## 8.5 How to permanently enable IP Forwarding

In TurboLinux the network process is started by executing this script during the server startup:

```
/etc/rc.d/init/network
```

The IP Forwarding is disabled by default. You can enable it by using the `turbonetcfg` tool described in Chapter 3, “Basic system administration” on page 39. Run `turbonetcfg`, then choose **TCP/IP Routing Config-->Options** and select **IPv4 Forwarding**. Then select **Done** and **Save/Exit**. `Turbonetcfg` will ask you to overwrite the existing files, and will restart the network. IP Forwarding is now enabled on this server.

Now your server is ready to act as a router. You can try this by pinging to the `tr0` interface 9.24.104.28 from the machine on 172.168.1.0 network. If the ping is successful your router is working correctly. You will see a window similar to Figure 175.

```
[root@x220 named]# ping 9.24.104.28
PING 9.24.104.28 (9.24.104.28): 56 data bytes
64 bytes from 9.24.104.28: icmp_seq=0 ttl=255 time=2.5 ms
64 bytes from 9.24.104.28: icmp_seq=1 ttl=255 time=1.1 ms
64 bytes from 9.24.104.28: icmp_seq=2 ttl=255 time=1.0 ms
64 bytes from 9.24.104.28: icmp_seq=3 ttl=255 time=1.0 ms
64 bytes from 9.24.104.28: icmp_seq=4 ttl=255 time=1.2 ms

--- 9.24.104.28 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 1.0/1.3/2.5 ms
```

Figure 175. PING after enabling IP Forwarding

---

## 8.6 Your first IP Chains rule

Now when your router is working, let us make use of it. It does not make sense to have a router without deploying it. We would like to access the external network 9.0.0.0 from the internal network 172.168.1.0. We can do this by using the IP Masquerading function of IP Chains. Follow these steps on the gateway server to set up the File Transport Protocol (FTP) access from internal network 172.168.1.0 to external network 9.0.0.0:

1. Create module dependency information for all modules by executing the command:

```
/sbin/depmod -a
```

2. Load the module for proper FTP masquerading:

```
/sbin/modprobe ip_masq_ftp
```

If you want to use another protocol, such as Real Audio and Internet Relay Chat (IRC), you can load the modules for them also.

3. Set up the timeout for IP Masquerading:

```
/sbin/ipchains -M -S 8000 20 200
```

The parameters have the following meaning:

8000 - timeout value for TCP sessions in seconds

20 - timeout value for TCP sessions after a FIN packet in seconds

200 - timeout value for UDP packets in seconds

You can adjust these settings to meet your needs.

4. Change built-in policy for forwarding by disabling it for all IP addresses:

```
/sbin/ipchains -P forward DENY
```

5. Add the policy for enabling the forwarding with masquerading for your internal networks:

```
/sbin/ipchains -A forward -s 192.168.1.0/24 -j MASQ
```

```
/sbin/ipchains -A forward -s 172.168.1.0/24 -j MASQ
```

You are ready to try your setup. From the computer on the network 172.168.1.0, execute the command:

```
/usr/bin/ftp server
```

Where `ftp server` is the FTP server on the external network (in our example 9.0.0.0). You will see a window similar to Figure 176.

```

[root@x220 named]# ftp 9.24.106.73
Connected to 9.24.106.73.
220 TPIV02 IBM TCP/IP for OS/2 - FTP Server ver 11:45:06 on Apr 17 2000 ready.
Name (9.24.106.73:root): ivo
331 Password required for ivo.
Password:
230 User ivo logged in.
Remote system type is OS/2.
ftp> 

```

Figure 176. FTP after IP Masquerading setup

You have just enabled access from internal networks to an external network.

## 8.7 How packets travel through a gateway

In this section we will explain how IP Chains work. You can see the path of a packet coming into your server in Figure 177.

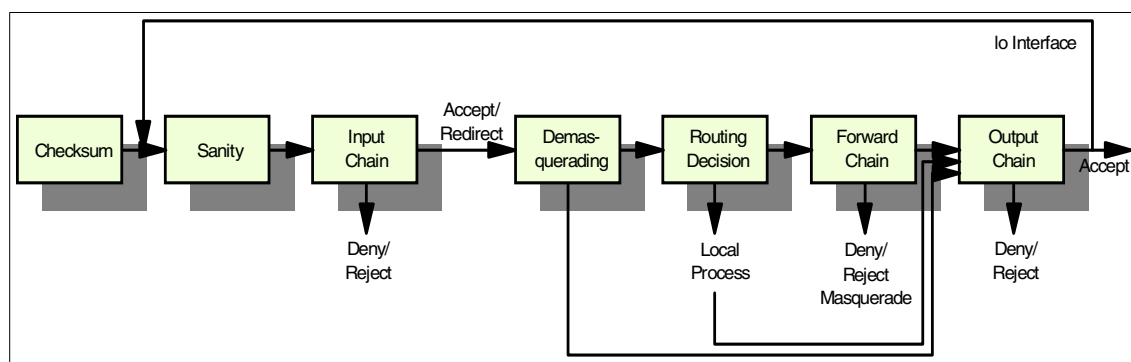


Figure 177. How the packet travels

Here are short descriptions of each stage:

- Checksum - this is to test if the packet is corrupted or not.
- Sanity - Malformed packets are denied here.
- Input chain - This is the first real packet checking point. Packets can be rejected, denied or accepted.
- Demasquerade - If the packet is a reply to a previously masqueraded packet, it is demasqueraded and goes directly from here to the output chain.
- Routing decision - Routing code decides if this packet is for a local process or should be forwarded to a remote machine.

- Local process - a process running on the server can receive packets after a routing decision step, and can then send the packets, which go through a routing decision step and then to the output chain.
- lo interface - if packets from a local process are destined for another local process, they will go through the output chain with interface set to “lo”, and will return to the input chain with interface “lo”. The “lo” interface is usually called the loopback interface.
- Local - if the packet is not created by the local process, then forward chain is checked.
- Forward chain - this is the checkpoint for all packets passing through this server to another.
- Output chain - this a checkpoint for all packets just before they are sent out.

As you can see from Figure 177, you have three places where you can check the packets in your server:

- a. Input chain
- b. Forward chain
- c. Output chain

With the `/sbin/ipchains` command you can set up your rules for packet checking.

**Note**

By default, all checking policies are set to Accept. This means that all packets can come in, go through or go out from your server without any restrictions.

You can see the current checking policies by executing:

```
/sbin/ipchains -L
```

You will see a window similar to Figure 178.

```
[root@client /root]# ipchains -L
Chain input (policy ACCEPT):
Chain forward (policy ACCEPT):
Chain output (policy ACCEPT):
[root@client /root]# █
```

Figure 178. Listing the default IP Chains policies

---

## 8.8 Using IP Chains

With the `/sbin/ipchains` command, you can create, change or delete your own policies for checking packets or you can modify built-in policies. You cannot delete the built-in chains, but you can append your rules to the existing chains or even create your own chains.

To manage whole chains you can use the parameters described in Table 19.

Table 19. Parameters for managing whole chains

Parameter	Description
-N	Create a new chain
-X	Delete an empty chain
-P	Change policy for a built-in chain
-L	List the rules in a chain
-F	Flush the rules out of a chain
-Z	Zero the packets and byte counters on all rules in a chain

For manipulating rules inside the chain you can use the parameters explained in Table 20.

Table 20. Parameters for managing rules in a chain

Parameter	Description
-A	Append new rule to a chain
-I	Insert a new rule in a chain at some position
-R	Replace a rule at some position in a chain
-D	Delete a rule at some position in a chain

And there are more operations for managing masquerading. They are described in Table 21.

Table 21. Parameters for managing masquerading

Parameter	Description
-M -L	List the currently masqueraded connections
-M -S	Set masquerading timeout values



### 8.8.1 How to create a rule

The most common syntax for creating a new rule is:

```
/sbin/ipchains -A input -s source -p protocol -j action
```

The parameters are described in Table 22.

Table 22. IPChains parameters

Parameter	Description
-A	Append a new rule to the chain
source	IP address or host name of the source
protocol	Type of the protocol to which one a rule is applied
action	What will happen with the packet: 1) ACCEPT - packet will be accepted 2) REJECT - packet will be rejected 3) DENY - packet is dropped since it was not received 4) MASQ - packet will be masqueraded 5) REDIRECT - packet is redirected to local port 6) RETURN - fail off the chain immediately

#### Note

Redirecting packets to a local port using the REDIRECT action makes sense only in combination with masquerading for a transparent proxy server.

For example, if you want to create a rule for denying the ICMP protocol packets, which are used when you execute the ping command, for a specific IP address you will do this by executing the command:

```
/sbin/ipchains -A input -s IP_address -p icmp -j DENY
```

If you omit the protocol definition, all the packets will be denied. So for example if you want to block the access to your machine from the network 172.168.1.0 with subnet mask 255.255.255.0 you can do this by executing the command:

```
/sbin/ipchains -A input -s 172.168.1.0/255.255.255.0 -j DENY
```

or with:

```
/sbin/ipchains -A input -s 172.168.1.0/24 -j DENY
```

As you can see, the subnet mask can be specified with the number of used bits for that mask.

The command for not allowing any traffic from your server to the network 172.168.1.0 with subnet mask 255.255.255.0 will look like this:

```
/sbin/ipchains -A output -d 172.168.1.0/24 -j DENY
```

Here we used the “-d” parameter for specifying the destination address.

#### 8.8.1.1 Using the inversion flag

With some of the parameters, you can use the inversion option “!”. This means that the rule will be applied to everything else except to the parameters specified after “!”. For example, if you want to deny packets that come from all IP addresses except from network 192.168.1.0 with subnet mask 255.255.255.0 you can do this by executing the command:

```
/sbin/ipchains -A input -s ! 192.168.1.0/24 -j DENY
```

#### Note

The rules you made are not permanent, so next time you restart the server they are gone.

### 8.8.2 Making the rules permanent

For making the rules permanent you have two scripts available that can make your life easier. To save all the rules you created, you can execute the command:

```
/sbin/ipchains-save > [filename]
```

You can then restore the saved rules by executing the command:

```
cat [filename] | /sbin/ipchains-restore
```

So if you want your saved rules to be enabled whenever you start your system, add the following line to the /etc/rc.d/rc.local file:

```
cat [filename] | /sbin/ipchains-restore
```

---

## 8.9 Sources of additional information

You can find more information on the official Linux IP Firewall Chains page at:

```
http://www.rustcorp.com/linux/ipchains
```

And there are always good how-to documents on the Linux Documentation Project home page:

```
http://www.linuxdoc.org/
```

---

## Chapter 9. sendmail

Communicating with other people is one of the most desirable experiences in life. Sending electronic mail is a way to communicate with people all over the globe. Electronic mail can be more reliable, cheaper, and faster than ordinary mail.

---

### 9.1 What is sendmail?

As you can tell from the name, sendmail is used to send mail. However, sendmail is not sending old-fashioned mail, but electronic mail, which becomes more important every day. But in spite of that, sendmail is basically acting as a post office. It receives mail from a sender and passes the mail on to the recipient post office. At the recipient post office a local postman delivers the mail to the recipient mailbox. Sendmail is a powerful Mail Transport Agent (MTA) and is used to pass the mail to another MTA, which can be sendmail or some other application capable of handling electronic mail.

---

### 9.2 What you can do with sendmail

With sendmail your Linux server can become a server for electronic mail. You can handle mail for users of a Linux server locally and users do not have to ask for mail accounts. The users on your Linux server will have their mailboxes locally and they will still be able to send mail to people anywhere. When you set up sendmail, you can also offer mail service to the users who have accounts on the other network servers that do not provide Internet mail service.

---

### 9.3 Before you begin

In the following sections we explain how to set up a mail server on your Linux server. In this explanation we will use our lab network setup, and all setup is related to this lab setup. You can easily adapt this to your existing installation. You can see our lab network in Figure 179.

The first step is to use the command `rpm -q [package name]` to confirm that the following packages are installed:

```
bind-8.2.2P5-3
bind-utils-8.2.2P5-3
sendmail-8.9.3-17
```

sendmail-cf-8.9.3-17

If any of these packages are not installed, mount the main TurboLinux CD and install them with the commands:

```
mount /mnt/cdrom
cd /mnt/cdrom/TurboLinux/RPMS
rpm -Uhr [each package listed above]
```

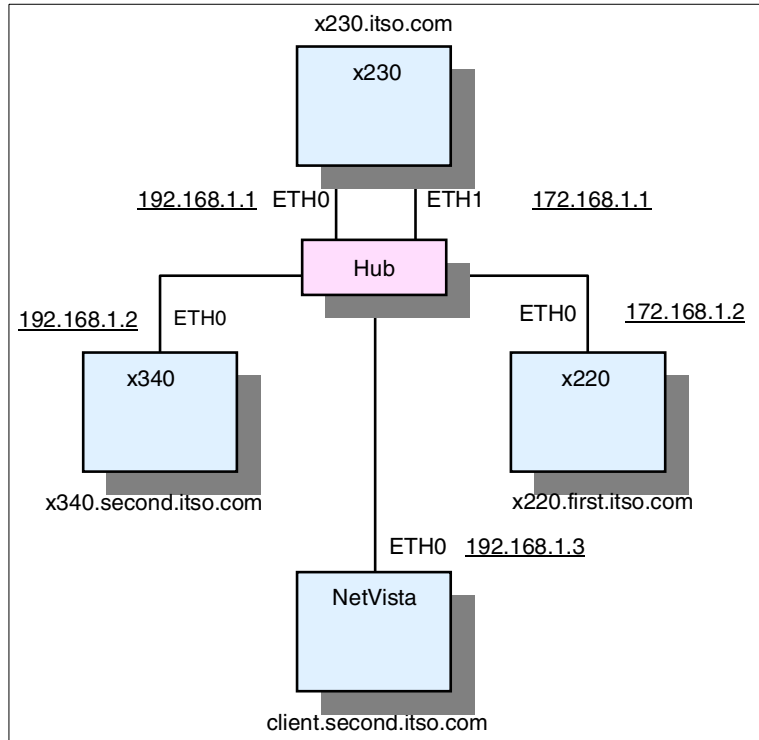


Figure 179. Lab network installation for sendmail setup

**Note**

For successful operation of sendmail you need the correct settings for the Domain Name System server (DNS). This means that you have to set up your own DNS correctly or have access to another DNS.

As you can see in Figure 179 our network consists of three domains:

1. Itso.com - this is the master domain. All computers in this domain have a .itso.com extension. In this domain we have a server running the master DNS.
2. First.itso.com - this is the first subdomain of the .itso.com domain. The main server in this domain is running the DNS for this domain and is also the mail (sendmail) server for the users of the domain.
3. Second.itso.com - this is the second subdomain of the .itso.com domain. The main server in this domain is running the DNS for this domain and is the mail (sendmail) server for the users of the domain.

All users of the mail server have a user name/password definition on their domain server even if they are using other physical servers or workstations. They need this user name/password for accessing the mail. Each defined user has a mailbox on the mail server. He can reach his mailbox over the network with his client connecting to the mail server. When the connection is established, the user can download his messages to his workstation and delete them from the server. Or he can remotely connect to the server and read his mail on the server, but in this case the mail stays in the mailbox on the server.

When the users are using the server for mail only and they have their own workstations, the mail servers are set up with limited space for each mail box. That means that users have to download their mail regularly from the server to make room for the new messages. In the environment where users use the server for their operations, the mailboxes are usually bigger. Users can still reach their mail remotely with the client; in which case they do not download the messages.

The most commonly used protocols for sending and delivering mail are SMTP/POP3. The Simple Mail Transfer Protocol (SMTP) is used for sending mail from the mail client and the Post Office Protocol (POP3) is for getting the mail from the mail server. Along with all other protocols, sendmail also supports the SMTP and POP3 protocols. In our setup we used SMTP/POP3 protocols for the mail exchange.

Before we start describing how to set up a mail server, we will describe how to set up DNS for our lab network. That is because the correct DNS setup is important for successful operation of the mail server.

---

## 9.4 Network configuration

Each subdomain is on its own network. The “first.itso.com” domain is on the network 172.168.1.0 and “second.itso.com” domain is on the network 192.168.1.0. The server running the “itso.com” domain is acting as the gateway for both subdomains (see Figure 180.)

We have the following network definitions:

1. The gateway server with two Network Interface Cards (NIC). The first NIC has an IP 192.168.1.1. The second NIC has the IP 172.168.1.1. The server has enabled IP Forwarding so it can act as a gateway for subnetworks. This server is also the DNS server for the “itso.com” domain.
2. The server for the “first.itso.com” domain has one NIC with the IP address 172.168.1.2.
3. The server for the “second.itso.com” domain has one NIC with the IP address 192.168.1.2.
4. The client in “second.itso.com” domain has one NIC with the IP address 192.168.1.3.
5. The client in “first.itso.com” domain has one NIC with the IP address 172.168.1.3.

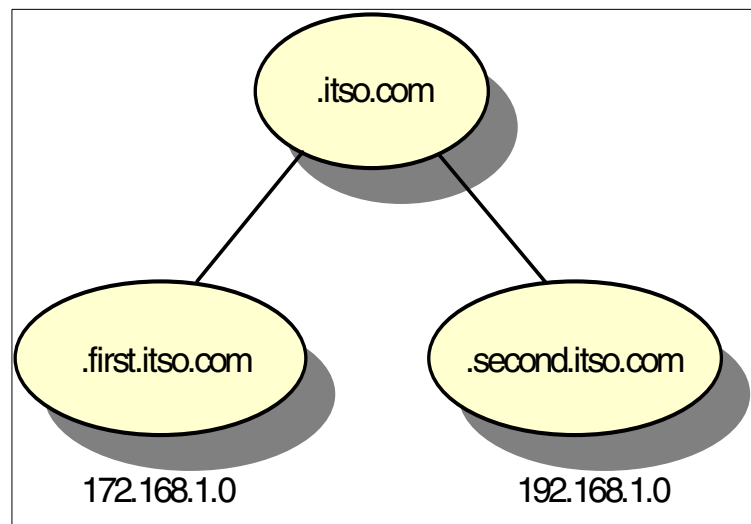


Figure 180. Domain setup

### 9.4.1 Setting up the master DNS

To set up the master DNS for “.itso.com” domain follow these steps:

1. Create the /etc/named.conf file with the following entries:

```
options {
directory "/var/named";
};
zone "." {
type hint;
file "root.hint";
};
zone "localhost" {
type master;
file "localhost.zone";
};
zone "0.0.127.in-addr.arpa" {
type master;
file "127.0.0";
};
zone "itso.com" {
notify no;
type master;
file "itso.com";
};
zone "1.168.172.in-addr.arpa" {
notify no;
type master;
file "172.168.1";
};
zone "1.168.192.in-addr.arpa" {
notify no;
type master;
```

Figure 181. Named.conf file

As you can see we defined the zone file for the “.itso.com” domain and the zone files for reverse address resolution for local networks, network 172.168.1.0 and network 192.168.1.0.

2. Create the directory /var/named.
3. Create the zone file /var/named/itso.com with the following entries:

```

@      IN      SOA      x230.itso.com. root.itso.com. (
                                1997022700 ; Serial
                                28800      ; Refresh
                                14400      ; Retry
                                3600000    ; Expire
                                86400 )    ; Minimum
;
NS x230 ; Name Server
MX 10 mail ; Mail Server
;
x230 A 172.168.1.1
mail CNAME x230
first A 172.168.1.1
second A 192.168.1.1
second.itso.com. 86400 IN NS x340.second.itso.com.
x340.second.itso.com. 86400 IN A 192.168.1.2
first.itso.com. 86400 IN NS x220.first.itso.com.

```

Figure 182. *Itso.com file*

We specified in this file that all requests for “first.itso.com” and “second.itso.com” go to the corresponding DNS servers in these domains.

4. Create the zone file `/var/named/172.168.1` with the following entries:

```

@      IN      SOA      x230.itso.com. root.itso.com. (
                                1997022700 ; Serial
                                28800      ; Refresh
                                14400      ; Retry
                                3600000    ; Expire
                                86400 )    ; Minimum
;
NS x230.itso.com. ; Name Server
1 PTR x230.itso.com.
2.1.168.172.in-addr.arpa. IN CNAME 2.0-255.0.168.172.in-addr.arpa.
3.1.168.172.in-addr.arpa. IN CNAME 3.0-255.0.168.172.in-addr.arpa.
;
0-255.0.168.172.in-addr.arpa. 86400 IN NS x220.first.itso.com.

```

Figure 183. *172.168.1 zone file*

As you can see, all requests for the reverse address resolution of the 172.168.1.0 network are passed on to the DNS in the “first.itso.com” domain. So when the DNS server gets a request for IP address in the network 172.168.1.0-255 it will pass this request to the DNS server that is serving this network. In our example, this is the DNS server for “first.itso.com” domain.

5. Create the zone file `/var/named/192.168.1` with the following entries:



```

@      IN      SOA      x230.itso.com. root.itso.com. (
                                1997022700 ; Serial
                                28800      ; Refresh
                                14400      ; Retry
                                3600000    ; Expire
                                86400     ) ; Minimum

NS x230.itso.com. ; Name Server
1 PTR x230.itso.com.
2.1.168.192.in-addr.arpa. IN CNAME 2.0-255.0.168.192.in-addr.arpa.
3.1.168.192.in-addr.arpa. IN CNAME 3.0-255.0.168.192.in-addr.arpa.
;
0-255.0.168.192.in-addr.arpa. 86400 IN NS x340.second.itso.com.

```

Figure 184. 192.168.1 zone file

As you can see, all requests for the reverse address resolution of the 192.168.1.0 network are passed on to the DNS in the “second.itso.com” domain. So when the DNS server gets a request for an IP address in the network 192.168.1.0-255, it will pass this request to the DNS server which is serving this network. In our example this is the DNS server for the “second.itso.com” domain.

6. Create the zone file /var/named/127.0.0 with the following entries:

```

@      IN      SOA      localhost. root.localhost. (
                                1997022700 ; Serial
                                28800      ; Refresh
                                14400      ; Retry
                                3600000    ; Expire
                                86400     ) ; Minimum

NS localhost.

```

Figure 185. 127.0.0 zone file

7. Create the zone file /var/named/localhost.zone with the following entries:

```

@      IN      SOA      localhost. root.localhost. (
                                1997022700 ; Serial
                                28800      ; Refresh
                                14400      ; Retry
                                3600000    ; Expire
                                86400     ) ; Minimum
NS localhost.

```

Figure 186. localhost.zone file

8. You need to set up the DNS client so it will point to the DNS server running on the server. You need to specify the address of the DNS server to be 127.0.0.1, an example is shown in 5.2, “DNS sample configuration” on page 131.
9. Using the Webmin configuration tool you need to set the domain name to “itso.com”.
10. Your server is ready to be powered on. To start the server without restarting the operating system, which is the case in another very popular operating system, execute the command:

```
/etc/rc.d/init/named start
```

Congratulations! You have just set up a fully functional DNS server. Now let’s set up the DNS servers for the subdomains.

#### 9.4.2 Setting up the DNS for the first subdomain

Before you start configuring DNS, you need to check if the DNS server is installed. You can do this following the instructions in 9.4.1, “Setting up the master DNS” on page 193.

After you have checked all prerequisites, follow these steps to set up DNS for the “first.itso.com” domain:

1. Create the /etc/named.conf file with the following entries:

```

options {
directory "/var/named";
forward only;
forwarders {172.168.1.1;};
};
zone "." {
type hint;
file "root.hint";
};
};
zone "localhost" {
type master;
file "localhost.zone";
};
zone "0.0.127.in-addr.arpa" {
type master;
file "127.0.0";
};
};
zone "first.itso.com" {
notify no;
type master;
file "first.itso.com";
};
zone "1.168.172.in-addr.arpa" {
notify no;
type master;
file "172.168.1";
};
};
zone "0-255.1.168.172.in-addr.arpa" {
type master;
};
};

```

Figure 187. *Named.conf* file

As you can see we defined the zone for the root domain “.” and the zone for “first.itso.com” domain. We also defined zones for reverse address resolution for local network 172.168.1.0 and for network 172.168.1.0-255, which serve the requests from the root server. You can see the zone files for network 172.168.1.0 and network 172.168.1.0-255 are the same. We need two definitions because the first one (1.168.172) is for local requests and the second (0-255.1.168.172) is for resolving requests from the master server in case someone else requests reverse resolution in this network (172.168.1.0) from the root server. The master server will ask the server serving this network (172.168.1.0) for the information (in our example, the server for the “first.itso.com” domain). Refer to 9.4.1, “Setting up the master DNS” on page 193 to see how the root server setup is done to pass requests to servers in subdomains. As you can see in options

section of Figure 187, we set up forwarding in the Named.conf file. We used this because the server is on the private network and it cannot reach the root servers. So any requests that are not for the “first.itso.com” domain will be passed to the master server of the “itso.com” domain.

2. Create the directory /var/named.
3. Create the zone file /var/named/first.itso.com with the following entries:

```
@      IN      SOA x220.first.itso.com. root.first.itso.com. (
                                1997022700 ; Serial
                                28800     ; Refresh
                                14400     ; Retry
                                3600000   ; Expire
                                86400    ) ; Minimum

;
NS x220 ; Name Server
MX 10 mail ; Mail Server
;
x220 A 172.168.1.2
client A 172.168.1.3
mail CNAME x220
```

Figure 188. first.itso.com file

In this file we create definitions for all the computers in network 172.168.1.0.

4. Create the zone file /var/named/172.168.1 with the following entries:

```
@      IN      SOA x220.first.itso.com. root.first.itso.com. (
                                1997022700 ; Serial
                                28800     ; Refresh
                                14400     ; Retry
                                3600000   ; Expire
                                86400    ) ; Minimum

NS x220.first.itso.com. ; Name Server
1 PTR x230.itso.com.
2 PTR x220.first.itso.com.
3 PTR client.first.itso.com.
```

Figure 189. 172.168.1 zone file

In this file we define reverse address resolution for network 172.168.1.0.

5. Create the zone file /var/named/127.0.0 with the following entries:

```

@      IN      SOA      localhost. root.localhost. (
                                1997022700 ; Serial
                                28800      ; Refresh
                                14400      ; Retry
                                3600000    ; Expire
                                86400     ) ; Minimum
NS 127.0.0.1

```

Figure 190. 127.0.0 zone file

6. Create the zone file `/var/named/localhost.zone` with the following entries:

```

@      IN      SOA      localhost. root.localhost. (
                                1997022700 ; Serial
                                28800      ; Refresh
                                14400      ; Retry
                                3600000    ; Expire
                                86400     ) ; Minimum
NS localhost.

```

Figure 191. localhost.zone file

7. You need to set up the DNS client so it will point to the DNS server running on the server. You need to specify the address of the DNS server to be 127.0.0.1, following the procedure described in 5.2, “DNS sample configuration” on page 131.
8. Using the Webmin configuration tool you need to set up the domain name to “first.itso.com”.
9. Start the server with the command:

```
/etc/rc.d/init/named start
```

### 9.4.3 Setting up the DNS for the second subdomain

Follow these steps to set up the DNS for the “second.itso.com” domain:

1. Create the `/etc/named.conf` file with the following entries:

```

options {
directory "/var/named";
forward only;
forwarders {192.168.1.1;};
};
zone "." {
type hint;
file "root.hint";
};
zone "localhost" {
type master;
file "localhost.zone";
};
zone "0.0.127.in-addr.arpa" {
type master;
file "127.0.0";
};
zone "second.itso.com" {
notify no;
type master;
file "second.itso.com";
};
zone "1.168.192.in-addr.arpa" {
notify no;
type master;
file "192.168.1";
};
zone "0-255.1.168.192.in-addr.arpa" {
type master;
file "192.168.1";
};

```

Figure 192. *Named.conf* file

As you can see, we defined the zone for the root domain “.” and the zone for “second.itso.com” domain. We also defined zones for reverse address resolution for local, network 192.168.1.0 and for network 192.168.1.0-255, which serves the requests from the root server. You can see the zone files for network 192.168.1.0 and network 192.168.1.0-255 are the same. We need two definitions because the first one (1.168.192) is for local requests and the second (0-255.1.192.168) is for resolving requests from the root server in case someone else requests reverse resolution in this network (192.168.1.0) from the root server. The root server will ask the server serving this network (192.168.1.0) for the information (in our example, the server for the “first.itso” domain). Refer to 9.4.1, “Setting up the master DNS” on page 193 to see how root server setup is done to pass requests

to servers in subdomains. As you can see in the options section of Figure 192, we set up forwarding in the `Named.conf` file. We used this because the server is on the private network and it can not reach the root servers. So any requests that are not for the “second.itso.com” domain will be passed to the master server of the “itso.com” domain.

2. Create the directory `/var/named`.
3. Create the zone file `/var/named/second.itso.com` with the following entries:

```
@      IN SOA x340.second.itso.com root.second.itso.com. (
                                1997022700 ; Serial
                                28800      ; Refresh
                                14400      ; Retry
                                3600000    ; Expire
                                86400 )    ; Minimum

;
NS x340 ; Name Server
MX 10 mail ; Mail Server
;
x340 A 192.168.1.2
mail CNAME x340
```

Figure 193. `Second.itso.com` zone file

In this file we create definitions for all computers in network 192.168.1.0.

4. Create the zone file `/var/named/192.168.1` with the following entries:

```
@      IN SOA x340.second.itso.com root.second.itso.com. (
                                1997022700 ; Serial
                                28800      ; Refresh
                                14400      ; Retry
                                3600000    ; Expire
                                86400 )    ; Minimum

NS x340.second.itso.com. ; Name Server
1 PTR x230.itso.com.
2 PTR x340.second.itso.com.
3 PTR client.second.itso.com
```

Figure 194. `192.168.1` zone file

In this file we define reverse address resolution for network 192.168.1.0.

5. Create the zone file `/var/named/127.0.0` with the following entries:

```

@      IN      SOA      localhost. root.localhost. (
                                1997022700 ; Serial
                                28800      ; Refresh
                                14400      ; Retry
                                3600000    ; Expire
                                86400     ) ; Minimum
NS 127.0.0.1

```

Figure 195. 127.0.0 zone file

6. Create the zone file /var/named/localhost.zone with the following entries:

```

@      IN      SOA      localhost. root.localhost. (
                                1997022700 ; Serial
                                28800      ; Refresh
                                14400      ; Retry
                                3600000    ; Expire
                                86400     ) ; Minimum
NS localhost.

```

Figure 196. localhost.zone file

7. You need to set up the DNS client so it will point to the DNS server running on the server. You need to specify the address of the DNS server to be 127.0.0.1, following the procedure described in 5.2, “DNS sample configuration” on page 131.
8. Using the Webmin configuration tool you need to set up the domain name to “second.itso.com”.
9. Start the server with the command:

```
/etc/rc.d/init/named start
```

You now have three DNS servers running. The network is ready for the setup of the mail server.

#### 9.4.4 Setting up sendmail

All documentation on sendmail will tell you that the sendmail configuration file /etc/sendmail.cf is a nightmare for a network administrator. This is not entirely true; when you do not need any special features offered by sendmail, the setup is fairly easy. You just need to modify the generic macro files slightly and recreate the new sendmail.cf with the “m4” macro processor. In this section we explain how to set up sendmail for handling mail in its own domain. TurboLinux 6 comes with a generic macro file for sendmail called



/usr/lib/sendmail-cf/turboLinux.mc. This file just needs a little modification to become a working file on your server. Follow these steps to set up your mail servers:

1. Make a copy of the generic macro file with the commands:

```
cd /usr/lib/sendmail-cf
cp turboLinux.mc mydomain.mc
```

2. Add the following lines to the mydomain.mc file:

- a. For the server in the “first.itso.com” domain:

```
dnl #####
dnl # Definitions for sample domain
dnl # we define PSEUDONYMS, DEFAULT_HOST
define(`PSEUDONYMS', `x220.first.itso.com' `first.itso.com')
define(`DEFAULT_HOST', 'x220.first.itso')
```

Figure 197. mydomain.mc file

- b. For the server in “second.itso.com” domain:

```
dnl #####
dnl # Definitions for sample domain
dnl # we define PSEUDONYMS, DEFAULT_HOST
define(`PSEUDONYMS', `x340.second.itso.com' `second.itso.com')
define(`DEFAULT_HOST', 'x340.second.itso.com')
dnl #####
```

Figure 198. mydomain.mc file

3. Create a new sendmail.cf file with your domain file by executing the sendmail create command:

```
/usr/bin/m4 /usr/lib/sendmail-cf/m4/cf.m4
/usr/lib/sendmail-cf/cf/mydomain.cf
/usr/lib/sendmail-cf/feature/relay_entire_domain.m4 >
/etc/sendmail.cf
```

Figure 199. sendmail create command

This can be a lot easier if your current directory is /usr/lib/sendmail-cf:

```
/usr/bin/m4 m4/cf.m4 cf/mydomain.cf feature/relay_entire_domain.m4 >
/etc/sendmail.cf
```

As you can see we used two more files in order to create the configuration file:

- a. cf.m4 must be used, or files will not be parsed correctly.
  - b. relay\_entire\_domain.m4 is used to enable clients, which access the mail server with remote clients, to send mail through this server.
4. Modify the /etc/sendmail.cw file:

- a. For the server in the “first.itso.com” domain add the line:

```
first.itso.com
```

- a. For the server in the “second.itso” domain add the line:

```
second.itso.com
```

The /etc/sendmail.cw includes all aliases for your mail server. You need to include the domain name; otherwise, mail will be undeliverable.

5. Start the server with the command:

```
/etc/rc.d/init.d/mta start
```

If you want sendmail to start automatically when the server is started, run `turboservice`, choose **Advanced**, and select sendmail for the runlevel you use on your server.

You need to execute all the previous instructions on the servers in both domains (first.itso and second.itso) if you want to send mail from one domain to another.

#### 9.4.4.1 Configure sendmail for mail routing

By default sendmail can deliver mail to a defined user if it can reach the mail server for the user’s domain. So, for example, if you are on an internal network and your sendmail server does not have direct connection to the Internet, you can configure sendmail to route mail through another reachable mail server, that is connected to the Internet. To do this you need to enter the appropriate values into the file /etc/mail/mailertable. For example, if you want to route the mail for `otherdomain.com` through `reachableserver`, then your mailertable file will look similar to this:

```
#
# /etc/mail/mailertable
#
# This file can be used to fine-tune sendmail's email routing.
# After you change this file, you must rebuild the DB file using
# /usr/sbin/makemap hash /etc/sendmail/mailertable < /etc/sendmail/maile
#
otherdomain. smtp:reachableserver
```

Figure 200. mailertable file

This means that all mail for `otherdomain` will be routed to the `reachableserver` mail server with the SMTP protocol. The domain name must always have a “.” dot at the end. With this setup you can route mail from server to server until it reaches its destination.

Whenever you modify the `/etc/mail/mailertable` file, you need to rebuild the `/etc/mail/mailertable.db` file, which is really the file used by sendmail to perform routing tasks. You can do this by executing the command:

```
/usr/sbin/makemap hash /etc/mail/mailertable < /etc/mail/mailertable
```

If you want all your mail to be routed to another server, you may do this by appending the following line to the `/etc/mail/mailertable` file:

```
smtp:smartserver
```

Where `smartserver` is the mail server that will handle all mail from your sendmail server.

### 9.4.5 Setting up the mail client

Each mail user needs a user name/password on the server that is running the mail server (in our case sendmail). After the user has a user name/password, he can reach his mailbox remotely. In this section we show how to set up the Netscape mail setting for sending and receiving mail.

**Note**

The POP3 server must be installed and running before you can set up the client.

Follow these steps to set up the Netscape mail properties:

1. Start Netscape.
2. Select **Edit > Preferences > Mail & Newsgroups > Identity** and you will see a window similar to Figure 201.

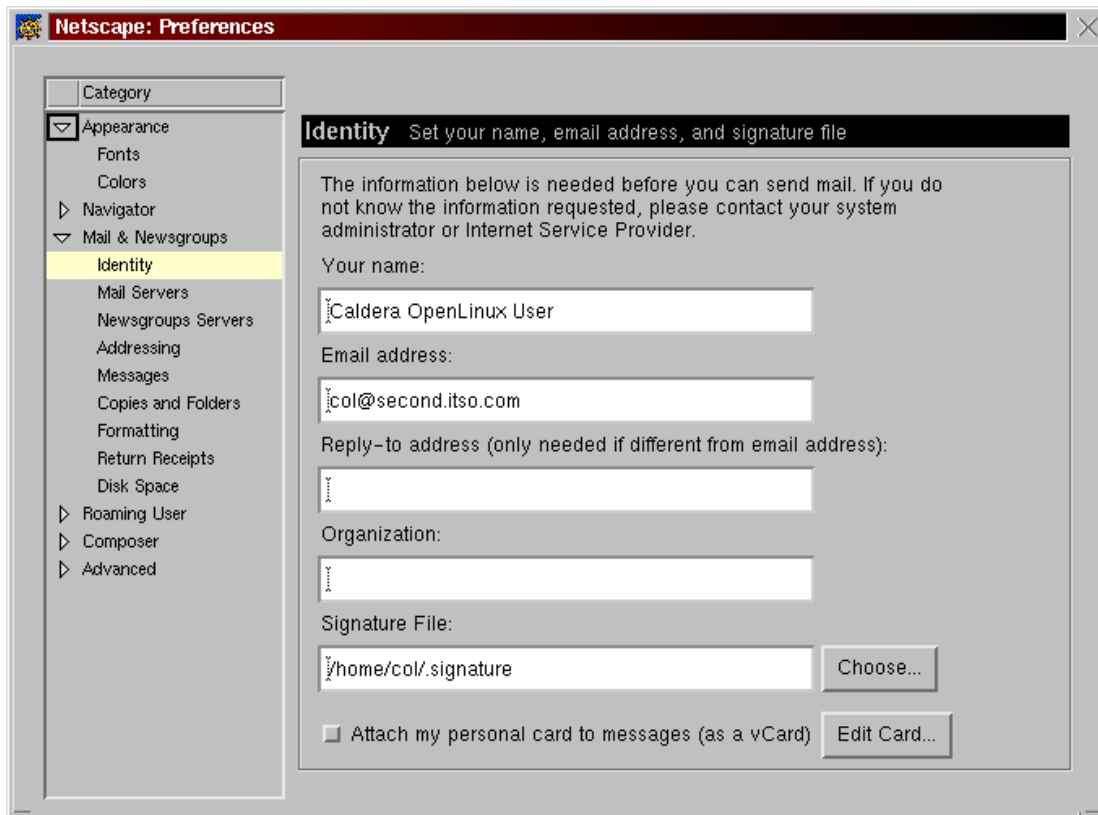


Figure 201. Setting the identity settings

3. Type in the required values and select **Mail Servers**. You will see a window similar to Figure 202.

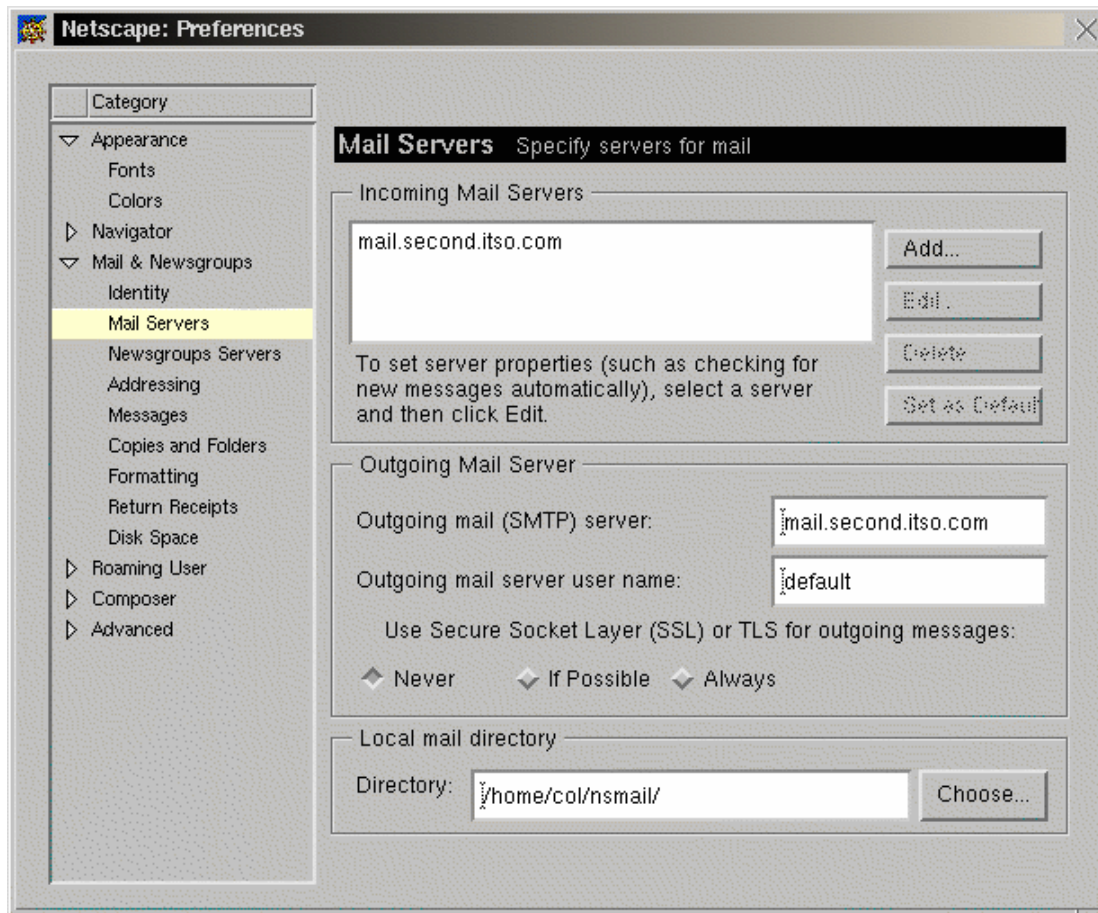


Figure 202. Setting the mail servers

4. In the Outgoing mail (SMTP) server field type your mail server address (in our example, “mail.second.itso.com”).
5. In the Incoming Mail Server section, select the current server and click **Edit**, and you will see a window similar to Figure 203.

