Linux Applications on pSeries

- AIX Toolbox for Linux Applications
- Recompiling Linux Applications on AIX
- Running Linux natively on pSeries

Jan-Rainer Lahmann
Janethe Co
Björn Rodén
Luis Ferreira

Redbooks

ibm.com/redbooks

International Technical Support Organization

Linux Applications on pSeries

February 2003

SG24-6033-01

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

Second Edition (February 2003)

This edition presents additional material and updated material not found in the original work. It reflects changes in installation procedures and usage of the AIX Toolbox for Linux Applications, adds a detailed description of the most-used tools in the Toolbox, and discusses various aspects of native Linux on IBM eServer pSeries hardware.

This edition applies to AIX Toolbox for Linux Applications for use with AIX 5L and AIX 4.3.3 operating systems. The chapter about native Linux applies to pSeries hardware and Linux distributions listed in 5.1, “Introduction” on page 116.

Contents

Figures

Notices

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

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

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

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

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

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

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

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

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

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

Trademarks

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

AIX®
AIX 5L™
DB2®
IBM®
IBM eServer™
iSeries™
Lotus®
MVS™
Netfinity®
Notes®
OS/2®
OS/390®
Perform™
PowerPC®
PowerPC 601®
PowerPC 603™
PowerPC 604™
PowerPC Reference Platform®
pSeries™
Redbooks™
Redbooks (logo)™
RS/6000®
S/390®
SAA®
SP™
TCS®
Tivoli®
WebSphere®
Word Pro®
xSeries™
z/VM™
zSeries™

The following terms are trademarks of other companies:

ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States, other countries, or both.

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

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

C-bus is a trademark of Corollary, Inc. in the United States, other countries, or both.

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

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

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

Preface

The strengths of the AIX operating system are well known among in the UNIX software community. Its reliability and great degree of scaling make AIX the perfect choice for hosting mission-critical applications. It is a robust and flexible operating system that meets all the requirements for the various demands of e-business environments. At the same time, Linux is maturing rapidly and generating excitement among software developers that has not been seen in many years.

With the adoption of Linux in early 2000, IBM became very interested in enabling Linux applications to run on the AIX operating system. Thus, the AIX Toolbox for Linux Applications was developed. The Toolbox provides the capability to easily recompile and port Linux applications to AIX and provides tools to work on those applications. In addition, Linux can be installed natively on pSeries and RS/6000 hardware. This opens the pSeries hardware platform for additional Linux and Open Source applications.

Countless developers around the world are completely focused on developing applications for Linux systems. You can easily port these applications and run them directly on AIX while taking advantage of all the features and benefits that the AIX operating system offers—or you can let them run under a native Linux operating system on pSeries.

This IBM Redbook will show you what you need in order to run Linux applications on pSeries. The AIX Toolbox for Linux Applications contains a large collection of Open Source and GNU software built for AIX 5L and AIX 4.3.3 for IBM eServer pSeries systems and IBM RS/6000. The book will help you comprehend and install the Toolbox, understand the procedure to follow for porting Open Source software utilizing the Toolbox, and give a detailed description of a large number of the tools delivered in the Toolbox. It will also describe the procedure to install a native Linux operating system on several pSeries hardware types in detail—including some common Linux applications.

The team that wrote this redbook

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

Dr. Jan-Rainer Lahmann is an IBM eServer Consultant and IT Architect in Germany. He holds a PhD in applied mathematics from the University of

Karlsruhe, Germany. He has ten years of experience managing RS/6000 and Linux clusters. Currently, he is a technical expert for Linux environments on both the pSeries and zSeries systems. The ITSO Austin invited him to be a guest Project Leader for this redbook project.

Luis Ferreira (also known as “Luix”) is a Software Engineer at IBM Corporation - International Technical Support Organization, Austin Center, working on Linux, AIX, and Grid computing projects. He has 19 years of experience with UNIX-like operating systems, and holds a MSc. Degree in System Engineering from Universidade Federal do Rio de Janeiro in Brazil. Before joining the ITSO, Luis worked at Tivoli Systems as a Certified Tivoli Consultant, at IBM Brazil as a Certified IT Specialist, and at Cobra Computadores as a Kernel Developer and Software Designer.

Janethe Co is a System Services Representative at IBM Philippines. She holds a Bachelor of Science Degree in Electronics and Communications Engineering from De La Salle University, Manila. She has three years of experience with supporting pSeries RS/6000 servers. Her areas of expertise include AIX, Linux, Security, and Firewall.

Björn Rodén is an AIX Expert from Sweden working as a Chief Technical Manager and Chief Programmer for the largest IBM pSeries/zSeries Business Partner in Sweden (Pulsen Systems). He has 13 years of experience with AIX and has worked with SP, HACMP, ADSM/Tivoli Storage Manager, and Open Source Software since the 1990s. He is currently certified as an AIX Advanced Technical Expert, Mid-Range Storage Technical Specialist, SP Specialist, HACMP Specialist, ADSM/Tivoli Storage Manager Specialist, Web Server Specialist, and Tivoli Instructor. He is a co-author of three previous redbooks: *RS/6000 SP System Performance Tuning Update,* SG24-5340, *AIX 5L Performance Tools Handbook,* SG24-6039, and *Tivoli Storage Manager Version 5.1 Technical Guide,* SG24-6554.

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

The team would like to express *special thanks* to the following people for their major contributions to this project:

David Clissold, Advisory Software Engineer
IBM Austin

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

Lupe Brown, Wade Wallace, Joanne Luedtke, Keigo Matsubara, Bart Jacob, Alex Jeong, Scott Vetter, Julie Czubik
International Technical Support Organization, Austin Center

Aaron Bolding, Mark Brown, Marc Stephenson, John Bissell, Jay Ashford, Reza Arbab, Matthew Davis, Dan McNichol
IBM Austin

Thomas Harrer and Frank Heynemann
IBM Germany

Olaf Hering
SuSE Linux AG

The Open Source Community
Especially Kevin Hendricks and Ken Foskey

Become a published author

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

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

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

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

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

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

  ibm.com/redbooks

- Send your comments in an Internet note to:

  redbook@us.ibm.com

- Mail your comments to:

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

Summary of changes

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

Summary of Changes
for SG24-6033-01
for Linux Applications on pSeries
as created or updated on February 21, 2003.

February 2003, Second Edition

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

New information

- An extensive description of the most-used Tools from the AIX Toolbox for Linux Applications has been added, including various examples.
- A new Chapter about native Linux on pSeries was added, discussing various aspects ranging from Linux installation on different pSeries hardware categories to software enablement.

Changed information

- The book title changed from “Running Linux Applications on AIX” to “Linux Applications on pSeries” because of the added content.
- This new edition reflects changes in installation procedures for the AIX Toolbox for Linux Applications, which include new scripts to ease installation and updates.
- Broader discussion of porting and recompiling standard Linux software (in form of SRPMs) to AIX.

1

Introduction

This chapter provides a brief history of IBM's AIX UNIX operating system and its current status. It also provides a brief introduction to another very popular UNIX-based operating system called Linux, its different flavors, and why it plays a major role in the IBM AIX roadmap.

In this chapter, the following topics are described:

- AIX, including a brief history of AIX
- Linux, including a brief history of Linux
- Linux in IBM
- The integration of AIX and Linux on pSeries

1.1 AIX

The IBM AIX operating system is a modern UNIX operating system for the pSeries and RS/6000 family that supports 32-bit and 64-bit applications. It provides a great degree of scaling, including uniprocessors, symmetric multiprocessors, clusters, and massively parallel systems, using a single consistent set of application and binary interfaces. AIX is highly reliable and supports high-availability storage systems and various cluster configurations for near fault-tolerant operations.

AIX is designed for mission-critical, core business applications, providing an integrated environment that is stable, highly scalable, and functionality rich. It inter-operates with many heterogeneous platforms and offers powerful management solutions for the enterprise.

IBM AIX is based on the ATT System V UNIX operating system but has incorporated enhancements developed by other organizations and institutions (for example, BSD UNIX (Berkeley Software Distribution)), included new features and POSIX IEEE 1003.1 standards conformance, and added many enhancements to produce the Advanced Interactive eXecutive (AIX).

Today, AIX runs on IBM's family of RISC workstations and servers, known as IBM RS/6000 or IBM @server pSeries models.

1.1.1 Current version and features

Over the past years, the vast growth of Internet users and the explosion of e-business has produced a large demand on server computing requirements. These trends are placing unprecedented demands on IT systems and creating the need for more advanced technology and standardization. The new e-business environments demand an operating system (OS) capable of handling the mission-critical functional requirements that allow businesses to focus on delivering a greater value to customers.

As the latest generation of AIX, AIX 5L fulfills the required needs of e-business environments and provides even more capabilities, including enhanced support for IBM POWER platforms, and a strong affinity with Linux.

IBM AIX 5L was designed with one simple goal in mind: To produce a single UNIX-based product line with broad industry support and to establish AIX as the leading open, industrial strength UNIX-based operating system.

AIX 5L is currently shipping and includes many new features, a description of which can be found at:

`http://www.ibm.com/servers/aix/os/index.html`

The benefits of these new features are:

- Flexibility
  - Affinity with Linux

    Helps deliver services across technology boundaries by allowing portable Linux applications to be combined with the scalability and robustness of AIX
- System scalability
  - JFS2 file system

    Efficient storage of large (16 Terabyte) files assists deployment of advanced applications and databases
  - Large pages

    16-MB pages help improve throughput for model p670/p690 compute-intensive workloads that require large amounts of data to be transferred between memory and storage
- Logical partition support for model p630/p670/p690
  - Dynamic LPAR

    Enables addition or removal of processors, adapters, or memory without system reboot, improving system availability and resource utilization
  - Dynamic Capacity Upgrade on Demand (CUoD)

    Allows activation of additional processors when needed—without a system or partition reboot—for greater flexibility and improved workload throughput
  - Dynamic Processor Sparing (with CUoD)

    Supports dynamic substitution of failing processors with spare, inactive processors to help keep systems available and processing their assigned workloads
- e-business and network performance
  - Virtual IP address (VIPA)

    Helps applications remain available if a network connection is lost
  - IP multipath routing

    Improves network availability by providing multiple routes to a destination
  - Multiple default gateways

    Keeps traffic moving through a network by detecting and routing around dead gateways

  - Mobile IPv6

    Extends internet connectivity to small, hand-held devices
  - Network tuning interface

    Helps reduce administrative effort associated with managing and tuning networks
- Security
  - Kerberos V5 Authentication

    Helps administrators simplify password authentication for users connecting to several machines
  - Pluggable Authentication Module (PAM)

    Permits the use of distributed security services to reduce administrative effort associated with linking users to multiple applications
  - Enterprise identity mapping

    Allows a user single-point access to a network comprised of heterogeneous server platforms
- Enhanced Java support
  - Included in base AIX

    Delivers a popular cross-platform programming language for e-business applications
- Systems and resource management
  - Fix Manager

    Provides reports that compare fix levels on a system to a reference system or base level of fixes for easier administration
  - Resource Monitoring and Control (RSCT)

    Delivers clustering technology to automate resource monitoring, improving system availability and performance
  - Distributed Command Execution Manager

    Offers centralized management of groups of AIX based servers through a web browser interface
  - Dynamic Workload Manager

    Adds time-based resource policies to allocate resources to applications within a whole system or in a partition

- Reliability, Availability, Serviceability (RAS)
  - Automated system hang recovery

    Helps systems remain available without administrator intervention
  - CPU-Gard

    Pro-actively checks processor integrity and removes failing processors so that systems are more available
  - System UE-Gard

    Improves system uptime by pro-actively managing checkstop errors at a thread level
  - Multipath I/O

    Enhances internal reliability of SCSI disk connections and permits maintenance deferral
- Storage
  - Split Mirror support for Logical Volume Manager

    Helps reduce any impact to system performance due to re-integrating the split mirror
  - JFS2 file system snapshot

    Helps administrators monitor and manage file system for action as needed
  - I/O size and alignment for Logical Volume Manager

    Removes size and alignment restrictions to help improve file system and overall system performance
  - Storage Area Network (SAN) boot

    Adds capability to initiate system boot from a single point of contact in a SAN
  - Migration via Alternate Disk Install

    Improves management of multiple operating system migrations in environments where downtime is critical

1.2 Linux

The Linux operating system is a free UNIX-like operating system that supports full multitasking, the X Window System, TCP/IP networking, and much more.

In the past few years, Linux has generated more excitement in the computer industry than any other development. Linux can run on a large variety of computer systems, turning them into powerful workstations and servers that give you the power of UNIX software at your fingertips.

1.2.1 Brief history

Linux is a freely distributed operating system based on a development approach that delivers innovation and portability. It was originally developed by Linus Torvalds, who started work on Linux in 1991 while he was a student at the University of Helsinki in Finland. Torvalds was inspired by Andrew Tanenbaum's Minix, a small UNIX-based operating system.

The initial release of Linux was distributed by means of the Internet, and generated one of the largest software development phenomena of all time.

The first official release of Linux, Version 0.02, took place on October 5, 1991; at this point, Torvalds was able to run bash (the GNU Bourne Again Shell) and **`gcc`** (the GNU C compiler). Basically, Linux was intended as a hacker's system. As Linux is one of the fastest maturing operating systems, the situation has now changed. The operating system provides a solid graphical environment, easy-to-install packages, and high-level applications.

Linux was initially developed for the Intel x86 architecture platform, but it is important to know that Linux now supports many other hardware platforms, such as PowerPC, S/390, SPARC, Alpha, and various processor types for embedded systems.

1.2.2 About Linux's copyright

The Linux kernel is written, distributed, and covered under the GNU General Public License (GPL), which means that its source code can be freely distributed and is available to the general public.

For information regarding GNU/Linux copyrights, the GNU Project, and the GNU General Public License (GPL), please refer to the following URL:

`http://www.gnu.org`

This is also discussed in 2.1.3, "Open Source Software" on page 17.

1.2.3 The GNU Project and the Linux kernel

By the 1980s, most of the operating systems were proprietary, which meant that you had to use the operating system provided for your specific hardware platform.

The initiative of the Free Software Foundation (FSF) and the GNU Project motivated and stimulated open development and worldwide user cooperation. The main goal of the GNU Project was to develop a UNIX-compatible operating system named GNU (GNU is not UNIX), capable of running on various hardware architectures. Calling it GNU was a way of paying tribute to UNIX-like systems while saying that GNU was something different. It was to be 100 percent free, which meant that users would be free to redistribute the whole system, and free to change and contribute to any part of it. It was decided to make it UNIX-compatible because UNIX had already been proven in terms of design and portability.

The GNU Project was founded by Richard Stallman, the founder of the Free Software Foundation, author of the GNU General Public License, and the original developer of some GNU software programs (for example, the **gcc** compiler and the Emacs text editor).

It took many years of hard work to write all the pieces of the GNU-based operating system, hundreds of programmers worldwide, and many hackers who worked very hard on the code and at the same time also used it. By 1990, most of the software pieces had been written except for the most important one: The kernel. The kernel is the core of the operating system. It is the piece of code that directly communicates and controls the interface between the user programs and the hardware devices (for example, disks, keyboard, mouse, and video). By that time, the free UNIX-based kernel developed by Linus Torvalds was combined with the GNU system, resulting in a complete operating system.

Today, the combination of GNU tools and commands and the Linux kernel is widely used around the world, and its popularity grows on a daily basis.

1.2.4 Different flavors of Linux

As a benefit of the source code for the Linux kernel being freely distributed, different companies have developed their own “flavor” or *distribution* of Linux. Each of these flavors has its own feature set, such as installation and administration procedures, software packages, and configurations. Many of them are configured for a specific type of computer system.

Some of the most popular distributions are:

Corel Linux	Developed by Corel Corporation
Debian GNU/Linux	Developed by The Debian Project
Linux Mandrake	Developed by MandrakeSoft, Inc.
Red Hat Linux	Developed by Red Hat, Inc.
SuSE Linux	Developed by SuSE, Inc.

UnitedLinux Developed by The SCO Group (formerly Caldera Systems, Inc.), Conectiva S/A, SuSE, Inc., and TurboLinux, Inc. (now part of SRA Japan)

As early as 1995, IBM Research and recognized experts in the Linux community ported Linux to the native PowerPC architecture platform and a Linux kernel (Version 2.2) for the IBM RS/6000 was developed. The initial RS/6000 support, following PowerPC Reference Platform (PReP) and Common Hardware Reference Platform (CHRP) specifications, was provided by Yellow Dog Linux on the IBM produced machines, such as the 7043-150, 7025-F50, and 7046-B50.

For more information regarding Linux on pSeries, please refer to:

`http://www.ibm.com/servers/eserver/pseries/linux`

Also to Chapter 5, “Native Linux on pSeries” on page 115.

1.3 Linux at IBM

IBM is focusing on Linux because of the increased mind share and market share that Linux is getting, the rapid market changes, and the customer needs. Also, Linux is a stable and reliable development and deployment platform for Internet applications. Its low cost and broad platform support allow applications to be developed on commodity hardware and deployed across a wide range of systems.

Linux can be acquired at no cost as a download from the Internet, and the kernel and most of the extensions are available as source code and can be improved by anyone willing to contribute.

Linux is a very popular operating system for Web servers and dedicated networking functions, such as Web infrastructure, file-and-print serving, firewalls, directory serving, e-mail serving, and so on. It is rapidly gaining popularity on application and even on database servers. Linux has also gained acceptance as an embedded OS for new Internet, file server, and other application appliances.

In the enterprise arena, AIX is IBM's strategic UNIX operating system for mission-critical, core business applications. The industrial-strength features and functions of AIX have been well proven over the years in a wide variety of server environments, from relatively small, single-processor systems to IBM's scalable, high-performance clusters, such as the IBM @server Cluster 1600 and RS/6000 Scalable POWERParallel (SP) servers. Features include 32-bit and 64-bit Application Programming Interface (API) support, state-of-the-art preemptive kernels, dynamic configuration and device attachments, a robust journaled file system, Logical Volume Manager (LVM) software, the simplified system administration commands System Management Interface Tool (SMIT) and Web-based System Manager (WebSM), industry standards compliance, high-availability cluster multiprocessing (HACMP) software products, and tens of thousands of supported customer applications.

1.3.1 IBM's UNIX-based operating system strategy

The IBM strategy for UNIX-based operating systems is built upon the great momentum that AIX is having, the establishment of AIX 5L as an enterprise class, industry leading, UNIX-based system with support for POWER architectures and a solid affinity with Linux.

Linux is being positioned as the strategic, high-volume UNIX-based operating system. Enabling Linux across all IBM @server platforms is also an important part of IBM's strategy. This allows porting applications to all of these platforms with little to no changes required to the source code.

In Figure Figure 1-1 on page 10 we show:

- IBM's commitment to the UNIX philosophy that gives reassurance that IBM is producing an open, industry standard platform.
- How AIX (on the pSeries and RS/6000 model) is gaining tremendous market momentum as the industrial-strength UNIX-based operating system for mission-critical environments.
- IBM's strategy calls for expanding the application base on Linux and establishes IBM as a leader enabling that transition for ISVs.
- Linux becoming a primary application development environment for all IBM platforms.
- Strong affinity between AIX and Linux that enables porting of Linux applications on AIX using the GNUPro development tools.
- The Linux compatibility that will help drive AIX to be more open, as opposed to being thought of as IBM's proprietary UNIX-based operating system.
- Dual-technology strategy satisfies widest range of customer needs.
- IBM has made an extensive commitment to support Linux as an open computing environment.
- IBM understands that the open computing business model requires customer flexibility and choice.