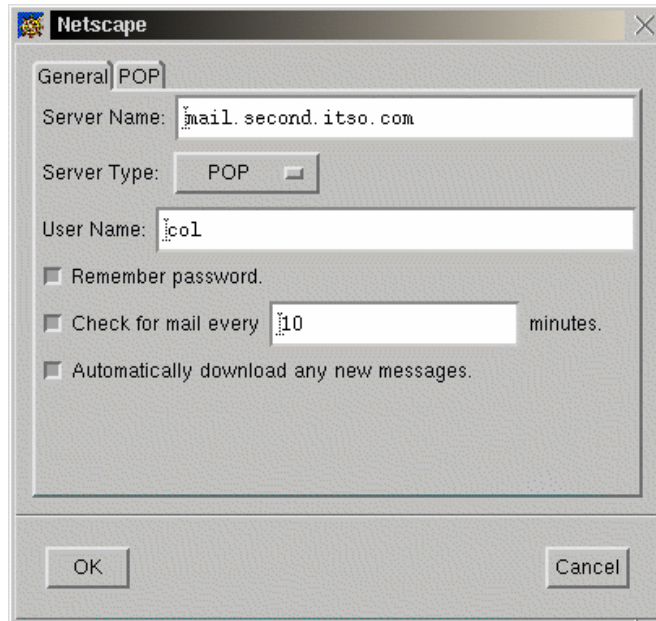


Figure 203. Setting POP3 server

6. In the Server Name field, type in the address of your mail server.
7. In the User Name field, type in your user name on the mail server. You can also configure some other options that will affect your mail reading. Click **OK** to continue.
8. When you are back in the Preferences window, click **OK** to store your new mail settings.

Now you are ready to send mail to all the users in the “first.itso.com” and “second.itso.com” domains.

---

## 9.5 Sources of additional information

You can find more information on the official Web site of the sendmail project:

<http://www.sendmail.org>

And there are always good how-to documents at the Linux Documentation project Web site:

<http://www.linuxdoc.org/>

---

## Chapter 10. DHCP - Dynamic Host Configuration Protocol

With the ever-decreasing number of IP addresses available, along with the headache of maintaining static IPs, DHCP has become a necessity in most TCP/IP computing environments.

---

### 10.1 What is DHCP?

DHCP stands for Dynamic Host Configuration Protocol. When using TCP/IP, a computer system needs a unique IP address to communicate with other computer systems. Without DHCP, the IP address must be entered manually at each computer system. DHCP lets network administrators distribute IP addresses from a central location without having to actively manage each individual address.

With DHCP, IP addresses are distributed through pools usually broken up by subnet. Leases are given out for a specific time period for each address. The process of managing leases is all done by the DHCP server. Once a lease has expired the DHCP server will try to contact the client or the client will contact the server to renew the lease. If the server cannot contact the client, the IP address is returned to the pool and available for the next client in need of an address.

---

### 10.2 Why should I use DHCP?

In the past, for every device on a network you had to have a static IP address. With the increasing number of computers accessing the Internet, the pool of available addresses is quickly diminishing. Network administrators can significantly reduce the number of IP addresses they need by using DHCP.

Even with smaller networks, keeping track of individual IP addresses can be maintenance intensive. With DHCP, the server does all of the maintenance, mapping IP addresses to MAC addresses and tracking lease times. Administrators can adjust lease times, expand or reduce pools, and change gateways or DNS addresses, all from a central location.

---

### 10.3 Implementation on TurboLinux 6

In this section we will discuss how to implement a DHCP server on Linux.

Install the DHCP server binaries. These packages are available on the TurboLinux Install CD. To install them, type the following command:

```
rpm -Uhv dhcp-2.0-11.i386.rpm
rpm -Uhv dhcp-client-2.0-11.i386.rpm
rpm -Uhv turbonetcfg-dhcp.1.6.21-1.i386.rpm
```

The following sample `dhcpd.conf` file is rather simple. We designate a default lease time of 600 seconds (10 minutes) but we will let clients request up to a 7200-second (2-hour) lease time. We include a recommended subnet mask of 255.255.255.0 and a broadcast address of 192.168.1.255. Other options we specify include a default gateway (router), two name servers, and the domain.

For our subnet specifics we are using the private 192.168.1.0 class C subnet. For our DHCP pool we will be giving out addresses numbered from 15 to 100 for a total of 85 addresses. The rest can be used by static devices.

```
#!/etc/dhcpd.conf
default-lease-time 600;
max-lease-time 7200;
option subnet-mask 255.255.255.0;
option broadcast-address 192.168.1.255;
option routers 192.168.1.254;
option domain-name-servers 192.168.1.1, 192.168.1.2;
option domain-name "ibm.com";

subnet 192.168.1.0 netmask 255.255.255.0 {
range 192.168.1.15 192.168.1.100;
```

Figure 204. `dhcpd.conf` file

You are not limited to a single subnet. You are allowed to have shared network-specific parameters, multiple subnet-specific parameters, group parameters, and host-specific parameters. You can define multiple ranges, assign specific IP addresses based on the hardware address of the client, and specify a WINS server if needed.

For more information, type `man dhcpd.conf`, or you can review the relevant RFCs and another example `dhcpd.conf` file in the directory `/usr/share/doc/packages/dhcp-2.0/`.

The DHCP server needs a place to keep track of leases. The `/var/state/dhcp/dhcpd.leases` file needs to be created to successfully start the DHCP daemon:

```
touch /var/lib/dhcp/dhcpd.leases
```

To start the DHCP daemon, type:

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



For debugging information, use the -d -f flags.



---

## Chapter 11. NFS - Network File System

The Network File System (NFS), developed by Sun Microsystems, allows you to share directories across the network. The directory mounts become transparent to you. You access the mounted directories just as you do any directory or file system on your computer. The mounting process is the same as for any file system or partition that you want to mount on your system. The basic foundation of this is the mount command.

In order to share directories across the network you will need two basic things:

- The system sharing the data must allow you to have access
- The system that is using the data must originate the request and allow the mount to happen

Both concepts will be discussed in this chapter.

---

### 11.1 The NFS process

First you need to verify that the NFS packages have been loaded. You can do this with the commands:

```
rpm -q knfsd
rpm -q knfsd-clients
rpm -q turbonetcfg-nfsexports
```

If they are not installed, you can mount the main TurboLinux CD and install them with the following commands

```
rpm -Uhv knfsd-1.4.7-2.i386.rpm
rpm -Uhv knfsd-clients--1.4.7-2.i386.rpm
rpm -Uhv turbonetcfg-nfsexports-1.6.21-1.i386.rpm
```

NFS makes use of several daemons. Those daemons are:

- **portmap**: This is the process that converts Remote Procedure Call (RPC) program numbers into Defense Advanced Research Projects Agency (DARPA) protocol port numbers. When a client wishes to make an RPC call to a given program number (for example, the NFS server), it will first contact portmap on the server machine to determine the port number where RPC packets should be sent.
- **rpc.mountd**: This handles the exporting of NFS file systems. It looks in the `/etc/exports` file to figure out what to do with mount requests from various hosts.

- **rpc.nfsd**: This provides the user level part of the NFS process.
- **rpc.rquotad**: This handles quotas for access to file systems. The quotas are based on disk usage and can be hard or soft limits.

You can verify that the `rpc.nfsd`, `rpc.mountd`, and `portmap` daemons are running as shown below.

```
# ps ax | grep nfs
323 ?      SW      0:00 [nfsd]
324 ?      SW      0:00 [nfsd]
325 ?      SW      0:00 [nfsd]
326 ?      SW      0:00 [nfsd]
327 ?      SW      0:00 [nfsd]
328 ?      SW      0:00 [nfsd]
329 ?      SW      0:00 [nfsd]
330 ?      SW      0:00 [nfsd]
662 ttypl  S       0:00 grep nfs
#
# ps ax | grep mount
313 ?      SW      0:00 [rpc.mountd]
673 ttypl  S       0:00 grep mount
#
# /etc/rc.d/init.d/portmap status
portmap (pid 192) is running...
```

Figure 205. Verifying the NFS daemons

If the `portmap` daemon is not running, you need to start it up first before you start up the NFS daemons. You can do this with the command:

```
/etc/rc.d/init.d/portmap start
```

Once the `portmap` daemon is running, you can start up the NFS daemons with the command:

```
/etc/rc.d/init.d/nfs start
```

```
# /usr/rc.d/init.d/nfs start
Starting NFS services: Starting NFS quotas: rpc.rquotad
Starting NFS mountd: rpc.mountd
Starting NFS daemon: rpc.nfsd
```

Figure 206. Starting up NFS

**Note**

If the `/etc/exports` file does not exist or is empty, the NFS daemons will not start. Information on setting up the `/etc/exports` file is in 11.3, “Sharing data with NFS: command-line process” on page 217.

To stop the NFS server use the command:

```
/etc/rc.d/init.d/nfs stop
```

The results are shown in Figure 207. You will notice that some processes are shut down, but not necessarily in the same order. The quota process is started up first because the quotas need to be established before the mounts take place. It is shut down last, so that nothing can slip past the quota process.

```
# /etc/rc.d/init.d/nfs stop
Shutting down NFS services:
Shutting down NFS mountd: rpc.mountd
Shutting down NFS daemon: nfsd
Shutting down NFS quotas: rpc.rquotad
#
```

Figure 207. Stopping the NFS server

You can restart the NFS process with the command:

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

This can also be used to restart the NFS process if you have made changes to the configuration files.

---

## 11.2 Using `turbonetcfg` to share data with NFS

TurboLinux 6 provides an easy, yet restricted, method of sharing directories with NFS. The following figures show you how to export a directory.

From the main `turbonetcfg` window, choose **NFS Export Config**. That will bring you to Figure 208. Here choose **Add** and proceed to the next window.

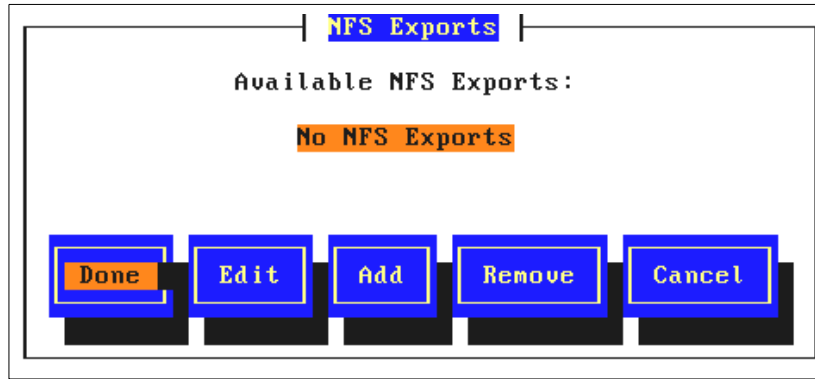


Figure 208. NFS Exports window

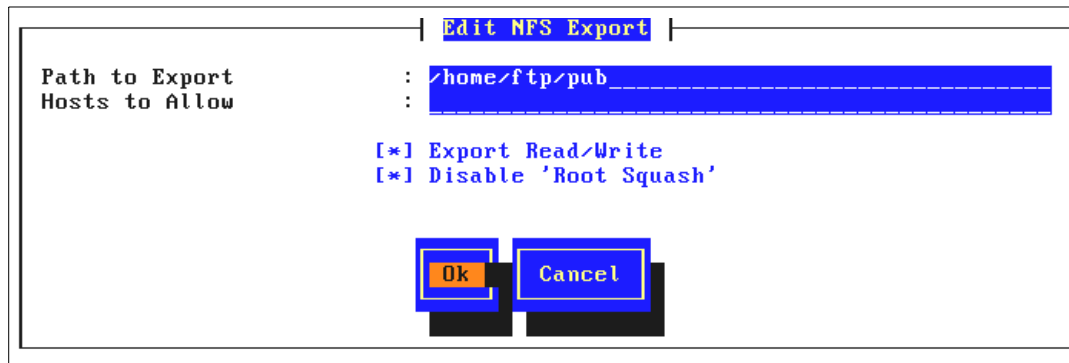


Figure 209. Edit NFS Export window

In Figure 209, we have exported the directory `/home/ftp/pub` to everyone (since the line **Hosts to Allow** is blank), and selected the option **Export Read/Write**. Note that even though we have allowed the NFS server to give read/write access to everyone, the NFS server will still have to get read access from the file system. For example, the directory we are exporting defaults to the permissions:

```
drwxr-sr-x root ftp
```

This means that root has write access, but no other user does. Since from a permissions point of view NFS is considered a user of the system, you will have to add write access to everyone in order to allow NFS users to write to this directory.

We also selected the option **Disable 'Root Squash'**, which access options are explained in Table 23 on page 218. After clicking **OK** to save this window, look at `/etc/exports`. It should contain the line

```
/home/ftp/pub (rw,no_root_squash)
```

This is the content of the export we have just added.

---

### 11.3 Sharing data with NFS: command-line process

Like most administration in Linux, administering from the command line allows more control over the details of the system. In this section we will give some examples of the configuration possible by editing the configuration file `/etc/exports` directly.

Create a sample file entry by opening the `/etc/exports` file. Then add the entry:

```
/usr/local/share myserver.mydomain.com(ro)
```

This says that the directory `/usr/local/share` is only accessible to the server `myserver.mydomain.com`.

#### Note

When exporting a file system you need to be sure that the exporting server can recognize and access the server that is in the `/etc/exports` file. You can verify this with the command:

```
ping server_name
```

Where `server_name` is the name of the server you are trying to access. Otherwise, the NFS commands may hang.

There are a number of options you can set up in the `/etc/exports` file. Some of them are listed in Table 23.

You need to be sure that the exporting server can recognize the server name.

The various options are explained in the table below.

Table 23. Access options

ro read only	Only permits reading
rw read write	Permits reading and writing. If both ro and rw are specified, rw takes priority.
root_squash client	Anonymous user (nobody) access from client.
no_root_squash client	Access request privileges per the privileges of the client root. Useful for diskless clients.
squash_uids and squash_gids	Specify a list of UIDs or GIDs that should be subject to anonymous mapping. A valid list of IDs looks like this: squash_uids=0-15,20,25-50
all_squash all access	Processes all requests for access as anonymous user.
anonuid=uid	root_squash or all_squash when options are set will assign a group ID to an anonymous user request.
anonuid=gid	root_squash or all_squash when options are set will assign a group ID to an anonymous user request.

A sample `/etc/exports` file is shown in the man pages for `exports` and in Figure 210.

```
# sample /etc/exports file
/          master(rw) trusty(rw,no_root_squash)
/projects  proj*.local.domain(rw)
/usr       *.local.domain(ro) @trusted(rw)
/home/joe  pc001(rw,all_squash,anonuid=150,anongid=100)
/pub       (ro,insecure,all_squash)
/pub/private (noaccess)
```

Figure 210. A sample `/etc/exports` file



The lines in the sample `/etc/exports` file are explained as follows:

- `# sample /etc/exports file`

This is just a comment. Any line or character string can be converted to a comment and disabled by entering a `#` symbol. Everything from that point to the end of the line is considered to be a comment.

- `/ master(rw) trusty(rw,no_root_squash)`

This says that the root directory (`/`) is exported to the servers:

- `master` - whose rights are read-write

- `trusty` - whose rights are read-write and the access rights of the client root can be the same as the server's root

- `/projects proj*.local.domain(rw)`

The directory `/projects` is read-write accessible to all servers whose names match the pattern `proj*.local.domain`. This includes `proj.local.domain`, `proj1.local.domain`, `projproj.local.domain` and so forth.

- `/usr *.local.domain(ro) @trusted(rw)`

Any systems whose hostname ends in `.local.domain` is allowed read-only access. The `@trusted` netgroup is allowed read-write access.

- `/home/joe pc001(rw,all_squash,anonuid=150,anongid=100)`

The directory `/home/joe` is accessible to `pc001` for read-write access; all requests for access are processed as anonymous users. The anonymous UID number is set to 150 and the anonymous group ID is set to 100. This is useful when using a client that is running PCNFS or an equivalent NFS process on the PC. Since the PC IDs do not necessarily map to the UNIX IDs, this allow the proper file attributes to be set.

- `/pub (ro,insecure,all_squash)`

The directory `/pub` is accessible as read-only. The option in this entry also allows clients with NFS implementations that don't use a reserved port for NFS and process all requests as an anonymous user.

- `/pub/private (noaccess)`

The directory `/pub/private` does not allow any NFS access.

---

## 11.4 Accessing data remotely with NFS - the command line view

To mount a remote filesystem on your local system the mount point must exist. The mount process does not create the mount point automatically. The

process of making the mount point is to use the Linux `mkdir` command. To make the `/usr/local/share` mount point, enter:

```
mkdir /usr/local/share
```

Typically you do not need to worry about file attributes and ownership when making an NFS mount point. The NFS access rights will usually supersede any rights established for the directory.

Once you have created the mount point then you can use the `mount` command as follows:

```
mount -t nfs nfs_host:share_dir local_mount_dir
```

where:

<code>-t nfs</code>	Says to do the mount as an NFS mount. This is now optional because if you explicitly specify the directory to be mounted as <code>host:directory</code> the <code>mount</code> command knows that it is an NFS mount.
<code>nfs_host</code>	Is the host that is exporting the file system to be shared.
<code>share_dir</code>	Is the actual directory that is to be shared.
<code>local_mount_dir</code>	Is the directory on the local host where the remote directory is going to be mounted. As mentioned earlier, this mount point must exist.

---

## Chapter 12. NIS - Network Information System

In a distributed computing environment, maintenance of password, group, and host files can be a major task. Consistency is possibly the biggest difficulty here. For example, when a user changes his password on one machine, ideally it would be propagated to any other machine he has accounts on. When a network is composed of hundreds or thousands of machines, this convenience becomes a necessity. NIS is one way of addressing some of these problems.

---

### 12.1 What is NIS?

The Network Information System (NIS) is a service designed to provide a distributed database system for common configuration files. It was formerly known as Sun Yellow Pages (YP). NIS servers manage copies of the database files. NIS clients request the information from the NIS server instead of using their own configuration files.

NIS is designed after the client/server model. A NIS server contains data files called "maps". These maps are owned by the NIS master and can only be updated by the master. There are NIS slave servers that replicate from the master. When there is a change to a master server's map, this change is then distributed to all the slave servers. Clients are hosts that request information from these maps but are not allowed to modify them locally.

NIS is commonly used in UNIX environments. However, it is also possible to integrate Windows NT clients in a NIS-based environment. NISGINA provides a NIS authenticated interactive logon for Windows NT 4.0 workstations. It supports changing UNIX passwords using a Windows NT dialog and some limited remote registry configuration. You can find it at the author's Web page at:

<http://www.dcs.qmw.ac.uk/~williams/>

---

### 12.2 How can I use NIS?

NIS is typically used to centrally manage commonly replicated configuration files. Examples of common configuration files are:

- /etc/hosts
- /etc/passwd
- /etc/group

---

## 12.3 Implementation on TurboLinux

To introduce the concepts behind NIS, we will create a map of our password file kept on the NIS master server. This will allow users to log in to NIS clients without having to maintain an account on each system. Centralized administration is a key benefit of using NIS.

A note on security: Before deciding to put NIS in a production environment, please consider the security implications of passing sensitive data across the network. You may wish to take a look at NIS+, which has strong encryption as well as additional maintenance implications. You may also consider LDAP, which is described Chapter 13, “LDAP - Lightweight Directory Access Protocol” on page 231. To use NIS, ensure that the proper packages have been installed by using the command `rpm -q [package name]`.

Packages that need to be installed to be an NIS client:

```
yptools-2.3-2.i386.rpm  
ypbind-3.3.16.i386.rpm
```

Packages that need to be installed to be an NIS server:

```
ypserv.1.3.9-2.i386.rpm  
make-3.78.1-3.i386.rpm
```

If any of these packages are not installed, mount the TurboLinux 6 CD and install them with the following commands:

```
mount /mnt/cdrom  
cd /mnt/cdrom/TurboLinux/RPMS  
rpm -Uhv [each package listed above]
```

### 12.3.1 Using the nsswitch file for lookups

The nsswitch file determines the order of lookups performed. A sample file of nsswitch.conf is shown in Figure 211:

```
# /etc/nsswitch.conf

passwd: files nis
shadow: files nis
group:   files nis
hosts:   files dns
bootparams: files
ethers:   files
netmasks: files
networks: files
protocols: files
rpc:      files
services: files
#netgroup: nisplus
#publickey: nisplus
automount: files
aliases:  files
```

Figure 211. *nsswitch.conf* file

### 12.3.2 NIS server

A key configuration file for the NIS master server is the `/etc/ypserv.conf` file. Uncomment the `passwd.byname` line, but leave the `passwd.byuid` commented. Figure 212 is a sample `ypserv.conf` file we used:

```

#ypserv.conf - In this file you can set certain options for the NIS
server,
#and you can deny or restrict access to certain maps based
#on the originating host.
#See ypserv.conf(5) for a description of the syntax.
dns: no
# The following, when uncommented, will give you shadow like
passwords.
# Note that it will not work if you have slave NIS servers in your
# network that do not run the same server as you.
# Host                : Map                : Security   :
Passwd_mangle
*                    : passwd.byname   : port       : yes
# *                   : passwd.byuid    : port       : yes
# Not everybody should see the shadow passwords, not secure, since
# under MSDOG everybody is root and can access ports < 1024 !!!
* : shadow.byname    : port        : yes
* : passwd.adjunct.byname : port      : yes
# If you comment out the next rule, ypserv and rpc.ypxfrd will
# look for YP_SECURE and YP_AUTHDES in the maps. This will make
# the security check a little bit slower, but you only have to
# change the keys on the master server, not the configuration files
# on each NIS server.
# If you have maps with YP_SECURE or YP_AUTHDES, you should create
# a rule for them above, that's much faster.
*                    : *           : none

```

Figure 212. *ypserv.conf* file

The other key configuration file is the `/var/yp/Makefile`. The only map we want to create is the `/etc/passwd` file, so the others can be commented out if you wish; however, the default Makefile works just fine. In Figure 213 is a sample `/var/yp/Makefile`:

```

# Makefile for the NIS databases
# This Makefile should only be run on the NIS master server of a
domain.
# All updated maps will be pushed to all NIS slave servers listed in
the
# /var/yp/ypservers file. Please make sure that the hostnames of all
# NIS servers in your domain are listed in /var/yp/ypservers.
# This Makefile can be modified to support more NIS maps if desired.
# Set the following variable to "-b" to have NIS servers use the
domain
# name resolver for hosts not in the current domain. This is only
needed,
# if you have SunOS slave YP server, which gets here maps from this
# server. The NYS YP server will ignore the YP_INTERDOMAIN key.
#B=-b
B=
# If we have only one server, we don't have to push the maps to the
# slave servers (NOPUSH=true). If you have slave servers, change this
# to "NOPUSH=false" and put all hostnames of your slave servers in
the file
# /var/yp/ypservers.
NOPUSH=true
# We do not put password entries with lower UIDs (the root and system
# entries) in the NIS password database, for security. MINUID is the
# lowest uid that will be included in the password maps.
# MINGID is the lowest gid that will be included in the group maps.
MINUID=500
MINGID=500
# Should we merge the passwd file with the shadow file ?
# MERGE_PASSWD=true|false
MERGE_PASSWD=true
# Should we merge the group file with the shadow file ?
# MERGE_GROUP=true|false
MERGE_GROUP=true
# These are commands which this Makefile needs to properly rebuild
the
# NIS databases. Don't change these unless you have a good reason.
AWK = /usr/bin/gawk
MAKE = /usr/bin/gmake
UMASK = umask 066
# These are the source directories for the NIS files; normally
# that is /etc but you may want to move the source for the password
# and group files to (for example) /var/yp/ypfiles. The directory

```

Figure 213. Makefile file

```

# for passwd, group and shadow is defined by YPPWDDIR, the rest is
# taken from YPSRCDIR.
YPSRCDIR = /etc
YPPWDDIR = /etc
YPBINDIR = /usr/lib/yp
YPSBINDIR = /usr/sbin
YPDIR = /var/yp
YPMAPDIR = $(YPDIR)/$(DOMAIN)
# These are the files from which the NIS databases are built. You may
edit
# these to taste in the event that you wish to keep your NIS source
files
# seperate from your NIS server's actual configuration files.
GROUP      = $(YPPWDDIR)/group
PASSWD     = $(YPPWDDIR)/passwd
SHADOW     = $(YPPWDDIR)/shadow
GSHADOW   = $(YPPWDDIR)/gshadow
ADJUNCT    = $(YPPWDDIR)/passwd.adjunct
#ALIASES   = $(YPSRCDIR)/aliases # aliases could be in /etc or
/etc/mail
ALIASES    = /etc/aliases
ETHERS     = $(YPSRCDIR)/ethers # ethernet addresses (for rarpd)
BOOTPARAMS = $(YPSRCDIR)/bootparams # for booting Sun boxes
           (bootparamd)
HOSTS      = $(YPSRCDIR)/hosts
NETWORKS   = $(YPSRCDIR)/networks
PROTOCOLS  = $(YPSRCDIR)/protocols
PUBLICKEYS = $(YPSRCDIR)/publickey
RPC        = $(YPSRCDIR)/rpc
SERVICES   = $(YPSRCDIR)/services
NETGROUP   = $(YPSRCDIR)/netgroup
NETID      = $(YPSRCDIR)/netid
AMD_HOME   = $(YPSRCDIR)/amd.home
AUTO_MASTER = $(YPSRCDIR)/auto.master
AUTO_HOME  = $(YPSRCDIR)/auto.home
YPSERVERS = $(YPDIR)/ypservers# List of all NIS servers for a domain
target: Makefile
@test ! -d $(LOCALDOMAIN) && mkdir $(LOCALDOMAIN) ; \
cd $(LOCALDOMAIN) ; \
$(NOPUSH) || $(MAKE) -f ../Makefile ypservers; \
$(MAKE) -f ../Makefile all
# If you don't want some of these maps built, feel free to comment
# them out from this list.

```

Figure 214. Makefile continuation



```
all: passwd group hosts rpc services netid protocols netgrp mail \  
#shadow publickey # networks ethers bootparams amd.home \  
auto.master auto.home passwd.adjunct  
#####  
####  
# DON'T EDIT ANYTHING BELOW IF YOU DON'T KNOW WHAT YOU ARE DOING !!!  
#  
#####  
####
```

Figure 215. Makefile end of file

Take a look at the `/var/yp/securenets` file, which defines the access rights to your NIS server. By default it is set to give access to everyone. Change it accordingly (see the man `securenets` file).

The Makefile will look for `/etc/netgroup`. We need to create this file so that the databases will be created successfully.

```
touch /etc/netgroup
```

We need to set our `nisdomain` name. This domain name should not be confused with DNS domain names! The YP domain name can be any generic name.

```
# nisdomainname nis.com
```

Edit the `/etc/sysconfig/network` file, adding or modifying the following entry:

```
NISDOMAIN="nis.com"
```

You also have to define which hosts should be allowed to contact the NIS server. In our example, we will allow all hosts from the local Class C network `192.168.99.0/24` to connect to the server.

Open `/etc/hosts.allow` in a text editor and add the following line:

```
ypserv: 127.0.0.0/255.0.0.0 192.168.99.0/255.255.255.0
```

It is imperative that the local host also be allowed to connect to the `ypserv` process via the loopback interface (`127.0.0.1`).

Now add the following line to `/etc/hosts.deny`:

```
ypserv: ALL
```

We are now ready to start the `ypserv` daemon:

```
ypserv (-d for debug information)
```

To test our NIS setup we will use the `rpcinfo` command:

```
rpcinfo -u localhost ypserv
```

You should see:

```
program 100004 version 1 ready and waiting
program 100004 version 2 ready and waiting
```

We will now create our NIS maps:

```
/usr/lib/yp/ypinit -m (to create the DB)
(ypinit -s masterhost to add a slave server)
```

The localhost will be selected as the master server.

Press `<ctrl> d`

Select `y` to confirm and begin building your maps.

To test our NIS master server, we need to set up a client to run `ypbind`. For simplicity we can use the master server to verify our configuration. The same steps should be followed to set up a remote client.

### 12.3.3 NIS Client

We need to edit the `/etc/yp.conf` file with our entries for the NIS domain and the NIS master server. For our test domain we used `nis.com` and our master server name is `nismaster`. The following is a sample `yp.conf` file:

```
# /etc/yp.conf - ypbind configuration file
# domain nis.com server nismaster
# Use server HOSTNAME for the domain NISDOMAIN.
# domain nis.com broadcast
# Use broadcast on the local net for domain NISDOMAIN
ypserver nismaster
# Use server HOSTNAME for the local domain. The
# IP-address of server must be listed in /etc/hosts.
```

Make sure the NIS master server is listed in the `/etc/hosts` file.

If you did not set the NIS domain name in the server section, you will need to do it now:

```
# nisdomainname nis.com
```

Edit the `/etc/sysconfig/network` file, adding or modifying the following entry:

```
NISDOMAIN="nis.com"
```

We are now ready to start the ypbind daemon:

```
ypbind (-d for debug information)
```

To test our NIS configuration we will use the `yppcat` command:

```
yppcat passwd
```

You should see output similar to Figure 216:

```
[root@test2 /root]# yppcat passwd
jakob:$1$J1uURHMH$GdSCfVKE1hCiZH3Gcgqdp.:503:503::/home/jakob:/bin/bash
ted:$1$Bmzad0.Y$7IEwMdTIFsrDMhuUc2L2n.:506:506::/home/ted:/bin/bash
rufus:$1$BsSuXAZQ$GJQ8t/Bwr3MHuzIFn43Z01:502:502::/home/rufus:/bin/bash
test:$1$gX.BvMF7$ozm1x8zLON7nRKUrq9J9K0:501:501::/home/test:/bin/bash
jhaskins:$1$o6QFi5X.$PSc/GgsdCS511Xp0LkF76/:500:500::/home/jhaskins:/bin/bas
georg:$1$qmQqrkKN$goWStNgshq046A8ZfpStQ/:507:507::/home/georg:/bin/bash
ivo:$1$ZeQWU09J$HuekZrXrw/3vJoxMAK6PH1:504:504::/home/ivo:/bin/bash
justin:$1$UVE2w0dR$/0/11CX4TBHhDXpo0d117/:505:505::/home/justin:/bin/bash
[root@test2 /root]# █
```

Figure 216. `yppcat passwd`

Now to really test the machine, log in to an NIS client using an account that is on the NIS master. When you log in, you should see the following:

```
[root@test1 RPHS]#telnet 192.168.1.2
Trying 192.168.1.2...
Connected to 192.168.1.2
Escape character is '^'_

Kernel 2.2.16 22 on an 1606
Login: jhaskins
Password:
No directory /home/jhaskins
Logging in with home - "/"
bash-2-04$
```

Figure 217. No home directory

Since `jhaskins`'s home directory is on `nismaster`, we get an error logging in. This can be fixed by creating a home directory for `jhaskins` on the client if necessary. Another option would be to use NFS in conjunction with NIS to automatically mount `jhaskins`'s home directory.

---

## 12.4 Sources of additional information

For further information or troubleshooting, *Managing NFS and NIS* by Hal Stern is good resource. The NIS how-to by Thorsten Kukuk is an excellent place to start. Find it at:

<http://www.metalab.unc.edu/pub/Linux/docs/HOWTO/NIS-HOWTO>.

---

## Chapter 13. LDAP - Lightweight Directory Access Protocol

LDAP has become a buzzword in the IT world. The exciting thing about LDAP and directory services is that they can be used for so many purposes. This chapter will give you a brief explanation of what LDAP is, what it can be used for, basic structures, and simple implementation on the Linux OS. This chapter merely scratches the surface of what is actually possible with LDAP.

---

### 13.1 What is LDAP?

LDAP stands for Lightweight Directory Access Protocol. LDAP has become an Internet standard for directory services that run over TCP/IP. LDAP is a client/server protocol for accessing a directory service. Originally designed as a front-end for X.500 databases, LDAP is now commonly used in a stand-alone capacity. IBM, Netscape, Sun, Novell, Microsoft, and many other companies are incorporating LDAP into their directory structures.

#### 13.1.1 Directory Services

A directory service is the collection of software, hardware, processes, policies, and administrative procedures involved in making the information in a directory available to the users of the directory.

A directory is similar to a database. However, directories and databases differ in the number of times they are searched and updated. Directories are tuned for being searched, while relational databases are geared toward maintaining data with a frequent number of updates.

Examples of directories would be the Yellow Pages, a card catalog, or an address book. Information is organized in a defined hierarchy and given attributes.

When we place a directory online, the data becomes dynamic in the sense that it can be easily updated and cross-referenced. Unlike printed material, any updates that occur are instantaneous for all users.

You can apply security to the directory so that only intended users can view, modify, or create data. This security can be based upon groups, individual users, or any other authentication scheme. The data can also be encrypted.

Directory services typically involve data distribution and replication. The advantages of distributing your directory services are performance, availability, and reliability. For a segmented network, distribution of servers containing the directory data improves performance by reducing network

traffic and load on individual servers. By replicating your data on multiple servers you increase availability in case a single server should go down.

### **13.1.2 X.500**

In the mid-1980s, the International Telecommunications Union (ITU, formerly the CCITT) and the International Organization for Standardization (ISO) merged their efforts on directory services standards and created X.500. The X.500 specifications consist of a series of recommendations on the concepts, models, authentication, distribution, attributes, objects, and replication that underlie an X.500 directory service.

Early X.500 implementations used a client access protocol known as DAP. DAP is thick, complicated, and difficult to implement for desktop computers. For all of these reasons other lighter-weight protocols were developed. As predecessors to LDAP, DIXIE and DAS were very successful. Out of this success a group from the Internet Engineering Task Force (IETF) began work on LDAP. The first Request for Comments (RFC 1487) describing LDAP was released in July 1993.

---

## **13.2 How can I use LDAP?**

LDAP can allow system and network administrators to centrally manage users, groups, devices, and other data. IT decision makers can avoid tying themselves to a single vendor for applications and operating systems. Developers can use LDAP-based standards to ensure cross-platform integration.

Some practical applications of LDAP-based directory services include:

- Corporate address book
- User authentication
- Domain Name System

---

## **13.3 LDAP basics**

The LDAP information model is based on objects. Objects can be people, printers, servers, or just about anything you can think of. The most basic unit of the LDAP model is the entry. An entry is a collection of information about an object. Each entry belongs to an object class that determines required and optional attributes. Each attribute has a type and one or more values. The type describes the kind of information contained in the attribute and the value contains the actual data.

An LDIF file is the standard way of representing directory data in a textual format. This format can typically be used for importing and exporting directory data. The following is an LDIF file for loading a basic LDAP directory and adding a user that we can use to test our authentication.

```
dn: ou=people, dc=ibm, dc=com
objectclass: top
objectclass: organizationalUnit
description: users who authenticate via the ldap server

dn: uid=ldaptest, ou=People, dc=ibm dc=com
uid: ldaptest
cn: ldaptest
objectclass: account
objectclass: posixaccount
objectclass: top
objectclass: shadowaccount
userpassword: test123
shadowLastchange:11263
shadowMax: 99999
shadowWarning: 7
loginShell: /bin/bash
uidNumber: 500
gidNumber: 500
homeDirectory: /home/testldap
```

Figure 218. LDIF file

Each LDAP entry must have a distinguished name (DN). The distinguished name is a unique key that refers to that entry specifically.

The first group of entries is to create the root or base entries in the directory. In this case we are using com (domain name or dn), ibm (dn), and people (organizational unit).

Within the organizational unit, people we will store the users and their corresponding authentication information. The second group of entries is to add the user ldaptest and the attributes that are necessary for authentication.

---

## 13.4 Implementation on TurboLinux

You can download the latest stable version of OpenLDAP from:

[ftp.openldap.org/pub/openldap](ftp://openldap.org/pub/openldap)

However, we will be using the rpms that come with the TurboLinux 6 distribution, so the first step is to use the command `rpm -q [package name]` to confirm that the following packages are installed:

```
openldap-1.2.10-1.i386.rpm
openldap-libs-1.2.10-1.i386.rpm
openldap-servers-1.2.10-1.i386.rpm
nss_ldap-97-3.i386.rpm
```

If any of these packages are not installed, mount the main TurboLinux CD and install them with the commands:

```
mount /mnt/cdrom
cd /mnt/cdrom/TurboLinux/RPMS
rpm -Uhv [each package listed above]
```

### 13.4.1 slapd.conf

Now edit the `/etc/openldap/slapd.conf` file. Replace `ibm.com` with the name of your organization.



```
#
# See slapd.conf(5) for details on configuration options.
# This file should NOT be world readable.
#
include /usr/local/etc/openldap/slapd.at.conf
include /usr/local/etc/openldap/slapd.oc.conf
schemacheckoff
pidfile      /var/run/slapd.pid
argsfile     /var/run/slapd.args

#####
# ldbm database definitions
#####

database      ldbm
suffix        "ou=people, dc=ibm, dc=com"
rootdn        "cn=admin, ou=people, dc=ibm, dc=com"
# cleartext passwords, especially for the rootdn, should
# be avoid. See slapd.conf(5) for details.
#rootpw       secret
rootpw        {crypt}wDE3PV/c1yRm.
# database directory
# this directory MUST exist prior to running slapd AND
# should only be accessible by the slapd/tools Mode 700 recommended.
directory     /var/lib/ldap
```

Figure 219. slapd.conf file

**Note**  
To generate an encrypted password for the rootpw or for adding a user to the LDAP directory for authentication, use the `openssl` command:

```
# openssl passwd -crypt secret
wDE3PV/c1yRm.
```

### 13.4.2 ldap.conf

When the `nss_ldap` rpm was installed it created the file `/etc/ldap.conf.rpmnew`. Copy `/etc/ldap.conf.rpmnew` to `/etc/ldap.conf`. Now edit the `/etc/ldap.conf` file. Modify the host and base entries. In this case the LDAP server we will be authenticating against is on the localhost. Replace `padl` with the name of your organization. In our example, we used `ibm`. All other entries can be left with the default values.

```

# This is the configuration file for the LDAP nameservice
# switch library and the LDAP PAM module.
# To contact the developers, mail support@padl.com.
# If the host and base aren't here, then the DNS RR
# _ldap._tcp.<defaultdomain>. will be resolved. <defaultdomain>
# will be mapped to a distinguished name and the target host
# will be used as the server.

# Your LDAP server. Must be resolvable without using LDAP.
host 127.0.0.1

# The distinguished name of the search base.
base ou=people,dc=ibm,dc=com

# The LDAP version to use (defaults to 2)
#ldap_version 3
# The distinguished name to bind to the server with.
# Optional: default is to bind anonymously.
#binddn cn=manager,dc=padl,dc=com
# The credentials to bind with.
# Optional: default is no credential.
#bindpw secret
# The port.
# Optional: default is 389.
#port 389
# The search scope.
#scope sub
#scope one
#scope base
# The following options are specific to nss_ldap.
# The hashing algorithm your libc uses.
# Optional: default is des
#crypt md5
#crypt sha
#crypt des
# The following options are specific to pam_ldap.
# Filter to AND with uid=%s
#pam_filter objectclass=account
# The user ID attribute (defaults to uid)
#pam_login_attribute uid
# Search the root DSE for the password policy (works
# with Netscape Directory Server)
#pam_lookup_policy yes

```

Figure 220. *ldap.conf* file

```
*# Group to enforce membership of
#pam_groupdn cn=PAM,ou=Groups,dc=padl,dc=com
# Group member attribute
#pam_member_attribute uniquemember
# Hash password locally; required for University of
# Michigan LDAP server, and works with Netscape
# Directory Server if you're using the UNIX-Crypt
# hash mechanism and not using the NT Synchronization
# service.
#pam_crypt local
```

Figure 221. *Ldap.conf* file end

### 13.4.3 nsswitch.conf

The `nsswitch.conf` file controls which type of name service your host will use to look up various types of entries. With LDAP you can easily set up Domain Name System and host lookup tables. In this example we need to modify the entry for `passwd`. Edit the `/etc/nsswitch.conf` file and add `ldap` to the entry for `passwd`:

```
passwd: files ldap
```

### 13.4.4 /etc/pam.d/login

PAM is a system of libraries that handle the authentication tasks of services on the system. PAM separates the tasks of authentication into four independent management groups: `account`, `authentication`, `password`, and `session`.

- `account`: provides account verification. Examples: Checking for password expiration or verifying that the user has permission to access the requested service.
- `authentication`: establish the user is who he claims to be, typically by prompting for a password.
- `password`: authentication update mechanism. Example: Prompting the user to enter a new password when the current password has expired.
- `session`: tasks that should be done prior to a service being given and after it is withdrawn. Examples: audit trail maintenance and mounting of the user's home directory.

To enable the PAM modules for accessing the LDAP directory edit the `/etc/pam.d/login` file:

auth	required	/lib/security/pam_securetty.so
auth	required	/lib/security/pam_nologin.so
auth	sufficient	/lib/security/pam_ldap.so
auth	required	/lib/security/pam_unix_auth.so try_first_pass
account	sufficient	/lib/security/pam_ldap.so
account	required	/lib/security/pam_unix_acct.so
password	required	/lib/security/pam_cracklib.so
password	required	/lib/security/pam_ldap.so
password	required	/lib/security/pam_pwdb.so use_first_pass
session	required	/lib/security/pam_unix_session.so
session	optional	/lib/security/pam_console.so

Figure 222. login file

For more information on PAM, see the PAM Administrators guide at:  
<http://www.kernel.org/pub/linux/libs/pam/Linux-PAM-html/pam.html>

### 13.4.5 Starting OpenLDAP

To start slapd, type `/usr/sbin/slapd (-d 255 for debugging information)`.

With slapd successfully running, we now need to load the initial database. Create an LDIF file like the one in Figure 218 on page 233. Replace `ldaptest` with your user name and `ibm` with your organization name.

Once you have created the `entries.ldif` file, load the LDAP server.

```
/usr/bin/ldapadd -f entries.ldif -D "cn=admin, ou=people, dc=ibm, dc=com"
-w secret
```

### 13.4.6 Testing authentication

Now that we've added our test user in the LDAP directory, we need to create the home directory we specified in our LDIF file along with the appropriate ownership.

```
mkdir /home/ldaptest
chown 500:500 /home/ldaptest
```

Now try to log in as `ldaptest` with the password `test`.

If the authentication failed, check the configuration files for typos. It can also be helpful to run `slapd` in debug mode to watch the output of the failed authentication.

Once you have logged in successfully either remove the test user or at the very least create a properly encrypted password.

### 13.4.7 Migrating /etc/passwd

With the installation of `nss_ldap` comes numerous scripts to help migration to LDAP. In keeping with our example for user authentication, we will look at the script that migrates the entire `/etc/passwd` into an LDIF structure which can then be easily put into an LDAP directory. Most of the migration scripts are written in Perl. Perl must be installed in order to run them.

Change the directory to `/usr/share/openldap/migration`:

```
cd /usr/share/openldap/migration/
ls
README                                migratefstab.pl
migrate_aliases.pl                    migrate_group.pl
migrate_all_netinfo_offline.sh         migrate_hosts.pl
migrate_all_netinfo_online.sh          migrate_netgroup.pl
migrate_all_nis_offline.sh             migrate_netgroup_byhost.pl
migrate_all_nis_online.sh              migrate_netgroup_byuser.pl
migrate_all_nisplus_offline.sh         migrate_networks.pl
migrate_all_nisplus_online.sh          migrate_passwd.pl
migrate_all_offline.sh                 migrate_protocols.pl
migrate_all_online.sh                  migrate_rpc.pl
migrate_automount.pl                   migrate_services.pl
migrate_base.pl                         migration-tools.txt
migrate_common.ph
```

Figure 223. `ls` command running on `/usr/share/openldap/migration` directory

The file `migrate_common.ph` keeps common definitions for all of the migration tools. We need to edit the following lines to be compatible with the directory we've set up:

```
# Default DNS domain
$DEFAULT_MAIL_DOMAIN = "padl.com";

# Default base
$DEFAULT_BASE = "dc=padl,dc=com";
```

PADL is the company that developed the Open Source `nss_ldap` module (see <http://www.padl.com> for more information). Replace `padl` with the name of your organization in both the default DNS domain and in the default base.

**Stop**

The LDIF file created by the migration script will have the encrypted passwords of all the users on the system, including root. Treat this file as you would the `/etc/shadow` file.

Now, as root, run the `migrate_passwd.pl` script against `/etc/passwd`.

```
./migrate_passwd.pl /etc/passwd /root/passwd.ldif
```

The final argument is the location and name of the LDIF file we want to create.

Edit the LDIF file and remove any users you do not want added to the LDAP directory.

Now to add the entire contents of the LDIF file, execute the following command:

```
ldapadd -f /root/passwd.ldif -D "cn=admin, ou=people, dn=ibm, dn=com" -w secret
```

Replace the `admin`, `ibm`, and `secret` with the appropriate entries.

After testing, you can remove those users from `/etc/passwd` and `/etc/shadow` and use LDAP to all of your authentication.

In a large environment you can set up your servers to authenticate against a single centralized LDAP directory server or create replicas. Either way, you have just made user administration a lot simpler.

---

## Chapter 14. General performance tools in Linux

Linux offers a great variety of ways to optimize your system for maximum performance. Apart from the general fact that it is always good to have as much RAM and the fastest CPU as possible, there are some additional parameters to tune a Linux system. This section is intended as a collection of useful hints and tools, but without getting into too much detail about them. Please refer to the respective documentation and references. You should also note that using some of these hints may render your system unstable; use them at your own risk and only if you know what you are doing.

---

### 14.1 General configuration hints

These are some general tips for tweaking your system to maximize performance.

Recompile your programs and the Linux kernel with all available compiler optimization flags (for example, `-funroll-loops`, `-fomit-frame-pointer`, `-O6`) and all architecture-specific compiler options for your hardware architecture. This may increase the size of binaries or make them unable to run on some processors, but you can gain a lot of speed in comparison with the binaries shipped in the distribution. Alternatively you could use special compilers for your architecture (for example, `pgcc`), which offer even more sophisticated optimization options.

Create swap partitions of equal priority but different hard disk drives to allow load balancing. Please note that they need to be different devices! Using two different partitions on one hard disk will have the reverse effect. Even better, try to avoid swapping at all by adding more memory. A busy server should never need to swap, as this would severely degrade the overall performance.

If you are running a heavily loaded server with a lot of parallel processes, you might run into the Linux kernel's limit of running processes (512 by default). This maximum number of tasks is configurable in the kernel sources, so you have to recompile the kernel after changing this value. This value is defined in the file `/usr/src/linux/include/linux/tasks.h`:

```
#define NR_TASKS          512
```

You can increase this value up to 4090 processes, if necessary.

Linux offers a filesystem mount option that is called `noatime`. The `atime` is a timestamp of the last access time (reading and writing) for a certain file. This option can be added to the mount options in the `/etc/fstab` file. When a file

system is mounted with this option, read accesses to files will no longer result in an update of the inode access time information. This information is usually not very interesting on a file or Web server, so the lack of updates to this field is not relevant. The performance advantage of the `noatime` flag is that it suppresses write operations to the filesystem for files that are simply being read. Since these write accesses add additional overhead, this can result in measurable performance gains. Instead of specifying this as a mount option that would apply to the whole filesystem, you can use the command `chattr` to set this flag on single files or directories. For example:

```
chattr -R +A /var/spool/news
```

This command would set the `noatime` flag recursively on all files below the news spool directory (a very common practice on busy news servers). See the manual page `chattr(1)` for more information.

You can use the `hdparm` tool to tune some hard disk drive parameters. Unfortunately most of them only work on IDE systems (which should be avoided in server systems, anyway), but the option `-a` works for SCSI, too. The manual page describes it as follows: “This option is used to get/set the sector count for file system read-ahead. This is used to improve performance in sequential reads of large files, by prefetching additional blocks in anticipation of them being needed by the running task. The default setting is 8 sectors (4 KB). This value seems good for most purposes, but in a system where most file accesses are random seeks, a smaller setting might provide better performance. Also, many drives have a separate built-in read-ahead function, which alleviates the need for a file system read-ahead in many situations.” For example, to set the sector count read-ahead of your first SCSI disk to 4 sectors (2 KB), you would use the following command:

```
hdparm -a 4 /dev/sda
```

See the `hdparm` manual page for a complete list of available options.

### 14.1.1 Powertweak

Powertweak is a general-purpose application to allow you to “tweak” different system parameters. Though it does not ship with TurboLinux, it can be downloaded from:

```
http://powertweak.sourceforge.net/download.html
```

You will have to download the source files and compile them on TurboLinux, as no RPMs are currently available. However, they may be available in the future.



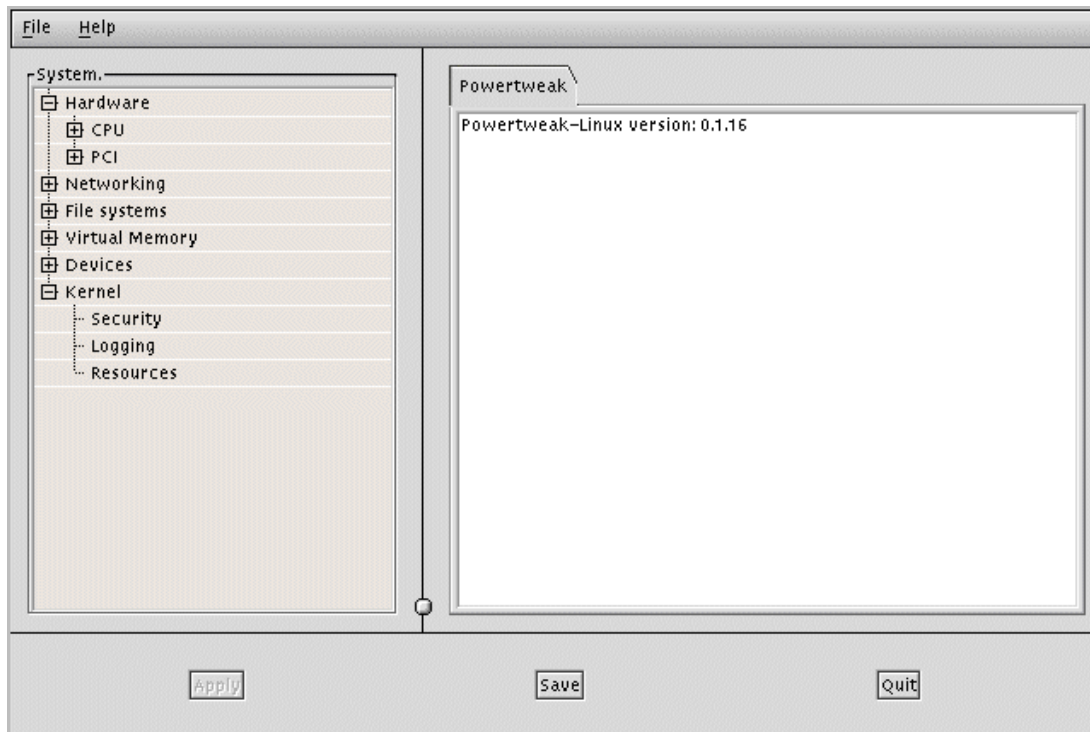


Figure 224. The main Powertweak window

As you can see in Figure 224, Powertweak allows tweaking of many system parameters. Some tabs will only display information about a device and will not allow you to tweak it. This is either because the setting for the device cannot be manipulated, or support has not been integrated with Powertweak yet.

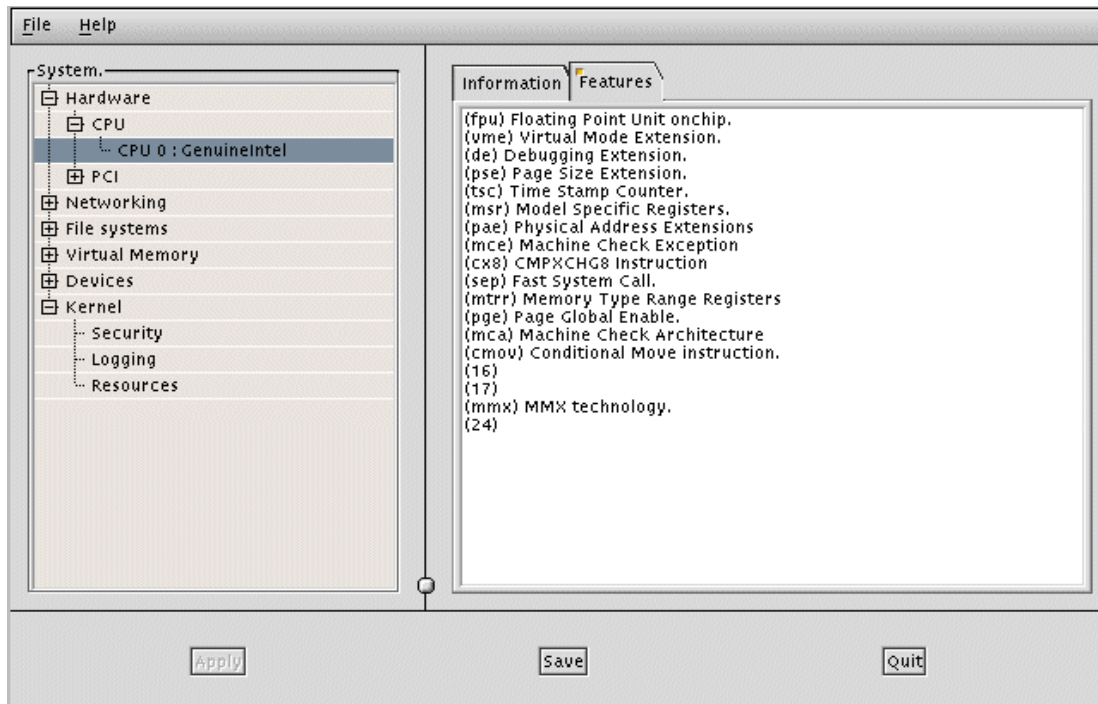


Figure 225. Powertweak: CPU information

The information about a device or a system resource is usually very detailed. This is ideal for troubleshooting the system, or reporting device IDs to vendors and so on.

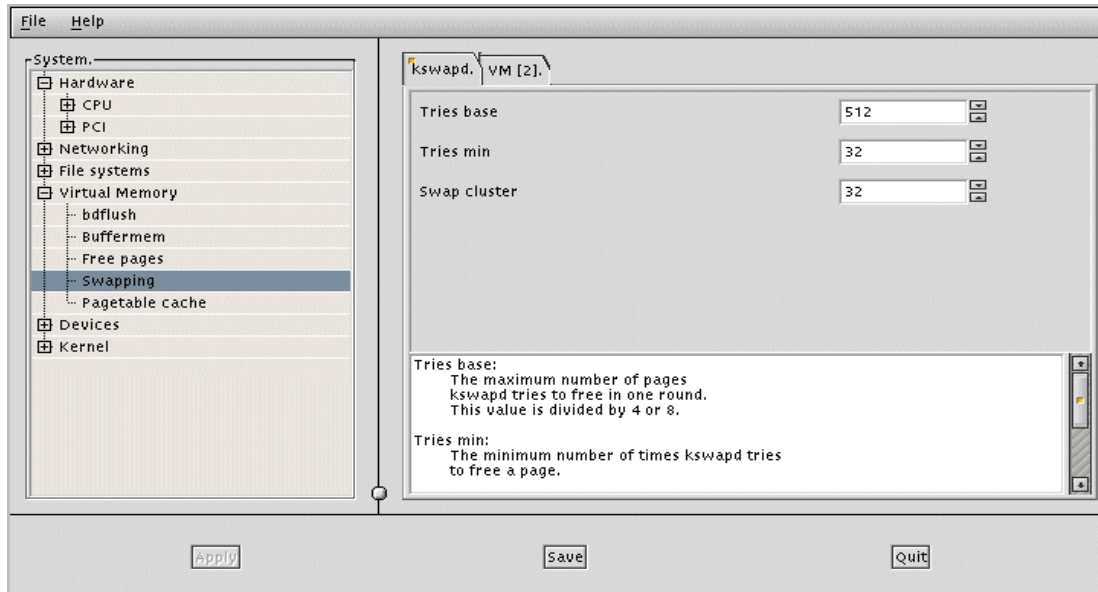


Figure 226. Powertweak swapping parameters

Details of what the settings do to is given for every dynamic resource that PowerTweak can manipulate. As with all tweaking it is a matter of trial and error. As such, do not try to manipulate the system in this way on a production/running system. It is advisable to test it out on a system of the same specification and then use the optimal settings as a basis for the production system.

### 14.1.2 Services

You should disable all unused services and daemons, especially network-related services. This has several advantages: fewer open services need fewer system resources (file descriptors, memory) and the system is less vulnerable to external attacks against known security holes. A good starting point is the `/etc/inetd.conf` file. Comment out all services you do not need, or disable `inetd` completely. This can also be done using `turboservice`.

The Linux `/proc` filesystem offers a lot of entry points for run-time optimization without recompiling the kernel. This directory does not physically exist on your hard drive; it is mapped as a virtual directory. Most of the files contained there are readable and contain various system information. Other files can be edited with a regular text editor to set a certain kernel parameter. See `/usr/src/linux/Documentation/sysctl/README` in the Linux kernel sources for

a detailed description of the tunable parameters (including file system, virtual memory, etc.).

There are some special TCP options that can be disabled in a local network with high signal quality and bandwidth, since they are mostly intended for lossy connections (see `/usr/src/linux/net/TUNABLE` in the Linux kernel sources for a detailed list):

To disable TCP timestamps, enter:

```
echo 0 > /proc/sys/net/ipv4/tcp_timestamps
```

To disable window scaling, enter:

```
echo 0 > /proc/sys/net/ipv4/tcp_window_scaling
```

To disable selective acknowledgments, enter:

```
echo 0 > /proc/sys/net/ipv4/tcp_sack
```

To tune the default and maximum window size (only if you know what you are doing), enter:

```
/proc/sys/net/core/rmem_default - default receive window  
/proc/sys/net/core/rmem_max     - maximum receive window  
/proc/sys/net/core/wmem_default - default send window  
/proc/sys/net/core/wmem_max     - maximum send window
```

The following Web sites offer a lot of additional helpful hints about tuning and performance issues on Linux:

```
http://tune.linux.com  
http://www.tunelinux.com
```

### 14.1.3 Kernel recompilation

Recompiling the kernel to include only the drivers and features needed by a machine can help to decrease the amount of memory used in the system.

Here are a few guidelines to follow when selecting the drivers and features to be used in a kernel:

- Drivers that are needed constantly by the server will compile directly into the kernel.
- Drivers that are needed by the system, but will not be in constant use, should be compiled as modules. This could be the case for an IDE CD-ROM drive.

- You are given the opportunity to set default values for a number of system resources, including timeouts. Set these to realistic values that can reduce times for device/resource access. Make sure you understand what you are changing with these values. You can usually look in the source code of the relevant driver to view comments about certain settings.
- Do not enable resources in the kernel that are not essential to the system. This includes framebuffer and sound support. These are nice things to have, but are not essential to the system that will be used as a server.
- If a new driver for a resource or device becomes available, use it in the kernel. You are usually given instructions on how to integrate these drivers into your system.
- With regards to the above comment, it is essential you upgrade to the correct ServeRAID driver when you upgrade the firmwares, not only for speed, but for the integrity of your system.

---

## 14.2 System monitoring and performance test tools

This section introduces a small collection of useful tools, among many available, to monitor your Linux system or to gather system information.

To get an overview about all running processes and the system load, run the command `top` in a terminal session.

```

10:45am up 2:54, 1 user, load average: 0.25, 0.27, 0.11
44 processes: 43 sleeping, 1 running, 0 zombie, 0 stopped
CPU states: 0.0% user, 1.1% system, 0.0% nice, 98.8% idle
Mem: 63108K av, 61524K used, 1584K free, 2276K shrd, 3348K buff
Swap: 124956K av, 5040K used, 119916K free 52196K cached

```

PID	USER	PRI	NI	SIZE	RSS	SHARE	STAT	LIB	%CPU	%MEM	TIME	COMMAND
1011	root	20	0	1056	1056	880	R	0	1.1	1.6	0:00	top
433	root	-7	-10	1848	876	740	S <	0	0.0	1.3	0:24	AgentMon
445	root	1	0	676	480	344	S	0	0.0	0.7	0:01	bash
233	root	0	0	136	96	72	S	0	0.0	0.1	0:00	cron
418	root	5	5	328	76	64	S N	0	0.0	0.1	0:00	AgentENL
204	root	0	0	128	64	52	S	0	0.0	0.1	0:00	syslogd
306	root	1	0	100	56	44	S	0	0.0	0.0	0:00	gpm
1	root	0	0	104	52	40	S	0	0.0	0.0	0:05	init
357	mysql	5	5	424	44	40	S N	0	0.0	0.0	0:00	mysqld
395	mysql	5	5	424	44	40	S N	0	0.0	0.0	0:00	mysqld
396	mysql	5	5	424	44	40	S N	0	0.0	0.0	0:00	mysqld
397	root	5	5	276	4	0	SWN	0	0.0	0.0	0:00	renld
2	root	0	0	0	0	0	SW	0	0.0	0.0	0:00	kflushd
3	root	1	0	0	0	0	SW	0	0.0	0.0	0:02	kupdate
4	root	0	0	0	0	0	SW	0	0.0	0.0	0:00	kpiod
5	root	0	0	0	0	0	SW	0	0.0	0.0	0:04	kswapd
6	root	-20	-20	0	0	0	SW<	0	0.0	0.0	0:00	mdrecoveryd
161	bin	0	0	68	0	0	SW	0	0.0	0.0	0:00	portmap

Figure 227. Example output of top

Top updates the process list in regular intervals. Press “?” to get an online help window about the available parameters. To change the refresh interval, press “s” and enter the desired number of seconds between each update. If you want to sort the processes by memory consumption, press “m”. To exit from top, press “q”. This will bring you back to the command line.

Similar to top, pstree displays a hierarchical structure of all currently running processes:



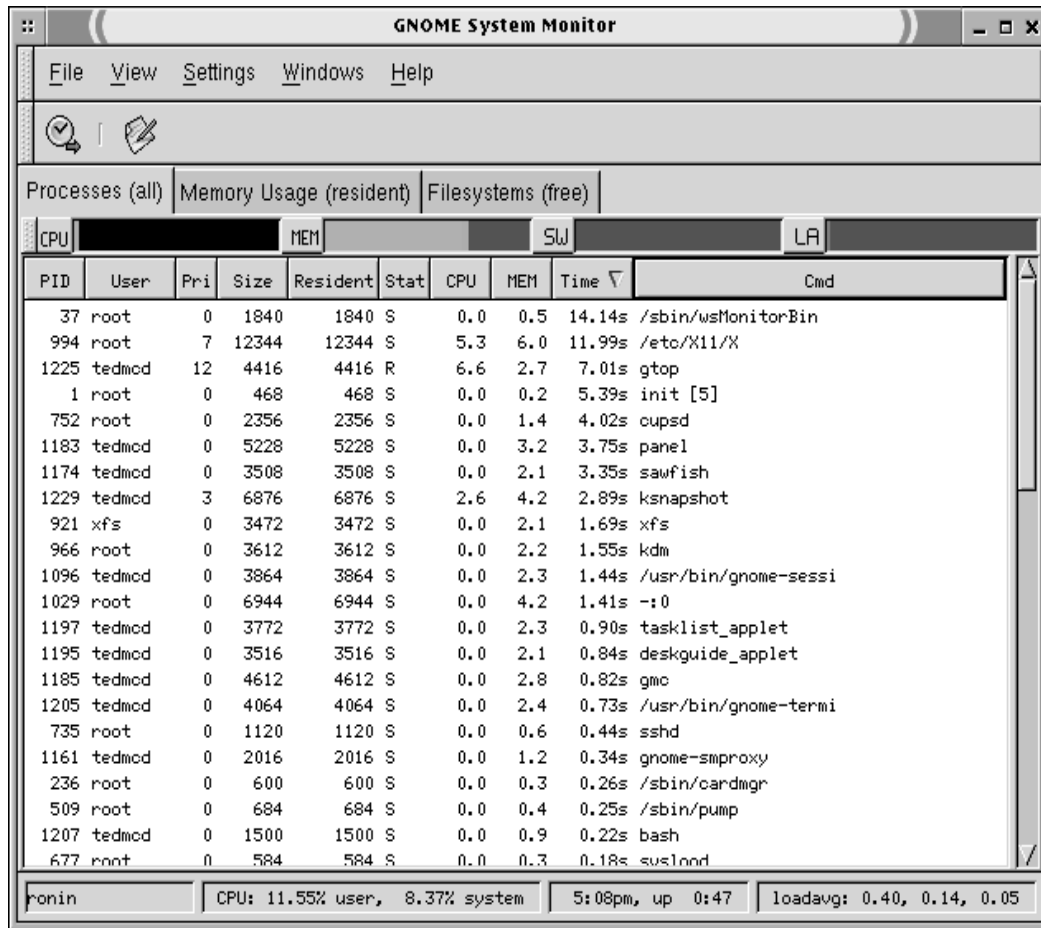


Figure 229. GNOME System Monitor window

Process (all) is the first tab displayed by gtop. All running processes are displayed, and the columns of information can be sorted in ascending or descending order. The information provided in the columns are:

- The process identifier, or PID, is the number assigned to the process by the kernel.
- The user who owns the process.
- The priority of the process.
- The size of the process in kilobytes.
- The amount of memory in resident memory (as opposed to swap).
- The Status.



- The percentage of CPU time being dedicated to this process.
- The percentage of memory being dedicated to this process.
- The amount of time the process has been running.
- The actual command being run is denoted by cmd.

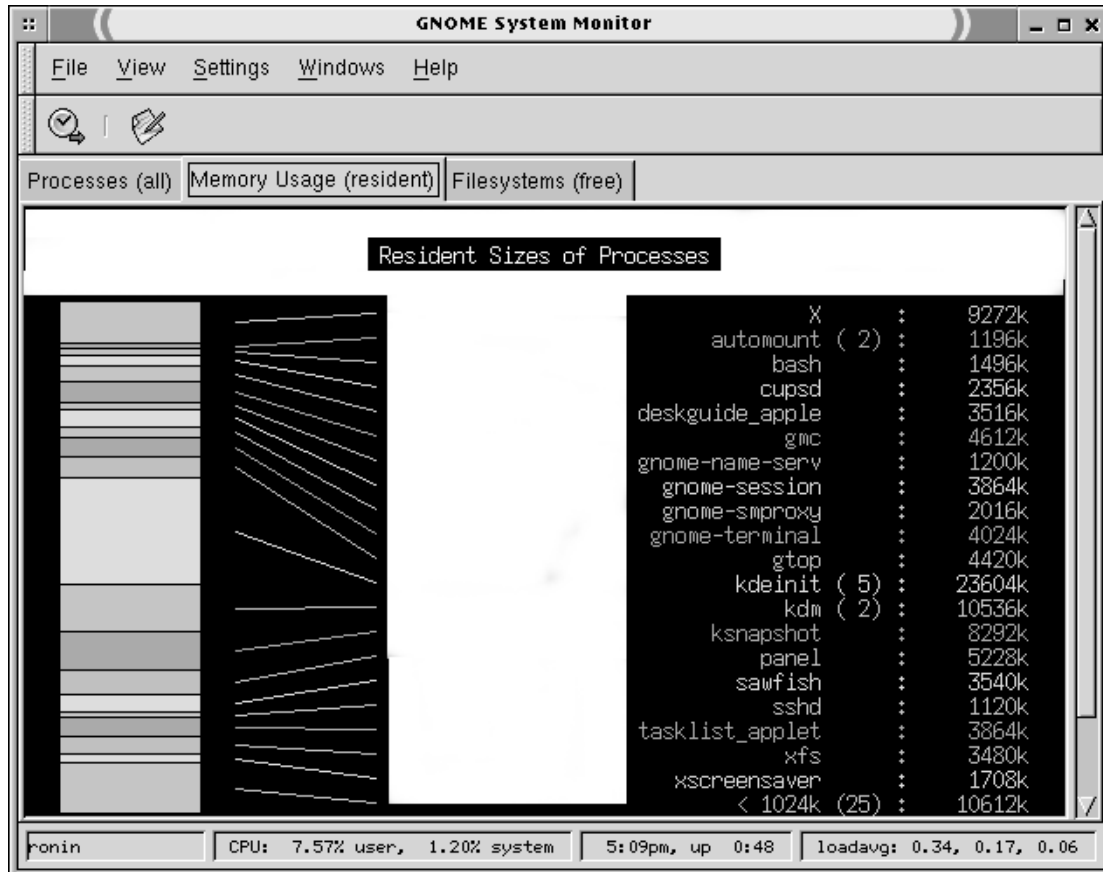


Figure 230. GNOME System Monitor window

The second tab in gtop, Memory Usage (resident), summarizes all the RAM being used by program. In this view, multiple instances of the same program show up as a single entry with their cumulative memory usage displaced. In the example above, there are two instances of automount and five instances of kdeinit running.

The `/proc/cpuinfo` file contains information about your CPU (that is, vendor, MHz, flags such as mmx). For example:

```

turbo:~ # cat /proc/cpuinfo
processor       : 0
vendor_id     : GenuineIntel
cpu family    : 6
model         : 5
model name    : Pentium II (Deschutes)
stepping      : 2
cpu MHz       : 513.953346
cache size   : 512 KB
fdiv_bug     : no
hlt_bug      : no
sep_bug      : no
f00f_bug     : no
coma_bug     : no
fpu          : yes
fpu_exception : yes
cpuid level   : 2
wp           : yes
flags        : fpu vme de pse tsc msr pae mce cx8 sep mtrr pge mca cmov pat p
se36 mmx osfxsr
bogomips     : 313.75

```

Figure 231. `cat /proc/cpuinfo` display

The `/proc/interrupts` file lists all interrupts used by Linux. Note that this shows interrupts only from devices that have been detected by the kernel! If a device can not be detected because of a resource conflict, you have to resolve this conflict manually (for example, by changing the BIOS setup). For example:

```

turbo:~ # cat /proc/interrupts
CPU0
0: 548029 XT-PIC timer
1: 557 XT-PIC keyboard
2: 0 XT-PIC cascade
8: 2 XT-PIC rtc
9: 371 XT-PIC PCnet/PCI II 79C970A
12: 68 XT-PIC PS/2 Mouse
13: 0 XT-PIC fpu
14: 198235 XT-PIC ide0
15: 3 XT-PIC ide1
NMI: 0

```

Figure 232. `cat /proc/interrupts` display

The `/proc/ioports` file contains all allocated device I/O ports. The same note as for interrupts applies here. Only devices that are actually detected by the kernel are listed here. For example:

```

turbo:~ # cat /proc/ioports
0000-001f : dma1
0020-003f : pic1
0040-005f : timer
0060-006f : keyboard
0070-007f : rtc
0080-008f : dma page reg
00a0-00bf : pic2
00c0-00df : dma2
00f0-00ff : fpu
0170-0177 : ide1
01f0-01f7 : ide0
02e8-02ef : serial(auto)
02f8-02ff : serial(auto)
0376-0376 : ide1
03c0-03df : vga+
03e8-03ef : serial(auto)
03f6-03f6 : ide0
03f8-03ff : serial(auto)
1000-101f : PCnet/PCI II 79C970A
1020-1027 : ide0
1028-102f : ide1

```

Figure 233. `cat /proc/ioports` display

The `/proc/meminfo` file displays information about memory (for example, memory used, free, swap size). You can also use the `free` command to display this information. For example:

```

turbo:~ # cat /proc/meminfo
          total:        used:        free:    shared: buffers:   cached:
Mem:  64569344 62578688 1990656 54308864 18792448 27807744
Swap: 129019904  102400 128917504
MemTotal:        63056 kB
MemFree:         1944 kB
MemShared:       53036 kB
Buffers:         18352 kB
Cached:          27156 kB
SwapTotal:      125996 kB
SwapFree:       125896 kB
turbo:~ # free
          total          used          free   shared    buffers     cached
Mem:      63056         61124         1932     53068     18352     27164
-/+ buffers/cache:    15608         47448
Swap:    125996           100       125896

```

Figure 234. `cat /proc/meminfo` display

The `/proc/mounts` file shows all currently mounted partitions. The `mount` command without parameters will display similar information. For example:

```

turbo:~ # cat /proc/mounts
/dev/root / ext2 rw 0 0
proc /proc proc rw 0 0
/dev/hda1 /boot ext2 rw 0 0
devpts /dev/pts devpts rw 0 0
turbo:~ # mount
/dev/hda3 on / type ext2 (rw)
proc on /proc type proc (rw)
/dev/hda1 on /boot type ext2 (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=0620)

```

Figure 235. `cat /proc/mounts` display

The `/proc/partitions` file displays all existing partitions on all devices. You can also use `fdisk -l` to display this information. For example:

```

turbo:~ # cat /proc/partitions
major minor #blocks name

 3      0   1023907 hda
 3      1     6016 hda1
 3      2   126000 hda2
 3      3   891072 hda3
 3     64   1023907 hdb
 3     65   1023088 hdb1
22      0 1073741823 hdc

turbo:~ # fdisk -l

Disk /dev/hda: 32 heads, 63 sectors, 1015 cylinders
Units = cylinders of 2016 * 512 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/hda1  *            1           6         6016+    83  Linux
/dev/hda2                7          131       126000    82  Linux swap
/dev/hda3           132          1015       891072    83  Linux

Disk /dev/hdb: 32 heads, 63 sectors, 1015 cylinders
Units = cylinders of 2016 * 512 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/hdb1                1          1015     1023088+    83  Linux

```

Figure 236. `cat /proc/partitions` display

The `/proc/pci` file gives information about all your PCI devices. You can also use the `lspci` command to provide output that is easier to read. Please note that `/proc/pci` is obsolete and will be replaced by `/proc/bus/pci/*` in the future. For example:

```

bash-2.04# cat /proc/pci
PCI devices found:
  Bus 0, device 0, function 0:
    Host bridge: Intel 82439TX (rev 1).
    Medium devsel. Master Capable. No bursts.
  Bus 0, device 7, function 0:
    ISA bridge: Intel 82371AB PIIX4 ISA (rev 8).
    Medium devsel. Master Capable. No bursts.
  Bus 0, device 7, function 1:
    IDE interface: Intel 82371AB PIIX4 IDE (rev 1).
    Medium devsel. Fast back-to-back capable. Master Capable. Latency=64.
    I/O at 0x1020 [0x1021].
  Bus 0, device 15, function 0:
    Display controller: Unknown vendor Unknown device (rev 0).
    Vendor id=15ad. Device id=710.
    Medium devsel. Fast back-to-back capable. Master Capable. Latency=64.
    I/O at 0x1030 [0x1031].
    Non-prefetchable 32 bit memory at 0xfc000000 [0xfc000000].
    Non-prefetchable 32 bit memory at 0xfb000000 [0xfb000000].
  Bus 0, device 16, function 0:
    Ethernet controller: AMD 79C970 (rev 16).
    Medium devsel. Fast back-to-back capable. IRQ 9. Master Capable. Latency=64.
    Min Gnt=6.Max Lat=255.
    I/O at 0x1000 [0x1001].
    Non-prefetchable 32 bit memory at 0xfd000000 [0xfd000000].
bash-2.04# lspci
00:00.0 Host bridge: Intel Corporation 430TX - 82439TX MTRX (rev 01)
00:07.0 ISA bridge: Intel Corporation 82371AB PIIX4 ISA (rev 08)
00:07.1 IDE interface: Intel Corporation 82371AB PIIX4 IDE (rev 01)
00:0f.0 Display controller: VMWare Inc Virtual SVGA
00:10.0 Ethernet controller: Advanced Micro Devices [AMD] 79c970 [PCnet LANCE] (rev 10)

```

Figure 237. `cat /proc/pci` display

The `/proc/swaps` file displays information about all active swap partitions. For example:

```

bash-2.04# cat /proc/swaps
Filename                                Type              Size    Used    Priority
/dev/hda2                               partition         125996  56     -1

```

Figure 238. `cat /proc/swaps` display

The `/proc/version` file displays some version information about the Linux kernel. The command `uname -a` will display similar information. For example:

```
bash-2.04# cat /proc/version
Linux version 2.2.14 (support@kernel.turbolinux.com) (gcc version 2.95.2 19991024 (release)
executing gcc version 2.7.2.3) #1
Fri May 26 19:07 PDT 2000
```

Figure 239. `cat /proc/versions` display

If you want to obtain some more information about your SCSI devices, have a look at the files below `/proc/scsi`.

A tool that is also gathering system information from the `/proc` filesystem is `vmstat`. It reports information about processes, memory, paging, block IO, traps, and CPU activity. The first report produced gives averages since the last reboot. Additional reports give information on a sampling period of length delay. The process and memory reports are instantaneous in either case. `vmstat` is very helpful for logging CPU and memory usage over a longer period of time.

Apart from configuring numerous parameters of your hard drive, the command `hdparm` can also be used to perform hard disk performance tests with the command `hdparm -tT <device>`. For example:

```
turbo:~ # hdparm -tT /dev/hda

/dev/hda:
Timing buffer-cache reads: 64 MB in 0.68 seconds =94.12 MB/sec
Timing buffered disk reads: 32 MB in 29.51 seconds = 1.08 MB/
turbo:~ # hdparm -c1 /dev/hda

/dev/hda:
setting 32-bit I/O support flag to 1
I/O support = 1 (32-bit)
turbo:~ # hdparm -tT /dev/hda

/dev/hda:
Timing buffer-cache reads: 64 MB in 0.67 seconds =95.52 MB/sec
Timing buffered disk reads: 32 MB in 12.92 seconds = 2.48 MB/sec
```

Figure 240. `hdparm -tT /dev/hda` display

Another popular hard disk performance test is `bonnie`, found at <http://www.textuality.com/bonnie/>. Note, however, that these tests are mostly useful for testing different parameter settings on one machine as a relative measure, not as a comparison between different systems.

To test the throughput of your network, you can either use `netperf`, found at <http://www.netperf.org/netperf/NetperfPage.html> or `bing`.

---

## Chapter 15. Backup and recovery

It may seem obvious that backing up and restoring data quickly is critical, but many administrators leave this task at the end of the “to do” list until it is too late. With the ease of use of the commercially available packages BRU (Enhanced Software Technologies), BackupEDGE/RecoverEDGE (MicroLite) or Arkeia (Knox Software), there is no need to wait.

### Note

We recommend that you do not connect tape devices to the IBM ServeRAID adapter. Use a separate SCSI controller for the tape devices.

---

### 15.1 BRU

BRU is a backup and restore utility with significant enhancements over other common utilities such as tar, cpio, volcopy and dump. BRU is designed to work with most backup devices, including cartridge, 4mm DAT, 8mm (Exabyte) and 9-track tape drives.

BRU includes incremental backups, full backups, multivolume archives, distribution and updates, error detection and recovery, random access capabilities, file comparisons, file overwrite protection, and increased speed over previous versions.

#### 15.1.1 Installing BRU

Before you begin, you need to know the following:

1. The device name of your tape drive. Typically under TurboLinux this will be `/dev/st0` for the rewinding and `/dev/nst0` for the non-rewinding drive.
2. The size of your backup medium in megabytes.

To install BRU from the floppy drive with the `tar` command, type:

```
cd /tmp
tar xvf /dev/fd0
./install
```

Follow the prompts regarding readme files and licenses, enter your *license data* and your *BRU serial number* when asked to do so until you come to the following window:

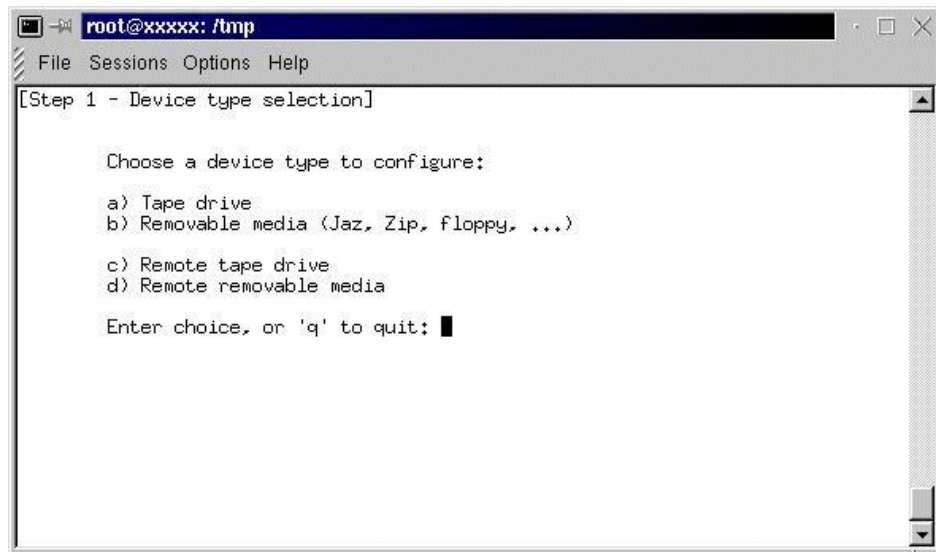


Figure 241. Selecting your backup devices

Enter the letter for your backup device and answer the following questions appropriate for your device.

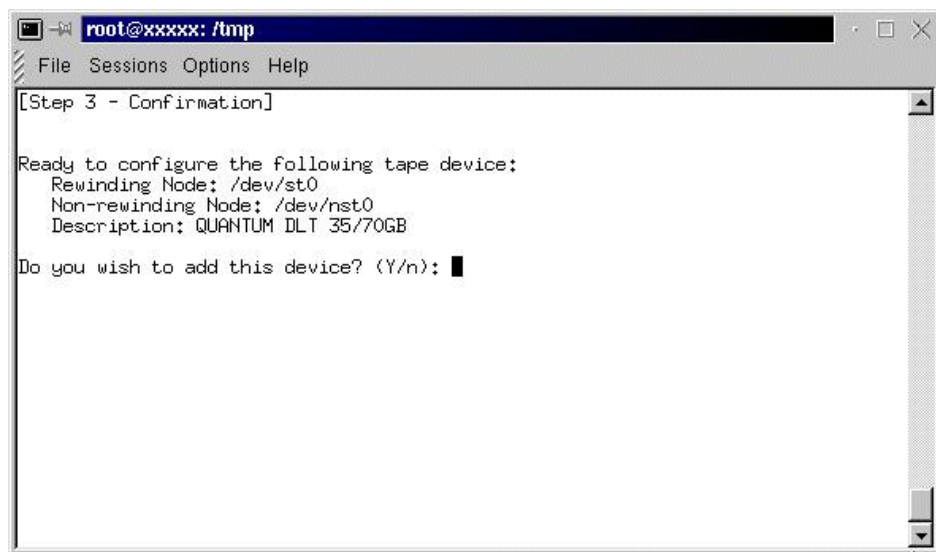


Figure 242. You have entered your backup devices



If you have entered the information for all your backup devices, you will be asked if you would like to install the X11 interface. Select **Y**.

The installation program needs to create an xbru directory. You can select a path or accept the default `/usr/local/`.

The installation program will install executables in a user-specified directory. The default is `/usr/local/bin`.

**Note**

The key configuration file is `/etc/brutab`. Consult the *BRU User's Guide* for advanced information. Do not edit unless you know what you are doing.

BRU is now installed.

### 15.1.2 Basic commands

The basic command structure for BRU is:

```
# bru modes [control options] [selection options] [files]
```

Where `bru` is the command or program followed by the mode specifying backup, restore, or various queries. `Control options` specify devices and buffer size. `Selection options` control which files or directories to work with. `Files` is the specified target of the `bru` command.

### 15.1.3 Basic backup

To back up a single file `/home/ayne/.profile`:

```
# bru -c -vvvv -G /home/ayne/.profile
```

To back up the complete directory `/home/ayne`:

```
# bru -c -vvvv -G /home/ayne
```

To back up the entire system:

```
# bru -c -vvvv -G /
```

### 15.1.4 Basic restore

To restore a single file `/home/ayne/.profile`:

```
# bru -x -vvvv -ua -w /home/ayne/.profile
```

To restore the complete directory `/home/ayne`:

```
# bru -x -vvvv -ua -w /home/ayne
```

To restore the entire system:

```
# bru -x -vvvv -ua -w /
```

### 15.1.5 Basic verification and listing commands

The `-i` mode can be used in conjunction with a backup command or by itself. The `-i` mode reads each block of data and verifies the checksum of the block. If used with the verbosity options (`-vvvv`), BRU will give a complete listing of the contents of an archive.

The `-G` mode displays the archive header block, which contains detailed information on the archive including the command used to create the archive. See the *BRU User's Guide* for more information.

The `-gg` mode displays the contents of the on-tape directory. This mode can only be used if the archive was created with the `-G` option.