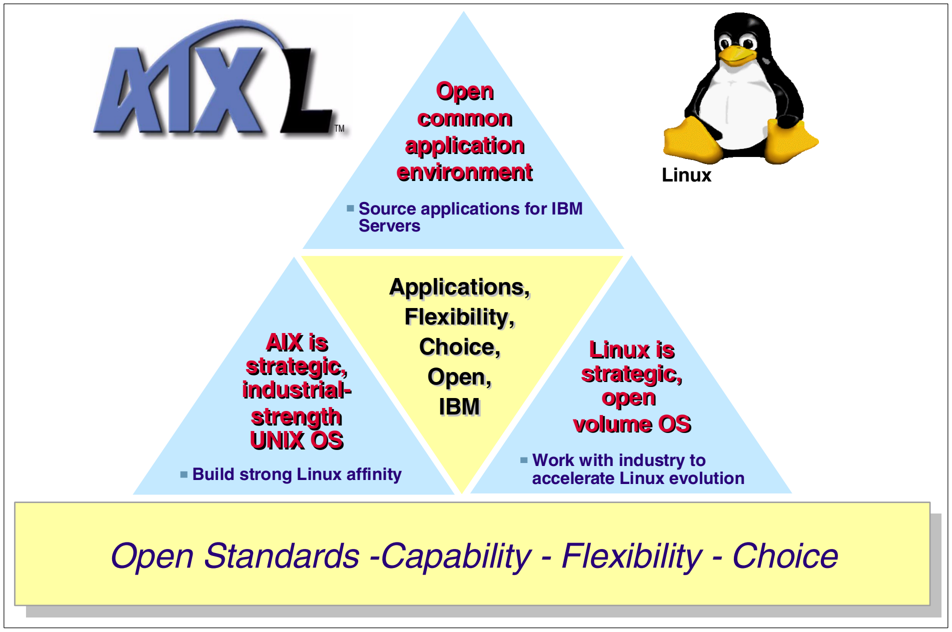

Figure 1-1 IBM's UNIX-based operating system strategy

1.4 The integration of AIX and Linux on pSeries

The high level of activity on the UNIX-based systems and Linux fronts during the past few years is allowing Linux to establish itself as a mainstream UNIX player. It looks as though Linux is going to be transformed into an enterprise class operating system over the years.

Figure 1-2 on page 11 lays out IBM's perception of the UNIX-based operating system environment. The direction for the IBM UNIX-based operation system evolution will be integration, coexistence, and interoperability of AIX and Linux, and a solid Linux affinity with the AIX 5L operating system.

Linux continues to scale and address larger computing tasks, and IBM is doing its part to speed this process along, while at the same time, optimizing IBM @server systems to offer customers the option of using Linux. IBM has taken on an active leadership role in Linux for the PowerPC. IBM developers are continuously working to enable pSeries systems in accordance with this strategy.

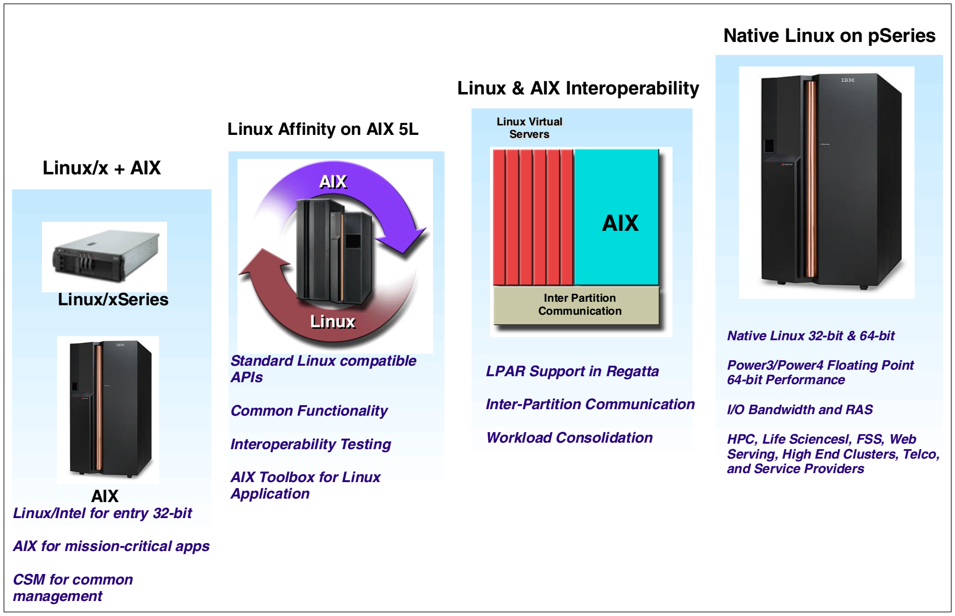

Figure 1-2 pSeries - AIX and Linux strategy

1.4.1 Linux on xSeries + AIX on pSeries

Linux and AIX are complementary systems. AIX is the strategic, proven, mission-critical operating system for the pSeries. Linux is a highly portable operating system that supports all IBM @server platforms. IBM expects to see many installations running Linux (on xSeries or pSeries hardware) as the front end to mission-critical AIX systems running DB2 and the other enterprise applications, and common management using CSM.

1.4.2 Linux affinity on AIX 5L

There is a strong affinity between Linux and AIX for applications. AIX has a long history of standards compliance and it is generally straightforward to rebuild Linux applications for AIX.

In order to enhance the interoperability between Linux and AIX, IBM ported a collection of Open Source and GNU software from the Linux world and bundled it into a toolbox for users of AIX. The AIX Toolbox for Linux Applications is a major step in IBM's effort to provide AIX and Linux interoperability.

The AIX Toolbox for Linux Applications contains a collection of Open Source and GNU software that works with both AIX 5L and AIX Version 4.3.3 on IBM

@server pSeries and RS/6000 systems. Linux users running Intel architecture machines will have the option to move up to more powerful systems. And for Linux developers, it introduces a way to expand the target for their applications to AIX.

All the tools are packaged using the easy-to-install RPM format. Once installed, the Linux applications can take advantage of the same scalability and performance as any other AIX applications.

1.4.3 Linux and AIX interoperability

Linux is supported running in one or more logical partitions on the @server pSeries model p670 and p690 systems and future LPAR-capable systems. AIX and Linux can run concurrently in separate partitions on an LPAR-enabled systems in any combination (that is, zero or more Linux partitions along with zero or more AIX partitions). This enables the customer to consolidate workloads from several separate servers onto a single system. Since the partitioning is controlled by the hypervisor firmware and the Hardware Management Console (HMC), AIX is never required and the entire system could be partitioned for use as a large Linux server.

1.4.4 Native Linux on pSeries

Linux for pSeries is a key element of the IBM @server Linux strategy. IBM's commitment to provide Linux for pSeries was announced as part of the IBM @server launch in October 2000. IBM intends to increase its growing server momentum by leveraging the power of Open Source in general and Linux in particular to offer new options and value to its customers.

Today, Linux is strong at the lower end of the scalability range, while pSeries has carved out a leadership position in the mid and high end of the enterprise server space. As Linux becomes more mature in enterprise reliability, availability, and scalability, Linux for pSeries will grow more compelling. As Linux scales over time, so too will the workloads for which it can be deployed.

Both 32-bit and 64-bit versions of Linux for pSeries are being provided in order to optimize customer choices and exploit pSeries hardware capabilities. The RS/6000 models B50 and 150 are 32-bit PowerPC systems supported by the Linux for pSeries 32-bit kernel and application environment. The 64-bit POWER3 and POWER4 systems have a 64-bit kernel (depending on the model and Linux distributions) but still maintain a 32-bit application address space. IBM is working closely with the Open Source community to provide the 64-bit technology for the PowerPC Linux libraries and support tools. This will result in a full 64-bit environment that fully exploits the increased addressability and performance of the 64-bit processors.

In Figure 1-3 we show the complementary AIX and Linux strategy:

- Using IBM's proven base of OS development and technologies to accelerate the maturation of Linux as an architecture-independent Enterprise server platform
- "Compile and Go" enabling for Linux applications on AIX
- Choosing Linux where it provides an advantage
- AIX taking advantage of the growing Linux skill base
- IBM expanding its available application portfolio by allowing Linux applications to be easily compiled on AIX with the Toolbox package

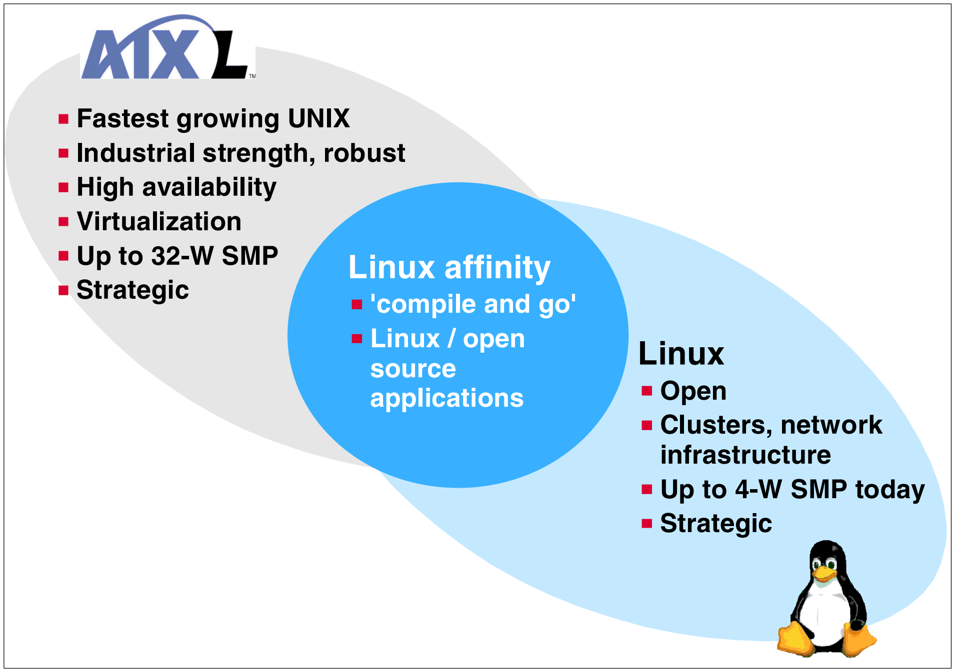

Figure 1-3 AIX and Linux integration

2

AIX Toolbox for Linux Applications

The AIX platform is, and will continue to be, the premier operating system from IBM for pSeries systems. In order to enhance the user interoperability between Linux and AIX, IBM has ported a collection of Open Source Software (OSS) tools and bundled them into a toolbox for users of AIX: The AIX Toolbox for Linux Applications.

For customers of AIX, it opens up a wide range of Open Source applications, development tools, and utilities. UNIX and Linux users running Intel architecture machines will have the option to move up to more powerful systems and still use the same tools. For UNIX and Linux developers, it introduces a way to expand the target systems for applications to AIX.

The Toolbox contains a collection of Open Source Software that works with both AIX 4.3.3 and AIX 5L.

This chapter is divided into the following sections:

- Overview of the Toolbox
- Design of the Toolbox
- Installing the Toolbox packages

2.1 Toolbox overview

AIX is a mission-critical operating system developed for scalability and stability. By porting and running Linux applications on AIX using the AIX Toolbox for Linux Applications, you will get the proven benefits of AIX together with Open Source.

2.1.1 UNIX/AIX/Linux standard compliance

AIX has a long history of standards compliance from international, national, and industry standards from organizations such as:

- International Organization for Standardization (ISO) standards
- American National Standards Institute (ANSI)
- Institute of Electrical and Electronics Engineers (IEEE)
- Internet Engineering Task Force (IETF)
- Federal Information Processing Standard (FIPS)
- The Open Group (a UNIX Vendor Consortium; IBM is a member)
- System V Interface Definition (SVID3) - a standard for System V UNIX products
- IBM Corporate standards (such as the SAA C definition)
- IBM AIX ABI (binary compatibility between releases)

Because of this history, there is a high degree of compatibility at the base system Application Programming Interface (API) level, the base utilities, interoperability, internationalization, multilingual support, and conformance testing between AIX and other flavors of UNIX-based systems.

Linux standards

Linux is not conforming to the IEEE Std 1003.1-2001 standard (Portable Operating System Interface for Computer Environments, or more commonly known as *POSIX*) nor the *The Single UNIX Specification*, which is necessary to conform to in order to brand an operating system as a UNIX operating system. The Single UNIX Specification is a de-facto and de-jure standard definition for the UNIX system application programming interfaces and integrates X/Open Company's XPG4, IEEE's POSIX Standards and ISO C. For more information please refer to the following Web sites:

http://www.opengroup.org/certification/index.htm
http://www.unix-systems.org/version3

Linux has its own standardization organization, the Free Standards Group. For more information please refer to the following Web site:

http://www.freestandards.org

The Free Standards Group has created a operating system conformity standard for Linux systems called the *Linux Standard Base* (LSB), for more information please refer to the following Web site:

http://www.linuxbase.org/spec

It is possible for Linux distributors and Linux-based developers to certify their distributions and applications against the LSB. The certification process is managed by the Open Group, for more information please refer to the following Web site:

http://www.opengroup.org/lsb/cert

For a list of certified distributions please refer to the following Web site:

http://www.opengroup.org/lsb/cert/cert_prodlist.tpl

2.1.2 Toolbox objective

The goal of the AIX Toolbox for Linux Applications is to provide ready-to-run, installable Open Source tools and facilitate recompilation of Open Source Software, without modifications, on AIX systems. These days many Open Source applications are created on Linux systems and are using Linux libraries and APIs. Since Linux is a UNIX clone and not a UNIX branded operating system ("It aims towards *POSIX* and *Single UNIX Specification* compliance"), supporting Linux Open Source Software requires that the Linux APIs are available in AIX to successfully recompile the sources.

For more information regarding the Linux kernel, please refer to the following Web site:

http://www.kernel.org

Once recompiled, the original Linux source applications become native AIX applications, meaning they can take advantage of the same scalability, reliability, and performance as any other AIX application. These applications are AIX binaries.

2.1.3 Open Source Software

Open Source Software (OSS) is software in source-code form that is often created and maintained by a collaborative, virtual community on the Internet and is usually downloadable for free over the Internet or available on CD-ROM at nominal cost. Open Source Software has several important features that distinguish it from other kinds of software:

- You cannot be prohibited from redistributing OSS.
- You cannot be prohibited from distributing modifications to OSS.
- No royalties can be imposed on you for using OSS.

It is important to understand that OSS is not Public Domain Software:

Public Domain Author *gives up copyright*; no restrictions on how the source code can be used

Open Source Author *retains copyright* to source code; allows author to distribute the code under a license that defines what you may (and may not) do with it

There are two distinctly different aspects to Open Source Software:

- The licensing model
- The development methodology

The Open Source Software community has a very precise definition of *Open Source*. That definition is embodied in the various Open Source Software licenses in use today.

The Linux kernel and the GNU software packages are some of the most well-known examples of Open Source Software. The software development tools in the Toolbox are the major ones that many Linux/UNIX application developers prefer to use. However, the Toolbox content is not limited to development tools only.

The Open Source Definition (OSD)

The Open Source Definition (OSD) is promoted and maintained by the not-for-profit Open Source Initiative (OSI). For more information about Open Source please refer to the following Web sites:

http://www.opensource.org/docs/definition_plain.html
http://www.gnu.org/philosophy/free-sw.html

Most of the popular Open Source Software licenses (for example, GPL, LGPL, CPL, MPL, BSD, and MIT) have been certified *OSD compliant* by the Open Source Initiative and are listed on the following Web site:

http://www.opensource.org/licenses

Currently there are 35 approved Open Source Software licenses. There are also over 100 licenses claiming to be Open Source Software licenses, each with its own set of terms and conditions.

The Open Source Definition defines nine criteria that a software license must meet in order to be considered an Open Source Software license:

1. *Free Redistribution:* The license can not prevent anyone from selling or giving away the software and can not impose any royalties or fees on use of the code.
2. *Source Code:* The source code must be made available in some well-publicized way for no more than a reasonable reproduction cost.
3. *Derivative Works:* The license must allow modifications and derivative works to be distributed under the same terms as the original software license.
4. *Integrity of The Author's Source Code:* The license can restrict source code from being distributed in modified form provided it allows the distribution of *patch files*. The license must allow distribution of software built from modified source code, but can require derived works to carry different names/version numbers from the original.
5. *No Discrimination Against Persons or Groups:* The license must not discriminate against any person or group of persons.
6. *No Discrimination Against Fields of Endeavor:* The license must not restrict anyone from making use of the program in a specific field of endeavor.
7. *Distribution of License:* The rights attached to the program must apply to all to whom the program is redistributed.
8. *License Must Not Be Specific to a Product:* The rights attached to the program must not depend on the program being part of a particular software distribution.
9. *The License Must Not Restrict Other Software:* The license must not place restrictions on any other software that is distributed along with the licensed software.

Open Source Software development and support

The heart of the Open Source Software development methodology is a virtual community of programmers, leveraging the Internet for communication, who create, debug, maintain, and evolve a source code base.

Open Source Software project management

Open Source Software projects are typically self-organizing, usually along the following lines:

- Someone determines a need and communicates that need to various USENET groups, bulletin boards, chat rooms, and user groups.
- If the project generates interest among a group of programmers, one or more of them takes the lead and begins writing code.

- Someone takes a leadership role and begins to map out a project road map (leaders can change as the project progresses).
- Interested programmers join the project and contribute new code or fine tune existing project code.
- A network of participants linked via the Internet is formed.
- Tiered participation emerges as programmers of many different skill levels contribute:
  - At the top level are a few project leaders who have earned the respect of the community and exercise ultimate decision making authority.
  - Below them are a small number of maintainers/secondary leaders responsible for major subsystems or modules of the code base.
  - At the third level is a larger group of contributors who identify and fix bugs and make small enhancements.
  - The lowest level consists of a large number of users and small contributors who submit bugs and feature requests.
- Evolution of the project is almost entirely user-driven since most contributors also use the software.
- The distributed working style encourages a clean, modular architectural design that minimizes interdependencies and development constraints.
- Work tasks are distributed down the chain of contributors whether on a volunteer basis or by delegation.
- Development takes place in various modules and at various levels simultaneously with asynchronous communication between co-developers.
- Workflow is iterative with incremental enhancements ("release early, release often" model).
- All work is ultimately accepted or rejected by the top two tiers of developers whose decisions informally enforce the "quality rules" of the community.

Open Source Software support management

Open Source Software support derives from the open-source development methodology and is based around two main principles:

- Release early, release often.
- Listen to your customers and peers.

The size and reach of the global Open Source Software community can support quick fixes that are routinely peer reviewed. Given a large beta-tester and co-developer base, almost any problem should be characterized quickly and the fix obvious to someone. In addition, there are companies who are involved with in OSS communities that are willing to provide free or fee-based support services, such as IBM, Red Hat, and SuSe.

2.1.4 Open Source Software in the Toolbox

The Toolbox contains Open Source applications, both recompiled versions and source code of the Gnome and KDE desktop environments and system utilities, including Samba, shells, GNU base utilities, and application development tools such as compilers and source code editors.

> **Note:** The AIX Toolbox packages can be obtained from the Toolbox Web site, the Toolbox FTP site, or from the Toolbox CD-ROM (shipped with AIX 5L).
>
> The Toolbox Web site:
>
> `http://www.ibm.com/servers/aix/products/aixos/linux`
>
> The Toolbox FTP site:
>
> `ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox`

For detailed information on the current content of the Toolbox, please refer to the following Web site:

`http://www.ibm.com/servers/aix/products/aixos/linux/rpmgroups.html`

> **Note:** The licenses associated with the various packages are available for viewing on the Toolbox CD and on the Toolbox Web site:
>
> `http://www.ibm.com/servers/aix/products/aixos/linux/altlic.html`

The following is a sample of the software that the AIX Toolbox for Linux Applications contains:

GNU base utilities	**tar**, **cpio**, **diffutils**, **fileutils**, **findutils**, and **sh-utils**
System utilities	**bzip2**, **gzip**, **ncftp**, **rsync**, **wget**, **lsof**, and **zip**
System shells	**bash**, **tcsh**, and **zsh**
Graphics applications	**xfig**, **xpdf**, **ghostscript**, **gv**, and **mpage**
Desktop environments	Gnome and KDE
Window managers	enlightenment and sawfish
Application development	**gcc**, **gplusplus**, **gdb**, **cvs**, **make**, **automake**, **autoconf**, **libtool**, **bison**, **flex**, and **m4**
Programming languages	PHP, Python, C, and C++ compilers

Libraries	ncurses, libtiff, libpng, libjpeg, db, gtk+, and qt

Note: Information regarding the development of the Toolbox, Toolbox problem reporting and Toolbox mailing list, is available from the IBM Developer Works Web site for the Toolbox:

http://oss.software.ibm.com/developerworks/projects/aixtoolbox

Cryptographic Content (SSL) for certain Toolbox packages can be obtained from the following Web site:

http://www6.software.ibm.com/dl/aixtbx/aixtbx-p

2.1.5 Other sources of Open Source Software

Apart from the Toolbox, there are other sources from where to obtain Open Source Software for AIX on the Internet. Some distribute ready-to-run executables of Open Source Software packages in Backup File Format (BFF), others in TAR format; normally they are given a file name extension of .bff and .tar, respectively. The BFF format is the standard installation packaging format for AIX. Additional information about the individual files delivered in the product are included inside the BFF package. This packaging is named Licensed Product Package (LPP).