### 15.1.6 X Interface

To use BRU's X interface, you will need to be in an X-Windows environment.  
Type:

```
xbrun
```



Figure 243. XBRU window

You will see a window similar to Figure 243.

From this interface you can:

- Create and restore backups.
- Create save, and load backup definitions.
- Schedule backups.
- List and verify the contents of archives.
- View the BRU log.

### 15.1.7 The big buttons in BRU

The three main buttons (Full, Level 1, and Level 2) are shortcuts to various levels of backing up your system, directories, or individual files.

- Select **Full** to back up all the files in the user's home directory, or, if the user is root, the entire system.

- Select **Level 1** to execute a backup for the same files as listed above, on the condition that files have been modified since the previous full backup. If no previous full backup has been done, this will be considered a full backup.
- Select **Level 2** to execute a backup for the same files as listed above, on the condition that files have been modified since the previous level 1 backup. If no previous level 1 backup has been done, this will be considered a level 1 backup.

### 15.1.8 Creating archives

Creating archives with BRU's X interface is simple. Click the **Backup** button to bring up the Backup File Selection interface (Figure 244).

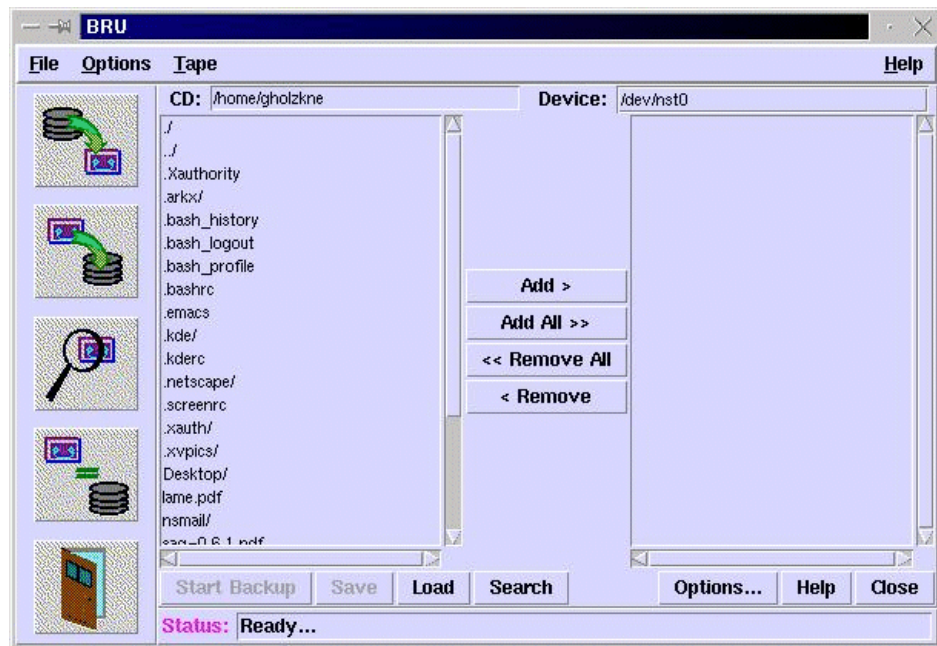


Figure 244. Creating an archive

The box on the left displays the contents of the current directory (CD:). You can change the current directory by editing the CD entry. Then press Enter.

You can add or remove files and directories from the backup list by selecting them and clicking the appropriate button.

BRU also provides a search function. Click the **Search** button to bring up a dialog box prompting you for a search string. This string can contain typical wildcards.

Backup Definitions are a way to define a set of commonly used backup options or preferences for use at a future time. You can create definitions for use with the backup scheduler or simply use the default selections.

After you have selected the files and directories that you wish to back up, you can click the **Options** button. In this dialog (Figure 245) you can set your preferences regarding different options. After you have made your decisions, click the **Close** button to return to the previous dialog. To start the backup click the **Start Backup** button.



Figure 245. Dialog for backup options

Enter in the next dialog, click **Enter Archive Label** and enter text to identify your new archive. Click **Create Backup** to proceed.

The backup will inform you of how many directories/files and which amount of data will be backed up. During backup, you see a window, informing you about the progress and the actual action. When the backup process has finished, click **Done** to return to XBRU's main dialog.

### 15.1.9 Scheduling

To access the scheduling feature, go to **File>Scheduler** on the menu.



Figure 246. Scheduler

BRU provides a scheduling utility to automate the backup process for the busy administrator. There are three predefined definitions: Full, Level 1, and Level 2. These are the same definitions used in 15.1.7, “The big buttons in BRU” on page 261. You can create your own definitions in the Creating Archives interface.

From the BRU for X11 Scheduler interface, you can set scheduled backups based on weekly, monthly, or single dates. The scheduler is very flexible. In order to take advantage of the scheduling options, you must save your desired schedule configuration and verify that the scheduler is being run from cron. To verify or add the cron entry, log in as root and type:

```
crontab -e
```

Insert the following line:

```
0/5 * * * * /usr/local/bin/bruschedule
```

If you chose a different path for the binaries during installation, change the entry accordingly.

Save the crontab entry. You can now schedule backups.

### 15.1.10 Restoring files

Restoring files with BRU's X interface is simple. BRU will retrieve the contents of the archive when you click the **Restore** button. After scanning the archive, the Restore File Selection interface (similar to Figure 244) will appear.

#### Note

If the on-tape directory is not in the archive, then BRU must scan the entire archive to get a listing. This can be very time consuming. When creating an archive, use the -G option to create the on-tape directory or chose **Create On-Tape Directory** in XBRU's **Options** dialog from the backup dialog.

The box on the left displays the contents of the current directory that is stored on the tape. You can change the current directory by editing the **CD:** entry and pressing Enter.

You can add or remove files and directories from the backup list by selecting them and selecting the appropriate button.

When you have selected all of the files and directories that you wish to restore, click the **Restore** button. A progress window will show each file as it is restored.

### 15.1.11 Listing and verifying archives

For listing the contents of an archive, BRU gives you three options:

1. Header - This option shows the archive header record, which lists the label, creation date, version, and serial number. For more information on the header, consult the *BRU User's Guide*.
2. Filenames only - This option displays the on-tape directory. If the archive was created without using the -G option, BRU will scan the entire archive to create a list of files. You will be prompted before this occurs, as this can be a lengthy process.
3. Full details - This option scans the entire archive for details such as file names, permissions, owners, size, modification times, etc. This process can be time consuming.

For verifying archives, BRU gives you two options:

1. Checksum Verification - When archives are written, a checksum is calculated for each block of data. The checksum is stored in the header of each block. Checksum verification will read each block, recalculate the checksum, and compare the checksum to the value in the header. Each file will be listed as it is verified, along with any errors found. If no errors are found, you know you have an accurate backup.
2. Compare Verification - BRU compares the files in the archive to the files on the hard drive. Any differences, such as modification times, size, or files in the archive that are nonexistent on the hard drive are noted. An *end of differences* notice will be posted when the verification is complete.

### 15.1.12 Summary

For information on advanced features consult your *BRU User's Guide* or the BRU Web site at:

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

---

## 15.2 Microlite BackupEDGE

BackupEDGE is a complete backup solution for the Linux platform. It is easy to use and still very robust. With BackupEDGE you can safely archive every file, directory, device node and special file on your file systems. Unlike the standard UNIX tar command, which ignores many important files, BackupEDGE also verifies every byte of data written to the tape to ensure the tape is an accurate reflection of your data. Below are the features provided by BackupEDGE backup software:

- Data Compression - automatic data compression is supported.
- Menu Interface - almost all functions can be accessed through an intuitive menu system.
- Remote Tape Drive Support - you can back up computers across the network.
- High Performance - advanced double buffering and variable block factors.
- Virtual File Support - you can back up virtual (sparse) files.
- Multi-Volume / Multi-Device Archives - automatic spanning across multiple volumes or devices.
- Wildcard Support - when selecting files you can use a wildcard.
- Raw Device Backups - you can archive an entire raw device/partition to tape.
- Master / Incremental Backups



- Unattended Operation - you can perform a master backup or back up only the changed files.

BackupEDGE is designed to operate on Linux kernels 2.x and there are available versions for several types of libraries.

In the following sections we describe how to install, configure and use the Microlite BackupEDGE backup software.

**Note**

We recommend that you do not connect tape devices to the IBM ServeRAID adapter. Use a separate SCSI controller for the tape devices.

### 15.2.1 Installing Microlite BackupEDGE

Before you install BackupEDGE you must identify the device entry for your backup device. Usually tape devices under Linux are assigned in device nodes `/dev/st0`, `/dev/st1`... A no-rewind device is created for each tape device, which is `/dev/nst0`, `/dev/nst1`... In our example, we used `/dev/st0` as tape device and `/dev/nst1` as the no-rewind device.

In our example, we used diskette as the installation medium. To install the product, follow these steps:

1. Log in as root.
2. Change the directory to root `/`.
3. Insert the diskette with the product in the floppy drive and execute the command:

```
tar xvf /dev/fd0
```

Where `/dev/fd0` is your floppy device.

4. Execute the following command to finish the installation:

```
/tmp/init.edge
```

You will see a window similar to Figure 247.

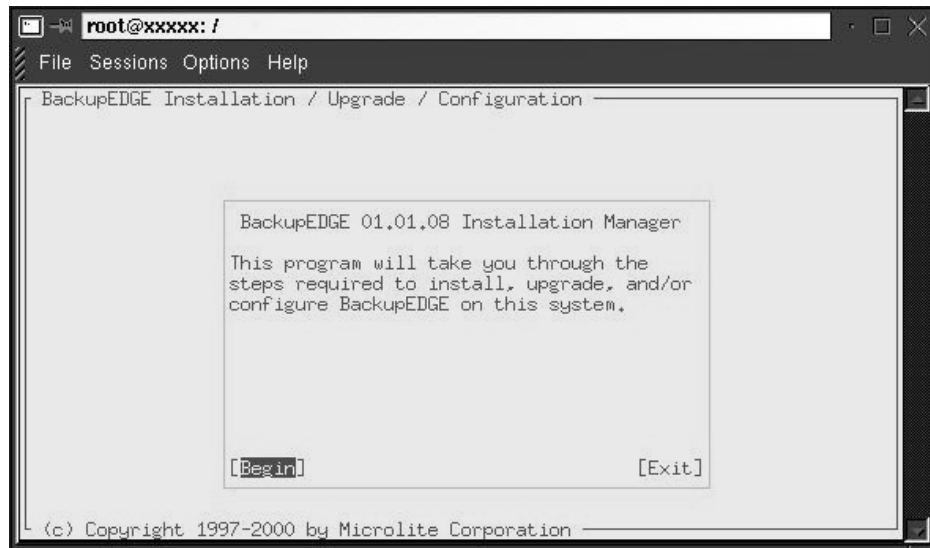


Figure 247. Start of installation dialog

The installation program guides you through the installation process. The windows are intuitive. During the installation process, you can also configure your backup device(s) and your scheduling schema for unattended operation. If information is needed during this process, you are asked to enter the appropriate data.

Now you are ready to use the product.

The actions *Resource Manager* and *Defining Devices* can be started by entering on the command line:

```
/usr/lib/edge/bin/edge.resmgr (Resource Manager) or  
/usr/bin/edge.config (Defining Devices)
```

You can also perform these actions, if you click **Admin** on BackupEDGE's main window.

### 15.2.2 Initializing the tape

Before you start making backups you should initialize the tape. To do this, you follow these steps:

1. Start the edgemenue program by executing command:

```
edgemenue
```

You will see a window similar to Figure 248.

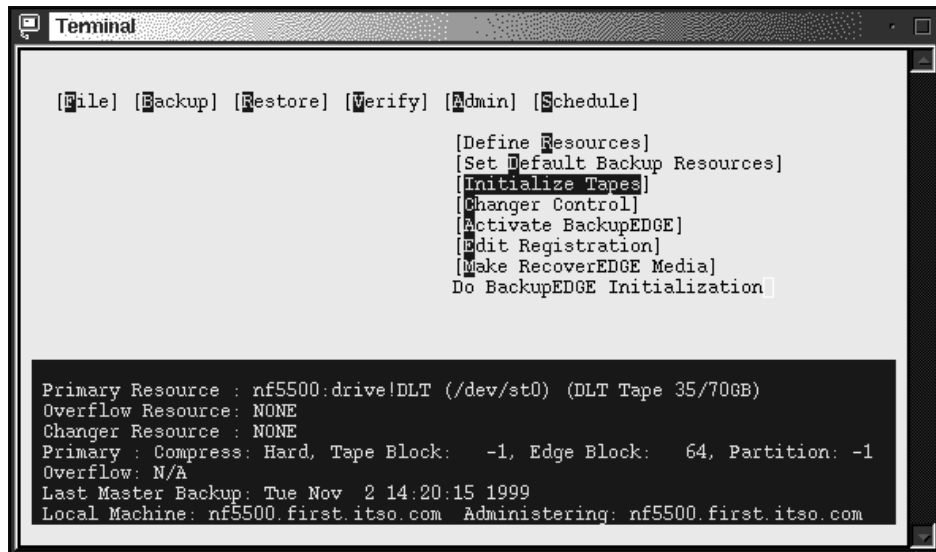


Figure 248. BackupEdge main menu

2. In the Admin menu select **Initialize Tapes**. You will see a window similar to Figure 249.

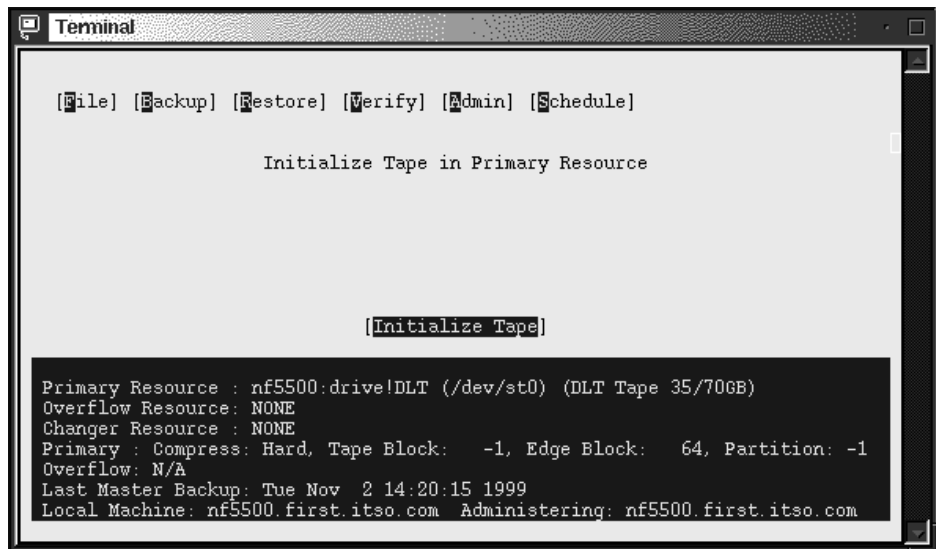


Figure 249. Initializing the tape

3. Select **Initialize Tape** and press Enter. The tape will be initialized. You will get a message that the tape is successfully initialized. Press Enter to continue.

You can check the tape properties by selecting **Show Tape Label** in the Verify menu. You will see a window similar to Figure 250.



Figure 250. Tape information

### 15.2.3 Your first backup

In this section we will show how to make backups of desired files or directories. You can perform backups in the edgemenu utility. Follow these steps to make a sample backup:

1. Start the edgemenu program by executing the following command:

```
/usr/bin/edgemenu
```

You will see a window similar to Figure 251.

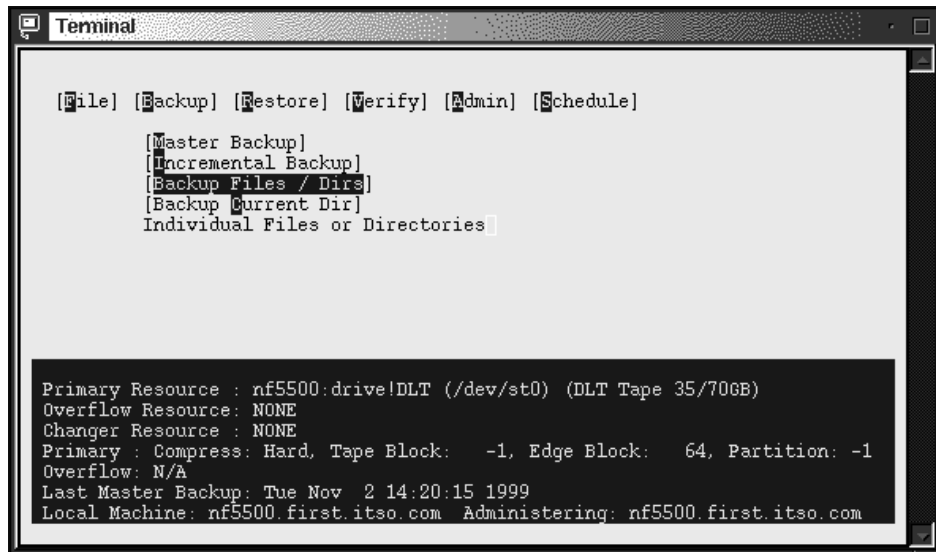


Figure 251. Starting the backup

2. In the Backup menu select **Backup Files / Dirs**, and you will see a window similar to Figure 252.

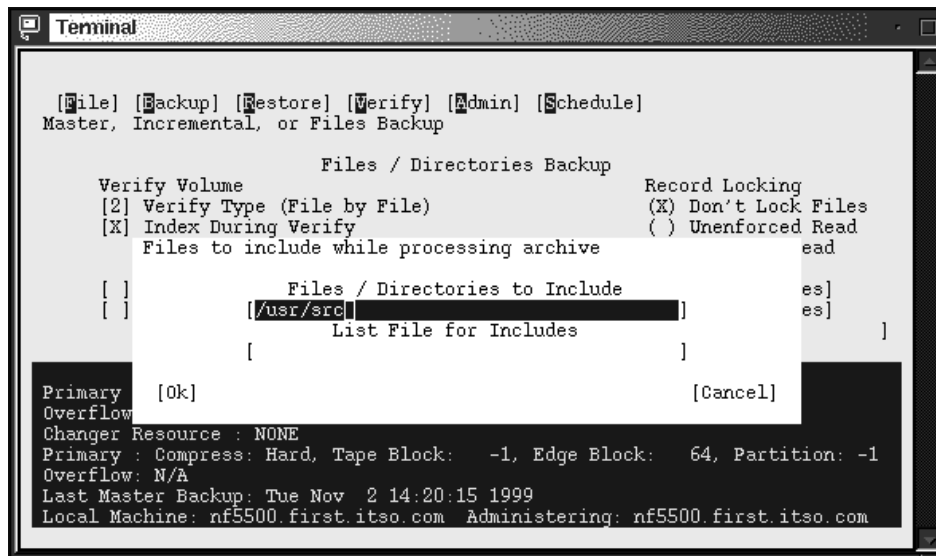


Figure 252. Selecting source for backup

3. In the Files / Directories to Include field, type in the files or directories you want to back up. In our example we want to make backups of the directory /usr/src. Select **OK** to continue. You will see a window similar to Figure 253.

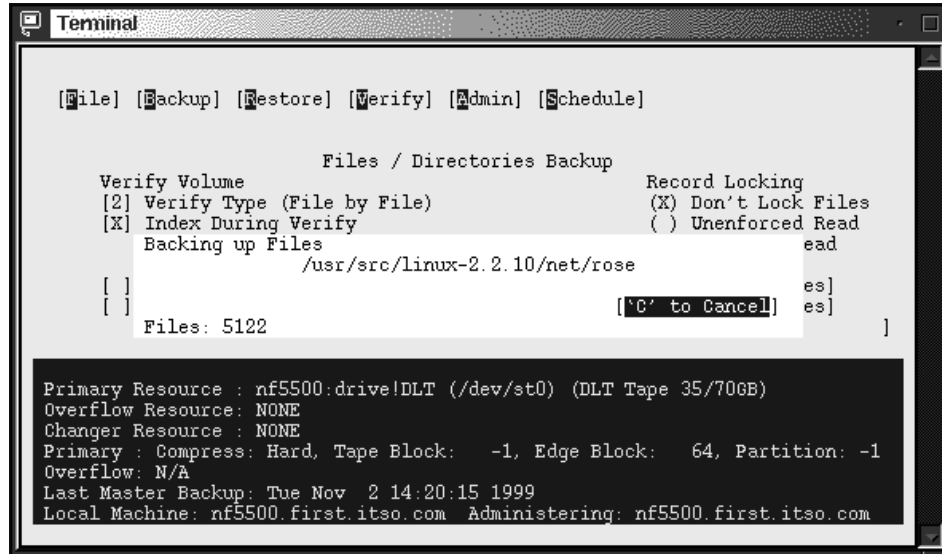


Figure 253. Backup in progress

After the backup is finished you will see a window similar to Figure 254.

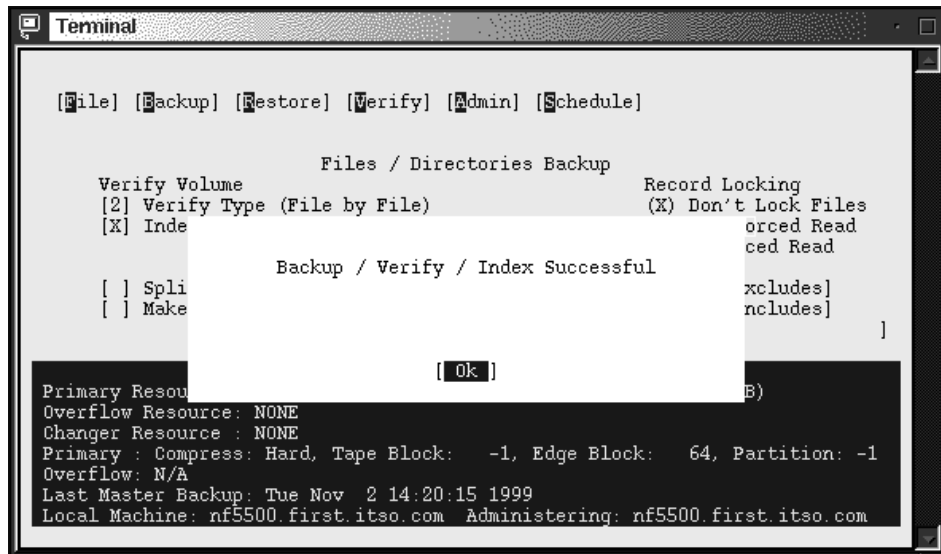


Figure 254. Backup completed

You will also see the backup report similar to Figure 255.

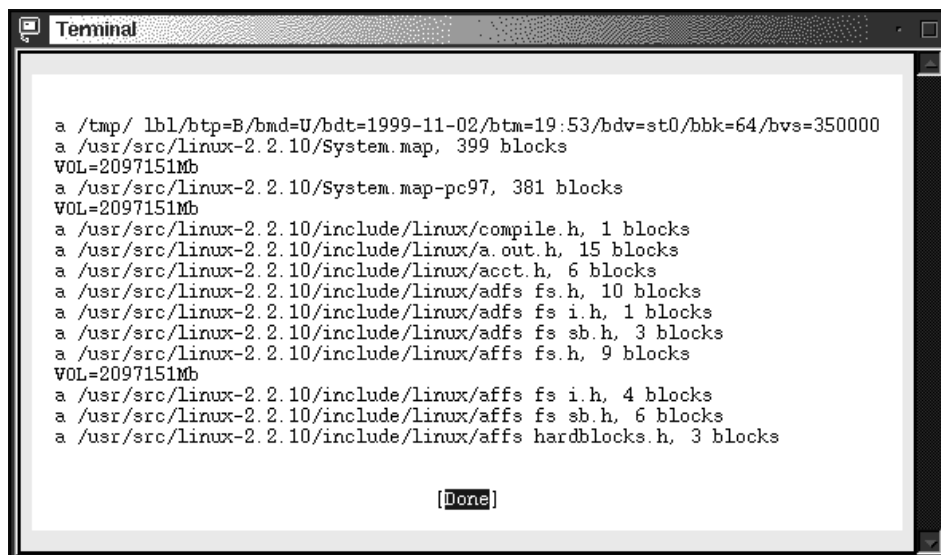


Figure 255. Backup report

You have just made your first backup and your files are safe now!

## 15.2.4 Restoring single files or directories

In this section we will show how to recover files from the backup. We are assuming that you are recovering files on the same server you made backups with the same user ID. You can perform recovery from the same utility as backups. Follow these steps to recover files:

1. Start the edgemenue program by executing the following command:

```
edgemenue
```

You will see a window similar to Figure 251. Select **Restore** and a window similar to Figure 256.

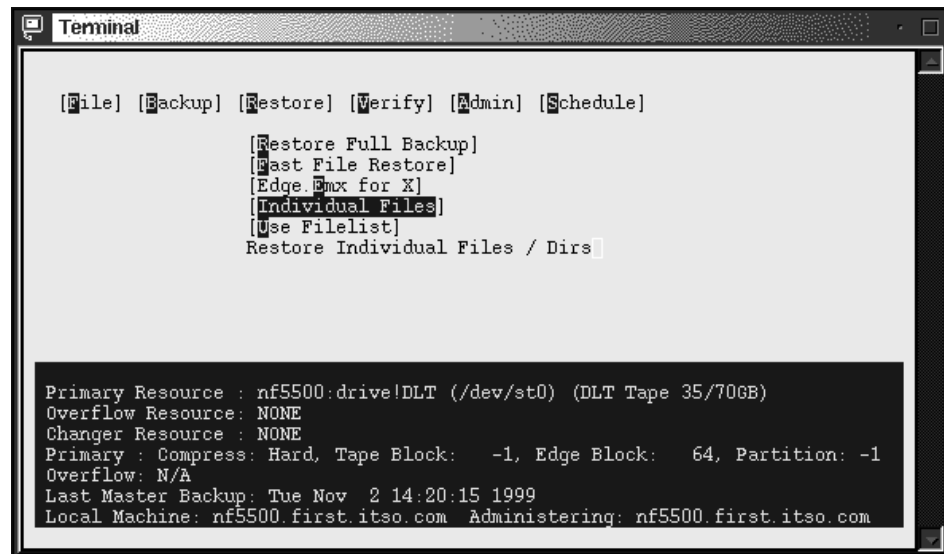


Figure 256. Starting the recovery

2. Select **Restore > Individual Files**, and you will see a window similar to Figure 252 on page 271.
3. Select the files or directories to restore. Select **OK** to continue, and you will see a window similar to Figure 257.



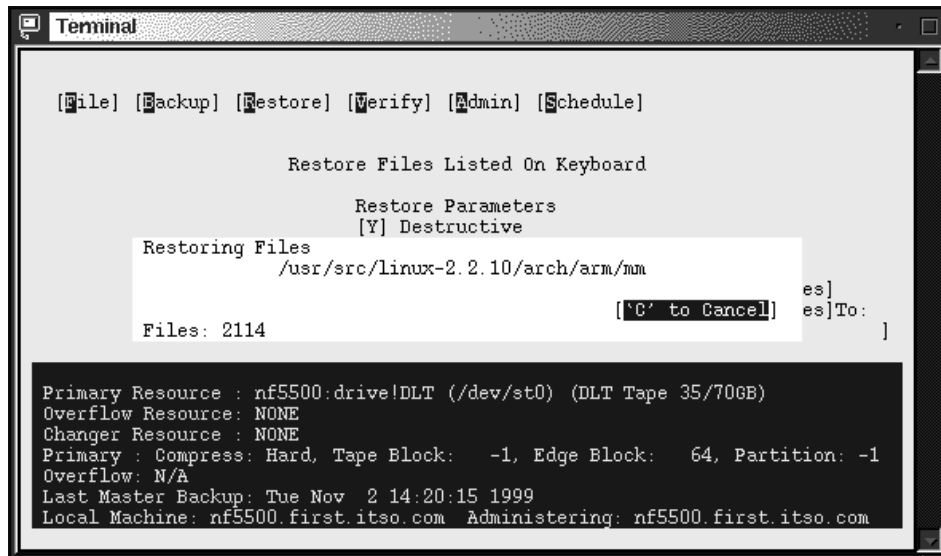


Figure 257. Recovery in progress

When the recovery is completed you will see a window similar to Figure 258.

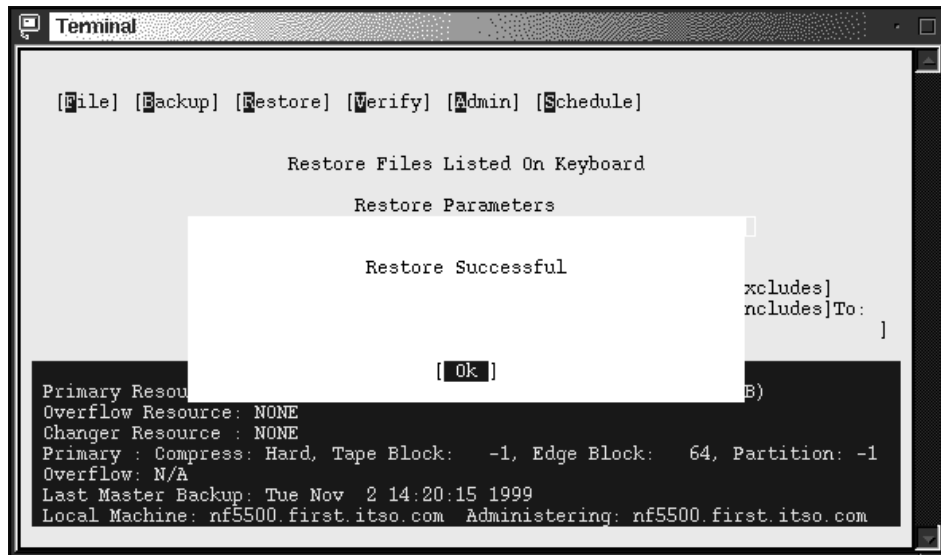
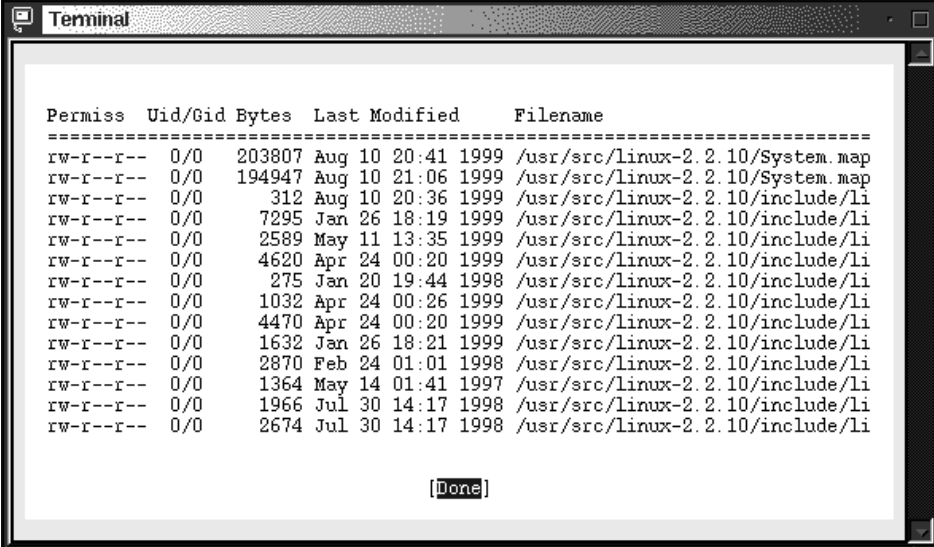


Figure 258. Recovery completed

Select **OK** to continue and you will see a recovery report similar to Figure 259.



```
Terminal
-----
Permiss Uid/Gid Bytes  Last Modified  Filename
-----
rw-r--r-- 0/0   203807 Aug 10 20:41 1999 /usr/src/linux-2.2.10/System.map
rw-r--r-- 0/0  194947 Aug 10 21:06 1999 /usr/src/linux-2.2.10/System.map
rw-r--r-- 0/0    312 Aug 10 20:36 1999 /usr/src/linux-2.2.10/include/li
rw-r--r-- 0/0   7295 Jan 26 18:19 1999 /usr/src/linux-2.2.10/include/li
rw-r--r-- 0/0   2589 May 11 13:35 1999 /usr/src/linux-2.2.10/include/li
rw-r--r-- 0/0   4620 Apr 24 00:20 1999 /usr/src/linux-2.2.10/include/li
rw-r--r-- 0/0    275 Jan 20 19:44 1998 /usr/src/linux-2.2.10/include/li
rw-r--r-- 0/0   1032 Apr 24 00:26 1999 /usr/src/linux-2.2.10/include/li
rw-r--r-- 0/0   4470 Apr 24 00:20 1999 /usr/src/linux-2.2.10/include/li
rw-r--r-- 0/0   1632 Jan 26 18:21 1999 /usr/src/linux-2.2.10/include/li
rw-r--r-- 0/0   2870 Feb 24 01:01 1998 /usr/src/linux-2.2.10/include/li
rw-r--r-- 0/0   1364 May 14 01:41 1997 /usr/src/linux-2.2.10/include/li
rw-r--r-- 0/0   1966 Jul 30 14:17 1998 /usr/src/linux-2.2.10/include/li
rw-r--r-- 0/0   2674 Jul 30 14:17 1998 /usr/src/linux-2.2.10/include/li

[Done]
```

Figure 259. Recovery report

Your files were recovered successfully!

### 15.2.5 Master and incremental backups

Usually system administrators perform so-called master and incremental backups. The master backup is a backup of all files on the system. Incremental backup is a backup of only those files that have changed from the last master backup. When you need to restore your data, restore the master backup and the last incremental backup. BackupEDGE can perform different types of incremental backups. Refer to the BackupEDGE manual for the explanation of them. Master and incremental backups can be performed from the edgemenue utility.

To perform a master backup follow these steps:

1. Start the edgemenue program by executing the following command:

```
edgemenue
```

You will see a window similar to Figure 251 on page 271.

2. Select **Backup > Master Backup**, and you will see a window similar to Figure 260.

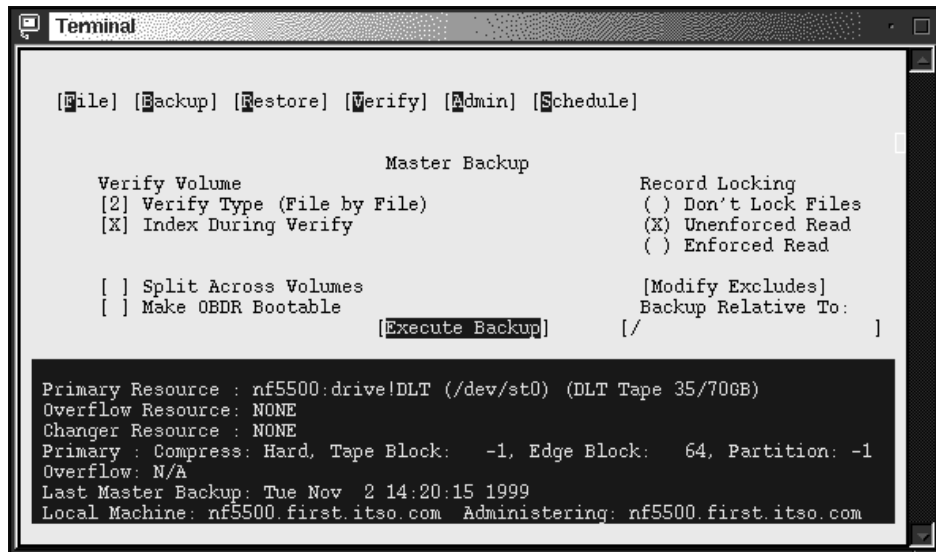


Figure 260. Starting the master backup

3. Choose the options you want and select **Execute Backup** to start the backup. You will see a window similar to Figure 261.

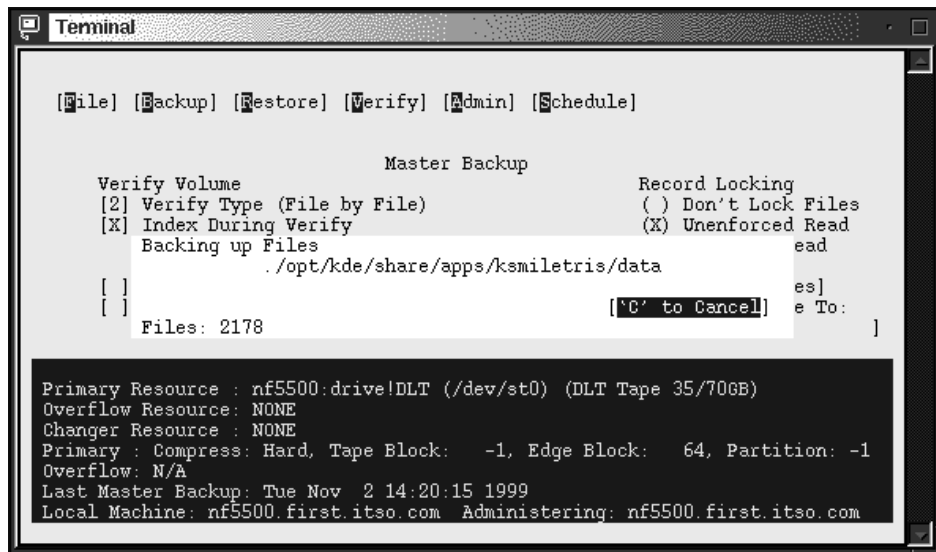


Figure 261. Master backup in progress

When the backup is finished you will see a window similar to Figure 262.

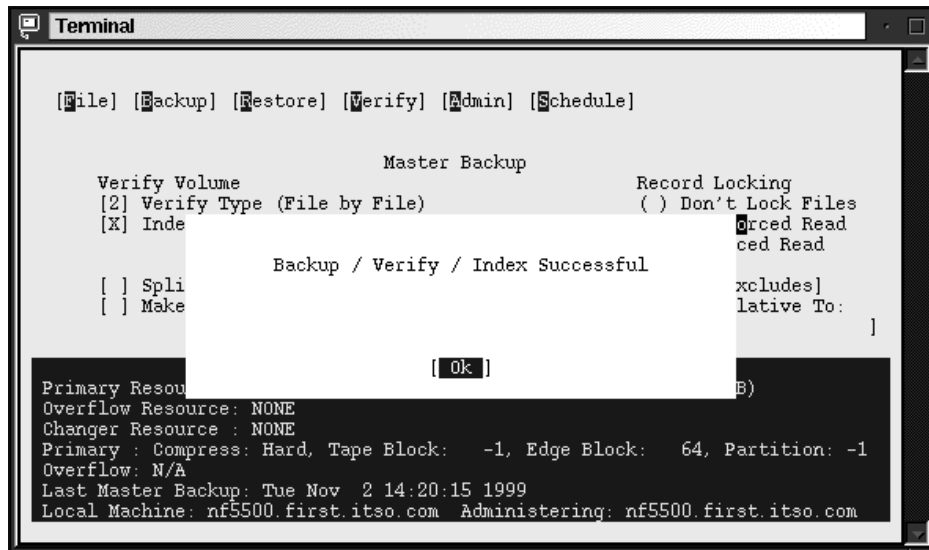


Figure 262. Master backup completed

Select **OK** to finish the operation, and you will see a backup report similar to Figure 263.

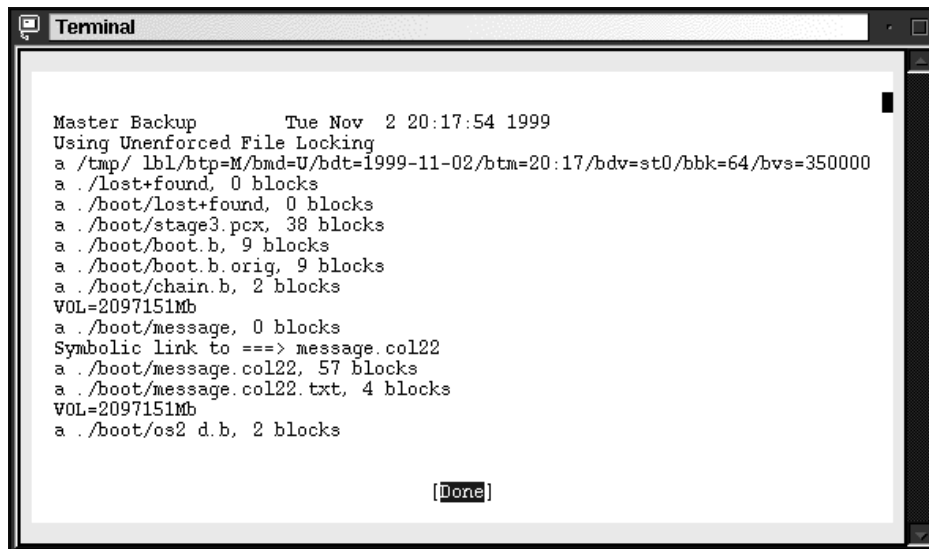


Figure 263. Master backup report

To perform incremental backups select **Backups > Incremental Backup**. Then follow the instructions in the window; they are similar to the ones for master backup.

### 15.2.6 Restoring master and incremental backups

To restore master and incremental backups you can use the edgemenue utility. When you start the utility and choose **Restore** you will see a window similar to Figure 264.

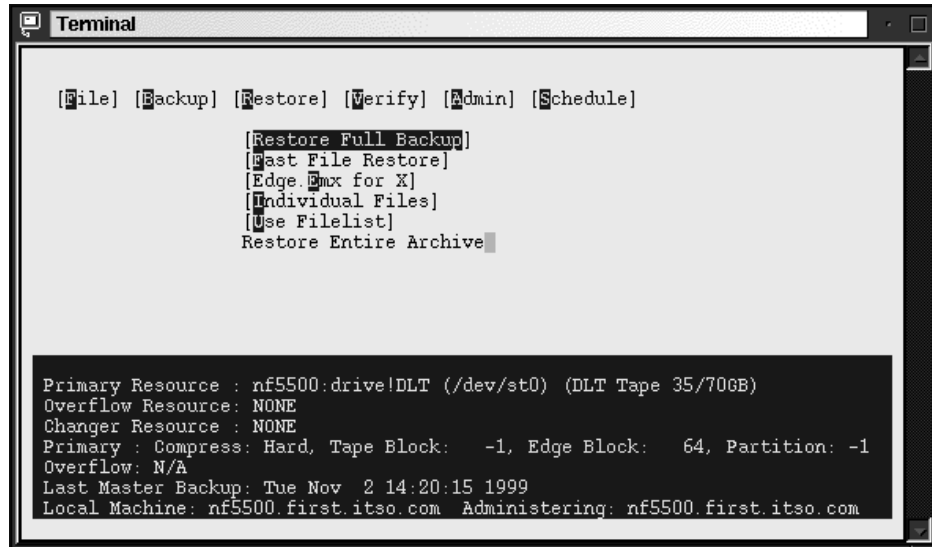


Figure 264. Starting restore full backup

Select **Restore > Restore Full Backup** and you will see a window similar to Figure 265.

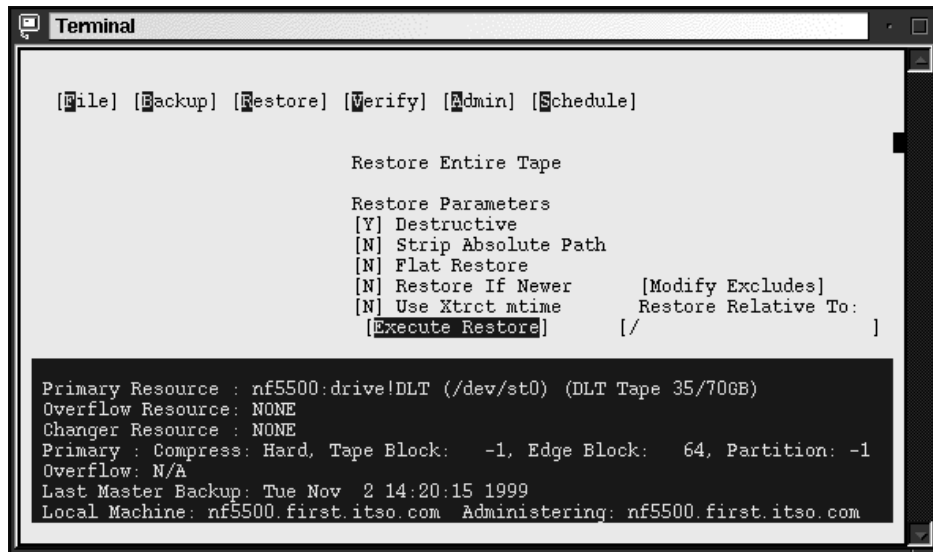


Figure 265. Full backup restore options

Choose your options and select **Execute Restore** to start restoring files.

### 15.2.7 Performing scheduled backups

To perform scheduled backups, you can use the `edge.nightly` utility included with BackupEDGE. To start this utility, execute the command:

```
/usr/lib/edge/bin/edge.nightly
```

But before you can use scheduled backups, you need to define them. To do this follow these steps:

1. Start the `edgemenu`.
2. Select **Schedule > Nightly Scheduling**. You will see a window similar to Figure 266.

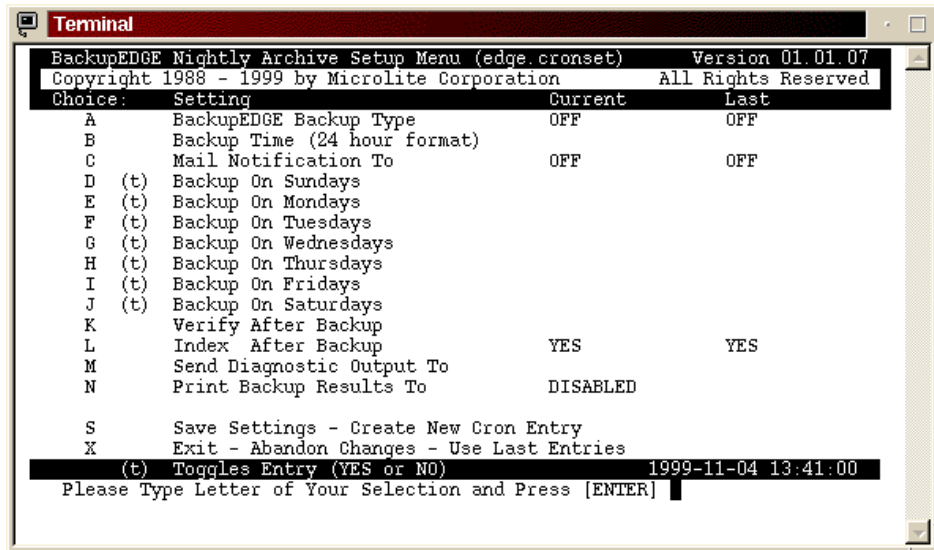


Figure 266. Schedule setup

- Here you can define the schedule for your backups. You need to define the type and time of the backup. To define the type of the backup select **A** and press Enter, and you will see a window similar to Figure 267.



Figure 267. Defining the type of backup

- Specify the type of backup you want to perform. In our example we selected **M** for master backup. You will be returned to the main window.

**Note**

You cannot mix master and incremental backups. If your master backup fits on one tape cartridge, we recommend that you do a master backup daily. If your master backup will not fit on one tape cartridge, do a manual master backup once a week and do incremental backups daily.

- Next you need to specify the time of everyday backup by selecting **B** and pressing Enter. You will see a window similar to Figure 268.

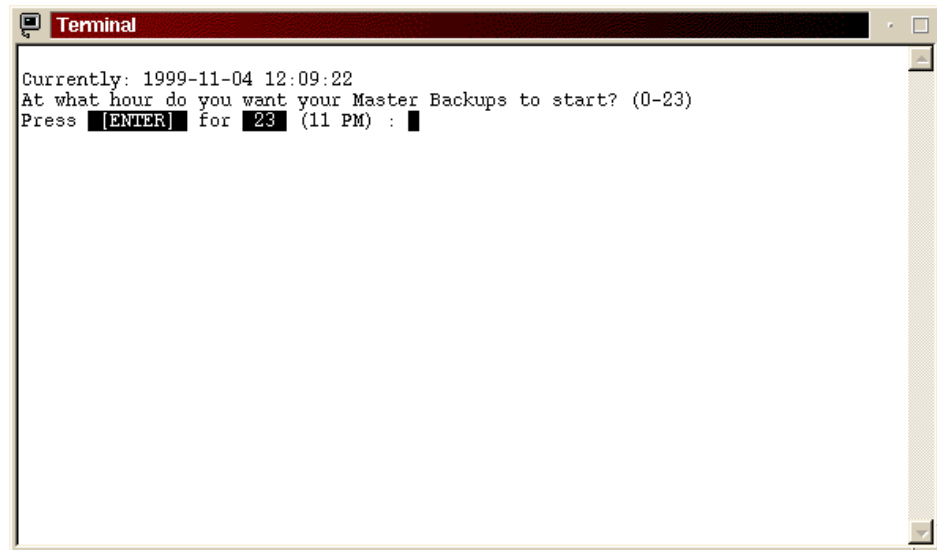


Figure 268. Setting the time

- Define the time for your backups. You will see a window similar to Figure 269.



```

Terminal <2>
BackupEDGE Nightly Archive Setup Menu (edge.cronset)          Version 01.01.07
Copyright 1988 - 1999 by Microlite Corporation              All Rights Reserved
Choice:  Setting                      Current                Last
A      BackupEDGE Backup Type         Master                Master
B      Backup Time (24 hour format)    12:30                 12:30
C      Mail Notification To            root                  root
D (t)  Backup On Sundays               YES                   YES
E (t)  Backup On Mondays               YES                   YES
F (t)  Backup On Tuesdays             YES                   YES
G (t)  Backup On Wednesdays           YES                   YES
H (t)  Backup On Thursdays            YES                   YES
I (t)  Backup On Fridays                YES                   YES
J (t)  Backup On Saturdays              YES                   YES
K      Verify After Backup              BIT                   BIT
L      Index After Backup                YES                   YES
M      Send Diagnostic Output To        /dev/null
N      Print Backup Results To          DISABLED

S      Save Settings - Create New Cron Entry
X      Exit - Abandon Changes - Use Last Entries
(t)    Toggles Entry (YES or NO)       1999-11-04 12:27:39
Please Type Letter of Your Selection and Press [ENTER]

```

Figure 269. After schedule definition

7. Select **S** and press Enter to save the settings. The configuration program will create an entry in the cron database for executing the `edge.nightly` utility. From now on, cron will execute the backup utility as you defined in the previous steps.

**Note**

Before you start using scheduled backups, check if you need to copy the file `/usr/lib/edge/bin/S88egde` to the `/etc/rc.d/rc2.d` directory. This script will clear all zombie PIDs from the `edge.nightly` on the system restart.

You can also start `edge.nightly` from your own scripts. When you start it from a command line or a script, you have to be logged in as root. After `edge.nightly` is started it will perform an immediate backup.

### 15.2.8 Configuring the tape devices

Any time after installation you can define or change your backup device. To accomplish this follow these steps:

1. Start the `edge.resmgr` resource manager by executing the command:

```
/usr/lib/edge/bin/edge.resmgr
```

You will see a window similar to Figure 270.

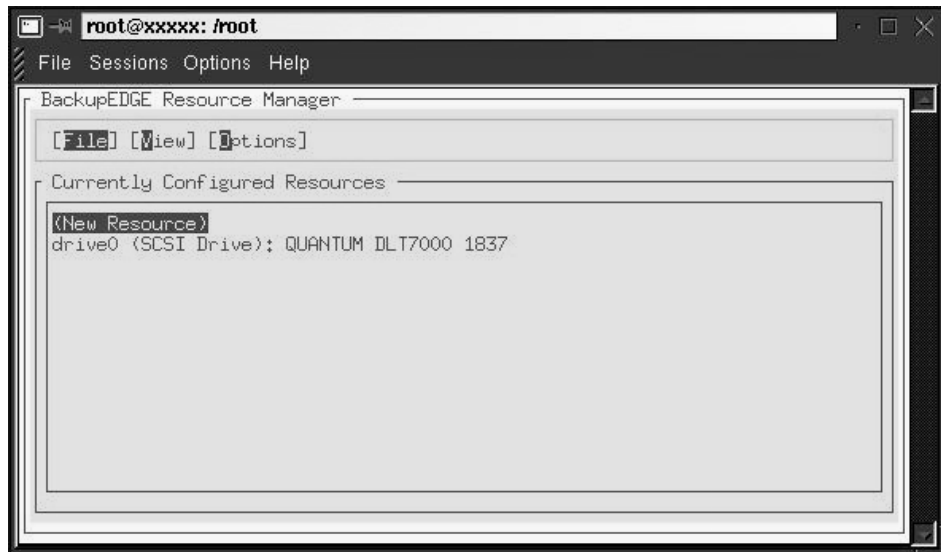


Figure 270. Starting the resource manager

2. Select **New Resource** and press Enter. You will see a window similar to Figure 271.

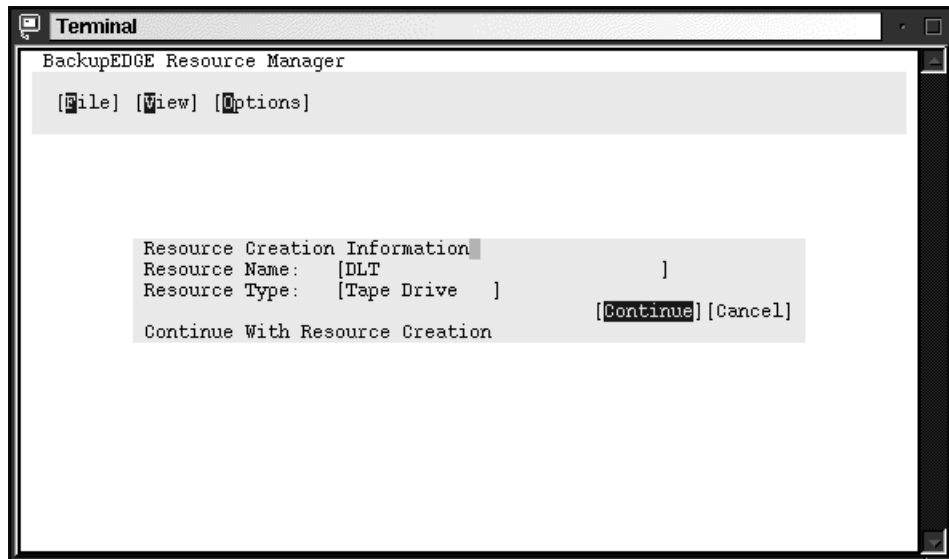


Figure 271. Defining the resource name

3. Type in the resource name and select a resource type. Select **Continue** to go on. You will see a window similar to Figure 272.

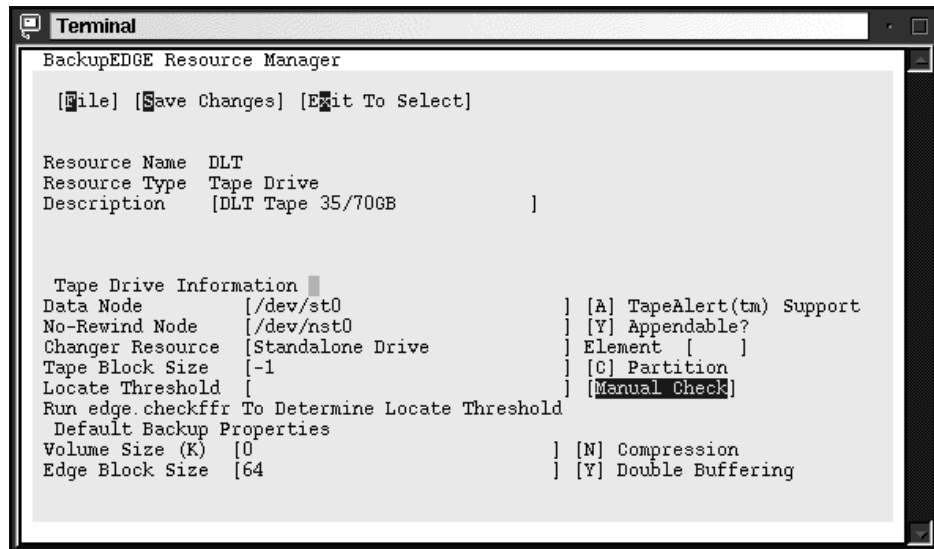


Figure 272. Parameters for the tape

4. Type in the description, data node and no-rewind node. In our example, the data node is `/dev/st0` and no-rewind node is `/dev/nst0`. You can leave all other fields as default.
5. Select **Manual Check** to define other parameters automatically. You will see a window similar to Figure 273.

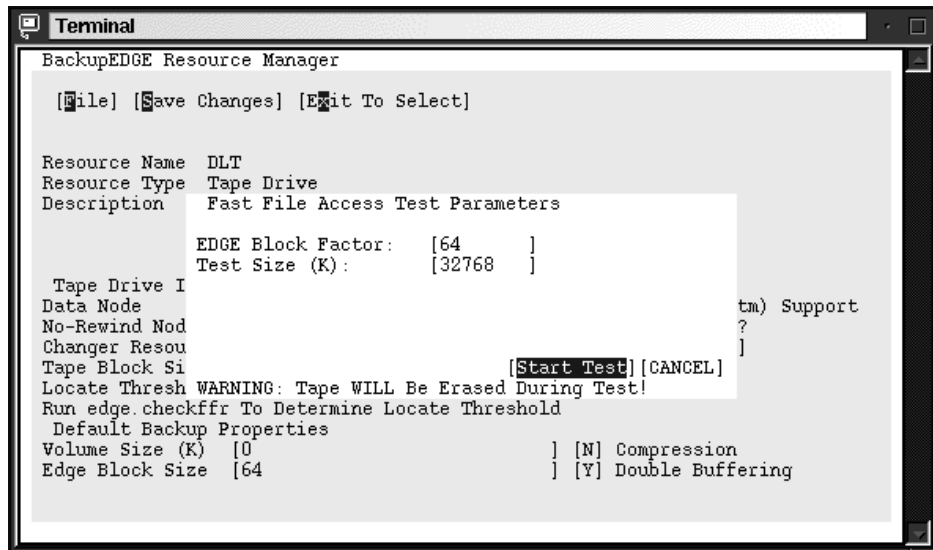


Figure 273. Setting the parameters for tests

- Here you can select the block factor and the test size. Select **Start Test** to continue. You will see a window similar to Figure 274.

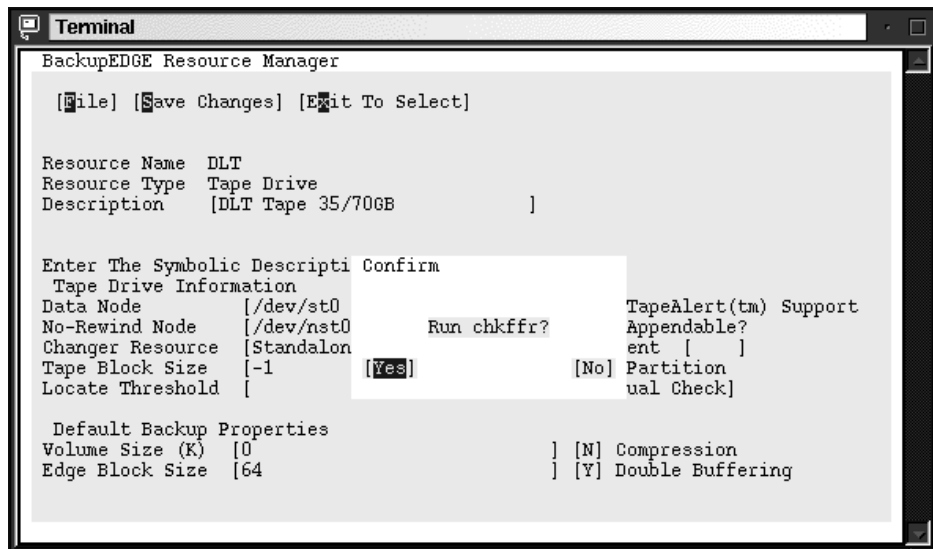


Figure 274. Starting the test

**Stop**

Performing this test will destroy all data on the tape.

7. Select **Yes** to continue. You will see a window similar to Figure 275.

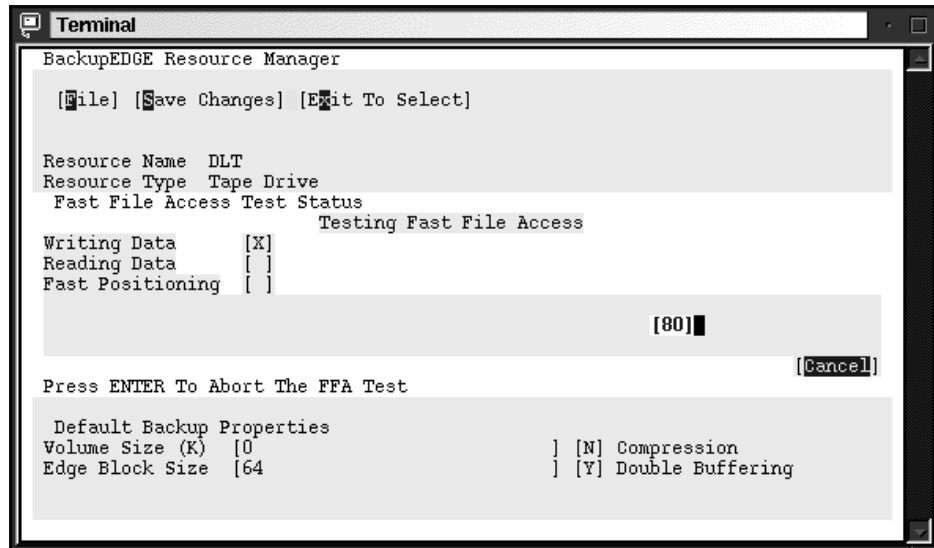


Figure 275. Performance test

After the test is done you will see a window similar to Figure 276.



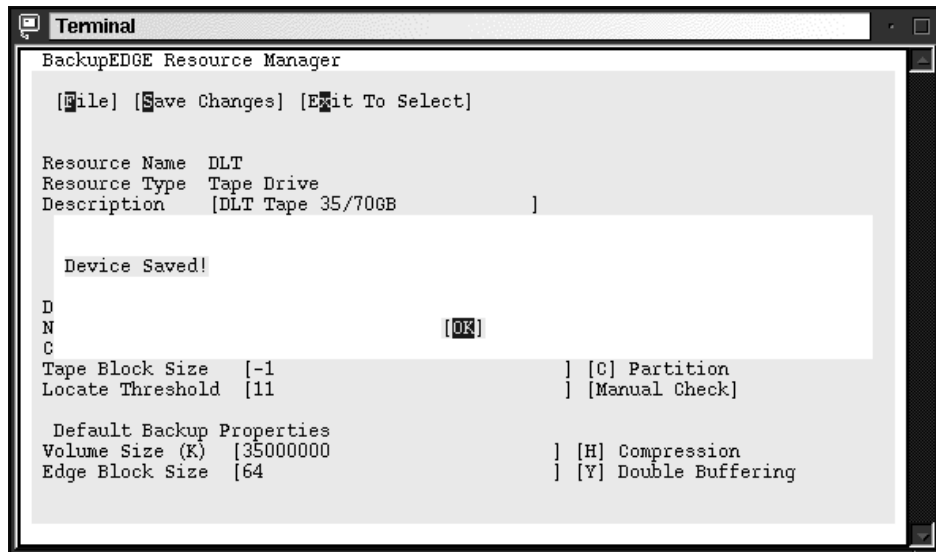


Figure 277. Saving the device definitions

### 15.2.9 Defining the devices for making backups

Any time after installation when you configured your backup hardware device, you can change which device the backup software uses for each user performing backups. If you are logged in as root, you will define devices for the root user. Usually this is the only user doing backups on the system. Follow these steps to enter the resource manager for backup:

1. Start the edge.config configuration menu by executing the command:

```
/usr/bin/edge.config
```

You will see a window similar to Figure 278.

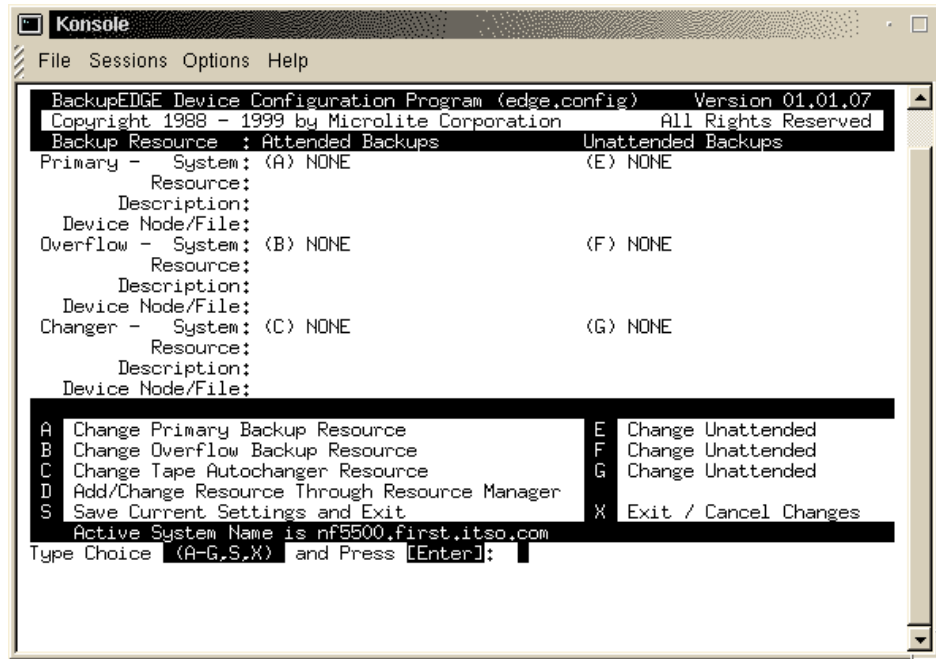


Figure 278. Device Configuration

2. Here you need to define the devices for attended and unattended backups.
3. Type in A and press Enter to define the device for attended backups. You will see a window similar to Figure 279.



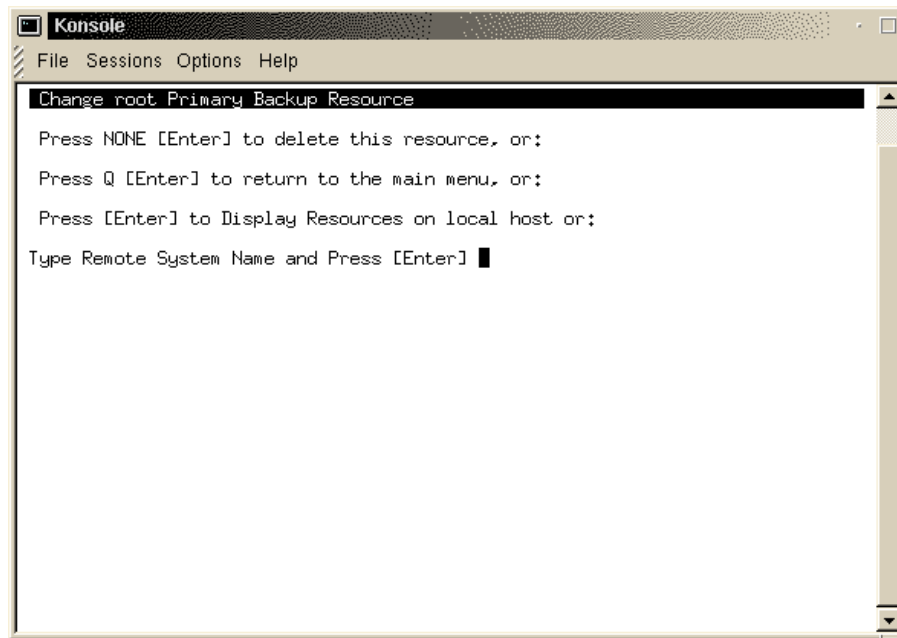


Figure 279. Selecting the device for backup

4. Press Enter to continue. In the next window you will see all defined backup devices. Type in the device you want and press Enter to continue. You will see a window similar to Figure 280.

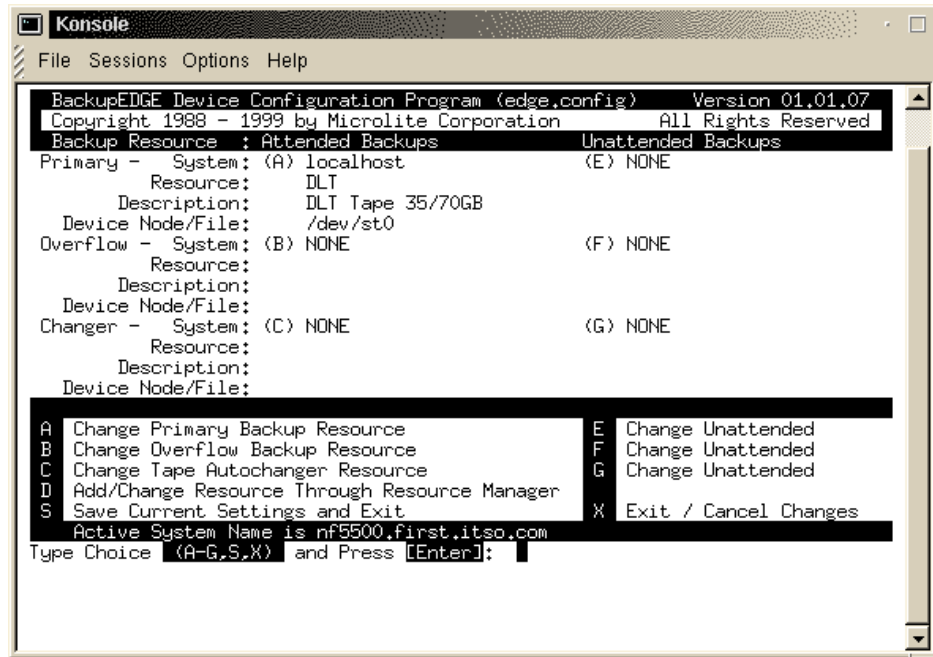


Figure 280. After definition of attended backup device

5. Follow the steps from 1- 4 for the unattended device also.

### 15.2.10 Microlite RecoverEDGE

By using the RecoverEDGE tools you can create emergency recovery diskettes to rebuild your system in the case of disaster. RecoverEDGE handles the details of reconstructing your FDisk, divvy, and/or slice tables, rebuilding your file systems and restoring your data, even if your hard drive size has changed. RecoverEDGE uses your live system backups, so there is no need to shut down your system in order to protect it. You can even restore your system over the network.

With RecoverEDGE restoring the system is very easy. To recover the system you should follow these tasks:

1. Identify and correct the cause of the failure.
2. Boot from the RecoverEDGE disks.
3. Reconfigure your file systems.
4. Restore your backups.
5. Shut down and reboot.

6. System is ready to use.

**Note**

RestoreEDGE uses your master and incremental backups for recovery, so the accuracy of the data depends on these backups.

### 15.2.10.1 Creating the RecoverEDGE boot disks

Before you can use RecoverEDGE for disaster recovery you should build a set of boot disks. To create the boot disks follow these steps:

1. Start the utility for creating the RecoverEDGE boot diskettes:

```
/usr/bin/re2
```

or go to **Admin>Make RecoverEDGE Media** in the menu.

You will see a window similar to Figure 281.

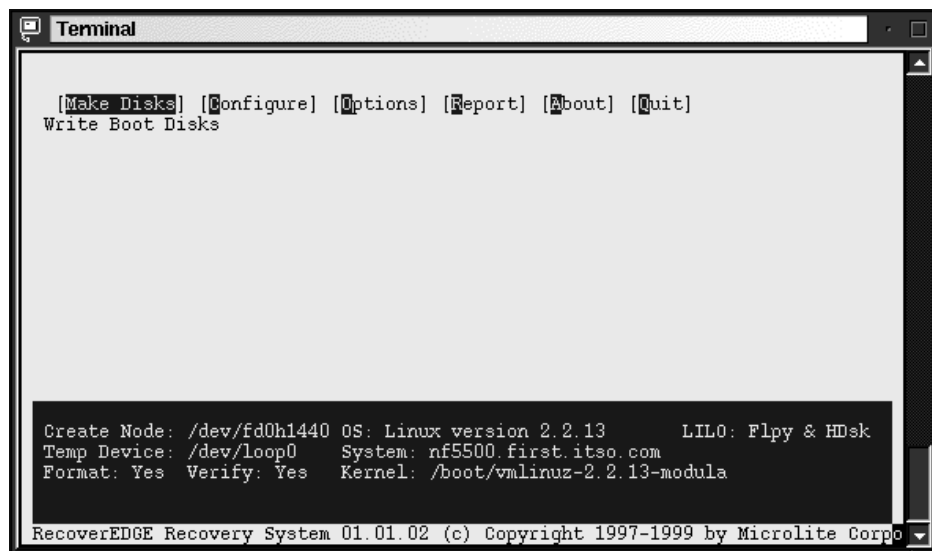


Figure 281. RecoverEDGE utility

2. Select the **Configure** option and press Enter, and you will see a window similar to Figure 282.

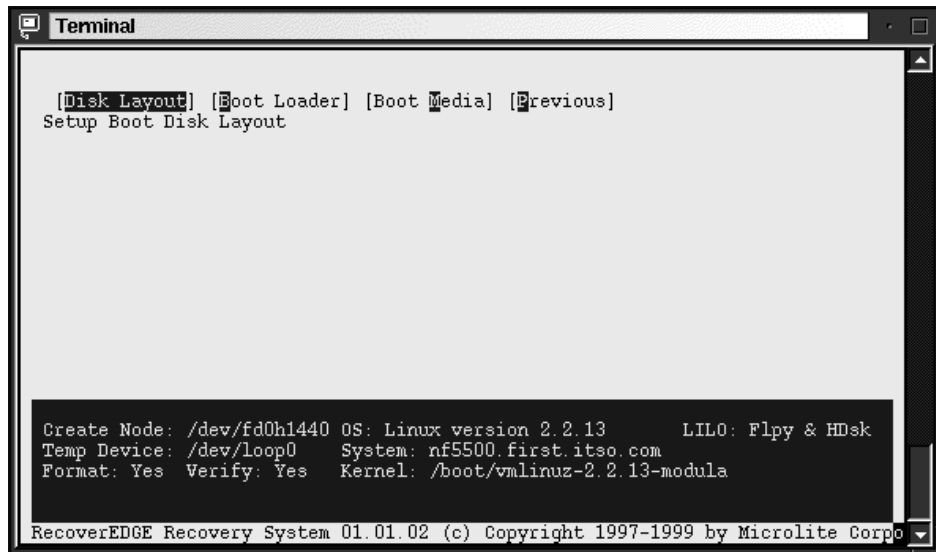


Figure 282. Configure menu

3. Select the **Disk Layout** option and press Enter, and you will see a window similar to Figure 283.

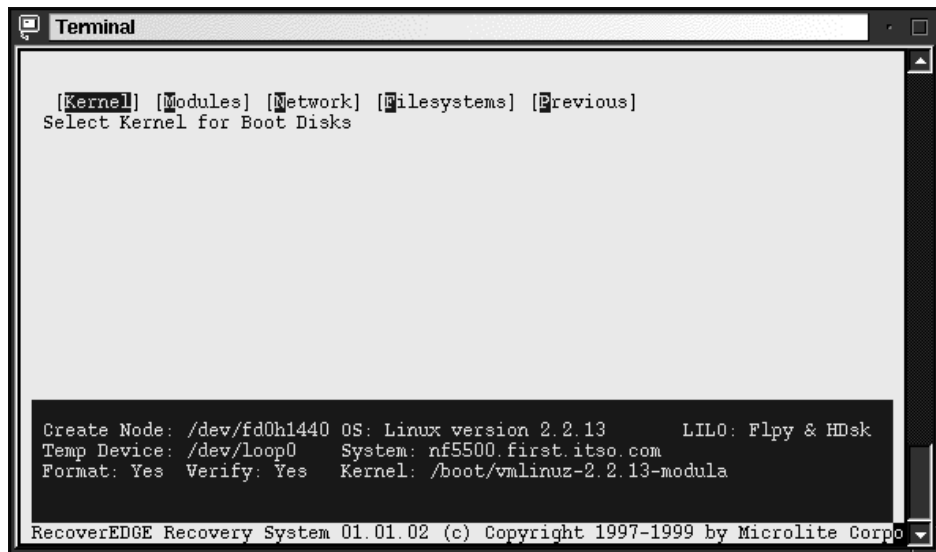


Figure 283. Disk layout menu

- Here you can configure the kernel, modules, network and the file systems for your RecoverEDGE boot disks. Select the **Kernel** option and you will see a window similar to Figure 284.

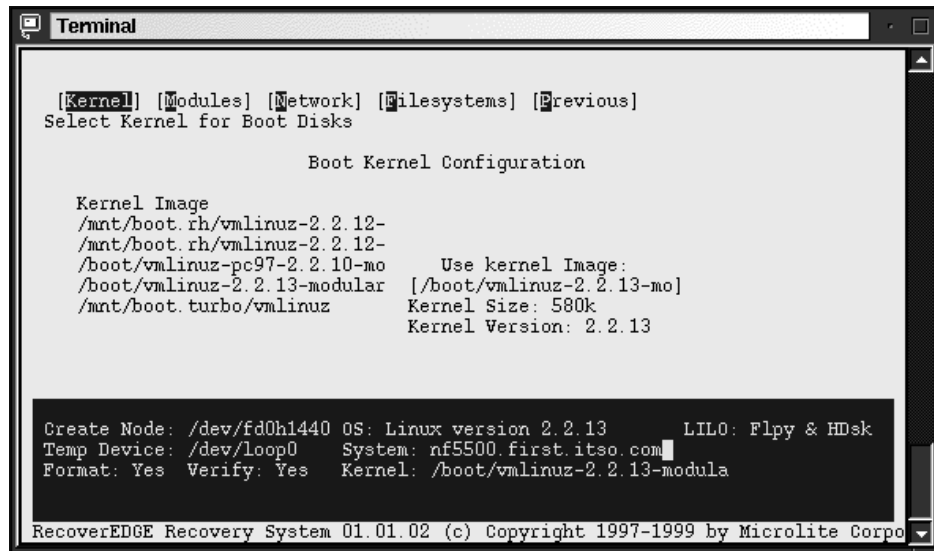


Figure 284. Kernel options

Here you define which kernel will be used for creating the diskette.

- Return to the previous stage and select **Modules** and press Enter, and you will see a window similar to Figure 285.

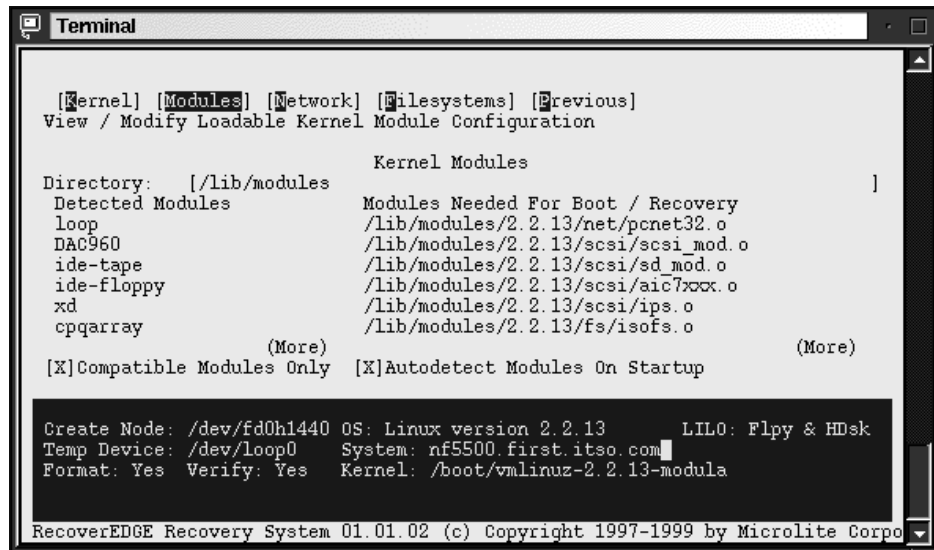


Figure 285. Modules options

Here you define which modules will be used for building the initial RAM disk for the recovery system. In the Directory field you can specify the path to the modules that corresponds to the kernel you defined for booting. If you choose the option **Autodetect Modules on Startup**, RecoverEDGE will load currently loaded modules.

**Note**

Do not forget to include the module for the tape drives.

- Return to the previous stage and select **Network** and press Enter, and you will see a window similar to Figure 286.

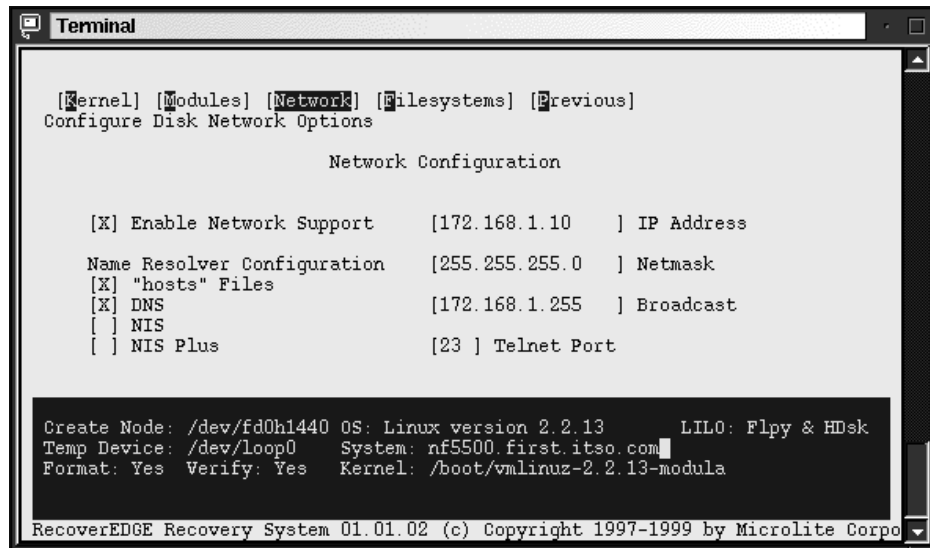


Figure 286. Network options

Here you define your network setup in case you will restore the system from a tape device on the network. You do not need this if you have a locally attached tape.

7. Return to the previous stage and select **Filesystems** and press Enter, and you will see a window similar to Figure 287.

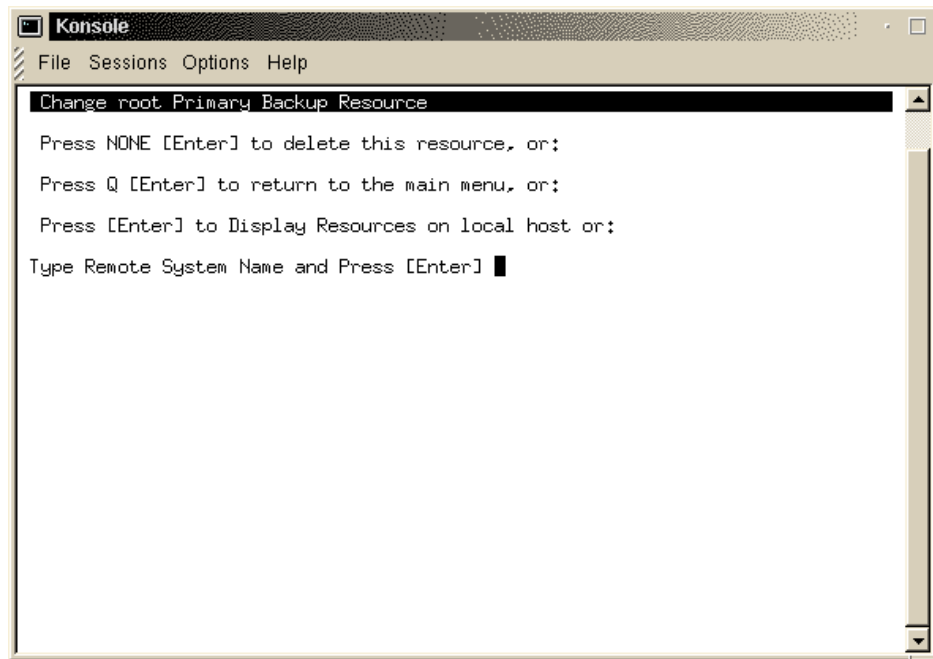


Figure 287. Filesystems options

Here you define which mounted file systems will be recovered.

8. Return to the configuration panel and select the **Boot Loader** option and press Enter. You will see a window similar to Figure 288.