Some examples of Open Source Software Web sites where a wide range of precompiled and packaged software for AIX can be downloaded from are:

Group Bull	http://www.bullfreeware.com
UCLA	http://aixpdslib.seas.ucla.edu/aixpdslib.html

For more information on how to obtain Open Source Software in source form, please refer to Chapter 3, “Porting Open Source Software to AIX” on page 57.

2.2 Design of the Toolbox

The Toolbox was designed to provide the best performance possible on AIX Version 4.3.3 as well as AIX 5L. All of the elements of the Toolbox were compiled as native AIX applications, with little or no porting of the original source code. This was done using standard Open Source tools, such as **`autoconf`** and **`automake`**, and was helped by the high affinity that AIX has with Linux APIs.

AIX 5L has been enhanced to include more Linux-compatible APIs, which were not available in previous versions of AIX. This has added more compatibility between the two operating systems, resulting in a higher degree of Linux application affinity.

2.2.1 Toolbox FTP site directory structure

The Toolbox FTP site directory structure can change, but the following is a short description of how it is currently organized.

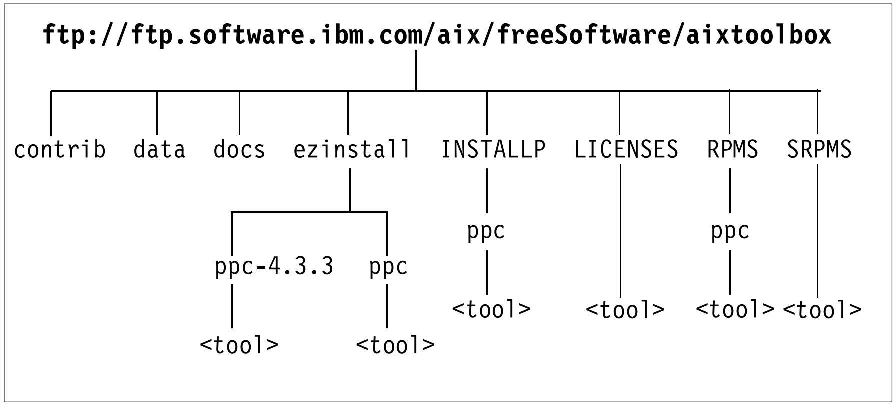

Figure 2-1 Toolbox Web site hierarchy

The current structure is:

contrib In the contrib directory you can find various scripts for installing and maintaining your Toolbox installation. In "contrib directory" on page 41 we explain some of the scripts that are available from this directory on the Toolbox Web site.

data In the data directory you find some documents regarding the available tools.

docs In the docs directory you find various documents regarding the *LSB APIs versus AIX APIs*. For differences between Toolbox commands and utilities compared to the same supplied with AIX, please refer to the following Web page for more information:
`ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/docs/index.html`

ezinstall In the ezinstall directory you find scripts that will allow you to download some collections of tool packages in a single batch to ease installation by acquiring the prerequisites for the desired tool at the same time as the main tool. In "ezinstall directory" on page 42 we explain some of the scripts that are available from this directory on the Toolbox Web site.

INSTALLP The INSTALLP directory is mainly for AIX 4.3.3 systems that do not have the `rpm` tool installed or systems that do not have it obtainable from the normal AIX 5L distribution media (CD-ROM).

LICENSES The LICENSES directory contains the licenses for all of the tools on the Toolbox Web site.

RPMS The RPMS directory contains subdirectories for each tools package (some tools may have several versions) in a ready-to-run form (executable or binary RPM). This is the top directory from where you will download the Toolbox binary RPM packages.

SRPMS The SRPMS directory contains subdirectories for each individual tools source RPM package (some tools may have several versions).

2.2.2 Creation of the Toolbox

The Toolbox contents were generally created on an AIX 4.3.3.0 system with no modified system headers. Most C applications were first built with the gcc compiler packaged with the Toolbox, then rebuilt with the Visual Age C Compiler for performance considerations.

Most C++ libraries and applications were built using g++ since C++ environments cannot generally be mix-and-match. g++ is used since it is the most commonly used C++ compiler in the Open Source community. Note that in future versions of the Toolbox C++ packages, the usage of the g++ compiler might change to the Visual Age C++ compiler.

> **Note:** GNU C and C++ binaries are significantly larger in size and, in some measured instances, they also execute slower than binaries created from the same source by using the Visual Age C and C++ compiler from IBM.

2.2.3 Toolbox RPM packaging format

The tools and applications that come with the Toolbox are all in RPM format. The RPM Package Manager is an open packaging system that can work on Linux systems and other UNIX-based systems. Originally it was developed as a tool by the Linux distributor Red Hat, and it is now an Open Source Software. It is easy to use and provides many features for installing, uninstalling, upgrading, deleting, and building packages.

The RPM Package Manager

The RPM Package Manager maintains a database of all installed packages and their corresponding files. It also stores information on all the packages that are installed or upgraded on the system. The database also reflects the configuration of the system on which it resides; thus, it could easily check if the RPM database has become corrupted or if the system configuration has changed. Normally the RPM database files are located in the /var/opt/freeware/lib/rpm directory.

The RPM Package Manager makes the process of building a package and distributing the software easy by taking the source code of the software and packaging it into source or binary form.

With the RPM database feature, you can perform queries and verification of the installed RPM packages in your system and determine what package a certain file belongs to.

Object Data Manager on AIX

AIX 5L and previous AIX versions normally store their installation information in Object Data Manager (ODM) databases. The ODM is a collection of databases intended for storing system information. Information is stored and maintained as objects with associated characteristics.

The ODM is also used to manage Vital Product Data (VPD) of application programs for installation and update procedures. Normally the ODM database files are located in the /etc/objrepos or /usr/lib/objrepos directories or in databases pointed to by the ODMDIR environment variable (set in the /etc/environment file to point to /etc/objrepos).

The standard AIX installation program **installp** updates the ODM VPD when installing AIX BFF images, much in the same way that is done with the **rpm** command and the RPM installation database.

RPM and ODM VPD integration on AIX

When you install a RPM package, all files in the package are recorded in the RPM database. That way, if any other packages require that library or shell, RPM knows that the library is available on the system because the database includes it. But all of the libraries and shells in the base AIX OS are installed by the **installp** command, so RPM does not know about them. They are not listed in the RPM database because they were not installed as part of any RPM.

To include the information about the already installed libraries and shells in AIX, the concept of a *virtual package* is used in the Toolbox.

The virtual package AIX-rpm is created and contains all libraries and shells from the base AIX OS. This is done by executing the **updtvpkg** script that determines what libraries and shells are provided by AIX, then it creates the virtual package so that RPM can record all of the system libraries in its own database. That way, a package that depends on a library, such as "/usr/lib/libc.a", will install well because RPM sees that the library is available.

For additional information on how to use the **updtvpkg** command please refer to “How to update the AIX-rpm with updtvpkg” on page 39.

RPM package types

The Toolbox software packages can be installed in either ready-to-run executables and configuration files (RPM) or as source code (SRPM). Both package types use the RPM Package Manager format.

A Source RPM (SRPM) does not contain compiled binaries, but instead contains the sources that a binary package can been built from. The SRPM packages in the Toolbox are marked by the file extension src.rpm. This source package file is an archive that contains the original compressed tar file(s) with source code, patches, and specification file(s).

The binary package file contains all files that make up the application, along with additional information that is needed to install, upgrade, and erase it. A binary RPM can be installed by using the **rpm** command without needing to do any recompilations.

SRPMs are important if you want to rebuild a RPM package for whatever reason. Rebuilding a SRPM file does not mean that the binary RPM package has been or will be installed on the particular system. To actually use the application, you have to install the binary RPM package that is being produced, during the rebuilding of the SRPM.

RPM package labeling

You can search through the entire database for packages or just certain files to get information about the system. Identification of these packages is done using package labels. Each label contains information that uniquely identifies the package. Even if the package *file* is renamed, the new file name will not confuse the RPM Package Manager because the package label is within the content of the package file. However, to ease management of packages on a file level, the RPM package naming convention shown in Figure 2-2 on page 27 is normally used.

Figure 2-2 Sample of RPM package naming convention

The three components in each package file name should be:

The software name All RPM package file names start with the software name. This may be derived from the application name or a description of the related programs grouped together in one package.

The software version This is an identifier that states the version of the packaged software.

The package release This is the most specific part of the package label, which shows the number of times the package has been rebuilt with the same software version. Rebuilds are normally done due to bugs uncovered after initial packaging or during use.

2.2.4 Toolbox directory structure

In AIX 5L, the /opt file system is created in the rootvg volume group and could be enlarged so that there will be enough space for installing Toolbox packages. The amount depends entirely on the amount and type of packages you wish to install.

For AIX 4.3.3 it is recommended that you create a separate file system for the /opt/freeware directory prior to Toolbox installation, or just create the /opt file system like it is done in AIX 5L (make sure you copy the existing content, if any, of the /opt directory to the /opt file systems logical volume prior to mounting it over /opt). However, avoid creating hierarchically dependent file systems since this can cause problems if you later will use NFS and `automounter` to mount the file systems from other host systems.

> **Note:** Make sure that the /var file system has enough free space after you have installed the Toolbox packages. Many Open Source applications use the /var file system to store temporary as well as permanent data.
>
> Also, make sure the /var/opt/freeware/tmp directory exists and is symbolically linked to /var/tmp.

The /opt/freeware directory will store the software packages you decide to install. When the Toolbox is installed on an AIX system, new directories and files are created and some library links are added:

AIX 5L On AIX 5L systems the /usr/opt/freeware directory is a symbolic link to the /opt/freeware directory.

AIX 4.3.3 On AIX 4.3.3 systems, under certain conditions, where /opt would normally be a part of the root (/) file system and no /opt/freeware file system has been created, /opt/freeware will be created as a symbolic link to /usr/opt/freeware in order to avoid filling the root file system.

The Linux binaries and libraries installed from the Toolbox will be placed in the /opt/freeware/bin and /opt/freeware/lib directories, with links being added to /usr/bin, /usr/linux/bin, /usr/lib, and /usr/linux/lib. This structure is set up in a way that avoids conflicts with AIX binaries and libraries. In some cases where the added Toolbox command has the same name as an existing AIX command, then no links are provided in /usr/bin, but are instead provided in /usr/linux/bin.

Figure 2-3 shows the directory structure after the Toolbox installation.

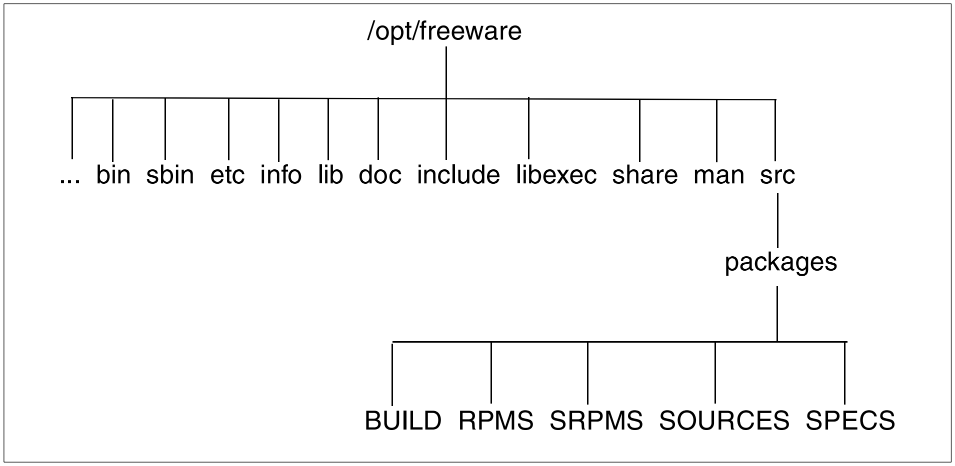

Figure 2-3 The /opt/freeware tree

The following is a short description of the /opt/freeware tree:

bin Primary directory of binary commands that may be used by both the administrator and users.

sbin Primary directory of binary commands that may be used by the system administrator.

etc Contains symbolic links to /etc.

info GNU information system's primary directory.

lib Contains shared libraries used by the Toolbox applications. It also contains object libraries, compiler program binaries, and other libraries.

doc Contains miscellaneous documentation.

include Contains include files for the Toolbox.

libexec Contains support programs and libraries for a particular set of programs that are not meant to be executed or linked directly by other applications.

share Contains architecture-independent files, such as timezone and terminfo information.

man Manual pages are organized into user programs (man1), library functions and subroutines (man3), file formats (man5), and system administration pages (man8).

src/packages The packages source directory is organized into binary RPM packages (RPMS), source RPM packages (SRPMS), source code and usually zipped TAR packages (SOURCES), RPM specification files (SPECS), and the directory where the building of RPM packages is done by the rpm command (BUILD). Please refer to Chapter 3, "Porting Open Source Software to AIX" on page 57.

2.2.5 How to enable the use of the Toolbox commands

To execute the Linux version of the command (the Toolbox version) after it is installed, you can either:

- Call it with its relative or absolute path.
- Create an alias for the command name.
- Change the PATH variable to have /usr/linux/bin in the beginning of the PATH.

Note: Changing the PATH variable may cause conflicts with and malfunctions in some AIX applications, specifically SMIT. It might be necessary to change the PATH, depending on the tasks to be performed.

Do not change the PATH environment variable in /etc/environment, /etc/profile, or any other user environment file that is used by applications started by **`init`** at IPL (boot-time).

For system administrators it is not recommended to set or use the /usr/linux/bin prior to AIX directories in the PATH environment variable at login time. Use an environment loading script that can be run manually or by specific applications, such as **`aixterm`** (using the ENV variable mechanism in Korn shell), or alias the desired Toolbox commands.

For users and developers, please use your own preference.

Using relative or absolute path

To use a Toolbox RPM package command with its relative or absolute path, you must of course know where it is stored in your file system hierarchy. In the next two examples we use the **`/usr/linux/bin/ls`** command from the fileutils RPM package downloaded from the Toolbox Web site, and our current directory is /home/work. Installation will be covered in greater detail in 2.3.3, “How to install and manage the Toolbox RPM packages” on page 36.

The first example is using the *absolute* path to the **`ls`** command with the --color option:

```
/usr/linux/bin/ls --color
```

The second example is using the *relative* path to the **`ls`** command with the --help option (our current directory is /home/work):

```
../../../usr/linux/bin/ls --help
```

Using PATH search preference

To have the Toolbox RPM package commands to be found first by the running Korn shell (or a similar shell), we can use the PATH variable and point it to /usr/linux/bin before the other directories to be searched. Below we show how to set the PATH variable in the current Korn shell environment:

```
export PATH=/usr/linux/bin:$PATH
```

In Example 2-1 on page 31, first we use the AIX **`nl`** command, then export the new PATH environment variable, pointing to /usr/linux/bin first, and then use **`nl`** again (/usr/linux/bin/nl is a symbolic link to /opt/freeware/bin/nl). As you can see from the output, it is two different commands that are used in each instance.

Example 2-1 Using the PATH environment variable to search for command

```
        0       print $PATH
                       455
        0       nl -?
  06!7  /!80     0     0 9
:   0  0;  0%  <0;  0=      50=      ><0;  0%  <0;  0%  <
00000000;  0$    <0;  0$    <0;  0?    <0;  <0;  0       <
00000000;  0$    <0;  0$    <0;?  <

        0       export PATH=/usr/linux/bin:$PATH
        0       print $PATH
                               455
        0       nl -?
          0     0    0  09
%  0@           0    A0  0   0
```

Using command aliasing

To set a command alias for a single command so that the Toolbox version is used instead of the one supplied with AIX, use the alias built-in function in the Korn shell (similar mechanisms can be found in other shells), as shown in the following syntax example:

```
      0  command2absolute path to command, with options if any
```

The following example shows how to create an alias for the **rm** command and point the alias definition to the **/usr/linux/bin/rm** command:

```
      0  2
```

In Example 2-2, first we use the AIX **nl** command, then alias **nl** to point to /usr/linux/bin/nl, and then use **nl** again (/usr/linux/bin/nl is a symbolic link to /opt/freeware/bin/nl). As you can see from the output, it is two different commands that are used in each instance.

Example 2-2 alias command

```
        0       print $PATH
                       455
        0       nl -?
  06!7  /!80     0     0 9
:   0  0;  0%  <0;  0=      50=      ><0;  0%  <0;  0%  <
00000000;  0$    <0;  0$    <0;  0?    <0;  <0;  0       <
00000000;  0$    <0;  0$    <0;?  <

        0       alias nl=/         /bin/nl
        0       nl -?
          0     0    0  09
```

```
Try `/usr/linux/bin/nl --help' for more information.
```

How to enable access to the Toolbox man pages

To access the man pages of the installed Toolbox applications, add /opt/freeware/man to your MANPATH variable. The MANPATH variable tells the **man** command where to search for information about commands. Below we show how to set the MANPATH variable in the current Korn shell environment:

```
export MANPATH=$MANPATH:/opt/freeware/man
```

In Example 2-3, we want to look at the manual page for the **ls** command. Since the AIX **man** command uses the native manual pages, if installed, even if the MANPATH environment variable is not set, we get the standard AIX manual page for the **ls** command. After setting the MANPATH environment variable to point to the Toolbox man directory, we get the Toolbox manual page for the **ls** command, when using the **man ls** command the second time.

Example 2-3 man command using MANPATH

```
root@fenris:/: man ls

                         Commands Reference, Volume 3

ls Command

Purpose

Displays the contents of a directory.

Syntax

To Display Contents of Directory or Name of File
ls [ -1 ] [ -A ] [ -C ] [ -F ] [ -L ] [ -N ] [ -R ] [ -a ] [ -b ] [ -c ] [ -d ]
[ -e ] [ -f ] [ -g ] [ -i ] [ -l ] [ -m ] [ -n ] [ -o ] [ -p ] [ -q ] [ -r ] [
-s ] [ -t ] [ -u ] [ -x ] [ File ... ]
...(lines omitted)...

root@fenris:/: export MANPATH=$MANPATH:/opt/freeware/man
root@fenris:/: man ls

     LS(1)            ls (fileutils) 4.1 (April 2001)            LS(1)

     NAME
          ls - list directory contents

     SYNOPSIS
          ls [OPTION]... [FILE]...
```

```
...(lines omitted)...
```

2.3 Installing the Toolbox RPM packages

In order to install Toolbox packages, the RPM Package Manager must be installed first. This is part of the default system installation for AIX 5L. For AIX 4.3.3, this package should be installed using the **installp** command directly or through the Systems Management Interface Tool (SMIT).

The installation of the Toolbox requires AIX 4.3.3 or later. You can use either AIX 5L or AIX 4.3.3. Use the **oslevel** command to determine the current installation level.

If you have AIX 4.3.3 with a lower Maintenance Level than ML8, you need to update the system with APAR IY15017. You can search for and download the IY15017 fix from:

http://techsupport.services.ibm.com/server/support

Follow the instructions for downloading and installing the fix. You could also check for the latest available maintenance level at:

http://techsupport.services.ibm.com/rs6k/fixdb.html

2.3.1 How to install rpm.rte

The following is a short description of the steps you need to take to install the RPM package.

First check if the rpm.rte is installed on your system by using the following **lslpp** command:

```
lslpp -L rpm.rte
```

To manually install the rpm.rte fileset, use the following **installp** command:

```
installp -d rpm.rte all
```

The following is the desired result from the installation (displayed in the end of the output from the **installp** command if it is successful):

```
Name                            Level          Part        Event       Result
-----------------------------------------------------------------------------
rpm.rte                         3.0.5.32       USR         APPLY       SUCCESS
rpm.rte                         3.0.5.32       ROOT        APPLY       SUCCESS
```

How to download the rpm.rte package

If you need the rpm.rte package and it cannot be found on any media available to you, you can download it from the Toolbox FTP server:

ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/INSTALLP/ppc/rpm.rte

It is also possible to download it by searching for rpm.rte on the following Web site:

http://techsupport.services.ibm.com/rs6k/fixdb.html

Remember to select **Fileset Name** and type `rpm.rte` in the search box in the AIX Fix Distribution Service Web form. Currently the version is 32 from this Web site:

ftp://techsupport.services.ibm.com/aix/fixes/v4/other/rpm.rte.3.0.5.32.bff

You can also obtain the IY15017 fix from the same Web site. Currently the version is 27 from this Web site:

ftp://techsupport.services.ibm.com/aix/fixes/v4/X11/X11.base.lib.4.3.3.27.b
ff

2.3.2 How to download Toolbox packages

To download Toolbox packages, you can select the individual packages from the Toolbox CD-ROM, or from the following URL in any Web browser:

http://www.ibm.com/servers/aix/products/aixos/linux/download.html

However, if you want to download several, or even all, packages at the same time, you should consider creating some FTP or HTTP script to do it. If you install the **`ncftp`** or **`wget`** packages first, you can use either of them to download the rest in one batch, supervised or unsupervised.