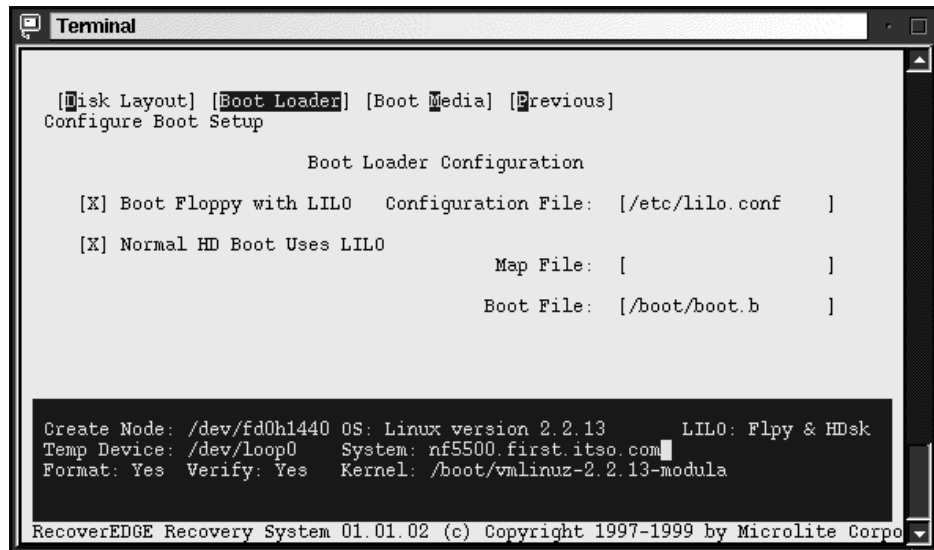


Figure 288. Boot Loader options

Here you define options for the Boot Loader.

- Return to the configuration panel and select the **Boot Media** option and press Enter. You will see a window similar to Figure 289.

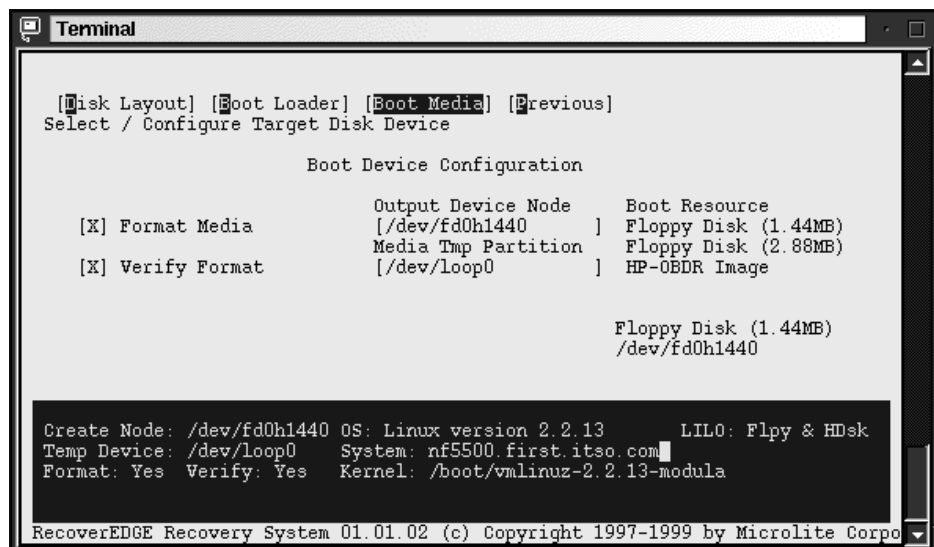


Figure 289. Boot Media options

Here you define how the boot diskettes will be created.

10. After you configured all settings return to the main window and select **Make Disks**. You will be prompted to insert three diskettes.

**Note**

If you get an error that diskettes cannot be created, the probable cause is that images are too big. Try to reduce the number of loaded modules or even make the special kernel just for this purpose, throwing out all unnecessary things.

After the diskettes are created you are ready to deal with disaster on your system. But before this really happens, try to boot from these diskettes and verify if your tape device is recognized.

#### 15.2.10.2 Verifying the RecoverEDGE boot diskettes

To verify the diskettes, boot from the first diskette and follow instructions on the window. When the system is started you will get the RecoverEDGE main menu. Select **Utilities > Tape Drive**.

In the Tape Device Node field, you see the defined tape device. Go to the Test Tape Drive field and test your tape device. If the test is successful your recovery set is ready to use.

#### 15.2.10.3 Recovering from a total crash

To recover from a disaster crash follow these steps:

1. Resolve all hardware problems.

**Note**

Before restoring the system, initialize the Master Boot Records of all disk drives.

2. Boot the server from the first RecoverEDGE boot diskette.
3. When you are prompted to insert the root diskette, insert the second RecoverEGDE boot diskette. After the diskette is loaded, RecoverEDGE will start and you will see a window similar to Figure 290.

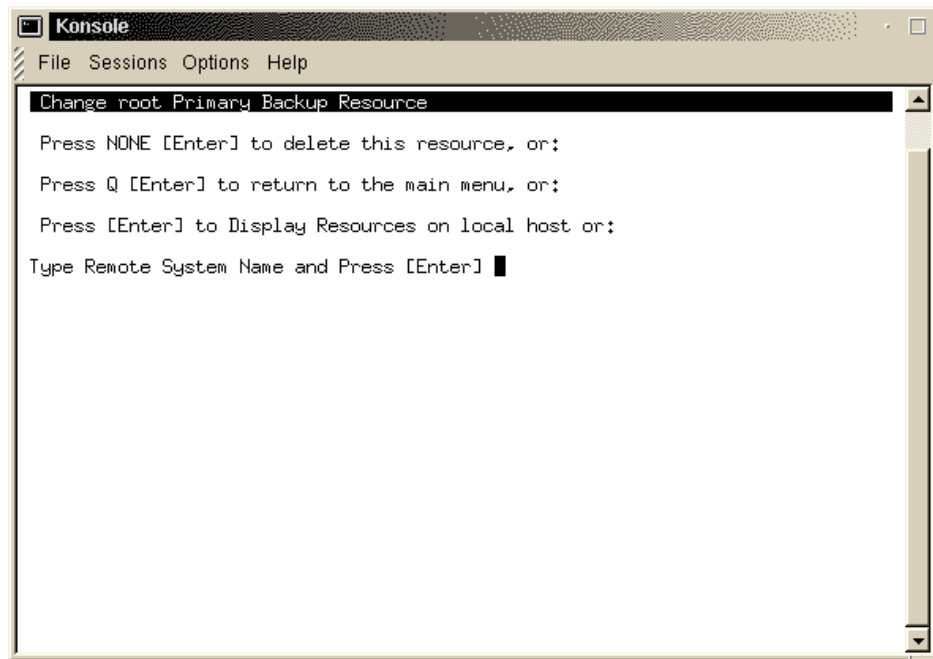


Figure 290. RecoverEDGE initial window

4. Select **Restore > One Touch**. Follow the instructions on the window to complete the recovery.



**Note**

For recovery you will use your master and incremental backups.

5. When all files are backed up, press a key to get back to the main window. All the file systems will be then synchronized and LILO will be set up and executed.
6. Before you reboot, switch to a console 2 with Alt+F2 and execute the following commands to check the fstab file for correct entries for your system:

```
mount /dev/sdb6 /mount  
cat /mount/etc/fstab
```

In our example `sdb6` is our root partition. You should use your root partition here.

That is all there is to it. Your restored system is ready to use.

### 15.2.11 More information on Microlite

For information on advanced features consult the *Microlite User's Guide* or the Microlite Web site at:

<http://www.microlite.com>

---

## 15.3 Arkeia

Arkeia is a complete client/server backup solution for Linux and other platforms. With Arkeia you can safely archive every file, directory, device node and special file on your file systems. Unlike standard UNIX tar command, which ignores many important files, Arkeia also verifies the data written to tape to ensure that the tape is an accurate reflection of your data. Below are the features provided by Arkeia backup software:

- Data Compression - automatic data compression is supported.
- GUI Interface. A CLI-interface is also available.
- The backup server may be your local system or a remote system.
- High Performance - advanced double buffering and variable block factors.
- Virtual File Support - you can back up virtual (sparse) files.
- Multi-Volume / Multi-Device Archives - automatic spanning across multiple volumes or devices.
- Wildcard Support - when selecting files you can use a wildcard.
- Raw Device Backups - you can archive an entire raw device/partition to tape.
- Master / Incremental Backups
- Unattended Operation - you can configure schemas to periodically perform full backups and/or incremental backups.

Arkeia is designed to operate on Linux kernels 2.x and there are available versions for several types of libraries (libc5 and libc6) and distributions.

Requirements for the server:

- A 486 processor or higher
- 32 MB RAM
- 1 GB disk space
- SCSI adapter card
- SCSI tape drive
- TCP/IP services
- Linux 2.0 or higher

Requirements for the client:

- A 486 processor or higher
- 5 MB disk space

In the following sections we describe how to install, configure and use the Arkeia backup software.

### 15.3.1 Installing Arkeia

Arkeia is available in different package formats (tar, rpm) for different distributions either on CD or downloadable from Arkeia's Web site (follow the link <http://www.arkeia.com>) in the DOWNLOAD AREA. To install Arkeia, we recommend that you follow the installation procedure described in the *Installation and Quick Start Manual*. You can find this manual on the Arkeia-CD or download it from Arkeia's Web site.

On the Arkeia server, you must also install the client and the GUI package. These packages are required to configure the backup server. After the installation of the client and GUI packages, you can install the server package.

### 15.3.2 Configuring Arkeia

Before you can configure Arkeia, check whether the Arkeia backup server is running. To do this, enter:

```
ps -ef | grep -v grep | grep nlservd
```

on the system which should be used as your backup server. If you see a line like

```
root 488 1 0 09:06 ? 00:00:00 /usr/knox/bin/nlservd start
```

the backup server is running. To begin with the configuration of Arkeia, be sure, you have X-Windows running. Then enter on the command line:

```
Arkeia
```

You will see a dialog like Figure 291:



Figure 291. Arkeia initial window

The field for the server name is by default filled in with the name of the system you currently work with. You must change this field if you have installed the server component on another system.

The field for the login name is by default filled in with `root`. Change it if you have changed the name of the Arkeia administrator.

The field for the password is empty by default. You have to enter the password when you have changed the password. The main dialog window of Arkeia appears (Figure 292):

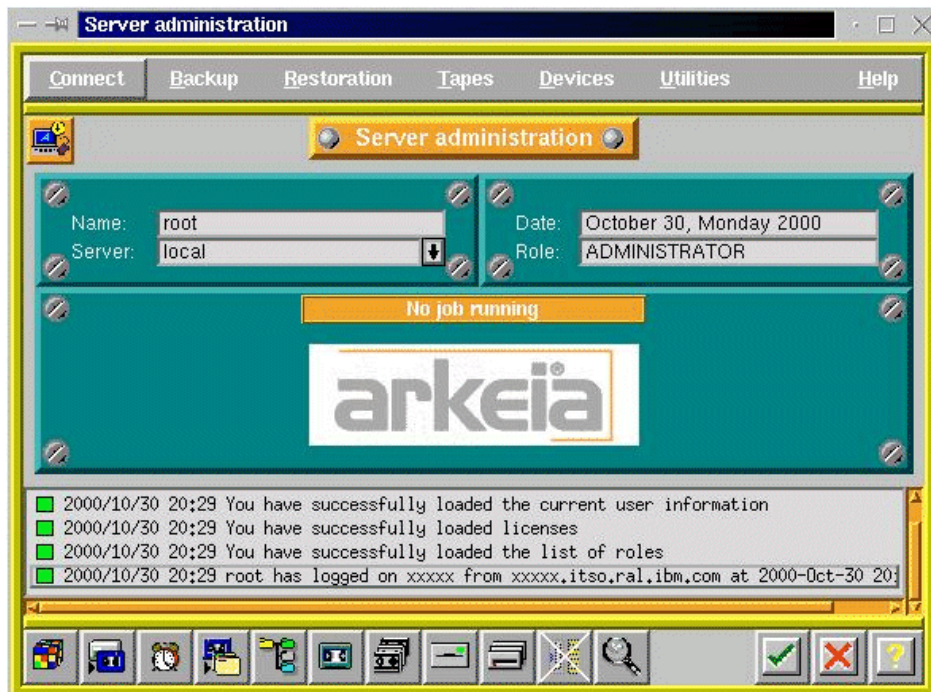


Figure 292. The Arkeia main dialog window

If you want a simpler layout of the window, go to **Utilities -> Setting** in the menu bar and modify the appearance of the windows. Click the **OK** button, save the new setting, and click the **OK** button again. Now, you will get a window similar to the window in Figure 293.

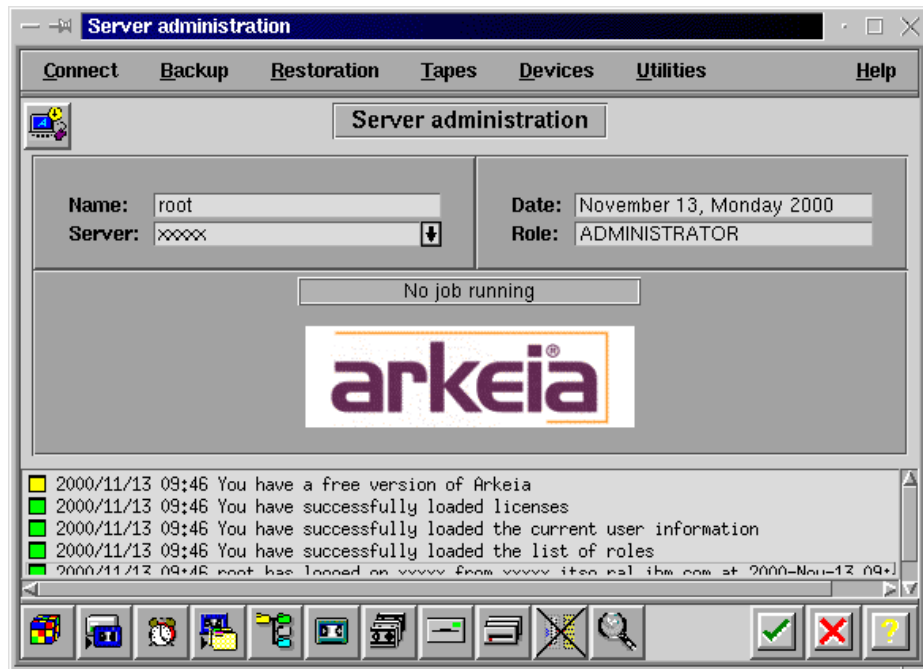


Figure 293. The new Arkeia main dialog window

At the bottom of the window you see push buttons shown in Figure 294:



Figure 294. Bottom part of main window

The meaning of these buttons is, from left to right:

- Refresh job
- Interactive backup
- Periodic backup
- Restoration
- Savepacks
- Tapes management
- Pools management
- Drives management
- Drivepacks
- Libraries management
- Backup done



- OK button. Clicking this button opens a new Welcome dialog.
- Cancel button. Clicking this button to leave Arkeia.
- Help

Before you can begin with your first backup, you must carry out the following configuration steps:

- Pool management
- Tape management
- Drives management
- Drivepacks management
- Savepacks management

Let us start with tape pool management. Click the pools management button on the bottom of the main dialog or click **Tapes -> Pools management** on the menu. The pools management window appears as in Figure 295:

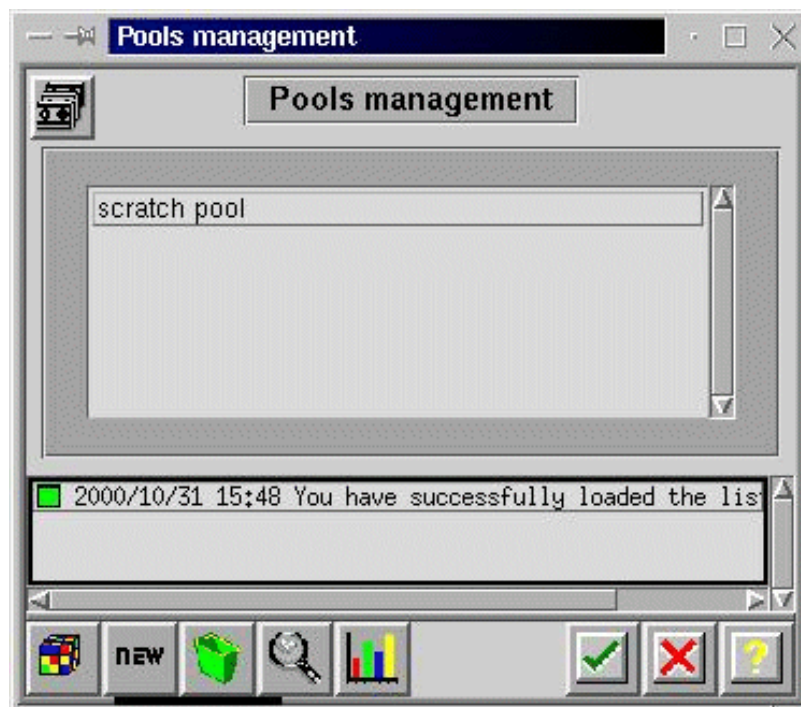


Figure 295. Pools management main dialog window

The scratch pool exists by default. To create a new tape pool, for instance for your backup tapes, click the **new** button. The pool creation dialog appears as in Figure 296:

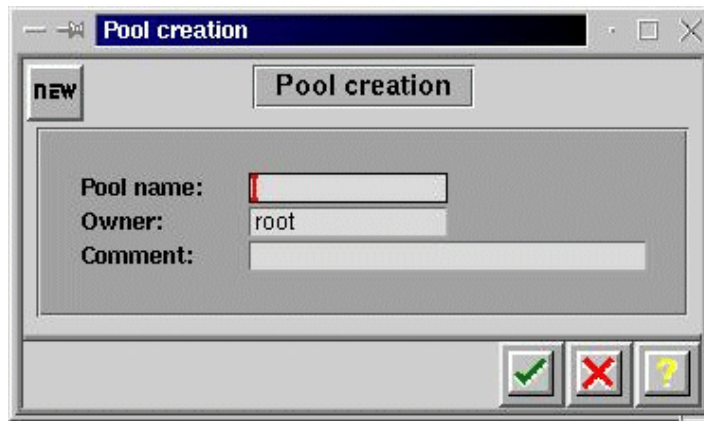


Figure 296. Pool creation window

Fill in the dialog fields with the appropriate information and click the **OK** button. The pools management main windows appears with the pool list updated as in Figure 297:

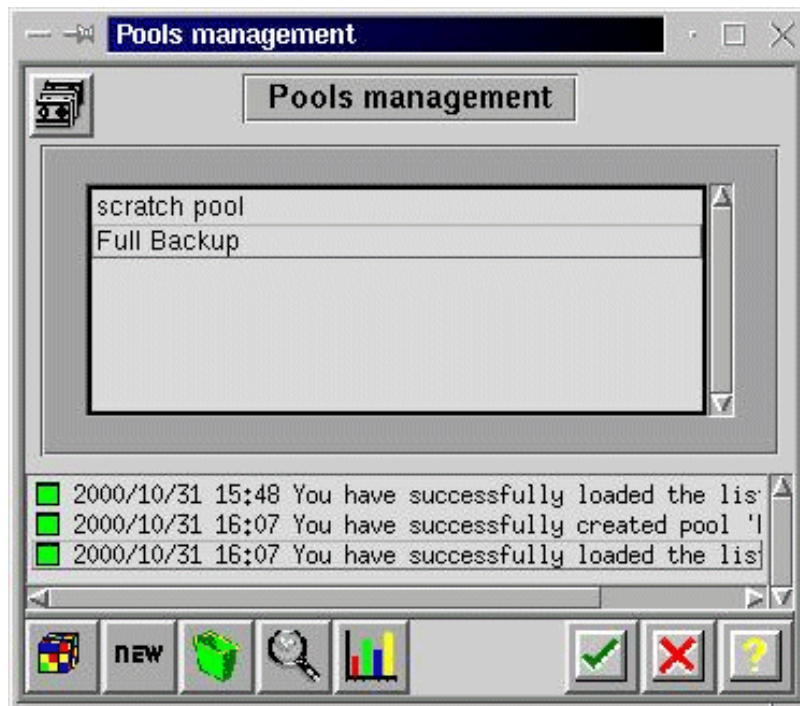


Figure 297. Pools management main window with updated pool list

To return to the main dialog, click the **OK** button. Now we can fill the Full Backup pool with tapes. To do this, click the tape management button or click **Tapes -> Tapes management** in the menu. The tapes management main window appears (Figure 298):

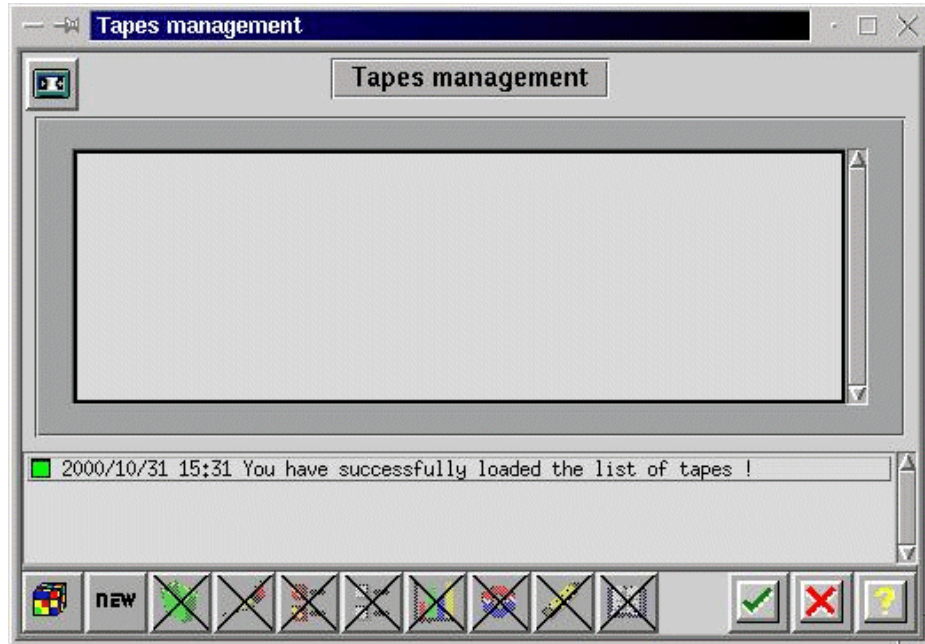


Figure 298. Tape management main window

Click the **new** button to enter new tapes (Figure 299):

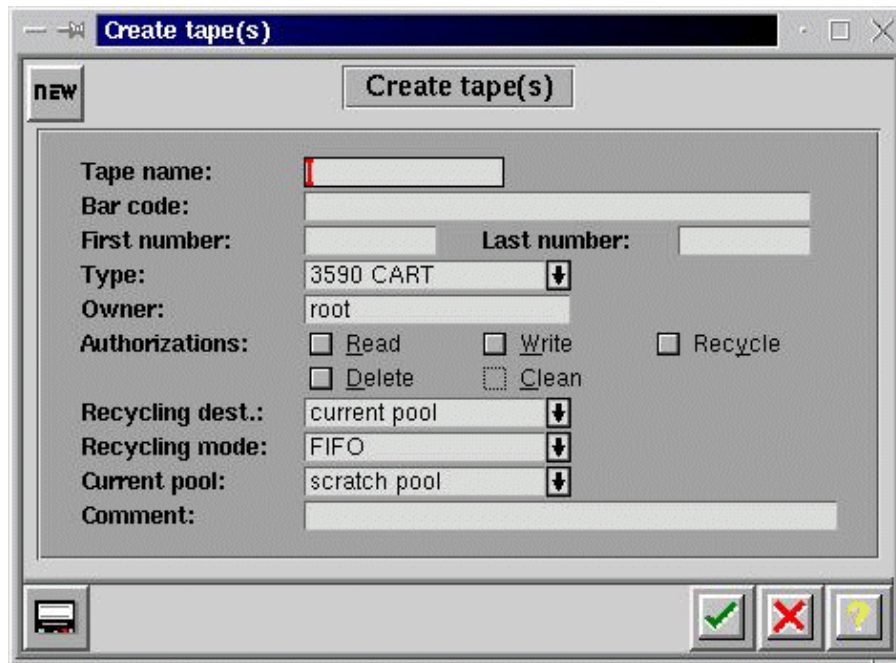


Figure 299. Create tape(s) window

The tape name consists of a fixed part and a variable part. The fixed part can be any text, while the variable part is a number. Enter the first part of the tape name, the first and the last number of the tapes to be used, and the tape type (DAT, DLT, etc.). Choose the pool these tapes should belong to and enter a comment in the comment line. Click the **OK** button to return to the tapes management main window. The tapes management main window appears with the updated list of currently created tapes. Click the **OK** button in this window to return to the main window.

After the creation of tape pools and tapes, we can create drives and drive packs.

Drives must be created first. To do this, click the drives management button in the main window or click **Devices -> Drives management** in the menu. The drives management window appears (Figure 300):

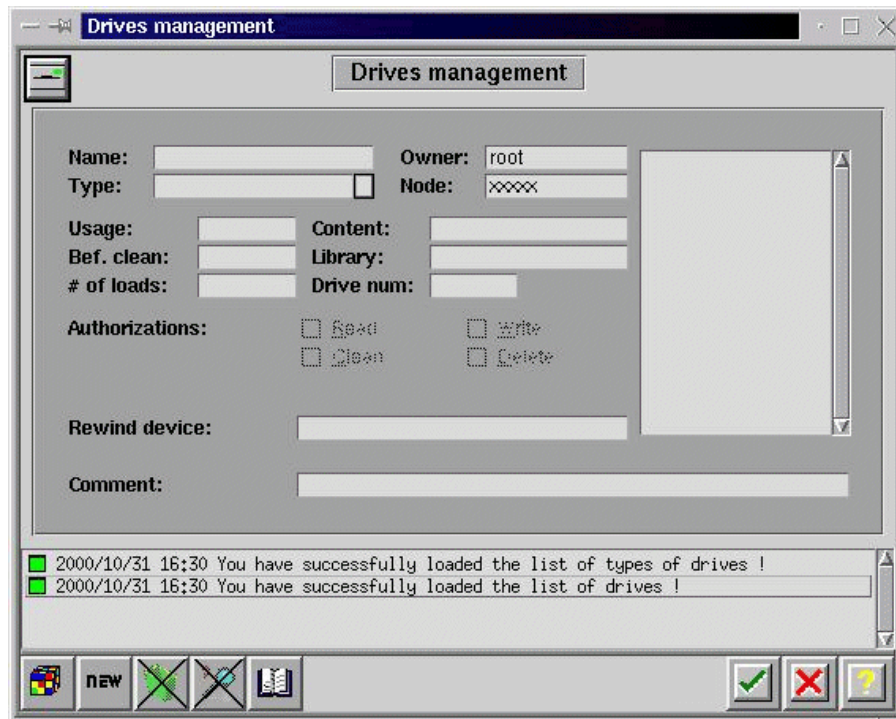


Figure 300. Drive management window

Click the **new** button to fill in the fields with the appropriate information. The fields Name and Rewind Device must be filled. Do not forget to choose the correct tape type in the Type field. To return to the Arkeia main window, double-click the **OK** button.

Now we can generate drivepacks. Press the drivepacks button or click **Devices -> Drivepacks** on the menu. The Drivepacks window appears (Figure 301):

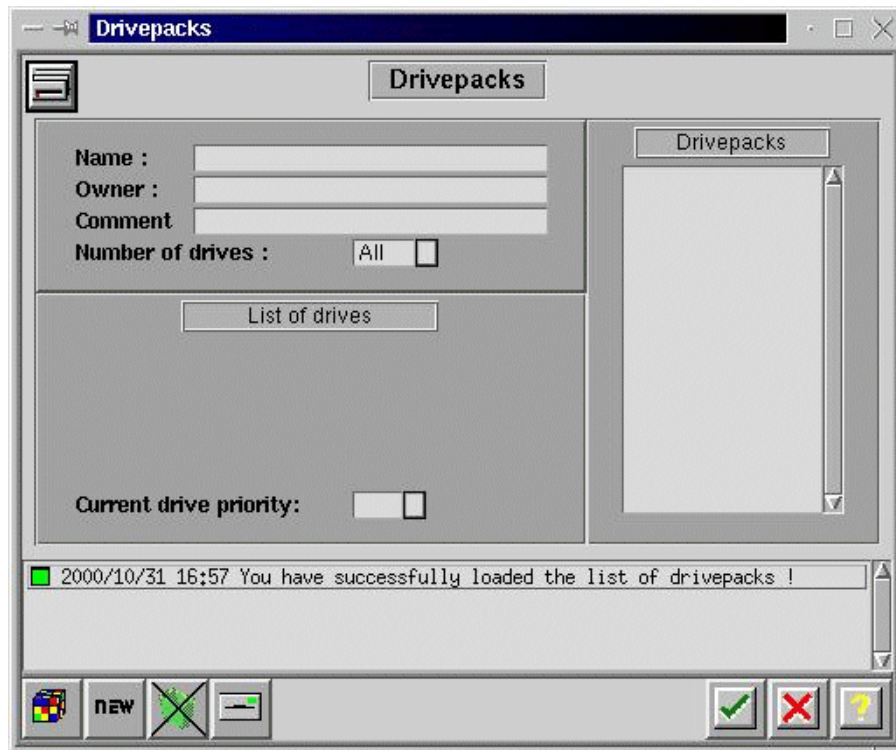


Figure 301. Drivepacks management window

Click the **new** button to fill in the fields. Fill in the Name field and choose one entry in the drives list and click the **OK** button to update the list of existing drivepacks on the right side of the window (Figure 302).

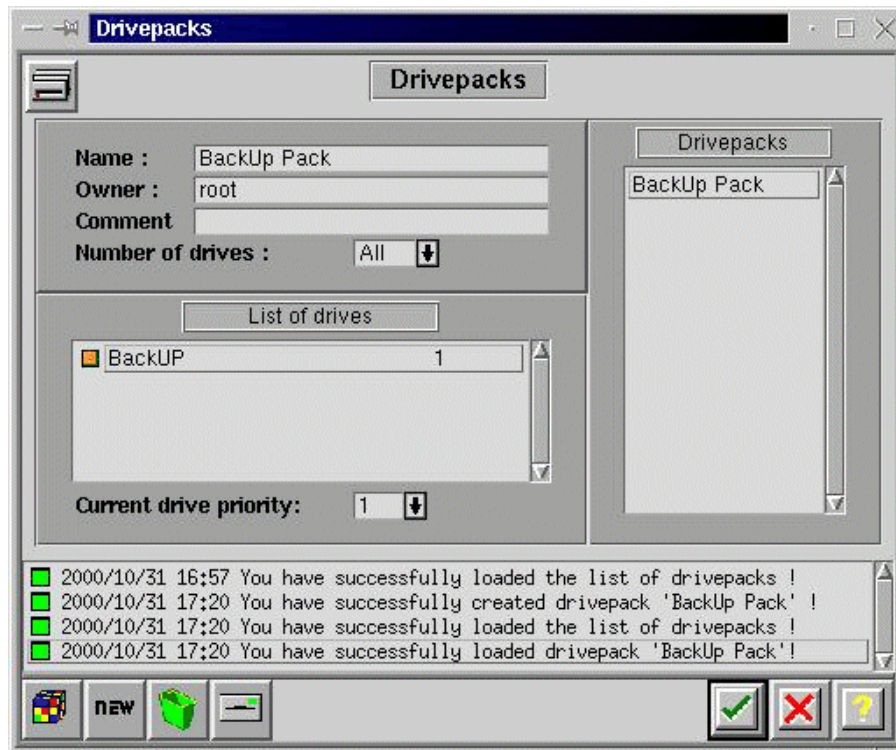


Figure 302. Updated drivepacks management window

Click the **OK** button again to return to the main dialog window.

The last step to be done before data can be saved is creating at least one savepack. You describe in savepacks which data should be saved. Different savepacks contain different sets of data to be saved.

To create savepack(s), click the Savepacks button or click **Tapes -> Savepacks** on the menu. You will see a window like Figure 303:

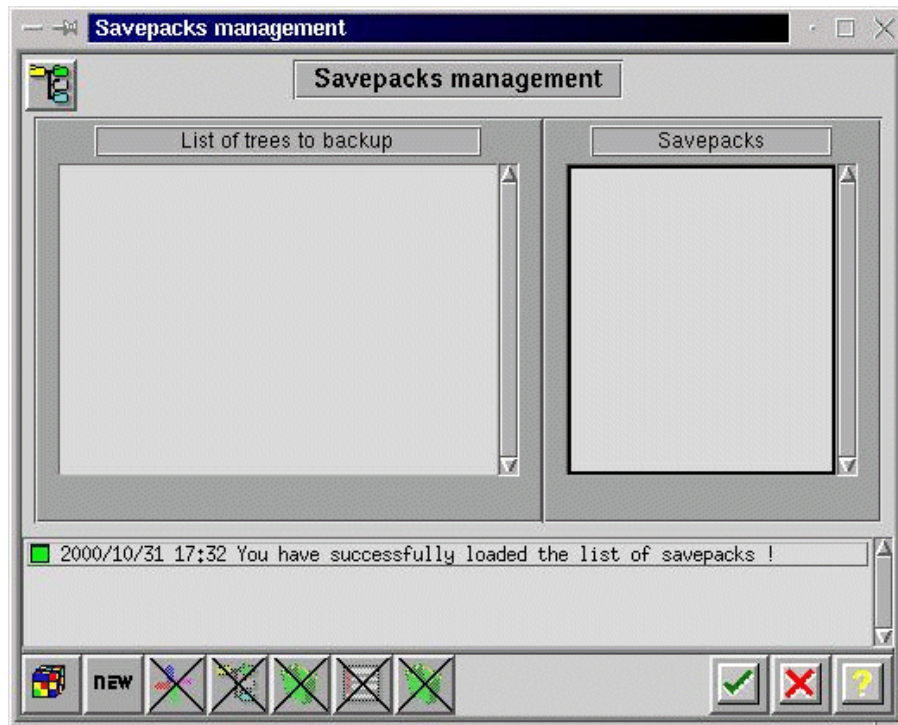


Figure 303. Savepacks management window

Click the **new** button to enter input mode. A window similar to Figure 304 appears.

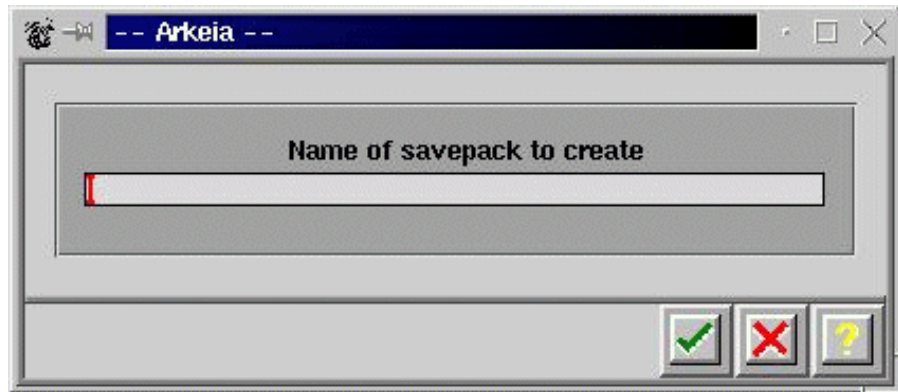


Figure 304. Window to create a new savepack



Enter the name of the new savepack and click the **OK** button to return to the updated savepacks management window (see the list of savepacks on the right side of the window). A window like in Figure 305 appears:

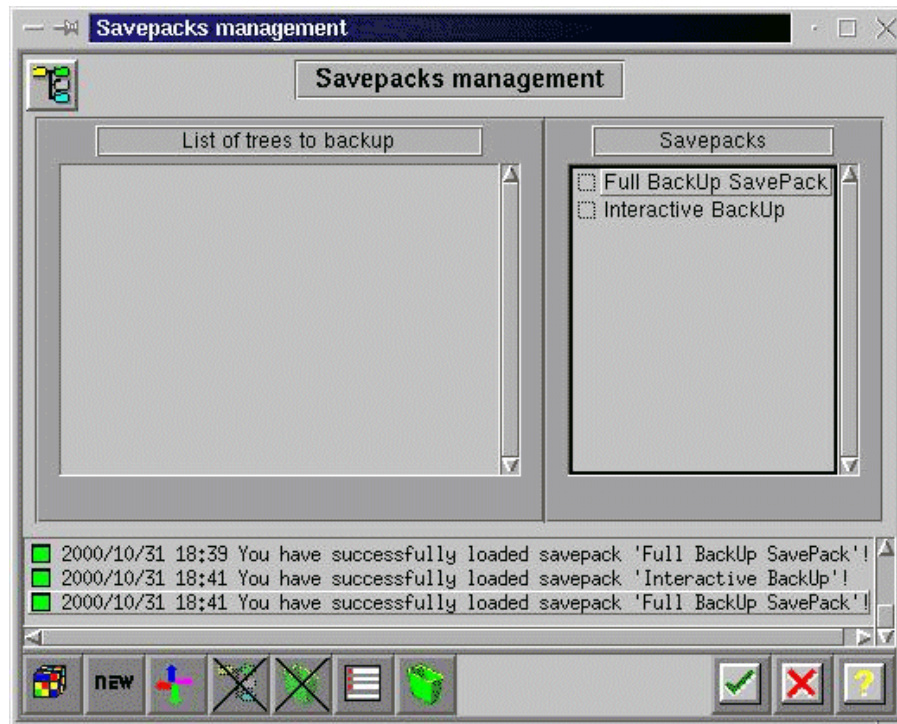


Figure 305. Updated savepacks management window

Now, you select the data that should be saved in every created savepack. Move the cursor over the name of the savepack you want to select the data for and click the left mouse button. You can see the selected savepack.

Now, move the cursor over the list of trees to back up (left listbox of this window), click the right mouse button and select **Navigator** in the upcoming pull-down menu. You will see a window similar to Figure 306.

To navigate through the directory tree of a system shown in this window, move the cursor over the system you want to select and double-click the left mouse button. A window similar to Figure 307 appears.

Double-clicking the left mouse button over a directory symbol opens this directory and shows the content of this directory.

Clicking once with the left mouse button in the checkbox to the left of a directory name or file name toggles the select/unselect status of this item. All selected items will be inserted in the list of trees to back up for the selected savepack. If you select a directory, the checkbox changes the color totally. If you select only a selection of the items in a directory, the checkbox for this directory changes color only in the right half of the checkbox.

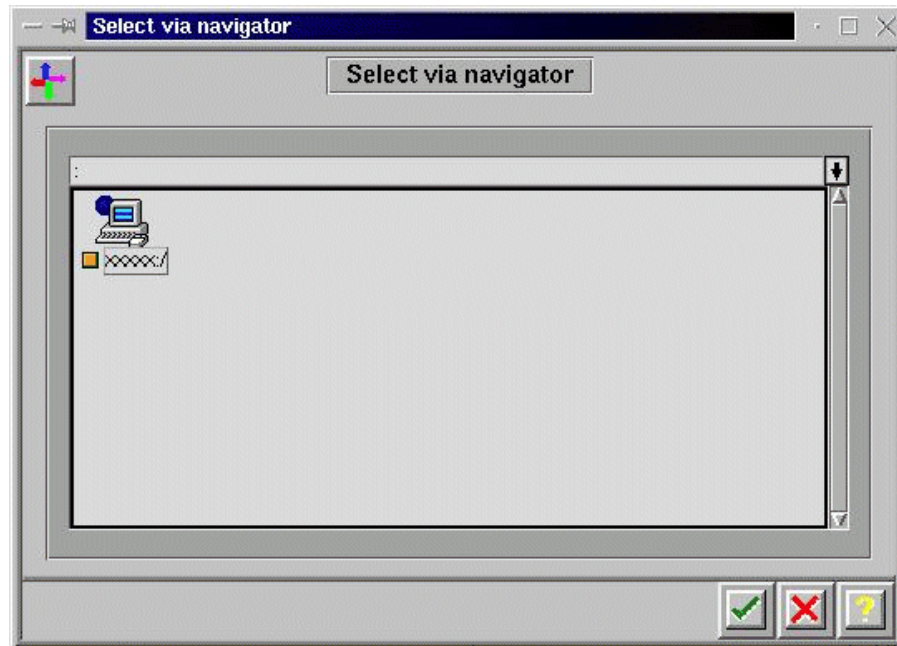


Figure 306. Navigator window



Figure 307. Updated navigator window

To return to the savepacks management window, click the **OK** button. You will see a window similar to Figure 308.

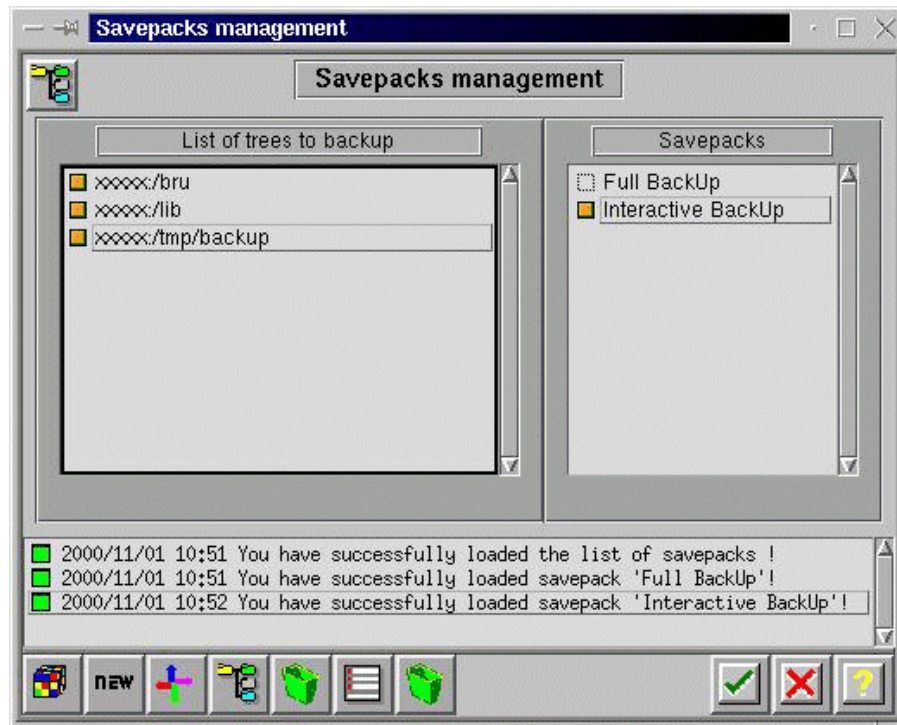


Figure 308. Updated savepacks management window

The basic configuration steps are now done.