Note: Cryptographic Content (SSL) for certain Toolbox packages can be obtained from the following Web site:

http://www6.software.ibm.com/dl/aixtbx/aixtbx-p

On AIX 5L the SSH packages are now part of the AIX Bonus Pack CD-ROM distribution. Please refer to the following Web site for additional information:

http://www.ibm.com/servers/aix/products/bonuspack/aix5l/bpcontent.html

FTP tools for batch download

wget is a command line tool that retrieves and recursively downloads files from the Web using the HTTP or FTP protocols. For more information please refer to the following Web site:

http://www.wget.org

ncftp is a replacement for the **ftp** command. It has a lot of usability enhancements, such as recursive file transfer from a hierarchical root directory on the remote FTP server. For more information please refer to the following Web site:

http://www.ncftp.com/ncftp

How to install and use the wget package

If you decide to use the **wget** tool, and do not have the corresponding RPM package yet, download it from:

ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/wget

Install it with the install option to the **rpm** command. The following example shows how to install the **wget** package (assuming there is only one **wget** RPM package in the current directory):

```
rpm -i wget*.rpm
```

You can now use the **wget** tool to recursively download other packages. To download all RPMs from the Toolbox Web site you would issue the following **wget** command:

```
wget -r ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/
```

How to install and use the ncftp package

If you decide to use the **ncftp** tool, and do not have the corresponding RPM package yet, download it from:

ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/ncftp

Install it with the install option to the **rpm** command. The following example shows how to install the **ncftp** package (assuming there is only one **ncftp** RPM package in the current directory):

```
rpm -i ncftp*.rpm
```

You can now use the **ncftp** tools to download other packages. To download all RPMs from the Toolbox Web site you would issue the following **ncftpget** command (on a single command line):

```
ncftpget -R
ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/
```

With the **ncftp** command, you can use standard FTP commands like **dir** or **get** (with automatic **reget**, in case the connection should end unexpectedly and you need to issue the **get** command again). **ncftp** also offers enhancements like word completion (press the Tab key) and retrieval of whole directory trees with **get -R**:

```
ncftp> get -R RPMS
```

Another way is to pipe the commands to the **ncftp** command. The following example shows the same **get -R**, but piped to **ncftp** standard input (on a single command line):

```
print get -R|ncftp
ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/
```

More complex commands can be enclosed with the **print** or **echo** commands with quotation marks ("or").

2.3.3 How to install and manage the Toolbox RPM packages

The installation for the Toolbox packages on AIX is performed the same way that RPM packages are installed on Linux, by using the **rpm** command.

On AIX 5L you can also use the **geninstall** command and the System Management Interface Tool (SMIT) to install RPM packages.

For more information on the **rpm** command, please refer to the **rpm** man page, **rpm --help**, or the following Web site:

http://www.rpm.org

For more information on the **geninstall** command, please refer to the following Web site:

http://www.ibm.com/servers/aix/library

The basic **rpm** command line options are:

-i	Install RPM package.
-e	Uninstall (erase) installed RPM package.
-q	Query installed RPM package.
-V	Verify installed RPM package.
-b	Build RPM packages.
--help	Usage information for the **rpm** command.

In the following sections we will describe the usage of the Install, Uninstall, Query, and Verify options of the **rpm** command. For more information on the Build options, please refer to Chapter 3, “Porting Open Source Software to AIX” on page 57.

2.3.4 How to install packages using the rpm command

To install Toolbox packages using the RPM-specific installation method, use the **rpm** command as shown below:

```
rpm -i package filename name
```

> **Note:** After installing SRPM packages, remember that you will find the SPEC file in the /opt/freeware/src/packages/SPECS directory, and the compressed TAR file with the source code in the /opt/freeware/src/packages/SOURCES directory. For more information please refer to Chapter 3, “Porting Open Source Software to AIX” on page 57.

The following example shows that the **rpm** command is true to the traditional UNIX way of progress and user feedback when everything works as it should (that is, no feedback what so ever).

Example 2-4 RPM package installation

```
root@fenris:/images/RPMS/zoo-2.10: rpm -i zoo-2.10-4.aix4.3.ppc.rpm
root@fenris:/images/RPMS/zoo-2.10:
```

For the previous and the next few examples we use the zoo compression utility package to demonstrate the installation procedure. To do this on your own system you would have to either uninstall the package between installations with the -e option or use the --force option to the installation command.

> **Note:** Although most RPM packages in the Toolbox are named aix4.3, this only means that the package (using the OSS package naming convention) requires *AIX 4.3.3 or later*, so they are equally suitable for AIX 5L, unless there are specific packages named aix5.1 or aix5.2.

Since RPM includes simple FTP and HTTP clients to simplify installing and querying packages that are available over the Internet, we will use this way of installing in the next examples. If we have the bandwidth, we do not need to set aside much storage for the RPM packages on all systems where we want to install the Toolbox RPM packages.

To install a Toolbox RPM package directly from the Toolbox FTP site use the following **rpm** command syntax:

```
rpm -i ftp://user:password@hostname:port/package path/package.rpm
```

If the :*password* portion is omitted, the password will be prompted for (once per *user@hostname* pair). If both the user and password are omitted, anonymous FTP is used.

> **Note:** When using **rpm** to install directly from the Toolbox FTP server, **rpm** temporarily stores the downloaded files in the /var/tmp directory. If this is too small, the **rpm** installation will fail and show error messages. After failed installations, **rpm** does not always remove the temporary files from the /var/tmp directory, so this has to be done manually with the **rm** command.
>
> Also, make sure that the /var/opt/freeware/tmp directory exists and is symbolically linked to /var/tmp.

The following three examples show the same installation of the zoo package from the Toolbox FTP site: Example 2-5 with the -ih install options (h prints 50 hash marks as the package archive is unpacked); Example 2-6 with the -iv option for some additional information; and Example 2-7 on page 39 with the -ivv option for much more information about the installation process.

Example 2-5 Using RPM to install from the Toolbox Web site with -ih option

```
root@fenris:/: rpm -ih
ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/zoo-2.10/zoo-2.
10-4.aix4.3.ppc.rpm
##################################################
```

Example 2-6 uses the -iv option and there is some information that reports that the installation has completed. Since we used the FTP protocol to transfer the RPM package, we are informed of this.

Example 2-6 Using RPM to install from the Toolbox Web site with -iv option

```
root@fenris:/: rpm -iv
ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/zoo-2.10/zoo-2.
10-4.aix4.3.ppc.rpm
Retrieving
ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/zoo-2.10/zoo-2.
10-4.aix4.3.ppc.rpm
zoo-2.10-4
```

Notice the FTP transfer in the beginning and the installation statistics at the end of the Example 2-7 on page 39 output. As you can see, the information that **rpm** displays with the -vv option is plentiful.

Example 2-7 Using RPM to install from the Toolbox Web site with -ivv option

```
root@fenris:/: rpm -ivv
ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/zoo-2.10/zoo-2.
10-4.aix4.3.ppc.rpm
D: counting packages to install
D: found 1 packages
D: looking for packages to download
Retrieving
ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/zoo-2.10/zoo-2.
10-4.aix4.3.ppc.rpm
D:  ... as /var/opt/freeware/tmp/rpm-xfer.7HnfMa
D: logging into ftp.software.ibm.com as ftp, pw (username)
D: retrieved 1 packages
D: New Header signature
D: Signature size: 68
D: Signature pad : 4
D: sigsize       : 72
D: Header + Archive: 101744
D: expected size   : 101744
D: opening database mode 0x102 in /var/opt/freeware/lib/rpm
D: found 0 source and 1 binary packages
D:  requires: libc.a(shr.o)  satisfied by db provides.
D: installing binary packages
D: New Header signature
D: Signature size: 68
D: Signature pad : 4
D: sigsize       : 72
D: Header + Archive: 101744
D: expected size   : 101744
D: package: zoo-2.10-4 files test = 0
D:    file: /opt/freeware/bin/fiz action: create
D:    file: /opt/freeware/bin/zoo action: create
D:    file: /opt/freeware/doc/zoo-2.10 action: create
D:    file: /opt/freeware/doc/zoo-2.10/Copyright action: create
D:    file: /opt/freeware/man/man1/fiz.1 action: create
D:    file: /opt/freeware/man/man1/zoo.1 action: create
D:    file: /usr/bin/fiz action: create
D:    file: /usr/bin/zoo action: create
D: running preinstall script (if any)
zoo-2.10-4
GZDIO:      96 reads,   251588 total bytes in 0.001 secs#
D: running postinstall scripts (if any)
```

How to update the AIX-rpm with updtvpkg

The **rpm** command generally handles dependencies between packages by auto-detecting shared library requirements. When the rpm.rte package is installed on AIX, the **/usr/sbin/updtvpkg** command is executed to create a *virtual package* called AIX-rpm (previously named SysProvides). The AIX-rpm catalogues all the known system libraries and shells, like /bin/sh and libc.a(shr.o), that have been installed by the **installp** program.

> **Note:** After installing or uninstalling LPPs (or other non-RPM packaged software) that contain API libraries or additional system shells, execute the **updtvpkg** command to synchronize the AIX-rpm package with what is actually installed on the system. There will be rare circumstances in a production system when this will be needed (most libraries and shells are installed during base installation of AIX). This applies to both AIX 4.3.3. and AIX 5L.

To show that the virtual image is installed use the **rpm** query command:

```
rpm -q AIX-rpm
```

To see what libraries and shells the virtual image has catalogued, use the following **rpm** query command:

```
rpm -q --provides AIX-rpm
```

To recreate the AIX-rpm virtual package, you can rerun the **updtvpkg** command as the *root* user. The more software and the older the hardware, the longer this process will take. But be patient; if nothing is wrong with your installation it will finish.

Example 2-8 Recreating the AIX-rpm virtual package with updtvpkg

```
root@fenris:/: updtvpkg
Please wait...
```

In the *rare cases* that the updtvpkg script does not finish, abort it (CTRL+C or **kill -15**) and review your VPD with the **lslpp** command. Check for file sets that are incompletely installed or BROKEN; check your AIX maintenance level and fix levels. Correct any problems, clean up the fileset installation with the **installp -C** command, and ensure that you have the proper AIX maintenance levels and required fixes.

You can also run the updtvpkg script in debug mode with **ksh -xv /usr/bin/updtvpkg**. If necessary, you can also add a `set -x` line after the start of the update_virtual_package function in the updtvpkg script. The main part of the updtvpkg script is the following command line pipe:

```
/usr/bin/lslpp -Ou -qfc | /usr/bin/cut -d':' -f3 |
/usr/bin/awk '{print $1}' | /usr/lib/rpm/find-provides
```

Try running this part of the script by hand to verify its output. If you are still stuck, use the AIX Toolbox mailing list and ask if someone has experienced the same problem.

How to use the installation scripts from the Toolbox Web site

On the Toolbox FTP site you can find a couple of very useful installation scripts to manage your Toolbox RPM package installation. You will find the installation scripts in the following directories on the Toolbox FTP site (`ftp://ftp.software.ibm.com`):

contrib /aix/freeSoftware/aixtoolbox/contrib

ezinstall /aix/freeSoftware/aixtoolbox/ezinstall

To download the scripts to your system, you can use the **ncftpget** command (as shown in Example 2-9) for downloading the entire contrib directory (including subdirectories, if there are any) from the Toolbox Web site.

Example 2-9 How to download the contrib or ezinstall directory from the Toolbox site

```
root@fenris:/images/tbox: ncftpget -R
ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/contrib
contrib/destroyRPMS:                                      2.75 kB   15.04 kB/s
contrib/installmissing.sh:                                2.45 kB   45.11 kB/s
contrib/installnewer:                                    11.40 kB  127.06 kB/s
contrib/installremoved.sh:                                5.62 kB   25.21 kB/s
contrib/kdeclean:                                       300.00 B     5.08 kB/s
contrib/outofdate.pl:                                     0.00 B     0.00 B/s
contrib/update_tbox:                                     25.14 kB  246.05 kB/s
contrib/xinitrc.sample:                                   1.65 kB    7.97 kB/s
```

contrib directory

In the contrib directory you will find the following installation scripts:

update_tbox This is an interactive script that finds all packages on the client's computer that need to be updated or installed, and gives you the option to do either an update or fresh install interactively or automatically. It downloads "serverlist" and "toolbox_dependencies" from the AIX Toolbox public site to figure out what is available and what dependencies are needed for each package. It does not query or update packages from the crypto site, only the main AIX Toolbox site. The **wget** command is needed by the script; it exits if **wget** is missing.

installnewer Given the RPM images installed on the system, examines the specified directory for newer versions of RPMS, and installs them along with any new prerequisite packages.

	Note that this script has some RPM package dependencies hardcoded, so the installation might fail if you are using an out-of-date version.
installmissing.sh	Attempts to install all RPMS from a given directory (containing RPM images) that are not already installed.
installremoved.sh	Attempts to install RPMS, which matches previously installed versions, that was previously removed by the destroyRPMS script.
destroyRPMS	Uninstalls all RPM images on the system. Removes the RPM database and rpm.rte, as well (unless KEEPRPM is set in the environment).

ezinstall directory

In the ezinstall/ppc directory you will find the following installation scripts:

getapp-dev.sh	Script to download app-dev bundle - Application Development Tools
getbase.sh	Script to download base bundle - Base Linux Affinity Support
getdesktop.base.sh	Script to download desktop.base bundle - Common Support Programs for Gnome and KDE
getgnome.apps.sh	Script to download gnome.apps bundle - Gnome Desktop Applications
getgnome.base.sh	Script to download gnome.base bundle - Gnome Desktop Base
getkde2.all.sh	Script to download kde.all bundle - KDE Desktop Base and Applications
getkde2.base.sh	Script to download kde.base bundle - KDE Desktop Base
getkde2.opt.sh	Script to download kde.opt bundle - KDE Desktop Applications

Note: Now it would be a good time to install all the tools and packages that you are interested in. You might find additional ideas in Chapter 6, “Tools in the Toolbox” on page 145.

2.3.5 How to uninstall packages using the rpm command

To uninstall or erase an installed package, use the -e option of the `rpm` command. The next example uninstalls a package (if no other packages are dependant upon its existence):

```
rpm -e package name
```

To remove all installed RPM packages (with their dependencies), you could use the destroyRPMS script from the contrib directory on the Toolbox Web site:

```
ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/contrib/destroyRPMS
```

To reinstall the RPM packages that were removed by the destroyRPMS script, use the installremoved.sh script from the contrib directory on the Toolbox Web site (`http://ftp.software.ibm.com`):

```
/aix/freeSoftware/aixtoolbox/contrib/installremoved.sh
```

2.3.6 How to verify installed packages using the rpm command

To verify installed packages (such as if there are broken dependencies to other packages) on the system, use the -V option with the **rpm** command. The first example verifies all installed packages with the -Va option:

```
rpm -Va
```

To only run verification on one single package, use the -V option:

```
rpm -V package name
```

The following **rpm** command verifies a specified RPM with extra verbose output by using two v options:

```
rpm -Vvv package name
```

2.3.7 How to query installed packages with the rpm command

To check which packages are installed on the system, use the following command:

```
rpm -qa
```

To check for specific packages using wildcards, you have to filter the output from the **rpm** command or use the AIX-specific **lslpp** command. The following examples show how to use **egrep**, **sed**, and **awk**, respectively, to search for all packages that start or end with the string “`ftp`”.

With **egrep**:

```
rpm -qa|egrep "^ftp|.*ftp-"
```

With **sed**:

```
rpm -qa|sed '/^ftp/p;/.*ftp-/p;d'
```

With **awk**:

```
rpm -qa|awk '/^ftp/||/.*ftp-/'
```

To make it easier, just create a command alias that takes the search value as an argument. The following example creates the **rpmls** alias (note the space between the **egrep** command and the end of the string):

```
alias rpmls='rpm -qa|egrep '
```

The **rpmls** command alias could then be used (as in the following example) to search for all package names that contain the string “ftp”:

```
rpmls ftp
```

Another option, instead of using the **rpm** command and **rpmls** alias, is to use the AIX **lslpp** command. To use the **lslpp** command, enclose the search string with wildcards in citation marks. In the following example, the **lslpp** command searches for all packages or filesets that start or end with the string “ftp” (note that the trailing dash used with the **rpm** command above should not be used with **lslpp** since it is not part of the RPM fileset name in the AIX software inventory):

```
lslpp -L “ftp*” “*ftp“
```

2.3.8 How to find the installed package a file belongs to

To find the installed package that a specific file belongs to, use the f option of the **rpm** query command:

```
rpm -qf filename
```

The next example shows how to find out which package a specific file, that we have installed on our system belongs to. In this case it is the GNU C compiler that we have previously installed.

Example 2-10 Finding which RPM package a file belongs to

```
root@fenris:/: rpm -qf /usr/bin/gcc
gcc-2.9.aix51.020209-1
```

2.3.9 How to find the uninstalled package a file belongs to

To find the uninstalled package that a specific file belongs to, you need to check each and every RPM package file that you have stored on your system or can find on the Web.

The following command will work in directories and subdirectories containing hundreds and thousands of RPM packages. However, the more files to search, the longer time it will take. Just substitute *filename* for the name of the file you are looking for:

```
find . -name '*.rpm'|
while read RPM_FILENAME; do
   rpm -qlp $RPM_FILENAME|grep filename && echo $f
done
```

If you do not find the RPM package that a specific file, such as a prerequisite file, belongs to on your system, you could use the search facility at the following Web site to search for it:

http://rpmfind.net/linux/rpm2html/search.php

Just go to the Web site, type the file name in the search box, and submit the query.

2.3.10 How to find the prerequisites for a package

To find the prerequisites for a RPM package, use the pR option to the **rpm** query command:

```
rpm -qpR package filename
```

The next example shows how to find out what the prerequisites are for a RPM package file that we have downloaded to our system. In this case it is the Samba server package that we have previously downloaded.

Example 2-11 Finding prerequisites for a RPM package

```
root@fenris:/: rpm -qpR samba-2.2.3a-2.aix4.3.ppc.rpm
samba-common = 2.2.3a
/bin/sh
/bin/sh
/usr/bin/perl
libc.a(shr.o)
libdl.a(shr.o)
```

In the example output in Example 2-11, we see that one prerequisite for this RPM is the samba-common package with Version 2.2.3a. The package also requires the /bin/sh and /usr/bin/perl programs, and the libc.a and libdl.a libraries in /usr/lib (/lib is a symbolic link to /usr/lib), or directories pointed to by the LIBPATH environment variable, if set.

2.3.11 How to query package information from the RPM file

To list the package information that is contained in an uninstalled RPM package, use the **rpm** query option pi in the following way:

```
rpm -qpi package filename
```

The next example shows how to find out more information about a RPM package file that we have downloaded to our system. In this case it is the GNU C compiler package that we have previously downloaded.

Example 2-12 Finding information about a package from the RPM file

```
root@fenris:/: rpm -qpi gcc-2.9.aix51.020209-1.aix4.3.ppc.rpm
Name        : gcc                Relocations: /opt/freeware
Version     : 2.9.aix51.020209   Vendor: (none)
Release     : 1                  Build Date: Wed Mar 20 12:28:40 CST 2002
Install date: (not installed)    Build Host: emperor.aixpclab.austin.ibm.com
Group       : Development/Tools  Source RPM: GNUPro-2.9.aix51.020209-1.src.rpm
Size        : 26069782           License: GPL
Summary     : The GNU gcc C compiler and headers
Description :
gcc is the GNU C compiler.  The gcc package contains the compiler and required
header files.
```

2.3.12 How to query package information from the RPM database

To list the package information for all installed packages, use the ai option to the **rpm** query command, as is shown in the next example:

```
rpm -qai
```

To list information about one installed package, use the following **rpm** command:

```
rpm -qi package name
```

Example 2-13 shows how to find out more information about a RPM package that we have installed in our system. In this case it is the GNU C compiler that we have previously installed.

Example 2-13 Finding information about a package from the RPM database

```
root@fenris:/: rpm -qi gcc
Name        : gcc                Relocations: /opt/freeware
Version     : 2.9.aix51.020209   Vendor: (none)
Release     : 1                  Build Date: Wed Mar 20 12:28:40 CST 2002
Install date: (not installed)    Build Host: emperor.aixpclab.austin.ibm.com
Group       : Development/Tools  Source RPM: GNUPro-2.9.aix51.020209-1.src.rpm
Size        : 26069782           License: GPL
Summary     : The GNU gcc C compiler and headers
Description :
gcc is the GNU C compiler.  The gcc package contains the compiler and required
header files.
```

2.3.13 How to query all files in uninstalled packages

To list all files that are contained in an uninstalled RPM package, use the **rpm** query option pl in the following way:

```
rpm -qpl package name
```

Example 2-14 shows how to list all files in a RPM package file that we have downloaded to our system. In this case it is the GNU C compiler package that we have previously downloaded.

Example 2-14 Finding all files that are included in a RPM package

```
root@fenris:/: rpm -qpl gcc-2.9.aix51.020209-1.aix4.3.ppc.rpm
/opt/freeware/GNUPro/COPYING
/opt/freeware/GNUPro/COPYING.LIB
/opt/freeware/GNUPro/COPYING.NEWLIB
/opt/freeware/GNUPro/CYGNUS
/opt/freeware/GNUPro/GNUPro.pdf
/opt/freeware/GNUPro/Install
/opt/freeware/GNUPro/bin/cpp
/opt/freeware/GNUPro/bin/gcc
/opt/freeware/GNUPro/bin/gcov
/opt/freeware/GNUPro/bin/powerpc-ibm-aix4.3.3.0-gcc
...(lines omitted)...
```

2.3.14 How to query all files in installed packages

To list all files from all installed packages, use the al option to the **rpm** query command, as is shown in the next example:

```
rpm -qal
```

To list all files from a single, installed package, use the following **rpm** command:

```
rpm -ql package name
```

Example 2-15 shows how to list all files in a RPM package that we have installed on our system. In this case it is the GNU C compiler package that we have previously installed.

Example 2-15 Finding all files that belong to a RPM package

```
root@fenris:/: rpm -ql gcc
/opt/freeware/GNUPro/COPYING
/opt/freeware/GNUPro/COPYING.LIB
/opt/freeware/GNUPro/COPYING.NEWLIB
/opt/freeware/GNUPro/CYGNUS
/opt/freeware/GNUPro/GNUPro.pdf
/opt/freeware/GNUPro/Install
```

```
/opt/freeware/GNUPro/bin/cpp
/opt/freeware/GNUPro/bin/gcc
/opt/freeware/GNUPro/bin/gcov
/opt/freeware/GNUPro/bin/powerpc-ibm-aix4.3.3.0-gcc
...(lines omitted)...
```

2.3.15 How to query specific information from the RPM database

To extract only specific information about packages with the query option, use the --queryformat, or the abbreviated --qf, option with the **rpm** command. The next example shows how to have only the name of each installed packaged reported:

```
rpm -qa --queryformat '%{NAME}\n'
```

Note the trailing '\n' (NewLine)[1] in the FORMAT specification. If this is left out, all information would come on the same line. The next example displays each package with its package name and the package version, but they will be separated with a colon (:):

```
rpm -qa --queryformat '%{NAME}:%{VERSION}\n'
```

Example 2-16 shows how to list the name and version number of all RPM packages that we have previously installed on our system, in alphabetical order.

Example 2-16 Using the --queryformat option with rpm -qa

```
root@fenris:/: rpm -qa --queryformat '%{NAME}:%{VERSION}\n'|sort
AIX-rpm:5.2.0.0
AfterStep:1.8.10
apache-manual:1.3.26
automake:1.5
bash:2.05a
binutils:2.9.aix51.020209
...(lines omitted)...
```

[1] In UNIX environments, each terminal output line, or text editor files, is normally terminated by a New Line (NL) character. This differs from Windows systems, which use a Carriage Return and New Line pair to terminate text editor lines.

All the queryformat tags can be found by using the --querytags option to the **rpm** command:

```
rpm --querytags
```

2.3.16 How to identify corrupt package files

One possible reason for a RPM installation to fail is if the package file is corrupt. You can usually spot this by using **rpm -ivv** instead of **rpm -i** when installing the package.

The first **rpm** command (in Example 2-17) results in an error message, but does not explain why the package could not be installed. The next **rpm** command shows that the expected and actual package size differs, which is the reason for the installation failure.

To fix the problem, download the package again from the Toolbox Web site and make sure that you use the BINARY or IMAGE, and not the ASCII, download method, and download and install the package again.

Example 2-17 Identifying corrupt RPM package files

```
# rpm -ivh kdebase*
error: kdebase-2.0.1-4.aix4.3.ppc.rpm cannot be installed

# rpm -ivv kdebase*
D: counting packages to install
D: found 1 packages
D: looking for packages to download
D: retrieved 0 packages
D: New Header signature
D: Signature size: 68
D: Signature pad : 4
D: sigsize        : 72
D: Header + Archive: 21639000
D: expected size   : 28631180
error: kdebase-2.0.1-4.aix4.3.ppc.rpm cannot be installed
D: found 0 source and 0 binary packages
```

2.3.17 How to extract files from a package

The **rpm** command uses a similar way as the GNU **cpio** command to archive files. However, RPM package files contain more than the CPIO archive with files, so you cannot use the **cpio** command on a RPM package file directly.

Use the **rpm2cpio** command to extract and create a GNU **cpio** readable CPIO archive file first. The **rpm2cpio** command uses either *stdin* or a specified file name to read the RPM package information from. Note that the **rpm2cpio** command does output the CPIO-formatted archive to *stdout*, so you need to redirect the output from the command to a file (do not use the package file name as the **cpio** archive file name).

In the following example, substitute *package filename* with the RPM package file name and *cpio filename* with the name of the package, or choose whatever name you like. The created CPIO formatted file will then be named *the name you choose*.cpio.

```
rpm2cpio package filename > cpio filename.cpio
```

Now you can use the **/usr/linux/bin/cpio** command to extract files from the *cpio filename*.cpio archive.

The following example shows how to use the GNU **cpio** command to list the content of the *cpio filename*.cpio file. Note that the GNU **cpio** command uses *stdin* to read the archive information unless the F flag is specified:

```
/usr/linux/bin/cpio -itvF cpio filename.cpio
```

To extract a specific file from the CPIO archive, remove the t option and append the path name for the desired file, as it is specified in the CPIO archive. Note that the d option (in Example 2-18) makes **cpio** create directories (if they do not exist already) for extracting the archive file path name:

```
/usr/linux/bin/cpio -idvF cpio filename.cpio archive file pathname
```

The next example shows how to use the **rpm2cpio** and **cpio** commands to extract a specific file from the zoo compression utility package. The --no-preserve-owner specifies that we do not want to keep the UID and GID from the archived file when it is stored in our file system, but that it should be replaced with the UID and GID from the extracting user instead.

Example 2-18 Using rpm2cpio and cpio to extract a specific file from a RPM package file

```
root@fenris:/: rpm2cpio zoo-2.10-4.aix4.3.ppc.rpm > zoo.cpio
root@fenris:/: /usr/linux/bin/cpio -itvF zoo.cpio
-rwxr-xr-x   1 snapp    staff       62096 Oct 31  2000 opt/freeware/bin/fiz
-rwxr-xr-x   1 snapp    staff      135536 Oct 31  2000 opt/freeware/bin/zoo
drwxr-xr-x   2 snapp    staff           0 Oct 31  2000 opt/freeware/doc/zoo-2.10
...(lines omitted)...

root@fenris:/: /usr/linux/bin/cpio -idv --no-preserve-owner -F zoo.cpio opt/freeware/bin/zoo
opt/freeware/bin/zoo
492 blocks
root@fenris:/: ls -l opt/freeware/bin/zoo
-rwxr-xr-x   1 root     system       135536 Oct 09 08:32 opt/freeware/bin/zoo
```

2.3.18 How to install RPM and BFF packages with SMIT or geninstall

On AIX 5L it is also possible to use the AIX-specific system administrative tool, the System Management Interface Tool (SMIT), in addition to the command line tools such as **rpm** and the AIX-specific **geninstall** command.

The following example shows how to install Toolbox RPM packages using the **geninstall** command:

```
geninstall -d package path package name | package filename
```

package path specifies the path to the directory where the package file is stored. The package name can be either the name of the package or the entire package file name, package file name.

The package name for the FTPCOPY package could be ftpcopy-0.3.9-1, and the file name would then be ftpcopy-0.3.9-1-aix4.3.ppc.rpm for the AIX 4.3 (and later) Power PC package.

If the FTPCOPY package file is available in the current directory, the **geninstall** command could be either the following, when using only the RPM package name:

```
geninstall -d . ftpcopy-0.3.9-1
```

Or it could be the following, by using the full RPM package file name:

```
geninstall -d . ftpcopy-0.3.9-1-aix4.3.ppc.rpm
```

How to check installed packages with the lslpp command

To check which packages are installed on the system, use the following command:

```
lslpp -L
```

RPM packages are marked with an `R` in the Type column. Normal AIX **installp** filesets are marked with an `F` in the Type column.

To check for specific packages using wildcards, enclose them in quotation marks (“ “). The following command searches for all packages or filesets that start or end with the string “`ftp`”:

```
lslpp -L “ftp*” “*ftp“
```

How to install packages with the smit command

The **smit** command will start the graphical user interface (GUI) version **msmit** if the **DISPLAY** variable is set; otherwise it will invoke **smitty**, the text based version. Refer to Figure 2-4 on page 52.

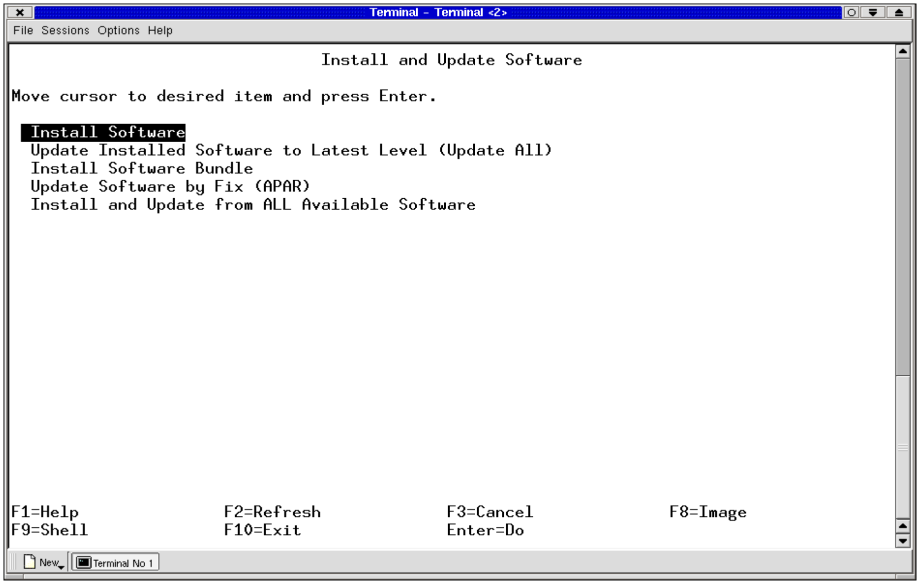

Figure 2-4 SMIT installp window (ASCII or text-based interface)

To get to the installation menu, simply type `smit` and choose the **Software Installation and Maintenance** option.

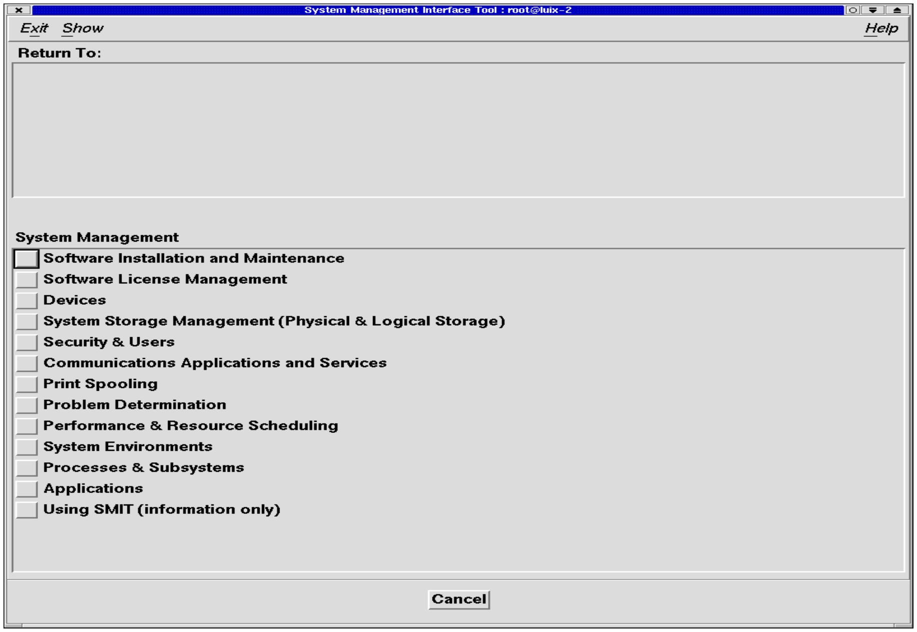

Figure 2-5 Main SMIT installation window (GUI interface)

It is also possible to get there using a *fast path*. A fast path is a shortcut method to display a menu directly from the command line. There is a fast path for each task/operation, such as managing the devices, security and users, applications, and more. The fast path will be displayed by pressing the F8 key in the desired menu screen.

The following shows the fast path for the installation of software:

```
smit install_latest
```

The Software Installation and Maintenance menu provides information that you can use for installing and updating software, and other tasks.

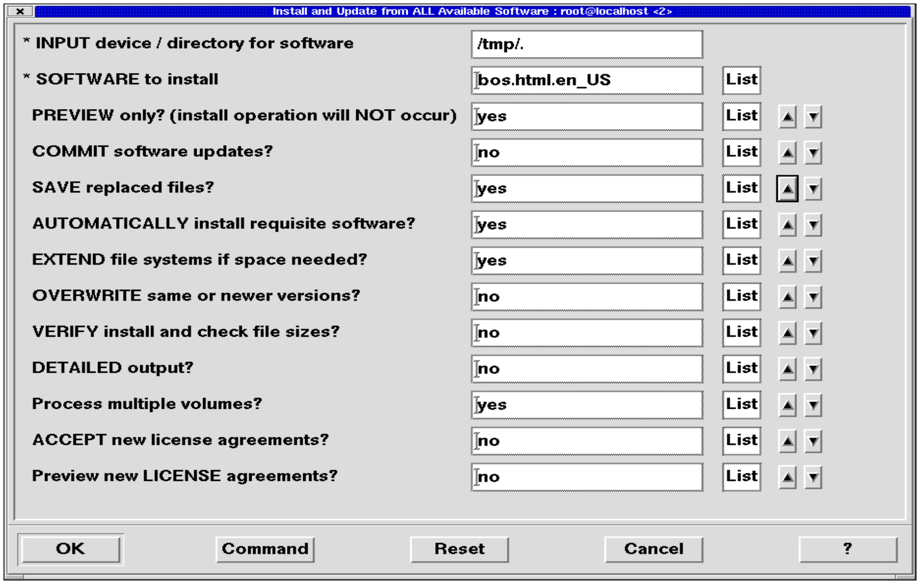

Figure 2-6 SMIT dialog window

The graphical interface for SMIT displays a hierarchy of menus. This was designed to simplify systems management tasks. There are several parts to the SMIT GUI:

Menu window Lower window of the primary SMIT screen. The functions available will be displayed in the menu bar and a list of menu items appears in the menu screen (refer to Figure 2-5 on page 53).

Path window Top window of the primary SMIT window. It shows menus that have been traversed to get to the current menu.

Dialog window A pop-up menu screen that appears each time a task is selected in the menu window. This is where you supply details of the task selected (refer to Figure 2-6).

Command output panel A display associated with the dialog screen when the **Do** button is selected. The output generated by the command will be displayed on this screen (refer to Figure 2-7 on page 55).

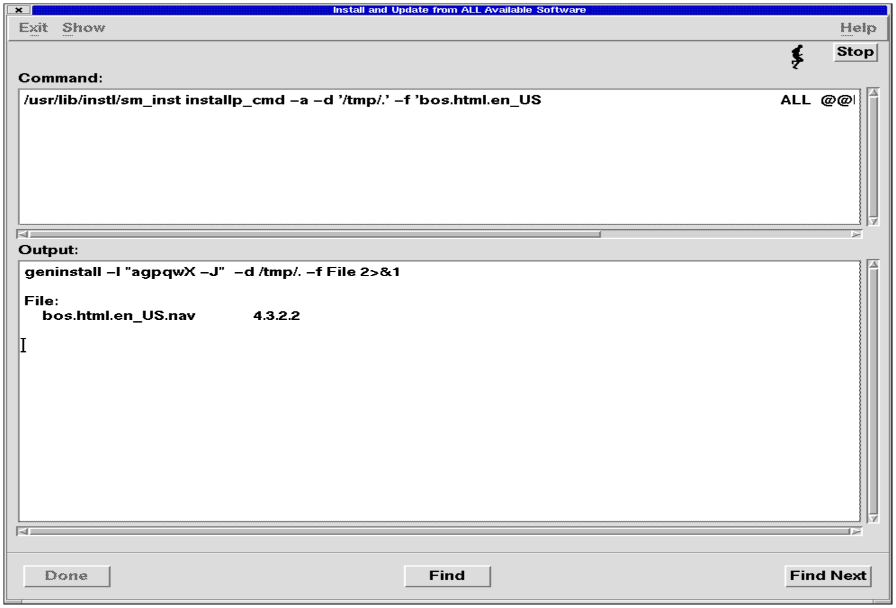

Figure 2-7 SMIT command output panel

3

Porting Open Source Software to AIX

This chapter discusses how to port UNIX and Linux based Open Source Software (OSS) to AIX with the aid of Toolbox packages.

This chapter is divided into the following sections:

- Why to port Open Source Software to AIX
- How to obtain Open Source Software
- How to install the GNUPro development environment
- How to build RPM packages
- How to compile Open Source Software from TAR packages
- How to create your own Open Source Software
- How to create SRPMs and RPMs from sources

3.1 Why to port Open Source Software

Open Source Software's popularity has risen to great importance. OSS is programs/applications whose licenses give users the freedom to run the program for any purpose, to modify the program, and to freely redistribute either the original or modified program. For a more detailed discussion of OSS, refer to 2.1.3, "Open Source Software" on page 17.

This alternative method of software development and distribution gives the Open Source community the possibility to contribute to the program by reporting errors and bugs or fixing problems of their own. Since different people have diverse techniques for tracing problems, the product/application is continuously enhanced and becoming more robust and reliable.

The AIX Toolbox contains many kinds of software that is commonly used in Linux systems. Because of the deep integration with the AIX operating system, the Toolbox should be used as the main source for Open Source Software.

However, if you have an application that is already running on a Linux system but is not included in the Toolbox, you have the option to recompile it and then run it natively on AIX 5L. The Toolbox contains GNU and other commonly used tools helpful for recompiling an application for use on AIX. This section will help you in utilizing these tools to port your Linux applications to AIX.

You can also get some ideas on how to port applications to AIX 5L in the *AIX 5L Porting Guide*, SG24-6034.

3.2 How to obtain Open Source Software

Most OSS software is nowadays developed on Linux. Here are a some good Web sites for finding the source for a particular Open Source Software and application:

GNU	`http://www.gnu.org/directory`
Freshmeat	`http://www.freshmeat.net`
SourceForge	`http://sourceforge.net/softwaremap`
Rpmfind	`http://www.rpmfind.net/linux/RPM`
Tuxfinder	`http://www.tuxfinder.org`
Linuxlinks	`http://www.linuxlinks.com/Software`
Freshrpms	`http://www.freshrpms.net`

Center for the Public Domain	http://www.ibiblio.org ftp://sunsite.unc.edu/pub

Unpacking the software source

To download the desired software source code, you should go to the home page and find the latest stable/production/released version. Sources will typically come either as a compressed, gzipped, or bzip2ed tar file or as a source RPM (SRPM). SRPMs are very convenient since they already contain the SPEC file used to produce a binary RPM.