Read the *Administrator's Manual* to get more information about the advanced possibilities of Arkeia.

### 15.3.3 Interactive backup

To start an interactive backup, click the interactive backup button or click **Backup>Interactive Backup** on the menu. A dialog like Figure 309 appears.

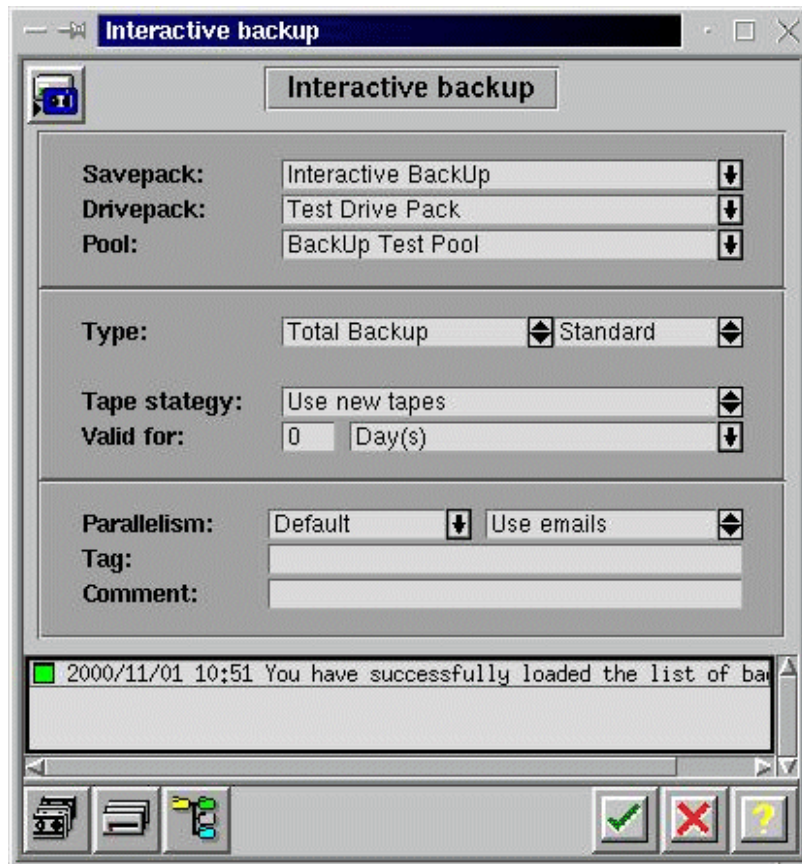


Figure 309. Interactive bckup start window

In the comboboxes Savepack, Drivepack and Pool fields, choose which data sets should be backed up on which tapes and on which tape drives.

In the **Type** box, choose between **Total Backup** and **Incremental Backup** and between **Standard** and **Continous**.

In the Tape Strategy field, choose between **Use new tapes** and **Complete existing tapes**.

In the **Valid for** field, decide how long the tape(s) for this backup should be valid.

Click the **OK** button to proceed. A window as in Figure 310 appears.

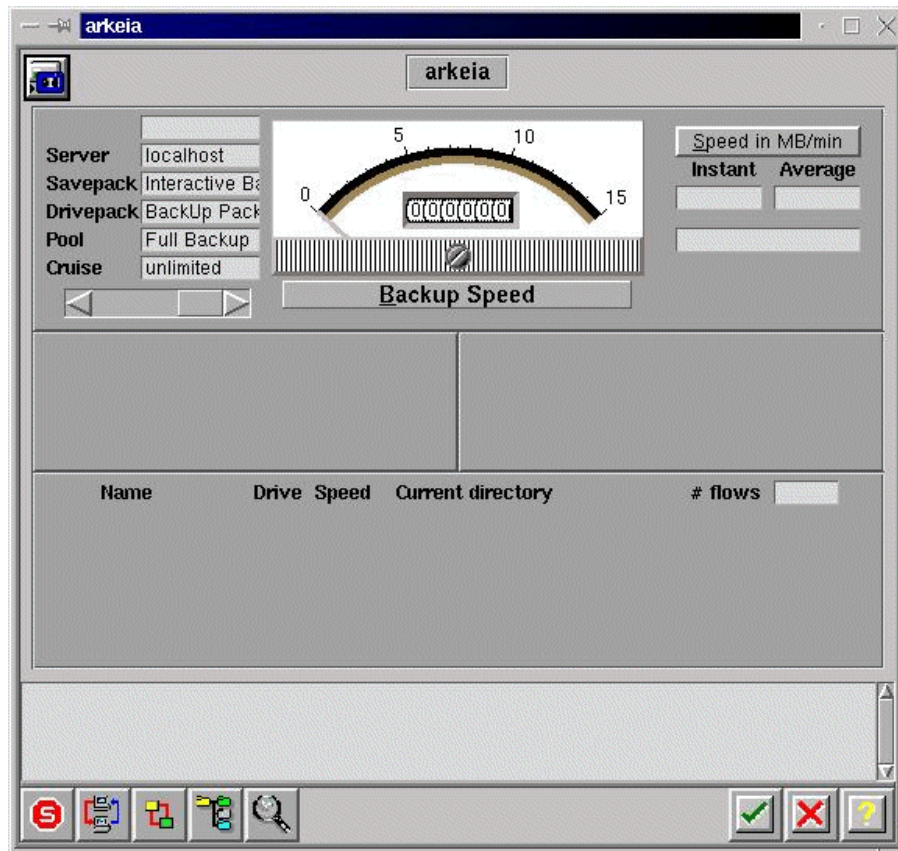


Figure 310. Arkeia's main window during backup

As the backup process proceeds, the content of this window will change. Most of the time, you will see a window like Figure 311.

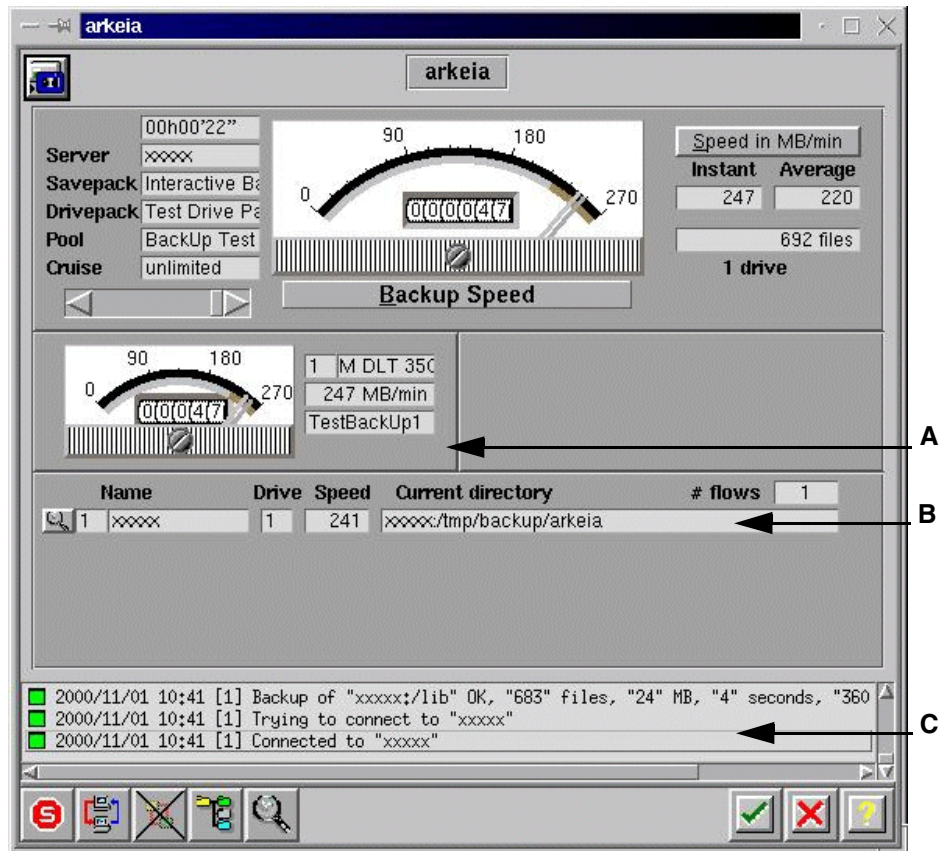


Figure 311. Main window during backup in progress

There are three areas in the window, marked **A**, **B** and **C** in Figure 311, which may require your attention:

In the area pointer **A** points to, you may sometimes see a push button labeled **OK**. Click this button when you have done the action, which was requested in the scroll list area **C**. In the line pointed to by **B**, you see the name of the file that actually is backed up.

You can leave this window by clicking the **OK** button. The backup process continues in the background.

If you want to connect again to this process or - as Arkeia calls it - job, go to Arkeia's main dialog window as shown in Figure 293. In this window you will see a box labeled either "No job running" or "List of jobs". If you see the text "List of jobs" and one or more lines under this box, move the cursor over the

line with the job you want to connect to and press the right mouse button. A pull-down menu as shown in Figure 312 appears.

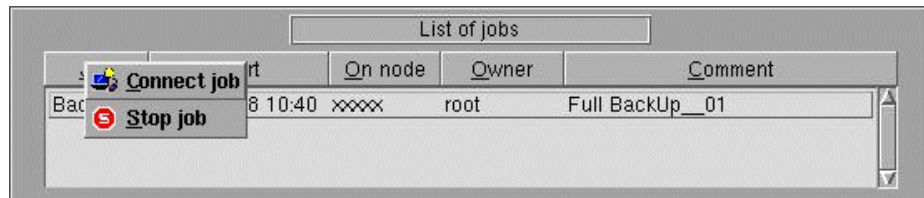


Figure 312. Connect job pull-down menu

Move the cursor over the line with the action you will perform and click the left mouse button. The requested action will be performed.

If you chose **Stop job**, you are asked in a new dialog whether you really want to stop this job.

If you select **Connect job**, you will see a window similar to Figure 311 again.

#### 15.3.4 Periodic Backup

To configure your scheme for periodic backups, press the periodic backup button or go to **Utilities>Periodic Backup** on the menu. You will see a window similar to Figure 313.



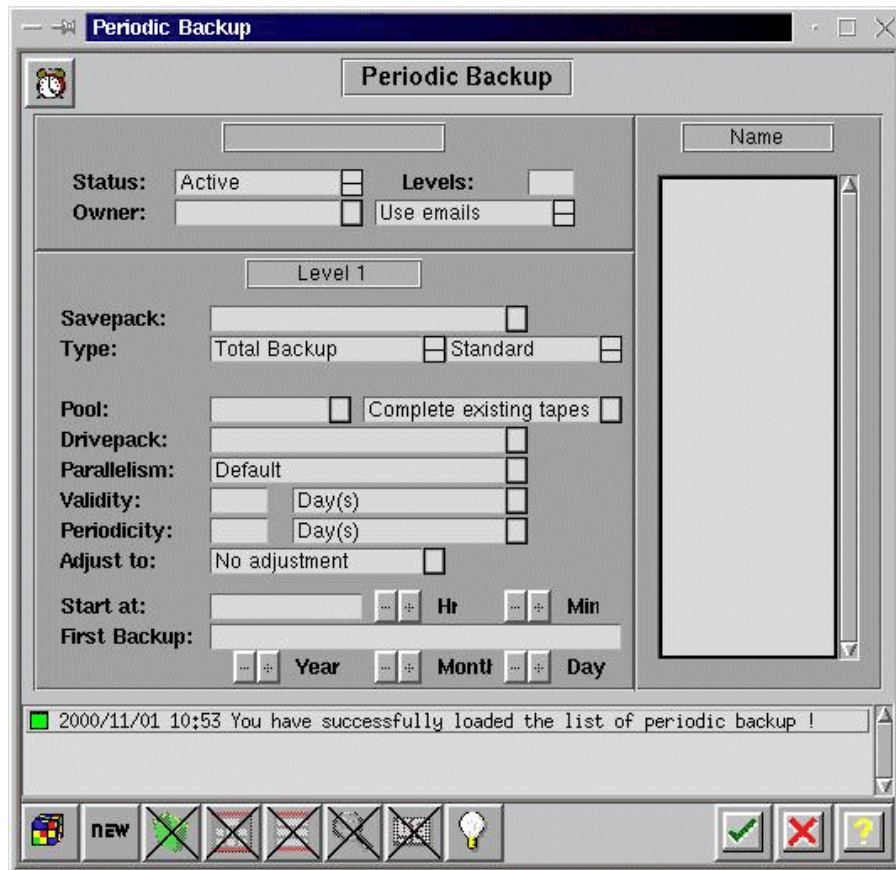


Figure 313. Periodic Backup window

To create a new entry for periodic backup, click the **new** button. You can now fill in the fields with the appropriate information. For more details, please consult the *Administrator's Manual*.

### 15.3.5 Restoration

To start restoration of data, click the restoration button or click **Restoration -> Restoration** on the menu. You will see a window like Figure 314.

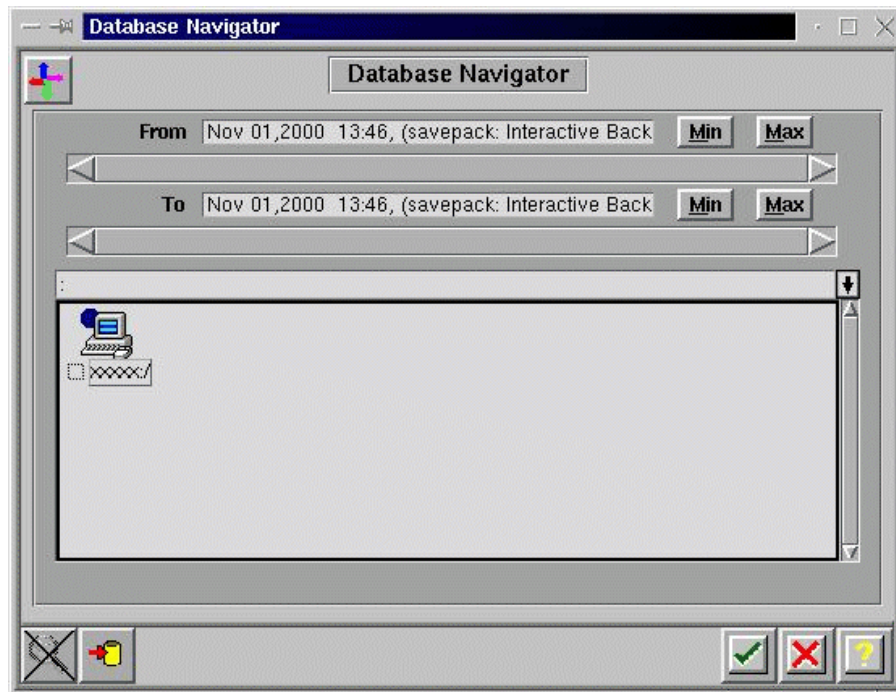


Figure 314. Restoration start dialog

Clicking with the left mouse button over the checkbox beside an item toggles the status of item between selected/not selected. By double-clicking over a symbol for a complete system or a directory, you can navigate through the tree of information, that this backup contains. If you are ready with your selection, click the **OK** button and a window like Figure 315 appears, containing a list of the files or directories that will be restored.

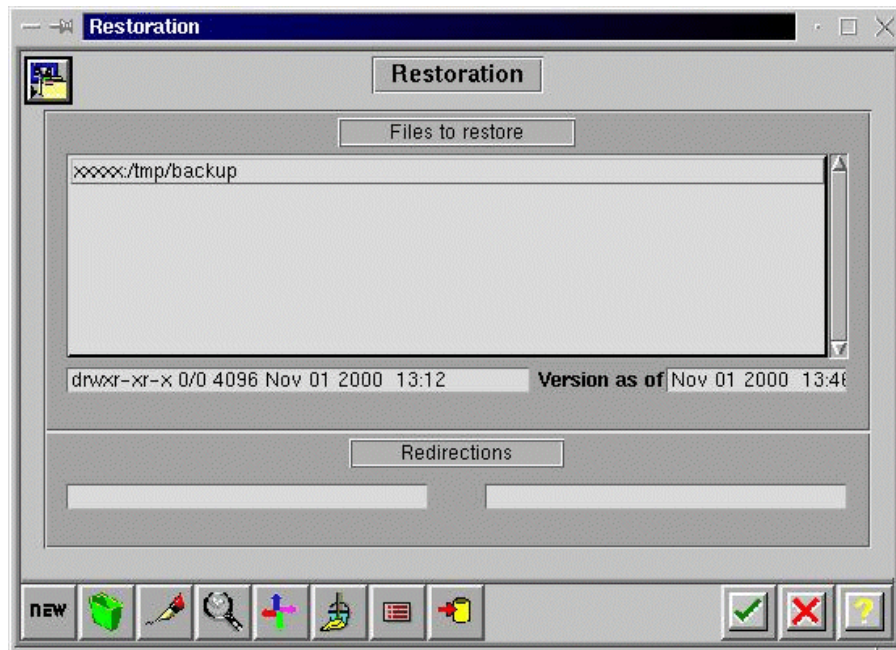


Figure 315. List of directories/files to store

Click the **OK** button in this window opens a new window, shown in Figure 316.



Figure 316. List of tapes used for restoration

You will see a list of the tape(s) that will be used during restoration. Click the **OK** button to proceed.

If the correct tape is already loaded to start the restoration with, you will see a window like Figure 317.

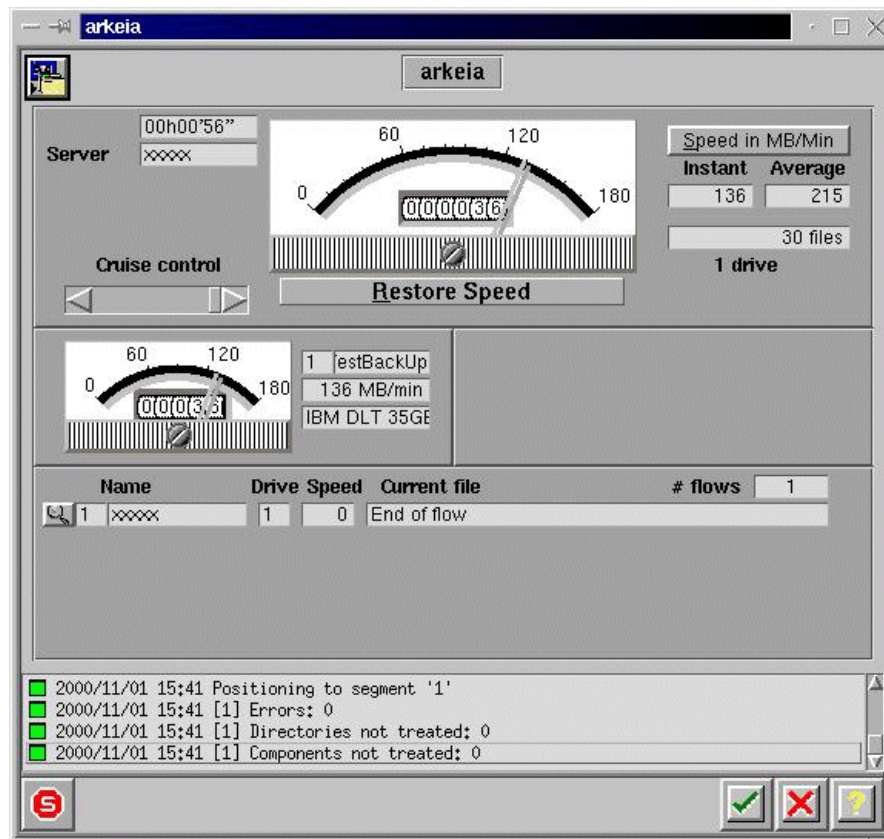


Figure 317. Restoration's main window

If the tape to start with must be mounted, a window like Figure 318 appears.

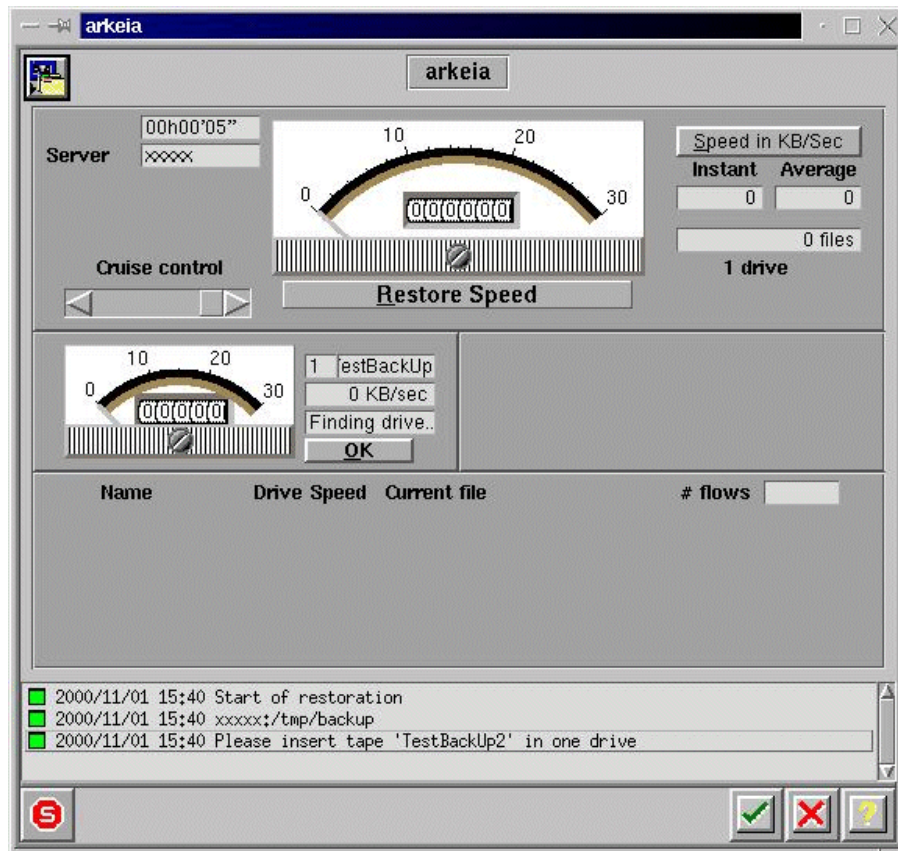


Figure 318. Window during restoration if manual intervention is required

Perform the action required and click **OK** to proceed. The appearance of the window changes. It is now like Figure 317.

### 15.3.6 Advanced features of Arkeia

For the advanced features of Arkeia, for example how to recycle or label tapes, please read the *Administrator's Manual*.

For more information, consult Arkeia's Web site at:

<http://www.arkeia.com>



---

## Appendix A. RAID levels

This appendix has been included for the convenience of our readers who are unfamiliar with the disk subsystem technology known as RAID. We anticipate that this will be a small percentage of our readership, because RAID is an important technology that most people implementing business-critical IT systems probably know about. RAID is mentioned in many places throughout this book and a basic appreciation of its features and benefits will help you to understand why.

Even those who know about RAID already will be interested to hear about the new RAID-5E level supported by the latest IBM ServeRAID adapter.

---

### A.1 What is RAID?

Although very commonly implemented using SCSI disks, RAID is independent of the specific disk technology being used. IBM Netfinity servers have RAID controllers that support SCSI, Fibre Channel, and SSA disk subsystems. In addition, Windows NT supports its own software-based RAID, although this is not often used, since much of the performance gained from having a dedicated hardware RAID controller is lost.

A typical RAID disk subsystem will have between two and six physical disks that are accessed by the processor by way of a specialized RAID controller adapter. The controller makes the array appear as a single large virtual disk to the processor. Because this disk has six completely independent head mechanisms for accessing data (in the case of a six-drive array), the potential for improved performance is immediately apparent. In the optimal situation, all six heads could be providing data to the system without the need for the time-consuming head-seeks to different areas of the disk that would be necessary were a single physical disk being used.

However, the primary intent of a RAID implementation is to prevent the system served by the array from being affected by critical hard disk failures. Several different implementations of RAID have been defined and are referred to as levels. Each level has different characteristics and these levels allow a choice to be made to best meet the cost, security, and performance desired. The three most common implementations are levels 0, 1, and 5. These are the levels available with all of IBM's disk subsystems supported by Netfinity servers, namely SCSI, SSA, and Fibre Channel. The Netfinity ServeRAID-3HB Ultra2 SCSI adapter introduces a new enhanced RAID-5 described in A.1.5, "RAID-5 enhanced" on page 337.

### A.1.1 RAID-0

RAID-0, sometimes referred to as disk striping, is not really a RAID solution since there is no redundancy in the array at all. The disk controller merely stripes the data across the array so that a performance gain is achieved. This is illustrated in Figure 319:

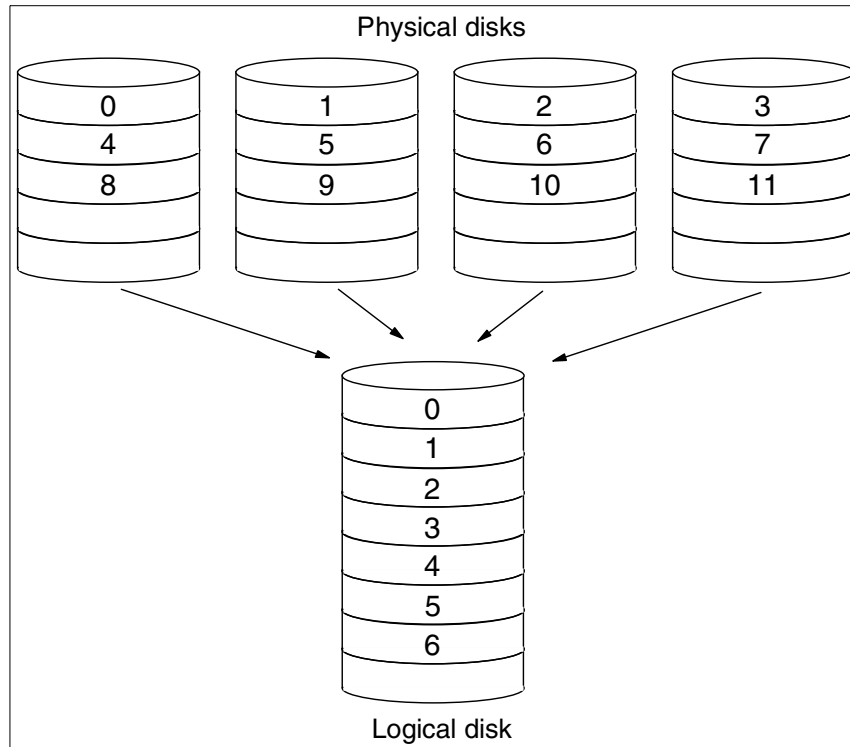


Figure 319. RAID-0 implementation

It is common for a striped disk array to map data in blocks with a stripe size that is an integer multiple of real drive track capacity. For example, the IBM ServeRAID adapters allow stripe sizes of 8 KB, 16 KB, 32 KB or 64 KB, selectable during initialization of the array. Applications get better performance if their data I/O size matches the stripe size of the array, so it is recommended that you take this into consideration when defining your RAID sets.

#### Advantages:

- Performance improvement in many cases.
- All disk space available for data.



**Disadvantages:**

- No redundancy.

**A.1.2 RAID-1 and RAID-1E**

RAID-1, or disk mirroring, offers true redundancy. Each stripe is duplicated, or mirrored, on another disk in the array. In its simplest form, there are two disks where the second is a simple copy of the first. If the first disk fails then the second can be used without any loss of data. Some performance enhancement is achieved by reading data from both drives. Certain operating systems, including Windows NT, provide direct support for disk mirroring. There is a performance overhead, however, as the processor has to issue duplicate write commands. Hardware solutions where the controller handles the duplicate writes are preferred.

When more than two disks are available, the duplication scheme can be a little more complex to allow striping with disk mirroring, also known as Enhanced RAID-1. An example is shown in Figure 320:

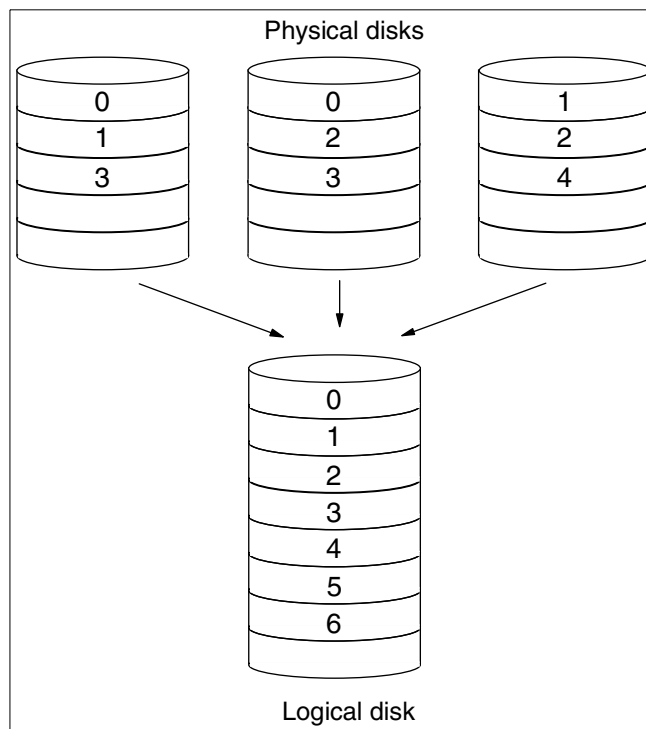


Figure 320. RAID-1E implementation

As you can see, any one disk can be removed from the array without loss of information because each data stripe exists on two physical disks. The controller detects a failed disk and redirects requests for data from the failed drive to the drive containing the copy of the data. When a drive has failed, the replacement drive can be rebuilt using the data from the remaining drives in the array.

When a disk fails, there is only one copy of the data that was on the failed disk available to the system. The system has lost its redundancy, and if another disk fails, data loss is the result. To avoid this, failed disks should be replaced as soon as possible. The controller then rebuilds the data that was on the failed disk from the remaining drives and writes it to the new disk, restoring the redundancy.

To avoid having to manually replace a failed disk, the IBM Netfinity ServeRAID controllers implement *hot spare* disks. A hot spare disk is held idle until a failure occurs, at which point the controller immediately starts to rebuild the lost data onto the hot spare, minimizing the time when redundancy is lost. The controller continues to provide data to the system while the rebuild takes place.

When you replace the failed drive, its replacement becomes the array's new hot spare.

**Advantages:**

- Performance improvement in many cases.
- Redundancy. A drive can fail without loss of data.

**Disadvantages:**

- Cost. The logical disk has only half the capacity of the physical disks.

### A.1.3 RAID-10

As we have seen, RAID-1 offers the potential for performance improvement as well as redundancy. RAID-10 is a variant of RAID-1 that effectively creates a mirror copy of a RAID-0 array.

In large disk subsystems that require, for example, two external storage enclosures, it would be beneficial to ensure that mirrored data exists in both units. This would allow an entire unit, including its power supply or connecting cables, to fail without interrupting operation. RAID-10 does just this by allowing one RAID-0 array to be contained in one of the enclosures and its mirror copy in the other. A diagram of a RAID-10 configuration is shown below:

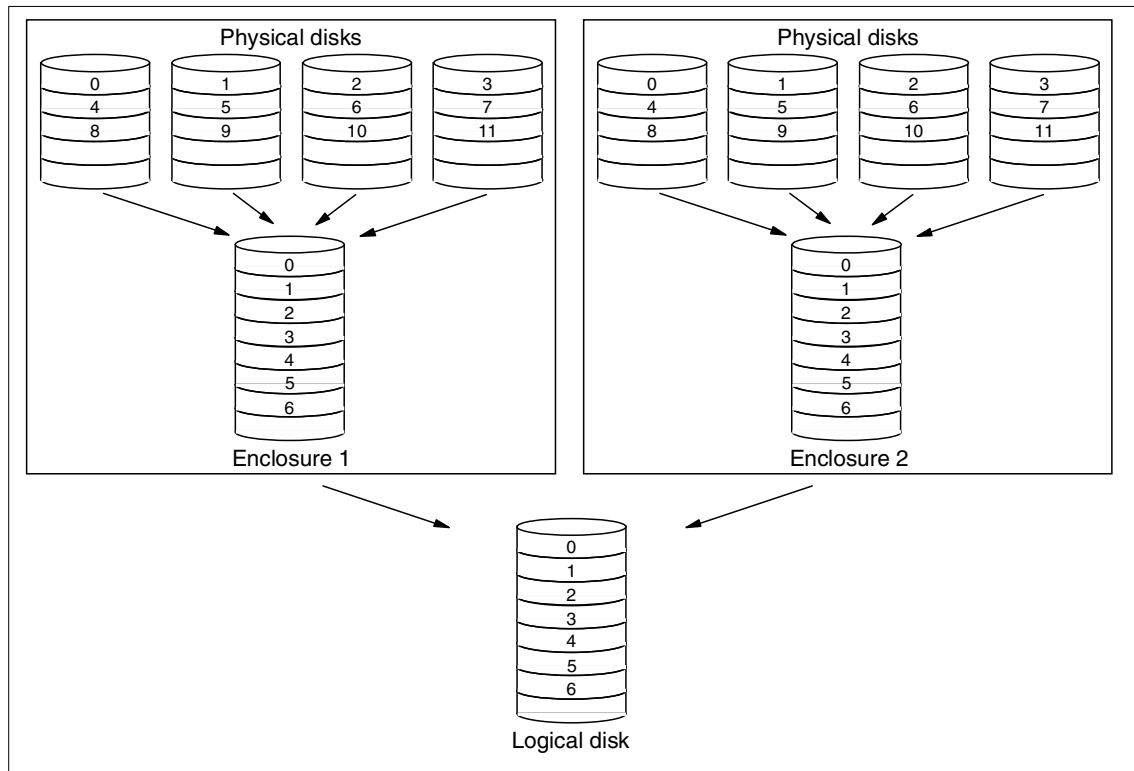


Figure 321. RAID-10 configuration

RAID-10 configurations are supported by the IBM Netfinity Fibre Channel RAID Controller Unit.

**Advantages:**

- Performance improvement in many cases.
- Redundancy. A drive can fail without loss of data.
- Provides fault tolerance for disk enclosures.

**Disadvantages:**

- Cost. The logical disk has only half the capacity of the physical disks.
- Slightly less flexible than RAID-1E (requires an even number of disks).

**A.1.4 RAID-5**

RAID-5 is one of the most capable and efficient ways of building redundancy into the disk subsystem. The way redundancy is implemented, capacity loss

is equal to one of the drives in the array and data striping provides the read performance gains from RAID-0 and RAID-1. The principles behind RAID-5 are very simple and are closely related to the parity methods sometimes used for computer memory subsystems. In memory, the parity bit is formed by evaluating the number of 1 bits in a single byte. For RAID-5, if we take the example of a four-drive array, three stripes of data are written to three of the drives and the bit-by-bit parity of the three stripes is written to the fourth drive.

As an example, we can look at the first byte of each stripe and see what this means for the parity stripe. Let us assume that the first byte of stripes 1, 2, and 3 are the letters A, B, and G respectively. The binary code for these characters is 01000001, 01000010 and 01000111 respectively.

We can now calculate the first byte of the parity block. Using the convention that an odd number of 1s in the data generates a 1 in the parity, the first parity byte is 01000100 (see Table 24). This is called Even Parity because there is always an even number of 1s if we look at the data and the parity together. Odd Parity could have been chosen; the choice is of no importance as long as it is consistent.

Table 24. Generation of parity data for RAID-5

Disk 1 "A"	Disk 2 "B"	Disk 3 "G"	Disk 4 Parity
0	0	0	0
1	1	1	1
0	0	0	0
0	0	0	0
0	0	0	0
0	0	1	1
0	1	1	0
1	0	1	0

Calculating the parity for the second byte is performed using the same method, and so on. In this way, the entire parity stripe for the first three data stripes can be calculated and stored on the fourth disk.

The presence of parity information allows any disk to fail without loss of data.

In the above example, if drive 2 fails (with B as its first byte) there is enough information in the parity byte and the data on the remaining drives to reconstruct the missing data. The controller has to look at the data on the remaining drives and calculate what drive 2's data must have been to maintain even parity. Because of this, a RAID-5 array with a failed drive can continue to provide the system with all the data from the failed drive.

Performance will suffer, of course, because the controller has to look at the data from all drives when a request is made to the failed one. However, that is better than losing the system completely. A RAID-5 array with a failed drive is said to be critical, since the loss of another drive will cause lost data. For this reason, the use of hot spare drives in a RAID-5 array is as important as in RAID-1.

The simplest implementation would always store the parity on disk 4 (in fact, this is the case in RAID-4, which is hardly ever implemented for the reason about to be explained). Disk reads are then serviced in much the same way as a level 0 array with three disks. However, writing to a RAID-5 array would then suffer from a performance bottleneck. Each write requires that both real data and parity data are updated. Therefore, the single parity disk would have to be written to every time any of the other disks were modified. To avoid this, the parity data is also striped, as shown in Figure 322, spreading the load across the entire array.

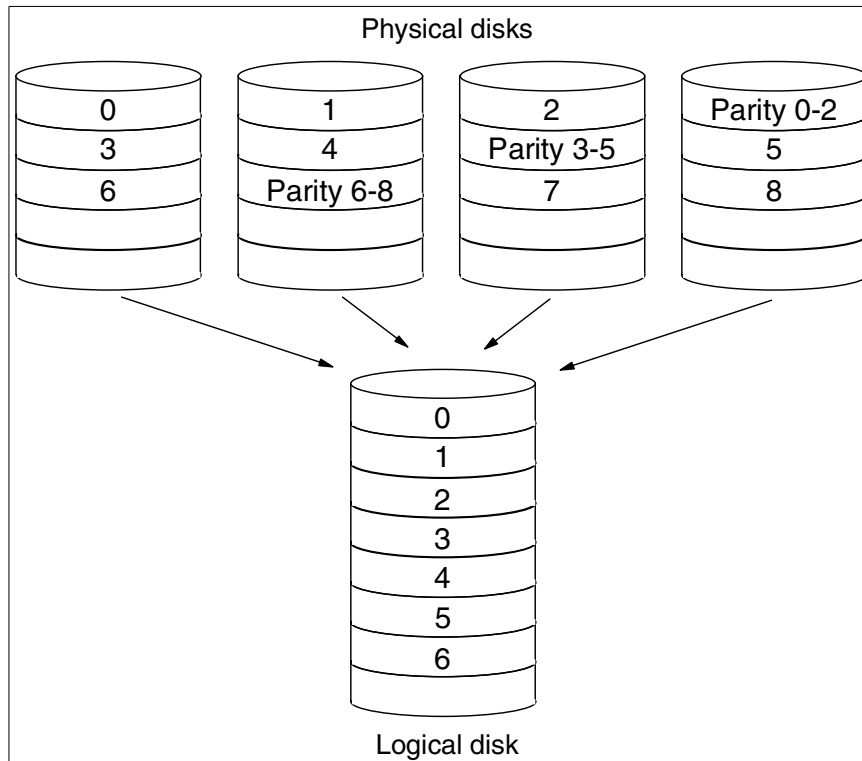


Figure 322. RAID-5 implementation

The consequence of having to update the parity information means that for every stripe written to the virtual disk, the controller has to read the old data from the stripe being updated and the associated parity stripe. Then the necessary changes to the parity stripe have to be calculated based on the old and the new data. All of this complexity is hidden from the processor, but the effect on the system is that writes are much slower than reads. This can be offset to a greater or lesser extent by the use of a cache on the RAID controller. The IBM ServeRAID controllers have cache as standard, which is used to hold the new data while the calculations are being performed. Meanwhile, the processor can continue as though the write has taken place. Battery backup options for the cache, available for some controllers, mean that data loss is kept to a minimum even if the controller fails with data still in the cache.

**Advantages:**

- Performance improvement in many cases.
- Redundancy. A drive can fail without loss of data.

- Storage overhead is equal to the size of only one drive.

**Disadvantages:**

- Overhead associated with writes can be detrimental to performance in applications where the write/read ratio is high. A controller cache can alleviate this.

**A.1.5 RAID-5 enhanced**

RAID-5 Enhanced (RAID-5E) puts hot spare drives to work to improve reliability and performance. A hot spare is normally inactive during array operation and is not used until a drive fails. By utilizing unallocated space on the drives in the array, a virtual distributed hot spare (DHS) can be created to improve reliability and performance. Figure 323 shows normal operation of a RAID-5E array. The data areas of the individual disks shown contain the application data and stripe parity data as for a normal RAID-5 array:

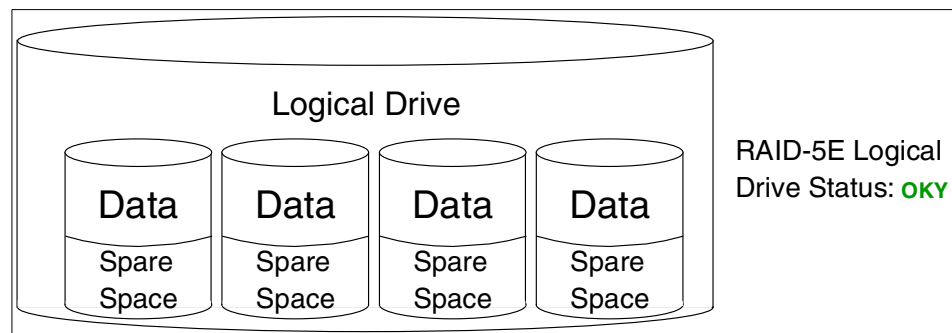


Figure 323. RAID-5E array: normal operation

In the event of a physical drive failing, its status will change to Defunct Disk Drive (DDD) and the ServeRAID adapter will start rearranging the data the disk contained into the spare space on the other drives in the array, provided there is enough space, of course.