Extract the source and it will typically create its own subdirectory. Here are some common extraction techniques for downloaded files with the following file name extensions:

***tar.gz**	`gunzip -c` *`filename`* `| tar xvf -`
***.tgz**	`gunzip -c` *`filename`* `| tar xvf -`
***tar.bz2**	`bzip2 -dc` *`filename`* `| tar xvf -`
***src.rpm**	`rpm -iv` *`filename`*

In the case of the SRPM, the sources, including the patches (if there are any), will be extracted into the /opt/freeware/src/packages/SOURCES directory and the SPEC file into the /opt/freeware/src/packages/SPECS directory. This will be discussed further in 3.4, "How to build RPM packages" on page 61.

3.3 The GNUPro development environment

The GNUPro development tools are a set of software development tools built around the Open Source GNU standard. The GNUPro Toolkit provides all the tools necessary for effective software compiling and debugging.

The GNUPro development environment images are available as RPM installable packages. They contain the **gcc** and **g++** compilers, the **gdb** debugger and associated utilities (such as **libtool** and **diff**). This can be downloaded on the Toolbox Web site in the same directory with the other RPMs at:

ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc

Before trying to build software packages, you need to install the appropriate GNUPro development environment. The following should be considered as a minimum and should be installed from the Toolbox:

gcc, gdb	C compiler and debugger

file-, find-, binutils	A set of GNU tools used frequently in the Open Source world that in some respects differ from their AIX counterparts
make	GNU **make** command
autoconf	GNU **autoconf** command
automake	GNU **automake** command
m4	GNU source code preprocessor
patch	GNU **patch** command
libtool	A set of GNU tools used frequently in the Open Source world to create and link with the various types of shared libraries in the UNIX world
info, **texinfo**	GNU tools (**install-text**) to view current tool documentation for GNU tools

You could also download the ezinstall script **getapp-dev.sh** and run it to download the base development tools for the GNUPro development environment. Running this script will download and install the whole set of recommended tools for application development and compiling. The **getapp-dev.sh** can be found in the following directory on the Toolbox FTP server (see 2.3.2, "How to download Toolbox packages" on page 34):

ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/ezinstall/ppc

For an installation of all images of the GNUPro toolkit, we need 80MB of disk space. The packages in the GNUPro toolkit will be installed under the /opt/freeware/GNUPro directory. Also, links to the executables may be created in /usr/bin and /usr/linux/bin.

This list does not avoid all "requisite missing" errors during the build phase of a particular software package. Sometimes, errors caused by the absence of Open Source versions of already installed AIX binaries are hard to debug because no significant error messages were generated, or the build process continues for some time after the incompatibility occurs.

Note: We recommend that you install the complete filesets for bos.adt and X11.adt prior to installing the compiler suite, although a subset might be enough in some cases. These filesets provide header files, libraries, and some other tools needed for building packages.

3.4 How to build RPM packages

In 2.3, "Installing the Toolbox RPM packages" on page 33, we use the **rpm** command provided by the rpm.rte package to install binary RPM packages into the system. But RPM can do more than that. It has additional options that allow you to install source packages or SRPMs, compile them, and produce new RPMs and SRPMs. A detailed description of the capabilities and usage of RPM can be found in the *Maximum RPM* book by Ed Bailey. Please refer to the following Web site:

http://www.rpm.org/max-rpm/index.html

The RPM's build process is controlled by the SPEC file, which is part of every SRPM. The SPEC file has several sections for the various stages in the build process. The eight sections/stages are:

preamble Information about the package and its history; intended to be read by human beings.

%prep To set up a clean build environment, expand archives, and so on.

%build The actual build step for the software (**make** is typically executed here).

%install Virtual installation of the software within the built environment (**make install** will be placed here).

%uninstall Consists of scripts that will be run on the user's system when the package is actually installed or removed.

%verify This script can be used to verify aspects of the package that are beyond RPM's capabilities.

%clean A script that cleans things up after the build process.

%files A list of all files belonging to the package.

Sometimes there are path variables that are hard coded in the SPEC file that have to be changed to include the correct prefix (/opt/freeware) used throughout the Toolbox. This can be done by using the %{_prefix} macro, which is one of the defined macros in the system default /usr/opt/freeware/lib/rpm/macros file.

Another change that might be necessary to ordinary SPEC files is adding links to binaries and libraries in the standard locations (/usr/bin and /usr/linux/bin or /usr/lib and /usr/linux/lib).

Here is a short summary of the options that be can use during the build process. The basic syntax for the BUILD option to the **rpm** command is:

rpm -b*stage* options *package name*.spec

Where *stage* can be:

p	Executes the %prep stage from the SPEC file. Normally this involves unpacking the sources and applying any patches.
l	Does a list check. The %files section from the SPEC file is macro expanded, and checks are made to verify that each file exists.
c	Does the %build stage from the SPEC file (after doing the %prep stage). This generally involves the equivalent of a **`make`**.
i	Does the %install stage from the SPEC file (after doing the %prep and %build stages). This generally involves the equivalent of a **`make install`**.
b	Builds a binary package (after doing the %prep, %build, and %install stages).
s	Builds just the source package (after doing the %prep, %build, and %install stages).
a	Builds binary and source packages (after doing the %prep, %build, and %install stages).

Additionally, the following options might be useful:

--short-circuit	Skips straight to the specified stage, that is, skips all stages leading up to the specified stage and are only valid with -bc and -bi options
--clean	Removes the build tree after the packages are made
--rmsource	Removes the sources and SPEC file after the build but may also be used stand alone, for example, **`rpm --rmsource foo.spec`**
--test	Does not execute any build stages; useful for testing out SPEC files
--vv	Displays debug information

3.4.1 How to build a RPM from a Toolbox SRPM

To install a software package with the **`rpm`** command, a binary RPM package is needed. In this section we will show you how to create a binary RPM package from a SRPM package that can then be installed with the **`rpm`** command. We will use the **`wget`** package from the Toolbox as an example.

First, download your preferred version of the **wget** SRPM package from the Toolbox FTP site. This task can be done using the **ncftpget** command (refer to 2.3.2, “How to download Toolbox packages” on page 34).

Next install the SRPM package with the **rpm** command, as shown in Example 3-1.

Example 3-1 Install SRPM package

```
[root@bayani]: /> rpm -iv wget-1.8.1-1.src.rpm
wget-1.8.1-1

[root@bayani]: /> ls -l /opt/freeware/src/packages/SOURCES
-rw-r--r--   1 root     system      1097780 Jan 08 2002  wget-1.8.1.tar.gz

[root@bayani]: /> ls -l /opt/freeware/src/packages/SPECS
-rw-r--r--   1 root     system         3030 Jan 28 2002  wget.spec
```

Issuing the **rpm -iv** command will extract the contents of the package and place the source in the /opt/freeware/src/packages/SOURCES directory and the SPEC file in the /opt/freeware/src/packages/SPECS directory.

Tip: Adding the vv option to the **rpm -i** command will give you more detailed output. The output will show you from where the files are extracted.

Now change to the directory /opt/freeware/src/packages/SPECS and rebuild the **wget** package with the **rpm -ba** command:

```
rpm -ba wget.spec
```

This should generate the wget-1.8.1-1.aix5.2.ppc.rpm RPM package in the /opt/freeware/src/packages/RPMS/ppc directory, which can then be installed with the **rpm** command:

```
rpm -iv wget-1.8.1-1.aix5.2.ppc.rpm
```

3.4.2 How to build RPMs from Toolbox SRPM with upgraded source

In this section we will demonstrate how to update a Toolbox SRPM package with the latest available version found on the Internet. We will discuss the changes necessary to a SPEC file when updating Toolbox packages to a newer release level. We will use the **wget** package as an example.

To update an installed package would be easy if “normal” Linux SRPM could be used. But, usually, some changes have to be made to the Linux SRPM SPEC file to make it work with the Toolbox.

The following procedures can be used as a step-by-step guideline to easily upgrade a Toolbox SRPM package:

1. Download the old SRPM file from the Toolbox FTP site.
2. Download the new source (.tar) and save it in the /opt/freeware/src/packages/SOURCES directory.
3. Change the SPEC file (from the old SRPM) to use the new version of the source code.
4. Build the binary RPM package using the **rpm -b** command.

Now let us go through all these steps. First, we download the corresponding **wget** SRPM package to be upgraded from the Toolbox FTP site. You can do this by using the **ncftpget** command as shown in 2.3.2, “How to download Toolbox packages” on page 34. For this example we downloaded the wget-1.8-1.src.rpm file.

Next we install the SRPM package with the **rpm** command, as shown below:

```
rpm -iv wget-1.8-1.src.rpm
```

This command will generate the following two files:

```
/opt/freeware/src/packages/SOURCES/wget-1.8.1.tar.gz
/opt/freeware/src/packages/SPECS/wget.spec
```

Next we want to update the source code of **wget** to the latest level. You can get the latest version of the source from the GNU FTP server, for example, the master server at:

ftp://ftp.gnu.org/pub/gnu/wget

Download the new software archive wget-1.8.2.tar.gz and save it to the /opt/freeware/src/packages/SOURCES directory.

Edit the SPEC file and change the line:

```
%define version 1.8.1
```

To:

```
%define version 1.8.2
```

Next build the binary and source packages by issuing the command:

```
rpm -ba wget.spec
```

This command will generate the following RPMs:

```
/opt/freeware/src/packages/SRPMS/wget-1.8.2-1.src.rpm
/opt/freeware/src/packages/RPMS/ppc/wget-1.8.2-1.aix5.2.ppc.rpm
```

The new version can now be installed with the -i, -U, or -F options of the **rpm** command. In the next example we install the RPM package with the -iv options of the **rpm** command (install and verbose):

```
rpm -iv /opt/freeware/src/packages/RPMS/ppc/wget-1.8.2-1.aix5.2.ppc.rpm
```

The -i option with **rpm** will only work if **wget** is not already installed on the system. If **wget** is already installed, use the -U or -F option with **rpm**.

3.4.3 How to build RPMs from patched Toolbox SRPMs

RPM packages generally get built from a series of source files and patches. The source images should be used in their original form as they came from the home site for that particular project. Any changes that are required to get the package building and running (other than those that can be accomplished through the RPM build process via the SPEC file itself) need to be packaged as patch files.

In short, a patch is an update to the source code of the program to make it work on your environment. Patches can fix a bug or security hole in a program, stop a program from crashing, or add extra features and functionality to a program.

The tool used to produce the patch is called **diff** (short for "difference"). Not only does a **diff** show the difference between two sets of source files (the original files and the patched ones), but it also contains the information necessary to change another set of original files into patched ones. Before building and installing the software, you then pass the **diff** file to a tool called **patch**. The patch tool can understand several different **diff** formats and automatically apply them to your source files.

A good way to create a patch is to keep the original code in one directory, and the modified code in another directory at the same level. You can then use **/usr/bin/diff** to create a "unified" **diff** that can be used as a patch file and applied with the **/usr/bin/patch** command.

For example, the original code is extracted into the foo directory, then foo is copied to foo.orig. Both the foo and the foo.orig directory now contain the same source code. If we now modify the source code in the foo directory, we can then create a patch file by running the GNU **diff** command:

```
/usr/linux/bin/diff -uNr foo.orig foo > foo-level-desc.patch
```

The patch will be applied to the SRPM with the **patch** command, as the source gets extracted during the RPM's build process.

How to apply a patch

To demonstrate the concept of the patch, we will use the application Apache (from the Toolbox SRPM) as an example.

First, we download the apache SRPM package from the Toolbox FTP site. For this example, we used the apache-1.3.20-1.src.rpm file.

Next we install the SRPM package with the **rpm** command, as shown below:

```
rpm -iv apache-1.3.20-1.src.rpm
```

The **rpm** command will generate the following files:

- /opt/freeware/src/packages/SOURCE/apache_1.3.20.tar.gz
- /opt/freeware/src/packages/SOURCES/apache_1.3.20-perlpath.patch
- /opt/freeware/src/packages/SPECS/apache.spec

The apache_1.3.20-perlpath.patch file could look like what is shown in Example 3-2.

Example 3-2 Patch file for Apache

```
Index: apache_1.3.19/htdocs/manual/search/manual-index.cgi
diff -c apache_1.3.19/htdocs/manual/search/manual-index.cgi:1.1
apache_1.3.19/htdocs/manual/search/manual-index.cgi:1.2
*** apache_1.3.19/htdocs/manual/search/manual-index.cgi:1.1     Mon Mar 19
10:26:27 2001
--- apache_1.3.19/htdocs/manual/search/manual-index.cgi Mon Mar 19 10:36:21
2001
***************
*** 1,4 ****
! #!/usr/local/bin/perl5 -w
  # ====================================================================
  # Copyright (c) 1995-2000 The Apache Group.  All rights reserved.
  #
--- 1,4 ----
! #!/usr/bin/perl -w
  # ====================================================================
  # Copyright (c) 1995-2000 The Apache Group.  All rights reserved.
  #
```

The patch changes the hardcoded path /usr/local/bin/perl5 to /usr/bin/perl in the manual-index.cgi file. This patch was generated by the Toolbox developers using the **diff** command as demonstrated earlier in this section.

To include this patch in the binary RPM package, it must be added in the SPEC file before building the binary RPM. To get the patches properly applied, we need to add two things in the SPEC file:

1. A patch tag line (Patch#) pointing to our patch file in the %preamble section
2. A %patch macro in the %prep section

Like the “source” tag lines, every patch tag is numbered starting at zero. You can have as many patch files as you want as long as they are properly tagged. The %patch macro, by default, applies the patch file named on the patch tag line (in this case, patch0).

The %patch macro has two ways to specify which patch tag line to use:

1. Append the number of the desired patch tag to the end of %patch macro, for example, for the third (3) patch it would be referenced as %patch3.
2. Use the -P option followed by the number of the desired patch tag line, for example, for the third (3) patch it would be referenced as %patch -P 3.

Here are the other options that you can use with the %patch macro:

-p# Strips the # leading slashes and directories from patch file names.

-b*name* Sets the backup file extension to *name*. This is normally done when multiple patches are to be applied to a given file. It will save a copy of the file as it existed prior to each patch.

-E Removes empty output files.

In our example, the apache SPEC file will look like in Example 3-3.

Example 3-3 apache.spec

```
# Use --define 'nossl 1' on the command line to disable SSL detection
%{!?nossl:%define SSL 1}
%{?nossl:%define SSL 0}
%define ssldir %{_prefix}

%define EAPI_vers  2.8.4
%define apache_vers 1.3.20
Summary: The most widely used Web server on the Internet.
Name: apache
Version: %{apache_vers}
Release: 1%{!?nossl:ssl}
Group: System Environment/Daemons
Source0: ftp://ftp.apache.org/apache/dist/apache_%{apache_vers}.tar.gz
%if %{SSL} == 1
```

```
Source1:
http://www.modssl.org/source/mod_ssl-%{EAPI_vers}-%{apache_vers}.tar.gz
%endif
Patch0: apache_%{apache_vers}-perlpath.patch
...(lines omitted)...

%description
Apache is a powerful, full-featured, efficient and freely-available
...(lines omitted)...

%prep
%setup -q -n apache_%{apache_vers}
%patch0 -p1

...(the patch is applied before the RPM starts building the package)...

%build
# Use the default compiler for this platform - gcc otherwise
if [[ -z "$CC" ]]
fi
...(lines omitted)...
```

Building SRPM with patches

Now let us try to build the package by using **rpm -bp** and see if the patch works. Refer to Example 3-4 for the output.

Example 3-4 rpm -bp output when building Apache

```
[root@bayani]: /opt/freeware/src/packages/SPECS> rpm -bp apache.spec --define
'nossl 1'
Executing(%prep): /bin/sh -e /var/opt/freeware/tmp/rpm-tmp.27922
+ umask 022
+ cd /opt/freeware/src/packages/BUILD
+ cd /opt/freeware/src/packages/BUILD
+ rm -rf apache_1.3.20
+ tar -xf -
+ /bin/gzip -dc /opt/freeware/src/packages/SOURCES/apache_1.3.20.tar.gz
+ STATUS=0
+ [ 0 -ne 0 ]
+ cd apache_1.3.20
+ /bin/id -u
+ [ 0 = 0 ]
+ /bin/chown -Rhf root .
+ /bin/id -u
+ [ 0 = 0 ]
+ /bin/chgrp -Rhf system .
+ /bin/chmod -Rf a+rX,g-w,o-w .
+ echo Patch #0 (apache_1.3.20-perlpath.patch):
```

```
Patch #0 (apache_1.3.20-perlpath.patch):
+ patch -p1 -s
+ 0< /opt/freeware/src/packages/SOURCES/apache_1.3.20-perlpath.patch
+ exit 0
```

At the end of the process, we can see that the %patch macro displays a message showing that the patch is being applied and then invoke the **patch** command to do its job. If we expand the **patch** command, it should produce an output that looks similar to Example 3-5.

Example 3-5 patch command output

```
Processing...  Looks like a new-style context diff...
The text leading up to this was:
--------------------------
|Index: apache_1.3.19/htdocs/manual/search/manual-index.cgi
|diff -c apache_1.3.19/htdocs/manual/search/manual-index.cgi:1.1
apache_1.3.19/htdocs/manual/search/manual-index.cgi:1.2
|*** apache_1.3.19/htdocs/manual/search/manual-index.cgi:1.1    Mon Mar 19
10:26:27 2001
|--- apache_1.3.19/htdocs/manual/search/manual-index.cgi        Mon Mar 19
10:36:21 2001
--------------------------
Patching file htdocs/manual/search/manual-index.cgi using Plan A...
Hunk #1 succeeded at 1.
done
```

After identifying all the patches that you need and creating them, you can now build the binary package with the **rpm -ba** command, in this case:

```
rpm -ba apache.spec
```

This should generate the apache-1.3.20-1.aix5.2.ppc.rpm RPM package in the /opt/freeware/src/packages/RPMS/ppc directory, which can then be installed with the **rpm** command:

```
rpm -iv apache-1.3.20-1.aix5.2.ppc.rpm
```

3.5 How to compile OSS from TAR packages

In this section we describe how to compile and install Open Source Software without using the **rpm** command, but with the basic **configure**, **make**, **cc**, and **install** commands.

In most cases, the build process will simply consist of the following steps:

```
cd projectdir
./configure --prefix=/opt/freeware
make
make install
```

Open Source Software often specifies installation under /usr/local, unlike the Toolbox packages that are installed under the /opt/freeware directory. To change this behavior, we could use the **--prefix=/opt/freeware** with the **configure** tool to use the same directory structure as the Toolkit does.

Not all tools use **configure**, and some may simply take the prefix as part of the make step, as in make prefix=/opt/freeware.

In this section we will use the GNU C compiler package as an example. First, we download the desired source package. Since we need a C compiler to build the GNU C compiler software, we have installed the **gcc** compiler version of the Toolbox, which we will then remove before installing the other version of **gcc**. We recommend that, once the build is completed and the newer version of the compiler is installed, you rebuild and install the source the second time. This allows the current **gcc** build level to be built with itself, the latest level.

In our case, we download the latest **gcc** package from a GNU FTP mirror site:

```
ftp://mirrors.rcn.net/pub/sourceware/gcc/releases/gcc-3.2/gcc-3.2.tar.gz
```

Next we unpack the software package with the appropriate **unzip** and **tar** commands, as shown below:

```
gunzip -c gcc-3.2.tar.gz|tar xf -
```

Then we need to create the makefile for our specific system with the **configure** command. In our example it will be $PWD/gcc-3.2:

```
./configure -prefix=/opt/freeware
```

After our platform-specific makefile has been created, we can create the binaries by running the **make** command:

```
make
```

Since we want to use the latest compiler to compile itself, we now have to uninstall the Toolbox GNU C compiler package with the **rpm** command:

```
rpm -e gcc
```

The next step would be to install the binaries and libraries of the new compiler in the AIX file system hierarchy. This can be done by using the install option of the **make** command:

```
make install
```

For most other SRPMs, this would be the *final step*, but we want to use the latest compiler to compile itself. We now have to recompile the GNU C compiler package after removing all binary files from the build directories. This is first done with the clean option of the **make** command, and then we can rebuild the binaries again:

```
make clean; make
```

Now we can finally install the binaries again with the install option of the **make** command. We could also uninstall the previously installed binaries with the uninstall option of the **make** command, and then install them again with the install option:

```
make uninstall; make install
```

Note: The last two steps are specific to this compiler build only. Other software is already installed after the first **make install** command.

If you do not have the GNU C compiler package source as a TAR or compressed TAR file, but instead as a SRPM package from any Linux distribution, you can extract the sources by one of the following two methods:

- Execute only the %prep section of the SRPM by using **rpm -bp** and retrieve the sources from /opt/freeware/src/packages/SOURCES.
- Extract the source archives out of the SRPM by using **rpm2cpio**. (This tool is discussed in detail in 2.3.17, "How to extract files from a package" on page 49.)

How to use libtool to handle shared libraries

Many Open Source applications make use of **libtool** to handle shared libraries. Libtool is a GNU software package that helps develop and maintain shared libraries. It simplifies the use of shared libraries by hiding the complexity. The tool is fully integrated with the GNU **autoconf** and **automake** utilities and encapsulates the platform-specific dependencies and the user interface in a single script. The **libtool** interface helps to hide the idiosyncrasies from the programmers.

This section will focus on the basic design and use of **libtool**. For a more detailed description, please refer to the **libtool** documentation:

http://www.gnu.org/software/libtool/manual.html

In order to use **libtool**, the following files are needed in the source tree of the software to be compiled:

config.guess	Attempts to guess a canonical system name (such as powerpc-ibm-aix5.2.0.0)
config.sub	Validation script for a canonical system name
ltconfig	Generates a libtool script for a given system
ltmain.sh	A generic script implementing basic **libtool** functionality

These files should not be included in the source tree of the application. Instead, the **libtoolize** program should be used, which is part of the **libtool** package itself. In this case (during a software build), the **libtool** package has to be installed on the system prior to running **libtoolize**. After copying the needed files with **libtoolize** to the source tree, the actual libtool script can be generated with the **ltconfig** command.

ltconfig generates the system-specific libtool script. This process is somewhat similar to running **configure** to generate a makefile. The resulting libtool script can then be used as an interface to generate appropriate compiler, linker, debugger, and installer calls.

> **Note:** Many Open Source projects embed their own versions of **libtool**, and most of those versions have, basically, non-functional sections of code for AIX. The relevant files are usually named ltconfig, ltmain.sh, config.sub, and config.guess. If your system has a **libtool** version lower than 1.3.5 and you are encountering problems during the build process, then you will want to invoke **libtoolize --force** to copy in the updated versions from the Toolbox. This would be done in the %build phase in the SPEC file prior to the configure step.

3.6 How to create your own Open Source Software

There are a couple of items you should consider when you start a new project of creating your own Open Source Software, mainly how you should ensure that your code is portable to other platforms. It is easy to say but more difficult to do. Different hardware, compilers, etc. can result in your code behaving in unexpected ways.

To avoid this sort of unpleasantness, and ensure portability in your Open Source coding effort, consider the following simple guidelines before you create too much code:

- For legal reasons, make sure you use a proper Open Source Software licensing template, and do not forget the correct copyright date.

- Be aware of standards when you write your code. This includes international (POSIX), national (ANSI C), industry (Single Unix Specification), and de-facto standards (LSB). Sometimes choosing a particular API subroutine can cause portability problems, which could have been avoided by selecting a similar but standardized subroutine.
- Consider whether you need to use compiled code for your whole project or if you can at least use some parts with scripted programming tools, such as **`ksh`**, **`perl`**, or PHP.
- It will help when finding a particular source at a later time if you follow a file naming convention when creating source files.
- Sometimes you have to use some form of #ifdef statements throughout the source code to separate code sections that are needed for the software to compile on different systems.
- You need to think about the performance of your code. You can get some ideas on how to maximize the speed of your program from the “Designing and Implementing Efficient Programs” section in the *AIX 5.2 Performance Management Guide* (which you can find as a PDF by doing a search on the `http://www.ibm.com` Web site).
- Like performance, security is a concern when coding software and especially so when considering portability issues, since many security features’ API subroutines are system specific.
- Decide how you will document your software and what tools to use and then stick to it.

Use the tools you are most familiar with, and do not neglect to choose a good development environment that allows you to compile and debug, and version control of your software.

Coding guidelines

If there are no coding guidelines defined in your coding project, the GNU coding standards would be a good start to ensure a consistent debugging and build environment. Please refer to the following Web site for more information:

`http://www.gnu.org/prep/standards_toc.html`

The GNU **`autoconf`** tool is a major building block in this framework to achieve portability to most UNIX-based systems, please refer to 6.4.2, “autoconf/automake” on page 218, and the following Web site for more information.

`http://www.gnu.org/software/autoconf`

Compiler and API considerations

Since Linux applications will probably be written using the GNU compilers, and the same compilers are available with the Toolbox on AIX, there will not be any language-specific errors in the code. But there might be errors because of missing and/or incompatible APIs. On the IBM Toolbox Web site there are documents illustrating differences between standard compliant AIX APIs and the Linux Standards Base (LSB).

The Toolbox documentation on API differences can be found on the following FTP directory. (Please note that this is only an aide and not a substitute for the appropriate system API subroutine documentation for the target systems.)

ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/docs/index.html

When coding in the C programming language, you can use **lint** to check the source code for possible syntax problems, and the **vgrind** command to format the source code for printing.

Other considerations

Code intended to be portable should not depend on any specific byte ordering. Remember that the i86 architecture is little endian and the POWER architecture is big endian. Also, alignment and size of scalar types may differ. National language support (NLS) and specific characteristics in the networking layers are also areas to watch out for when creating portable code.

Of course, certain prerequisites, such as both libraries and other software (like databases and middleware), have to be available for the target platform, especially if it is proprietary software and the source code is not available.

3.7 How to create SRPMs and RPMs from sources

Using **gnugo** (the GNU program that plays the game of Go) as an example, we will demonstrate how to build packages from their source code (TAR archive) and use our own SPEC file to build the SRPM and binary RPM. We will analyze the important sections in the respective SPEC file and produce our first .rpm and .src.rpm files.

The first thing we need to do in order to build a RPM and SRPM package for **gnugo** is to obtain the source. Download the latest version of the **gnugo** source code from:

http://www.gnu.org/software/gnugo/gnugo.html

The version of the **gnugo** gzipped TAR archive file used in this example is 3.2.

Move the **gnugo** source code into the appropriate build directory, in our case to the /opt/freeware/src/packages/SOURCES directory:

```
mv gnugo-3.2.tar.gz /opt/freeware/src/packages/SOURCES
```

Before we can start building the package, we need to create a SPEC file to process the build. The SPEC file should be located in the /opt/freeware/src/packages/SPECS directory.

> **Tip:** Some of the OSS applications have SPEC files available on the Web. It will be easier to download and change a SPEC file to make it work in your environment than to write a new one from scratch. When doing this, you could use one of the Toolbox SPEC files as a reference for how directories and macros can be changed to fit in the Toolbox directory hierarchy.

A SPEC file is basically a makefile for **rpm**. As discussed in 3.4, "How to build RPM packages" on page 61, the SPEC file contains eight major sections, most of which are required. There are also %pre, %post, %preun, and %postun sections for run-time processing. Similar to the AIX-specific installation program **installp**, RPM packages are supposed to install and uninstall non-interactively.

> **Note:** RPM requires that the entire build process completes from top to bottom without an error before it can produce an installable RPM as output.

Example 3-6 is how the whole SPEC file (gnugo.spec) that we created for the GNU go application will look.

Example 3-6 GNUGo SPEC file (gnugo.spec)

```
Summary:        GNU Go is a free program that plays the game of Go
Name:           gnugo
Version:        3.2
Release:        1
Copyright:      GPL
Group:          Games/Boards
Source:         ftp://ftp.gnu.org/gnu/gnugo/gnugo-3.2.tar.gz
Url:            http://www.gnu.org/software/gnugo
%description
Go is a game of strategy between two players usually played on a 19x19 grid
called goban. GNU Go plays a game of Go against the user.
You need to install the CGoban package to use the GNU Go graphical Interface in
an X Window System.

%prep
%setup -q
```

```
%build
%configure --enable-color --with-curses
make
cd doc
makeinfo --html gnugo.texi

%install
make install

%files
%defattr(-,root,root)
%doc AUTHORS README INSTALL NEWS THANKS TODO COPYING ChangeLog doc/gnugo.html
%{_infodir}/gnugo*
%{_bindir}/*
%{_mandir}/man6/*

%clean
rm -rf $RPM_BUILD_ROOT
```

Now, let us go through each of the important sections that are needed in the **gnugo** SPEC file:

- Preamble
- %prep
- %build
- %install
- %files
- %clean

Preamble section

The preamble should contain entries with information about the package. This is a short description of the package software that can be displayed using the **rpm -qi** command, and other information as needed. The order of the lines is not important as long as they are in the preamble section. In this case, our preamble ends after the %description paragraph.

%prep section

We will then create the build environment of the **gnugo** software in this section. Refer to Example 3-7 to see how the %prep section will look in our SPEC file.

Example 3-7 gnugo.spec %prep section

```
%prep
%setup -q
```

The %prep script (%prep section) is the first script that RPM executes during a build:

- *Create* the top-level build directory, in our example the /opt/freeware/src/packages/BUILD/gnugo-3.2 directory.
- *Unpack* the original source into the build directory.
- *Apply* patches to the sources, if necessary.
- *Perform* any other actions required to get the sources in a ready-to-build state.

Instead of using scripts in this section, we used the %setup macro. The use of macros will make life easier especially in building more complex packages.

There are two macros that can be used exclusively in the %prep script that simplify routine functions:

- The %setup macro, which is used to unpack the original sources
- The %patch macro, which is used to apply patches to the original sources

In this example, there is no patching required so we only used the %setup macro. The %setup macro gets the name of the source archive from the source tag specified earlier in the preamble section of the SPEC file, and unpacks it in preparation for the build process. If the %setup macro is used with no added options, it will expand into the commands shown in Example 3-8.

Example 3-8 %setup macro

```
cd /opt/freeware/src/packages/BUILD
rm -rf package name
tar -xvvf -
/bin/gzip -dc /opt/freeware/src/packages/SOURCES/package name.tar.gz
if [ $? -ne 0 ]; then
   exit $?
fi
cd package name
/bin/id -u
/bin/chown -Rhf root .
/bin/id -u
/bin/chgrp -Rhf system .
/bin/chmod -Rf a+rX,g-w,o-w
```

Now let us try the **rpm -bp** command and check whether our SPEC file turns out right for the %prep section. Refer to Example 3-9 for the command output.

Example 3-9 rpm -bp output

```
[root@bayani]: /opt/freeware/src/packages/SPECS> rpm -bp gnugo.spec
```

```
Executing(%prep): /bin/sh -e /var/opt/freeware/tmp/rpm-tmp.26864
+ umask 022
+ cd /opt/freeware/src/packages/BUILD
+ cd /opt/freeware/src/packages/BUILD
+ rm -rf gnugo-3.2
+ tar -xf -
+ /bin/gzip -dc /opt/freeware/src/packages/SOURCES/gnugo-3.2.tar.gz
+ STATUS=0
+ [ 0 -ne 0 ]
+ cd gnugo-3.2
+ /bin/id -u
+ [ 0 = 0 ]
+ /bin/chown -Rhf root .
+ /bin/id -u
+ [ 0 = 0 ]
+ /bin/chgrp -Rhf system .
+ /bin/chmod -Rf a+rX,g-w,o-w .
+ exit 0
```

Looking at the output, we see that the sources are now unpacked. We are now ready to build.

%build section

The %build section is the part of the SPEC file that is responsible for the build process. Refer to Example 3-10 for the build process that we created.

Example 3-10 gnugo.spec %build section

```
%build
%configure --enable-color --with-curses
make
cd doc
makeinfo --html gnugo.texi
```

In here, the %build changes its directory into the software's top-level build directory (RPM_BUILD_DIR). If we read through the INSTALL file included in the sources, we just need to run the **configure** and **make** commands to build GNU Go.

> **Tip:** It is advisable to use macros included in the macros file /usr/opt/freeware/lib/rpm/macros for your SPEC file whenever possible. This will facilitate architecture-independent SPEC files and ease recompiling on other operating systems.

Since the **configure** command is already included as a macro, we will use this instead and then issue **make**. You can also make HTML documentation from the

TEXINFO files included in the sources, but this is optional. To do this, just issue the **makeinfo** command like in the last line of Example 3-10 on page 78.

Now that the %build script is ready, let us verify the SPEC file again by issuing the following **rpm** command:

```
rpm -bc --short-circuit
```

This will skip the %prep phase and stop things after the %build script completes. Refer to Example 3-11 for the output.

Example 3-11 rpm -bc output

```
[root@bayani]: /opt/freeware/src/packages/SPECS> rpm -bc --short-circuit
gnugo.spec
Executing(%build): /bin/sh -e /var/opt/freeware/tmp/rpm-tmp.15788
+ umask 022
+ cd /opt/freeware/src/packages/BUILD
+ cd gnugo-3.2
      [...lines ommitted...]
+ ./configure ppc-ibm-aix5.2 --prefix=/opt/freeware --exec-prefix=/opt/freeware
--bindir=/opt/freeware/bin --sbindir=/opt/freeware/sbin
--sysconfdir=/opt/freeware/etc --datadir=/opt/freeware/share
--includedir=/opt/freeware/include --libdir=/opt/freeware/lib
--libexecdir=/opt/freeware/libexec --localstatedir=/opt/freeware/var
--sharedstatedir=/opt/freeware/com --mandir=/opt/freeware/man
--infodir=/opt/freeware/info --enable-color --with-curses
checking for a BSD compatible install... /usr/linux/bin/install -c
      [...the usual configure tests are being done here...]
config.status: config.h is unchanged
      [...now, make is called...]
+ make
cd . \
  && CONFIG_FILES= CONFIG_HEADERS=config.h \
     /bin/sh ./config.status
config.status: creating config.h
config.status: config.h is unchanged
make  all-recursive
make[1]: Entering directory `/opt/freeware/src/packages/BUILD/gnugo-3.2'
      [...skipping compiler calls...]
      [...doing last phase of the build stage...]
+ cd doc
+ makeinfo --html gnugo.texi
+ exit 0
```

The source code passed the build stage successfully. Now let us proceed to the install stage.

%install section

Since we used the **make** utility for the %build section and it has an install target, this section will also be straightforward (refer to Example 3-12).

Example 3-12 gnugo.spec %install section

```
%install
make install
```

If the application does not have the means of automatically installing itself, it will be necessary to create a script in this section. This section is responsible to do whatever is necessary to install the newly built software. In this case, a **make install** is the only thing that is needed.

Now let us see how this works before we wrap up the SPEC file. To verify, run the following **rpm** command:

```
rpm -bi --short-circuit
```

This will force the build to start at the %install stage and skip the first few stages (refer to Example 3-13).

Example 3-13 rpm -bi output

```
[root@bayani]: /opt/freeware/src/packages/SPECS> rpm -bi --short-circuit
gnugo.spec
Executing(%install): /bin/sh -e /var/opt/freeware/tmp/rpm-tmp.13431
+ umask 022
+ cd /opt/freeware/src/packages/BUILD
+ cd gnugo-3.2
+ make install
Making install in utils
make[1]: Entering directory `/opt/freeware/src/packages/BUILD/gnugo-3.2/utils'
      [...lines ommitted...]
make[1]: Leaving directory `/opt/freeware/src/packages/BUILD/gnugo-3.2'
+ exit 0
Processing files: gnugo-3.2-1
```

Now it is time to create the %file section. Make sure that you save the output of the **rpm -bi** command, since this will help us generate the %files section.

%files section

The %files section contains the list of the files that are part of the package. To fill out this section, we need the list of files that **gnugo** installs on the system. At present, the best way to do this is to read the output from the **make install** during the %install stage.

Here is a list of the configuration files and documentation that need to be included:

- /opt/freeware/bin/debugboard
- /opt/freeware/bin/gnugo
- /opt/freeware/info/gnugo.info
- /opt/freeware/info/gnugo.info-1
- /opt/freeware/info/gnugo.info-2
- /opt/freeware/info/gnugo.info-3
- /opt/freeware/info/gnugo.info-4
- /opt/freeware/info/gnugo.info-5
- /opt/freeware/info/gnugo.info-6
- /opt/freeware/info/gnugo.info-7
- /opt/freeware/info/gnugo.info-8
- /opt/freeware/info/gnugo.info-9
- /opt/freeware/info/gnugo.info-10
- /opt/freeware/info/gnugo.info-11
- /opt/freeware/info/gnugo.info-12
- /opt/freeware/man/man6/gnugo.6
- The gnugo.html that we generated during the %build stage
- And documentation files, such as AUTHORS, README, INSTALL, NEWS, THANKS, TODO, COPYING, and Changelog

The **rpm** command processes these files according to their type but does not have a method of automatically determining their file types. This can done by using directives.

Let us start by making sure that these files have the correct permissions and ownership set. For this task, we will use the %attr directive. The main reason for using %attr is to permit users without *root* user access to build packages. The %attr macro has the following format:

```
%attr (mode, user, group) file
```

Replacing an attribute with a dash (-) would mean that %attr will not change it. To simplify our SPEC file, let us set all files with the same attribute:

```
%attr (-, root, root)
```

For the documentation files, the %doc directive can be used. This will help the users find information about an installed package since the **rpm** command keeps track of documentation files in its database. Let us add this in our SPEC file:

```
%doc AUTHORS README INSTALL NEWS THANKS TODO COPYING ChangeLog \
doc/gnugo.html
```

When the package is installed, the **rpm** command creates a directory (gnugo-3.2.1) in the /opt/freeware/doc directory, and copies all the files listed in the %doc line. The newly created directory and all its files will then be marked as documentation.

For the other files, we can list each one of them in the %file section by writing the whole path. Another option is to use a defined macro from the macros file /usr/opt/freeware/lib/rpm/macros to refine the list and to simplify the SPEC file. This will prevent the %file list from looking all cluttered up, especially with complex software that contains a lot of files to be packaged.

After Looking at the macros file, here are the macros that we need:

%_bindir	For files in /opt/freeware/bin
%_infodir	For files in /opt/freeware/info
%_mandir	For files in /opt/freeware/man

%clean section

The %clean section is used to clean up the software's build directory tree. Since we did not define a build root directory for our package, using the **rm -rf** command in the RPM_BUILD_ROOT environment variable should be enough.

After completing the SPEC file, it is time for a complete build of the package. Change to the directory /opt/freeware/src/packages/SPECS and build the **gnugo** package using the **rpm** command. Example 3-14 shows the final creation of both binary and source RPMs.

Example 3-14 Binary and source RPMs creation

```
[root@bayani]: /opt/freeware/src/packages/SPECS> rpm -ba gnugo.spec
      [...skipping output already seen in the previous examples...]
ibtiff.a(shr.o) libxml.a(libxml.so.1) libxml2.a(libxml2.so.2) libz.a(libz.so.1)
libzvt.so
Wrote: /opt/freeware/src/packages/SRPMS/gnugo-3.2-1.src.rpm
Wrote: /opt/freeware/src/packages/RPMS/ppc/gnugo-3.2-1.aix5.2.ppc.rpm
Executing(%clean): /bin/sh -e /var/opt/freeware/tmp/rpm-tmp.18670
+ umask 022
+ cd /opt/freeware/src/packages/BUILD
+ cd gnugo-3.2
+ rm -rf
+ exit 0
```

The **rpm -ba** command will generate the following RPMs:

```
/opt/freeware/src/packages/SRPMS/gnugo-3.2-1.src.rpm
/opt/freeware/src/packages/RPMS/ppc/gnugo-3.2-1.aix5.2.ppc.rpm
```

The **gnugo** binary RPM package can now be installed with this command:

```
rpm -iv gnugo-3.2-1.aix5.2.ppc.rpm
```

4

Graphical desktops

In this chapter, we provide a general overview of the graphical desktops available in the AIX and the AIX Toolbox for Linux Applications environment. We also provide information on how to install and use both the K Desktop Environment (KDE) and Gnome desktop environments.

This chapter contains the following:

- Introduction to the X Window Systems
- Window managers
- Desktops and graphical environment
- The Common Desktop Environment (CDE) desktop
- The KDE desktop (version 2.2.1)
- The Gnome desktop (version 1.4)
- How to install KDE and Gnome

4.1 Desktop and graphical applications

The graphical environment in most UNIX/AIX and Linux systems is based on the X Window System. The X Window System (sometimes referred to as “X” or “X11”) is an open, cross-platform, client/server system for managing a graphical user interface in a distributed network.