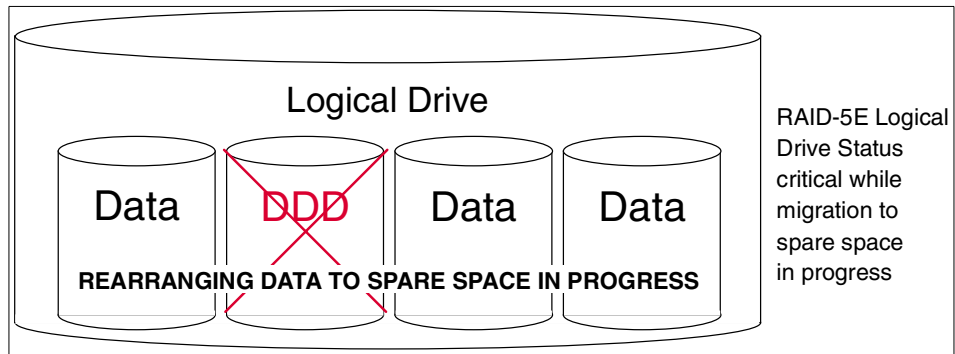


Figure 324. RAID-5E array: single physical disk failure

During the migration of data, the logical drive will be in a critical, nonredundant state. As soon as all the data is rearranged, the logical drive will be marked OKY (Okay) and have full redundancy again. This is illustrated in Figure 325.

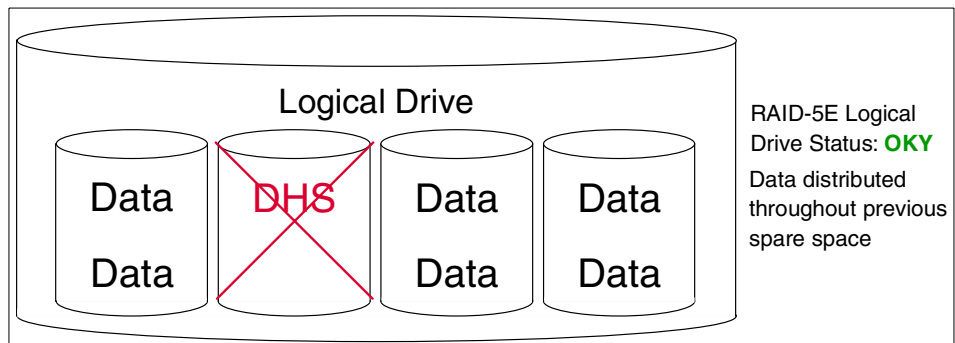


Figure 325. RAID-5E array: data distributed throughout previous spare space

In the event of a second physical disk failure before the previously failed disk has been replaced, illustrated in Figure 326, normal RAID-5 procedures will be taken to provide service to the system through the checksum calculations described in A.1.4, “RAID-5” on page 333.



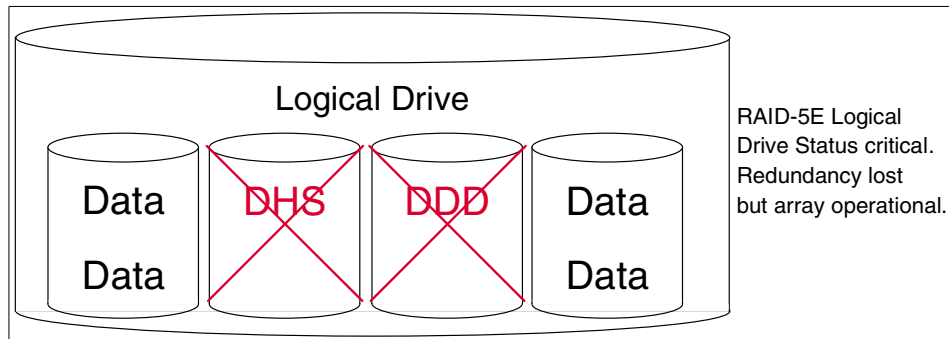


Figure 326. RAID-5E array: second physical disk failure

**Advantages (compared to RAID-5):**

- 15 - 20% performance improvement for smaller arrays with typical data transfer size.
- Protects data, even in the event of a two-drive failure.

**Disadvantages:**

- Migration time.

**Design characteristics:**

- One RAID-5E logical drive per array.
- Minimum of four physical drives in array configured for RAID-5E logical drive.

**A.1.6 Orthogonal RAID-5**

Orthogonal RAID-5 is an enhancement of RAID-5 in the sense that it is powered by more than one disk controller and hence improves both reliability and performance.

The performance of a disk subsystem depends on more than just the underlying performance of the disks. Multiple requests to one disk or across one adapter will typically take longer to satisfy than the same number of requests to multiple disks across multiple adapters.

In addition, the overall reliability of a standard RAID-5 system is dependent on the reliability of the one disk adapter to which all of the disks are connected. Orthogonal RAID-5 solves both of these concerns by grouping the disk arrays orthogonally to the disk adapters, SCSI buses, and power cables.

This would normally be implemented as a four-drive orthogonal RAID-5 array, where each disk would be connected to a different adapter and SCSI bus.

The result of this is that any one component of the disk subsystems, not just a disk drive, can fail with no loss of data and no interruption to system operation.

### **A.1.7 Performance**

With different parameters affecting your RAID solution it is virtually impossible to find the perfect combination without measuring live throughput. Increasing redundancy also increases price and possibly lowers performance due to added overhead, which could be solved with more or faster controllers, again increasing the price.

As you can see in Figure 327 on page 341, speed is a significant issue when deciding on RAID level. The numbers shown in this figure and in Figure 328 on page 342 are based on benchmark testing performed by the xSeries and Netfinity server development team. Specific systems may not show precisely the same performance ratios but the figures are representative of typical performance data.

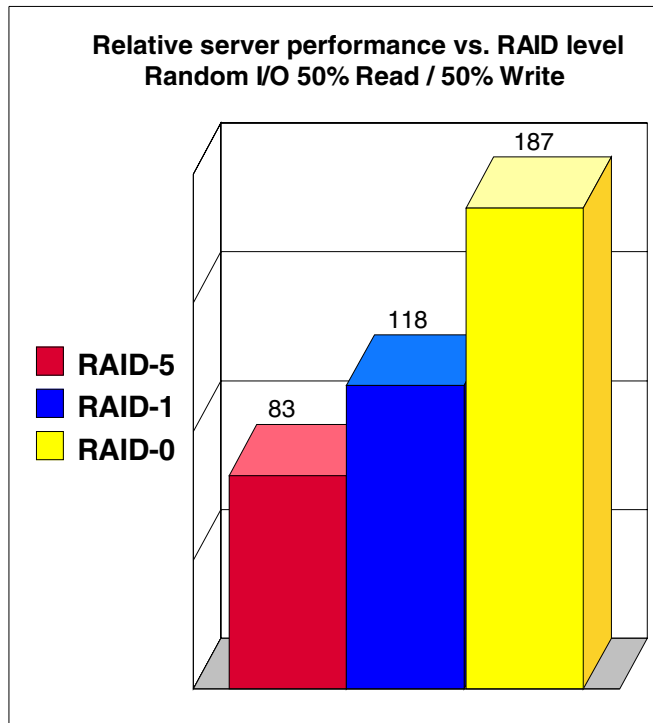


Figure 327. Relative server performance versus RAID strategy

It is important to point out that the speed difference in Figure 327 is mainly due to the same number of drives being used for all tests. Generally, the more drives you use in your array, the faster it gets, but it also requires your RAID controller to be able to attach more drives when using RAID-1 or RAID-5 to get optimal performance.

Using the same number of drives:

- RAID-0 gives up to 50% more throughput than RAID-1.
- RAID-1 gives up to 50% more throughput than RAID-5.

The above test was done using a worst-case scenario with 50% reads and 50% writes. A high write/read ratio adversely affects the performance of RAID-1 and RAID-5 arrays, so throughput improves with a higher percentage of reads, which is generally more common in a real-world environment.

- While increasing the number of drives boosts performance, it also increases the price. Figure 328 on page 342 shows what happens with I/O throughput when we add drives to a RAID-0 array.

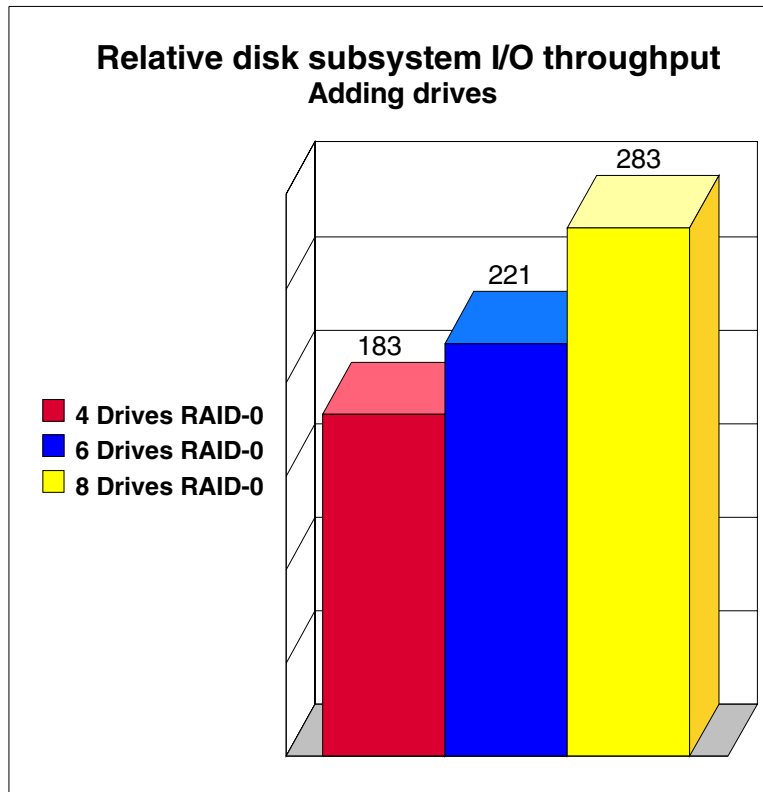


Figure 328. Adding drives to an array

Server throughput improves up to 50% when the number of drives is doubled for a RAID-0 and similar gains are shown for RAID-1 and RAID-5.

### A.1.8 Recommendations

Before configuring your array you have to decide on a stripe size for the array. When configuring for maximum performance, Table 25 shows some rules of thumb:

Table 25. Recommended stripe configurations for ServeRAID adapters

Environment	Stripe size	Read-ahead
Groupware (Lotus Notes, Exchange)	16 KB	ON
Database Server (Oracle, SQL Server, DB/2)	16 KB	OFF
File Server (Windows NT 4.0, NetWare 4.1x)	16 KB	ON

Environment	Stripe size	Read-ahead
Web Server	8 KB	OFF
Other	8 KB	ON

### A.1.9 Summary

RAID is an excellent and proven technology for protecting your data against the possibility of hard disk failure. IBM has a range of RAID controllers that bring the benefits of the technology to xSeries and Netfinity servers. As Intel-based servers become more and more critical to customers' businesses, they are demanding the reliability provided by RAID.

Here is a quick summary of the different RAID levels we have covered in this appendix:

**RAID-0:** Block interleave data striping without parity

- Best performance of all RAID levels
- Drive seek times and latencies effectively reduced by parallel operation
- Significantly outperforms single large disk

**RAID-1:** Disk mirroring

- Fast and reliable but requires 100% disk space overhead
- Two copies of data maintained
- No performance degradation with a single disk failure
- Writes are slower than a single disk, reads are quicker

**RAID-1E:** Data stripe mirroring

- All the benefits of RAID-1
- Provides mirroring with an odd number of drives

**RAID-10:** Mirrored RAID-0 arrays

- All the benefits of RAID-1
- Can provide fault tolerance for entire storage enclosures

**RAID-5:** Block interleave data striping with distributed parity

- Best for random transactions
- Poor for large sequential reads if request is larger than block size
- Block size is the key to performance; must be larger than typical request size

- Performance degrades in recovery mode, that is, when a single drive has failed

**RAID-5E:** RAID-5 with distributed hot spare

- All the benefits of RAID-5
- 15 - 20% performance improvement for smaller arrays
- Protects data, even in the event of a two-drive failure

**Orthogonal RAID-5:** RAID-5 with multiple orthogonal disk adapters

- All the benefits of RAID-5
- Improved performance (due to load being spread across disk adapters)
- Improved reliability due to redundancy of disk adapters and disks

Table 26 gives you a summary of RAID performance characteristics:

*Table 26. Summary of RAID performance characteristics*

RAID level	Capacity	Large transfers	I/O rate	Data availability
RAID-0	Excellent	Very Good	Very Good	Poor <sup>1</sup>
RAID-1/1E	Moderate	Good	Good	Good
RAID-10	Moderate	Good	Good	Very Good
RAID-5	Very Good	Very Good	Good	Good
RAID-5E	Very Good	Very Good	Good to Very Good	Very Good
Orthogonal RAID-5	Very Good	Very Good	Good	Very Good

<sup>1</sup> Availability = MTBF of one disk divided by the number of disks in the array

If you want to learn more about RAID, the RAID Advisory Board, of which IBM is an active member, exists to standardize terminology and provide information about RAID technology. Its Web site can be found at the following URL:

<http://www.raid-advisory.com/>

---

## Appendix B. Working video modes for IBM Netfinity servers

In this Appendix you can find some working modes for Xfree86 servers for IBM Netfinity servers. All working graphics cards in IBM Netfinity servers use the XFree86 SVGA server. So before you do a video card probe, select the SVGA server. The following are the modes tested in our working environment:

- Netfinity 3000  
Modeline "1152x864/70Hz" 92 at 24bpp
- Netfinity 3500M10  
Modeline "1024x768/70Hz" 75 at 16bpp
- Netfinity 5000  
Modeline "800x600/72Hz" 50 at 16bpp
- Netfinity 5600  
Modeline "800x600/85Hz" 60.75 at 24bpp
- Netfinity 5500 M10  
Modeline "1024x768/70Hz" 75 at 8bpp
- Netfinity 5500 M20  
Use XFree86 VGA server
- Netfinity 7000 M10  
Use XFree86 VGA server
- Netfinity 8500R  
Modeline "1152x864/70Hz" 92 at 24bpp





---

## Appendix C. Sample smb.conf Samba configuration file

```
# This is the main Samba configuration file. You should read the
# smb.conf(5) manual page in order to understand the options listed
# here. Samba has a huge number of configurable options (perhaps
# too many!) most of which are not shown in this example
#
# Any line which starts with a ; (semi-colon) or a # (hash)
# is a comment and is ignored. In this example we will use a #
# for commentry and a ; for parts of the config file that you
# may wish to enable
#
# NOTE: Whenever you modify this file you should run the command "testparm"
# to check that you have not many any basic syntactic errors.
#
#===== Global Settings =====
[global]

# workgroup = NT-Domain-Name or Workgroup-Name
# workgroup = LINUXRLZ

# server string is the equivalent of the NT Description field
# server string = Samba Server on TurboLinux

# This option is important for security. It allows you to restrict
# connections to machines which are on your local network. The
# following example restricts access to two C class networks and
# the "loopback" interface. For more examples of the syntax see
# the smb.conf man page
; hosts allow = 192.168.1. 192.168.2. 127.

# If you want to automatically load your printer list rather
# than setting them up individually then you'll need this
# load printers = yes

# you may wish to override the location of the printcap file
; printcap name = /etc/printcap

# It should not be necessary to specify the print system type unless
# it is non-standard. Currently supported print systems include:
# bsd, sysv, plp, lprmng, aix, hpux, qnx
# printing = lprmng

# Uncomment this if you want a guest account, you must add this to /etc/passwd
```

```

# otherwise the user "nobody" is used
; guest account = pcguest

# this tells Samba to use a separate log file for each machine
# that connects
; log file = /var/log/samba.d/smb.%m

# Put a capping on the size of the log files (in Kb).
    max log size = 50

# Security mode. Most people will want user level security. See
# security_level.txt for details.
    security = user
# Use password server option only with security = server
; password server = <NT-Server-Name>

# Password Level allows matching of _n_ characters of the password for
# all combinations of upper and lower case.
; password level = 8
; username level = 8

# You may wish to use password encryption. Please read
# ENCRYPTION.txt, Win95.txt and WinNT.txt in the Samba documentation.
# Do not enable this option unless you have read those documents
    encrypt passwords = yes
    smb passwd file = /etc/samba.d/smbpasswd

# The following are needed to allow password changing from Windows to
# update the Linux sytsem password also.
# NOTE: Use these with 'encrypt passwords' and 'smb passwd file' above.
# NOTE2: You do NOT need these to allow workstations to change only
#         the encrypted SMB passwords. They allow the Unix password
#         to be kept in sync with the SMB password.
; unix password sync = Yes
; passwd program = /usr/bin/passwd %u
; passwd chat = *New*UNIX*password* %n\n *ReType*new*UNIX*password* %n\n
*passwd:*all*authentication*tokens*updated*successfully*

# Unix users can map to different SMB User names
; username map = /etc/samba.d/smbusers

# Using the following line enables you to customise your configuration
# on a per machine basis. The %m gets replaced with the netbios name
# of the machine that is connecting
; include = /etc/samba.d/smb.conf.%m

```

```

# Most people will find that this option gives better performance.
# See speed.txt and the manual pages for details
    socket options = TCP_NODELAY

# Configure Samba to use multiple interfaces
# If you have multiple network interfaces then you must list them
# here. See the man page for details.
;   interfaces = 192.168.12.2/24 192.168.13.2/24

# Configure remote browse list synchronisation here
# request announcement to, or browse list sync from:
# a specific host or from / to a whole subnet (see below)
;   remote browse sync = 192.168.3.25 192.168.5.255
# Cause this host to announce itself to local subnets here
;   remote announce = 192.168.1.255 192.168.2.44

# Browser Control Options:
# set local master to no if you don't want Samba to become a master
# browser on your network. Otherwise the normal election rules apply
;   local master = no

# OS Level determines the precedence of this server in master browser
# elections. The default value should be reasonable
;   os level = 33

# Domain Master specifies Samba to be the Domain Master Browser. This
# allows Samba to collate browse lists between subnets. Don't use this
# if you already have a Windows NT domain controller doing this job
;   domain master = yes

# Preferred Master causes Samba to force a local browser election on startup
# and gives it a slightly higher chance of winning the election
;   preferred master = yes

# Use only if you have an NT server on your network that has been
# configured at install time to be a primary domain controller.
;   domain controller = <NT-Domain-Controller-SMBName>

# Enable this if you want Samba to be a domain logon server for
# Windows95 workstations.
;   domain logons = yes

# if you enable domain logons then you may want a per-machine or
# per user logon script

```

```

# run a specific logon batch file per workstation (machine)
; logon script = %m.bat
# run a specific logon batch file per username
; logon script = %U.bat

# Where to store roving profiles (only for Win95 and WinNT)
# %L substitutes for this servers netbios name, %U is username
# You must uncomment the [Profiles] share below
; logon path = \\%L\Profiles\%U

# All NetBIOS names must be resolved to IP Addresses
# 'Name Resolve Order' allows the named resolution mechanism to be specified
# the default order is "host lmhosts wins bcast". "host" means use the unix
# system gethostbyname() function call that will use either /etc/hosts OR
# DNS or NIS depending on the settings of /etc/host.config, /etc/nsswitch.conf
# and the /etc/resolv.conf file. "host" therefore is system configuration
# dependant. This parameter is most often of use to prevent DNS lookups
# in order to resolve NetBIOS names to IP Addresses. Use with care!
# The example below excludes use of name resolution for machines that are NOT
# on the local network segment
# - OR - are not deliberately to be known via lmhosts or via WINS.
; name resolve order = wins lmhosts bcast

# Windows Internet Name Serving Support Section:
# WINS Support - Tells the NMBD component of Samba to enable it's WINS Server
; wins support = yes

# WINS Server - Tells the NMBD components of Samba to be a WINS Client
# Note: Samba can be either a WINS Server, or a WINS Client, but NOT both
; wins server = w.x.y.z

# WINS Proxy - Tells Samba to answer name resolution queries on
# behalf of a non WINS capable client, for this to work there must be
# at least one WINS Server on the network. The default is NO.
; wins proxy = yes

# DNS Proxy - tells Samba whether or not to try to resolve NetBIOS names
# via DNS nslookups. The built-in default for versions 1.9.17 is yes,
# this has been changed in version 1.9.18 to no.
dns proxy = no

# Case Preservation can be handy - system default is _no_
# NOTE: These can be set on a per share basis
; preserve case = no
; short preserve case = no

```

```

# Default case is normally upper case for all DOS files
; default case = lower
# Be very careful with case sensitivity - it can break things!
; case sensitive = no

#===== Share Definitions =====
[homes]
    comment = Home Directories
; this gives access to a 'Public' sub-directory in each user's home...
; (it is named 'public' as it is intended to be used by other sharing
; technologies (like NetWare, appletalk) too and may get disclosed due
; to weak protocols! -- hmm, are there less secure protocols than NFS? :)
    path = %H
    valid users = %S
%    only user = yes
    browseable = no
    writable = yes
    create mask = 0750

# Un-comment the following and create the netlogon directory for Domain Logons
; [netlogon]
;    comment = Samba Network Logon Service
;    path = /home/samba/netlogon
;    guest ok = yes
;    writable = no
;    share modes = no

# Un-comment the following to provide a specific roving profile share
# the default is to use the user's home directory
; [Profiles]
;    path = /home/samba/profiles
;    browseable = no
;    guest ok = yes

# NOTE: If you have a BSD-style print system there is no need to
# specifically define each individual printer
[printers]
    comment = All Printers
    path = /var/spool/samba
    browseable = no
# Set public = yes to allow user 'guest account' to print
    guest ok = no
    writable = no

```

```

printable = yes
create mask = 0700

# A publicly accessible directory, but read only, except for people in
# the "users" group
[public]
    comment = Public Stuff
    path = /home/public
    browseable = yes
    public = yes
    writable = yes
    printable = no
# access may be controlled by these options
; read list = user1, user2, @group
; valid users = user1, user3
    write list = @users

# Other examples.
#
# This one is useful for people to share files, BUT
# access to '/tmp' or '/var/tmp' should *not* be given lightly,
# as this can (still) pose a security threat!
# Better use a dedicate sub-directory to /(var/)tmp or something
# like a [public] share!
;[tmp]
;    comment = Temporary file space
;    path = /tmp
;    read only = no
;    public = yes

# A private printer, usable only by fred. Spool data will be placed in fred's
# home directory. Note that fred must have write access to the spool directory,
# wherever it is.
;[fredsprn]
;    comment = Fred's Printer
;    valid users = fred
;    path = /homes/fred
;    printer = freds_printer
;    public = no
;    writable = no
;    printable = yes

# A private directory, usable only by fred. Note that fred requires write
# access to the directory.
;[fredsdir]

```

```

; comment = Fred's Service
; path = /usr/somewhere/private
; valid users = fred
; public = no
; writable = yes
; printable = no

# a service which has a different directory for each machine that connects
# this allows you to tailor configurations to incoming machines. You could
# also use the %u option to tailor it by user name.
# The %m gets replaced with the machine name that is connecting.
;[pchome]
; comment = PC Directories
; path = /usr/pc/%m
; public = no
; writable = yes

# A publicly accessible directory, read/write to all users. Note that all files
# created in the directory by users will be owned by the default user, so
# any user with access can delete any other user's files. Obviously this
# directory must be writable by the default user. Another user could of course
# be specified, in which case all files would be owned by that user instead.
;[public]
; path = /usr/somewhere/else/public
; public = yes
; only guest = yes
; writable = yes
; printable = no

# The following two entries demonstrate how to share a directory so that two
# users can place files there that will be owned by the specific users. In this
# setup, the directory should be writable by both users and should have the
# sticky bit set on it to prevent abuse. Obviously this could be extended to
# as many users as required.
;[myshare]
; comment = Mary's and Fred's stuff
; path = /usr/somewhere/shared
; valid users = mary fred
; public = no
; writable = yes
; printable = no
; create mask = 0765

```





---

## Appendix D. Special notices

This publication is intended to help you install and configure TurboLinux on your IBM xSeries or Netfinity servers. The information in this publication is not intended as the specification of any programming interfaces that are provided by TurboLinux, IBM xSeries and Netfinity. See the PUBLICATIONS section of the IBM Programming Announcement for IBM xSeries or Netfinity for more information about what publications are considered to be product documentation.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter 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 the IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers

attempting to adapt these techniques to their own environments do so at their own risk.

Any pointers in this publication to external Web sites are provided for convenience only and do not in any manner serve as an endorsement of these Web sites.

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

IBM ®	Redbooks
IBM	Redbooks Logo 
Netfinity	Lotus
NetVisa	Lotus Notes
PS/2	Notes
ServeRAID	xSeries
@server	Domino

The following terms are trademarks of other companies:

Tivoli, Manage. Anything. Anywhere., The Power To Manage., Anything. Anywhere., TME, NetView, Cross-Site, Tivoli Ready, Tivoli Certified, Planet Tivoli, and Tivoli Enterprise are trademarks or registered trademarks of Tivoli Systems Inc., an IBM company, in the United States, other countries, or both. In Denmark, Tivoli is a trademark licensed from Kjøbenhavns Sommer - Tivoli A/S.

Red Hat, RPM, and all Red Hat-based trademarks and logos are trademarks or registered trademarks of Red Hat, Inc. in the United States and other countries.

TurboLinux and its log are trademarks of TurboLinux, Inc. Linux is a trademark of Linus Torvalds.

C-bus is a trademark of Corollary, Inc. in the United States and/or other countries.

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

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

PC Direct is a trademark of Ziff Communications Company in the United

States and/or other countries and is used by IBM Corporation under license.

ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States and/or other countries.

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

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.



---

## Appendix E. Related publications

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

---

### E.1 IBM Redbooks

For information on ordering these ITSO publications see “How to get IBM Redbooks” on page 363.

- *Linux for WebSphere and DB2 Servers*, SG24-5850
- *Red Hat Linux Integration Guide for IBM @server xSeries and Netfinity*, SG24-5853
- *Caldera OpenLinux Integration Guide for IBM @server xSeries and Netfinity*, SG24-5861
- *SuSE Linux Integration Guide for IBM @server xSeries and Netfinity*, SG24-5863

---

### E.2 IBM Redbooks collections

Redbooks are also available on the following CD-ROMs. Click the CD-ROMs button at [ibm.com/redbooks](http://ibm.com/redbooks) for information about all the CD-ROMs offered, updates and formats.

CD-ROM Title	Collection Kit Number
IBM System/390 Redbooks Collection	SK2T-2177
IBM Networking Redbooks Collection	SK2T-6022
IBM Transaction Processing and Data Management Redbooks Collection	SK2T-8038
IBM Lotus Redbooks Collection	SK2T-8039
Tivoli Redbooks Collection	SK2T-8044
IBM AS/400 Redbooks Collection	SK2T-2849
IBM Netfinity Hardware and Software Redbooks Collection	SK2T-8046
IBM RS/6000 Redbooks Collection	SK2T-8043
IBM Application Development Redbooks Collection	SK2T-8037
IBM Enterprise Storage and Systems Management Solutions	SK3T-3694

---

### E.3 Other publications

These publications are also relevant as further information sources:

- *Understanding and Deploying LDAP Directory Services*, by Timothy Howes, Mark Smith, and Gordon Good, ISBN: 1578700701

- *The Linux NIS(YP)/NYS/NIS+ HOWTO* by Thorsten Kakuk, found at:  
<http://metalab.unc.edu/pub/Linux/docs/HOWTO/NIS-HOWTO>.
- *Managing NFS and NIS*, by Hal Stern, ISBN 0937175757

---

## E.4 Referenced Web sites

- <http://www.netcraft.com/survey>
- <http://www.apache.org>
- <http://modules.apache.org>
- <http://www-4.ibm.com/software/web servers/httpservers>
- <http://www-4.ibm.com/software/web servers/httpservers/download.htm>
- [http://www-4.ibm.com/software/web servers/httpservers/doc/v136/readme\\_html](http://www-4.ibm.com/software/web servers/httpservers/doc/v136/readme_html)
- <http://www.apache.org/docs/misc/perf-tuning.html>
- <http://www.suse.com/Support/Doku/FAQ>
- <http://www.redbooks.ibm.com>
- <http://www.turbolinux.com>
- <http://www.pc.ibm.com/support>
- [http://www.pc.ibm.com/us/netfinity/tech\\_library.html](http://www.pc.ibm.com/us/netfinity/tech_library.html)
- <http://www.networking.ibm.com>
- <http://www.xfree86.org/FAQ>
- <http://www.rpm.org>
- <http://www.developer.ibm.com/welcome/netfinity/serveraid.html>
- <http://www.linuxdoc.org/HOWTO/DNS-HOWTO.html>
- <http://www.samba.org>
- <http://www.rustcorp.com/linux/ipchains>
- <http://www.isc.org/dhcp-contrib.html>
- <http://www.sendmail.org>
- <http://metalab.unc.edu/pub/Linux/docs/HOWTO/NIS-HOWTO>
- <http://linuxdoc.org>
- <http://www.openldap.org/incoming/roaming-073099.tar.gz>
- [http://help.netscape.com/products/client/communicator/manual\\_roaming2.html](http://help.netscape.com/products/client/communicator/manual_roaming2.html)
- <http://www.OpenLDAP.org>
- <http://linux.powertweak.com>
- <http://tune.linux.com>
- <http://www.tunelinux.com>

- <http://www.linux-mandrake.com/lothar>
- <http://www.textuality.com/bonnie>
- <http://www.netperf.org/netperf/NetperfPage.html>
- <http://www.raid-advisory.com>
- <http://www.xfree.org>





---

## How to get IBM Redbooks

This section explains how both customers and IBM employees can find out about IBM Redbooks, redpieces, and CD-ROMs. A form for ordering books and CD-ROMs by fax or e-mail is also provided.

- **Redbooks Web Site** [ibm.com/redbooks](http://ibm.com/redbooks)

Search for, view, download, or order hardcopy/CD-ROM Redbooks from the Redbooks Web site. Also read redpieces and download additional materials (code samples or diskette/CD-ROM images) from this Redbooks site.

Redpieces are Redbooks in progress; not all Redbooks become redpieces and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

- **E-mail Orders**

Send orders by e-mail including information from the IBM Redbooks fax order form to:

	<b>e-mail address</b>
In United States or Canada	pubscan@us.ibm.com
Outside North America	Contact information is in the "How to Order" section at this site: <a href="http://www.elink.ibm.com/pbl/pbl">http://www.elink.ibm.com/pbl/pbl</a>

- **Telephone Orders**

United States (toll free)	1-800-879-2755
Canada (toll free)	1-800-IBM-4YOU
Outside North America	Country coordinator phone number is in the "How to Order" section at this site: <a href="http://www.elink.ibm.com/pbl/pbl">http://www.elink.ibm.com/pbl/pbl</a>

- **Fax Orders**

United States (toll free)	1-800-445-9269
Canada	1-403-267-4455
Outside North America	Fax phone number is in the "How to Order" section at this site: <a href="http://www.elink.ibm.com/pbl/pbl">http://www.elink.ibm.com/pbl/pbl</a>

This information was current at the time of publication, but is continually subject to change. The latest information may be found at the Redbooks Web site.

### IBM Intranet for Employees

IBM employees may register for information on workshops, residencies, and Redbooks by accessing the IBM Intranet Web site at <http://w3.itso.ibm.com/> and clicking the ITSO Mailing List button. Look in the Materials repository for workshops, presentations, papers, and Web pages developed and written by the ITSO technical professionals; click the Additional Materials button. Employees may access MyNews at <http://w3.ibm.com/> for redbook, residency, and workshop announcements.



---

## Index

### Symbols

/boot 15  
/home 16  
/tmp 16  
/var 16

### A

Apache 167  
    features 167  
    performance tips 176

### B

backup 257  
backup and restore utility 257  
BackupEDGE  
    unattended operation 267  
bad blocks 21  
base address 3  
bind4 131  
bind8 131  
BIOS 4  
block interleave data striping 343  
bootable 19  
Bootp 25  
Bourne Again Shell 67  
Bourne Shell 67  
BRU 261  
    basic backup 259  
    basic restore 259  
    commands 259  
    installation 257  
    restore 265  
    scheduling 264

### C

C Shell 67  
Caldera Systems 1  
CD-ROM 3, 9  
CFDISK 16, 17, 24  
Cyclades 14

### D

DAP 232  
DARPA 213

data compression 266  
data stripe mirroring 343  
default gateway 25  
Defunct Disk Drive (DDD) 337  
DHCP 209, 210  
    to get IP addresses automatically 25  
directory service 231  
disk mirroring 331, 343  
disk partitioning 15  
disk striping 330  
disk subsystem  
    *See also* RAID  
    RAID performance 340  
display adapter 3  
distributed hot spare 344  
distribution and updates 257  
DNS 129, 130, 131, 190, 209  
    configuration 131  
    installation 131  
    name server 25  
    YaST 132  
drive partition 10

### E

e-business 1  
error detection 257  
error recovery 257  
Ethernet card 8  
even parity 334  
extended verbosity 13

### F

FDISK 16  
file comparisons 257  
file overwrite protection 257  
file/print 139  
FQDN 129  
FTape 14  
FTP server 11, 12  
full backups 257

### H

hard drives 3  
hardware 3, 79  
hdparm 256  
host name 10

hot spare drives 337

## I

I/O address 3  
iBCS 14  
IBM Development and Competency Centers for Linux 1  
IBM HTTP Server 167, 168  
    administration server 172  
    features 168  
    installation 170  
    LDAP 168  
    machine translation support 168  
    performance tips 177  
    remote configuration 168  
    SNMP support 168  
    SSL secure connection 168  
IBM networking products 4  
IBM Technology Center 1  
IETF 232  
incremental backups 257, 266  
installing Linux  
    ServerAID 91  
Intel Binary Compatibility Standard (iBCS) 14  
interrupt level 3  
IP Address 25  
IP address 10  
ISA PnP cards 34  
ISO 232

## K

kernel 1  
Korn Shell 67

## L

LDAP 168, 231, 233  
LDIF 233  
LILO (Linux Loader) 19  
    boot process and file 30  
Linus Torvalds 1  
Linux 1  
Linux filesystem 13  
Linux kernel 241  
Linux Redbooks 359  
local printer 33  
lp application 14

## M

master backups 266  
Master Boot Record 19, 30  
memory location 3  
Microlite  
    BackupEDGE 266, 268  
        backup 270  
        features 266, 302  
        incremental backup 276  
        installation 267  
        master backup 276  
        master restore 279  
        restore 274  
        schedule backup 280  
        tape device 283  
    RecoverEDGE 292  
        boot disks 293  
        features 292  
        total crash 300  
Microsoft Windows and Linux 19  
mirrored RAID-0 arrays 343  
monitor 3  
mount table 24  
mouse 3  
multivolume archives 257

## N

name resolution 130  
NetBIOS 139  
Netfinity brand 3, 4, 29  
Netmask 25  
netperf 256  
network card 3  
Network Interface Card (NIC) 13  
network options 13  
networking 13  
NFS 9, 213, 214  
    portmap 213  
    process 213  
    remote access 219  
    rpc.mountd 213  
    rpc.nfsd 214  
    rpc.rquotad 214  
NIS 221  
    client 228  
    installation 221, 222  
    server 223  
NIS+ 222

NISGINA 221  
normal verbosity 12

## O

odd parity 334  
Open Source development lab 1  
orthogonal 344

## P

package management using RPM 61  
packet filtering with IP chains  
    additional information 188  
    dial-up Internet connection 179  
    FTP masquerading 183  
    gateway 179, 184  
        checksum 184  
        demasquerade 184  
        forward chain 185  
        input chain 184  
        lo interface 185  
        local process 185  
        output chain 185  
        routing decision 184  
        sanity 184  
    IP chains 186  
    IP forwarding 182  
    NAT 179  
    network configuration 180  
parallel-port 8  
performance  
    of RAID subsystems 340  
performance tools  
    Powertweak 242  
performance tools in Linux 241  
Personal Systems Reference (PSREF) 4  
PHP 167  
Point to Point Protocol (PPP) 13  
POP3 191, 205  
primary partition 20  
printer setup 33  
pstree 248

## R

RAID 343  
    described 329  
    level 0 (RAID-0) 330  
    level 1 (RAID-1, RAID-1E) 331

    level 10 (RAID-10) 332  
    level 5 (RAID-5) 333  
    level 5 enhanced (RAID-5E) 337  
    orthogonal RAID-5 339  
    performance 340  
    RAID Advisory Board 344  
    recommendations 342  
    software-based 329  
    summary of RAID levels 343  
    support for two disk failures 337  
    supported disk technologies 329  
RAID level 344  
RAID performance characteristics 344  
RAM 3  
random access 257  
raw device backup 266  
recompile 241  
recovery 257  
Red Hat 1  
Red Hat Package Manager (RPM) 56, 61  
remote LPD queue 33  
remote tape drive support 266  
RFC 232  
Riscom/8 14  
root filesystem 22  
RPC 213  
RPM 97  
    See Red Hat Package Manager

## S

Samba 139  
    configuration 140  
    global settings 141  
    NetBIOS 141  
    printer shares 147  
    security 142  
    shares 145  
    start 148  
    stop 148  
    SWAT 148  
        logon 149  
    WINS 139  
scheduling utility 264  
SCSI adapter 3  
SCSI generic support 14  
sendmail  
    additional information 208  
    configuration 202

- mail client 205
- mail routing 204
- MTA 189
- network configuration 192
- packages 189
- POP3 191, 205
- SMTP 191
- ServeRAID 91
  - adapter firmware 4
  - configuration 91
  - driver 92
  - hot swap rebuild 106
  - ipssend 99
    - devinfo 105
    - getconfig 100
    - getstatus 104
    - hsrebuild 106
    - rebuild 111
    - setstate 108
    - synch 110
    - unattended 110
  - manager utility 120
  - rebuild drive 111
  - replace drive 112
  - RPM 97
  - synchronize logical drives 110
  - unattended mode 110
  - utility 92
- ServeRAID command line tool 97
- SMB 139
- SMB/LAN manager printer. 33
- SMB/LAN Manager support 13
- SMTP 191
- SNMP 168
- source code 1
- SSL 168
- static IP address 209
- Sun Yellow Pages (YP) 221
- SuSE 1
- swap file 16
- swap partition 20
- swap partitions 241
- swap space 24
- SWAT 148
  - global settings 149, 151
  - logon 149
  - printers 159
  - restart Samba 158
  - Samba passwords 164

- Samba status 162
- shares 153
- system monitoring 247
- system performance 247

## T

- top 247
- TurboLinux 1, 2, 3, 5
  - adding new groups 62
  - adding new users 65
  - auto-probing 7
  - clustered server version 5
  - drivers 7
  - installation 5
    - check for bad blocks 21
    - choosing a printer 33
    - configureTCP/IP 25
    - disk partitioning 15
    - four methods 9
  - language 6
  - PCMCIA support 6
- type 19

## U

- units 19
- UNIX 1, 139, 221

## V

- VFAT(Win95) 13
- virtual file support 266
- vmstat 256

## W

- Web site
  - RAID Advisory Board 344
- Web sites referenced 360
- working modes for Xfree86 servers 345
- write 19

## X

- X Server setup 35
- X.500 231, 232
- X-architecture 2
- Xfree86 servers 345
- xSeries brand 2, 3, 4, 29

---

## IBM Redbooks review

Your feedback is valued by the Redbook authors. In particular we are interested in situations where a Redbook "made the difference" in a task or problem you encountered. Using one of the following methods, **please review the Redbook, addressing value, subject matter, structure, depth and quality as appropriate.**

- Use the online **Contact us** review redbook form found at [ibm.com/redbooks](http://ibm.com/redbooks)
- Fax this form to: USA International Access Code + 1 845 432 8264
- Send your comments in an Internet note to [redbook@us.ibm.com](mailto:redbook@us.ibm.com)

<b>Document Number</b>	SG24-5862-01
<b>Redbook Title</b>	TurboLinux Integration Guide for IBM @server for IBM xSeries and Netfinity
<b>Review</b>	        
<b>What other subjects would you like to see IBM Redbooks address?</b>	   
<b>Please rate your overall satisfaction:</b>	<input type="radio"/> Very Good <input type="radio"/> Good <input type="radio"/> Average <input type="radio"/> Poor
<b>Please identify yourself as belonging to one of the following groups:</b>	<input type="radio"/> Customer <input type="radio"/> Business Partner <input type="radio"/> Solution Developer <input type="radio"/> IBM, Lotus or Tivoli Employee <input type="radio"/> None of the above
<b>Your email address:</b> The data you provide here may be used to provide you with information from IBM or our business partners about our products, services or activities.	<input type="checkbox"/> Please do not use the information collected here for future marketing or promotional contacts or other communications beyond the scope of this transaction.
<b>Questions about IBM's privacy policy?</b>	The following link explains how we protect your personal information. <a href="http://ibm.com/privacy/yourprivacy/">ibm.com/privacy/yourprivacy/</a>







Redbooks

**Turbolinux Integration Guide for IBM @serverSeries and Netfinity**

(0.5" spine)

0.475" <-> 0.875"

250 <-> 459 pages







# TurboLinux Integration Guide for IBM **@server** xSeries and Netfinity

**The complete guide  
to running  
TurboLinux on  
xSeries and Netfinity**

**Netfinity  
server-specific  
coverage you can't  
find anywhere else,  
including ServeRAID  
configuration**

**Plan, configure, and  
install key services,  
step-by-step:  
Samba, Apache,  
sendmail, DNS,  
DHCP, LDAP, and  
more**

Here's all the information you need to maximize TurboLinux performance and reliability on state-of-the-art IBM **@server** xSeries and Netfinity server platforms. In this book, a team of IBM's top Linux experts presents start-to-finish, Netfinity server-specific coverage of TurboLinux 6.0 deployment and system administration throughout the entire system life cycle!

This redbook is aimed at beginners and intermediate Linux users and for all Windows users who are used to the safe and convenient graphical user interface.

The book covers the installation of TurboLinux 6.0. Once the installation has been completed, the book discusses some basic system administration tools that can help you manage your Linux system. Furthermore, this book provides an introduction to a wide range of services, such as Samba, NFS, and Apache among others. You will learn what each service is, what it is capable of, and how to install it. The services are not covered in detail, since they are very comprehensive. We recommend that you consult additional sources if you need more detailed information. These sources are mentioned during the chapters or at the end of each chapter.

## **INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION**

### **BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE**

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

**For more information:  
[ibm.com/redbooks](http://ibm.com/redbooks)**

SG24-5862-01

ISBN 0738419850