We will focus on the specifics of the KDE and Gnome desktop environments that are provided by the AIX Toolbox for Linux Applications, their functionality, and how they interact with the AIX graphical environment, which is based on the Motif 2.1 toolkit and X11R6.

4.1.1 The X Window System

The X Window System is the standard graphics interface for UNIX and Linux operating systems. When using X, the user can have multiple terminal windows on the screen at once, and each window can contain a different login session to the same or different host systems.

One of the great advantages of using the X Window System is that its functionality is achieved through the cooperation of different components, rather than everything being packed into one single large collective. The X Window System provides the opportunity to mix and match components from different X-based graphical packages.

In the X Window System, the software that manages a single screen, and the keyboard and mouse are known as an X server. A client is an application that displays on the X server and is usually termed an X client or simply the application. The X client sends requests to the X server, for example, a drawing or information request. The X server accepts requests from multiple clients and returns to the X client replies for information requests, user input, and errors.

All desktop applications and window managers are considered to be X clients or applications in the X Window System framework. It is the X server that decides what and how requests (“hints”) from the clients will be displayed on the user’s screen.

In most cases users run the X server, X Window manager, and X client applications on their workstations. However, only the X server is required to run on the user’s computer; all other applications can either run locally or on another host system. Figure 4-1 on page 87 shows the basic flow of information between the major components of the X Window System.

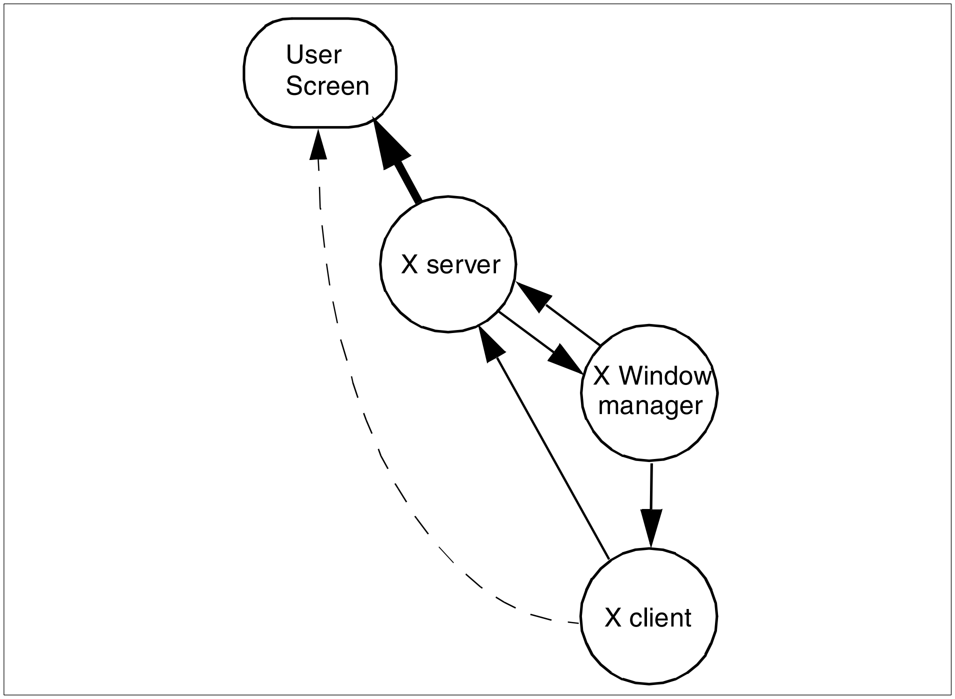

Figure 4-1 X Window System

In Figure 4-1, the solid arrows mark the conceptual flow of the graphical display requests from the X client to the user's screen. The client communicates with the X server, the X server communicates with the X Window manager, and the X Window manager communicates with the X client. The X client can also perform "pass-through" operations in the AIX implementation of the X server (used by OpenGL and graPHIGS in some cases).

The communication (*very simply* described) between the X client and the X server can be such simple things as "open window". When the X server receives the "open window" request, it performs many things; among them reserving an area of the screen and telling the window manager that this area should be decorated. The X Window manager informs the X server how it wants the window to be decorated. The X Window manager can also inform the client, through the X server, that the window size has been changed.

For more information on the X Window System, please refer to the following Web site:

`http://www.X.org`

4.2 X Window managers

A very important part of the X Window System is the window manager, regardless of which desktop is being used. The window manager provides the look and feel of the X interface. This program is in "charge" of the placement of windows and the user interface, and is used for resizing, iconifying, moving, and changing the appearance of the window frames. Table 4-1 describes the window managers used by the desktops.

Table 4-1 Desktops and window managers

Desktop	Window Manager
KDE	Any NETWM-compliant window manager, such as the default window manager **kwin**. Alternative window managers can be, but are not limited to: **enlightenment**, **fvwm**, **IceWM**, and **WindowMaker**. It is even possible to use **twm** and **mwm**.
Gnome	Any NETWM-compliant window manager, such as the default window manager **sawfish**. Alternative window managers can be, but are not limited to: **enlightenment**, **fvwm**, **IceWM**, and **WindowMaker**. It is even possible to use **twm** and **mwm**.
CDE	**dtwm**

4.2.1 The kwin window managers

kwin is the window manager of choice for the KDE desktop and is part of the kdebase package. It provides:

- Complete integration with KDE.
- Complete keyboard control and configuration.
- The ability to be reconfigured at runtime without restarting.
- Session management and working session management proxy for legacy applications. This proxy is able to restore applications to their previous state, including window properties and the virtual desktop that the GUI was running.

4.2.2 The enlightenment window manager

The **enlightenment** window manager (also known as E) design goal is to be as configurable as possible when it comes to the look and feel. The **enlightenment** window manager can also be used as a standalone window manager.

You can enable **enlightenment** for KDE, once it has been installed, by changing a line on the **startkde** script, located in /opt/freeware/kde/bin. This is done by appending --windowmanager enlightenment to the line starting with ksmserver --restore.

Some of the features that the **enlightenment** window manager provides are:

- Fully configurable window borders
- A graphical pager that takes miniature snapshots of your screen
- Theme support
- Window transparency when moving windows (opaque)
- Virtual desktops
- KDE hint support
- Gnome hint support
- Tooltips

4.2.3 The sawfish window manager

The highly configurable **sawfish** window manager, previously known as **sawmill**, uses a Lisp-like scripting language. The user interface policy is controlled through the Lisp language. User configuration is possible either by writing Lisp code in a personal .sawfishrc file, or through the integrated customizing system.

4.2.4 The dtwm window manager

The **dtwm** window manager is based on the Open Group's Motif window manager, **mwm** (IEEE 1295). The **dtwm** is an integral part of the CDE desktop; it communicates and facilitates access to other components in the environment, such as the session and style manager. In addition to this functionality, **dtwm** provides workspace management capabilities. Workspaces allow us to group together related windows, and each workspace is independent of the other workspaces.

One part of CDE is the **dtksh** command that provides a Korn shell 93 scripting language and built-in XLib and Motif subroutines.

4.3 The Toolbox graphical environment

The Toolbox provides a wide range of graphical tools that were ported from Linux. The graphical desktops available in the Toolbox are composed of different elements that provide a specific graphical development environment, which differs between the desktops.

Figure 4-2 show us the framework layout of the graphical library layers being used in regards to each of the desktops.

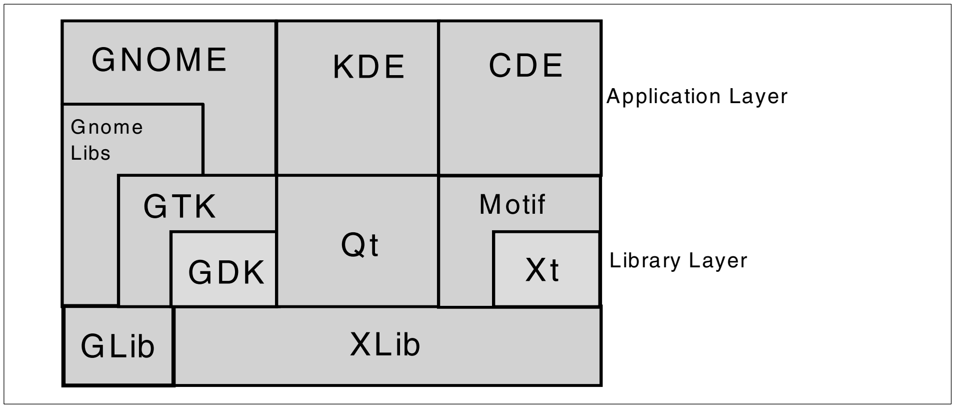

Figure 4-2 The Toolbox graphical framework

The X Window System-based libraries provide a software development framework that enables cross-platform developing and porting between heterogeneous graphical environments. By using these libraries, developers can create a single code base for different platforms.

In order to develop a graphical client application, a platform needs to provide the basic XLib libraries that allow the application to communicate with an X server locally or across a network.

Other libraries are:

- Qt
- GLib
- GDK
- GTK
- Gnome libs

For more information regarding GDK, GTK, and Gnome please refer to the following Web sites:

http://www.gtk.org
http://www.Gnome.org

For more information regarding Qt, please refer to the following Web site:

http://www.trolltech.com

4.4 The CDE desktop

The Common Desktop Environment (CDE) is an integrated graphical user interface for open systems desktop computing, combining X Window System, Motif, and the Common Desktop Environment technologies. CDE is designed to work across a large range of client/server platforms, support small workgroups to large enterprises, and support simple text and data, as well as advanced collaborative multimedia applications.

Many of the familiar AIX tools can be launched directly from the CDE desktop. Other products, including third-party applications, can be installed into the desktop's application manager folder, where they will appear as icons.

The most visible part of the CDE GUI is the front panel.

4.4.1 The front panel

The CDE front panel is a special desktop window that contains a set of controls for performing common tasks. The front panel is divided into the main panel and subpanels.

The front panel moves with you as you switch workspaces. Figure 4-3 on page 92 shows the front panel of the desktop; along the top of the front panel is a row of arrow buttons, which open the subpanels.

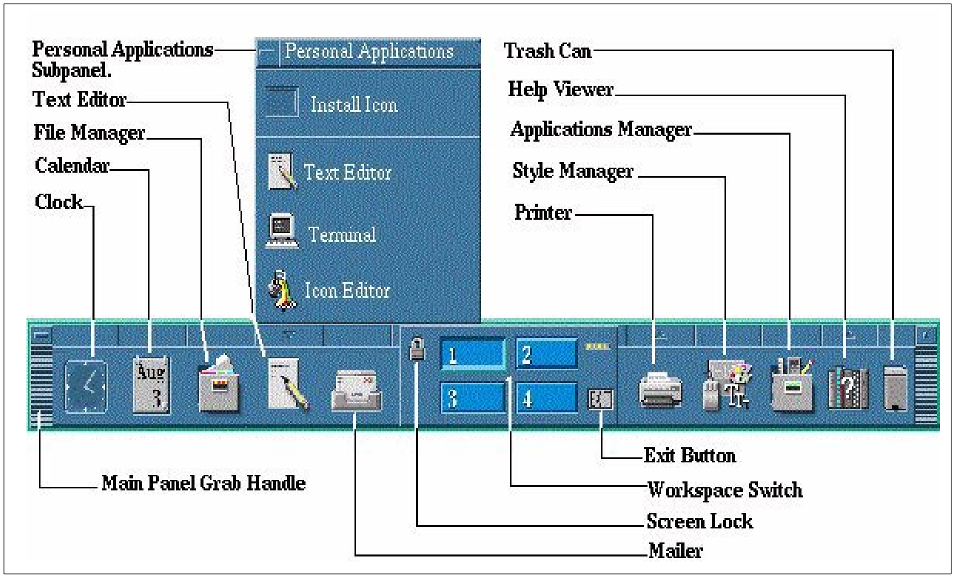

Figure 4-3 CDE front panel

Main panel

The main panel, part of the front panel, is the horizontal window at the bottom of the display. It contains a number of frequently used controls, including the workspace switch, which contains buttons for changing to other workspaces. In Figure 4-4 on page 93 we provide a brief description of the main panel action tasks.

Clock: Displays the current time.

Calendar: Provides facilities for the user to manage appointments.

File Manager: Provides facilities to graphically navigate file systems.

Text Editor: Provides a simple, easy-to-use editor.

Mailer: Enables the user to send, receive, and manage e-mail.

Workspace Switch: Enables the user to switch workspaces.

Print Manager: Allows the user to view and manage print jobs.

Style Manager: Allows the user to customize the desktop.

Applications Manager: Enables the user to manage applications.

Help Manager: Provides a help guide for the desktop.

Trash Can: Allows the user to place files without deleting them.

Figure 4-4 CDE main panel action tasks

Subpanels

If a control in the main panel has an arrow button on top of it, then that control has a subpanel. Below these arrows are icons that allow us to execute different desktop administrative tasks. A subpanel example is provide in Figure 4-5 on page 94.

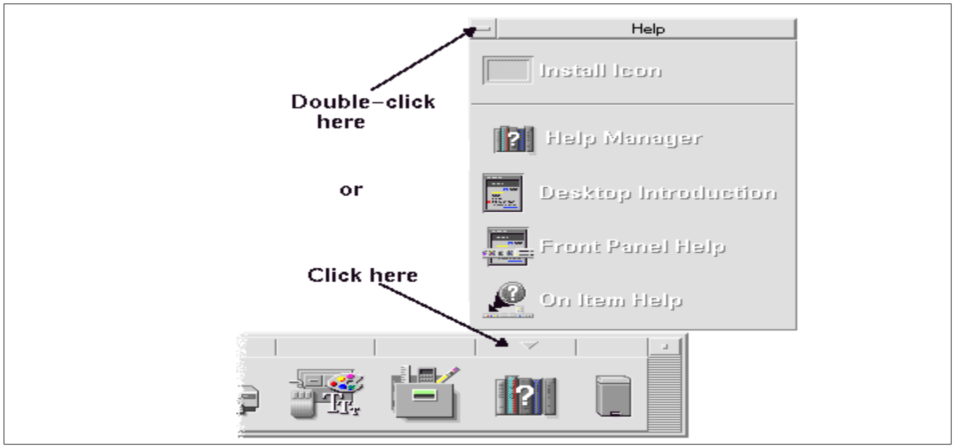

Figure 4-5 CDE subpanel example

4.5 The KDE desktop

The KDE desktop is an Open Source graphical desktop environment and a set of desktop applications. KDE is highly configurable and can be adapted to look and feel like other graphical interfaces. The KDE package also provides a collection of X applications and an office application suite called Koffice.

The user interface part of KDE is built on top of the X Window System and consists of the following application groups:

KDE-Libs Various run-time libraries, such as **`kdecore`**, **`kdeui`**, and **`khtwm`**

KDE-Core **`KPanel`**, **`Kfm`**, **`Kcontrol`**, **`Konqueror`**, **`Kdisplay`**, **`Organizer`**, and **`KDEHelp`**

KDE-Graphics **`Kpaint`**, **`Kdvi`**, **`KGhostview`**, and **`Kfax`**

KDE-Utilities **`Kedit`**, **`Kcal`**, and **`Knotes`**

KDE-Admin **`KPackage`**, **`KUser`**, and KDE System Guard (Task Manager and a Performance Monitor)

KDE-Network **`kmail`**, Windows Shares (SMB client), **`Korn`** (KDE mail checker), and **`KNode`** (news reader)

KDE-Games **`Kasteroids`**, **`Konquest`**, **`Tron`**, **`Smiletris`**, and **`SnakeRace`**

For more information about KDE, please refer to the KDE Web site:

`http://www.kde.org`

4.5.1 KDE desktop layout

The KDE desktop consists of the *main panel* with the *taskbar* and the *desktop* itself.

Main panel

The KDE main panel is a special desktop window that contains a set of controls for performing common tasks. The main panel at the bottom of the screen is used to start applications and switch between desktops. A large K icon (on the left side of the panel) displays a menu of applications to start when clicked.

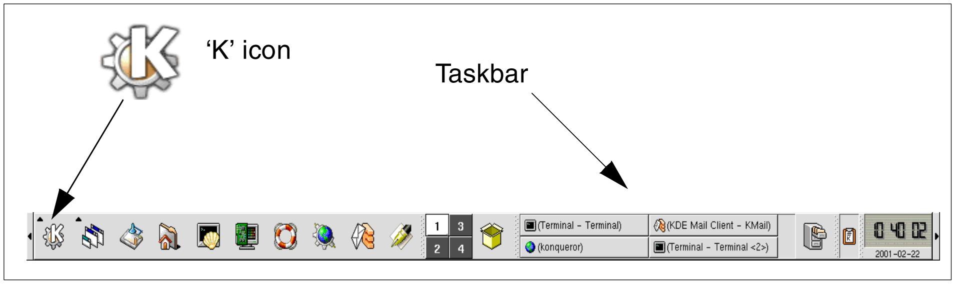

Figure 4-6 KDE desktop main panel and taskbar

Taskbar

The taskbar at the bottom-center of the screen is used to switch between and manage currently running applications. Click an application on the taskbar to switch to the application, as shown in Figure 4-6.

Desktop

The desktop itself, on which frequently used files and folders may be placed. KDE provides multiple desktops, each of which has its own or shared windows. Click the numbered buttons on the panel to switch between desktops, as shown in Figure 4-7 on page 96.

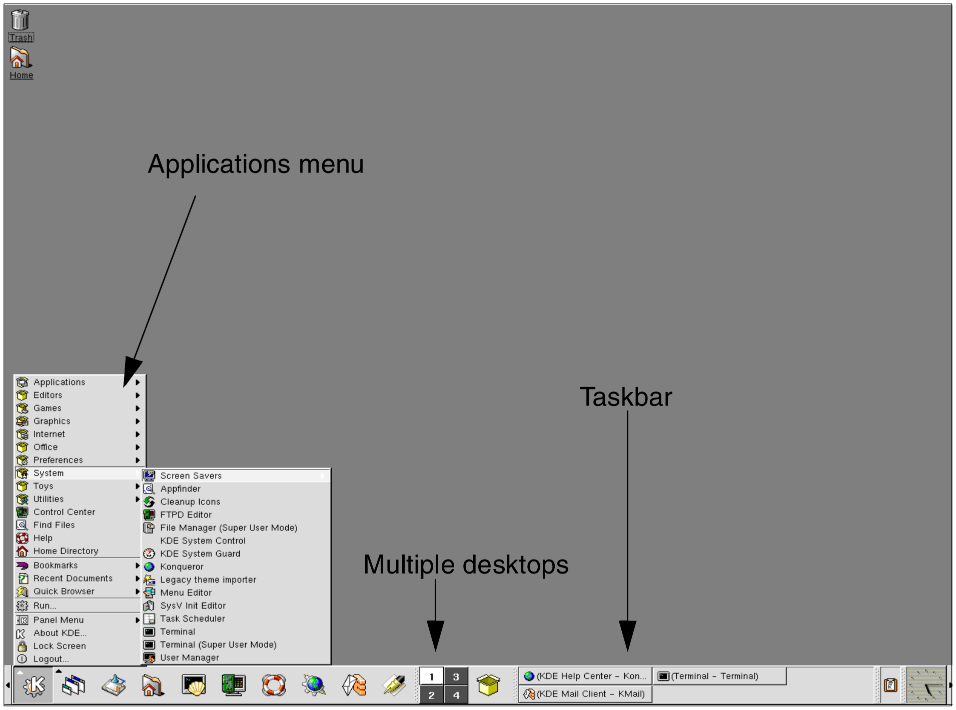

Figure 4-7 KDE desktop and its main panel

4.5.2 KDE applications

KDE provides a set of applications to customize the behavior, functionality, look, and feel of the desktop. The office applications (KOffice) provided with KDE include:

KWord	A frame-based word processor capable of professional-standard documents
KSpread	A powerful spreadsheet application
KPresenter	A full-featured presentation program
KChart	An integrated graph and chart drawing tool
KFormula	A powerful formula editor
Kivio	A flowcharting application
Kontour	A vector-drawing application
Kugar	A tool for generating business-quality reports

Figure 4-8 and Figure 4-9 show sample snapshots from some of the KOffice applications. The following samples where taken using Ksnapshot (a KDE utility used to capture images).

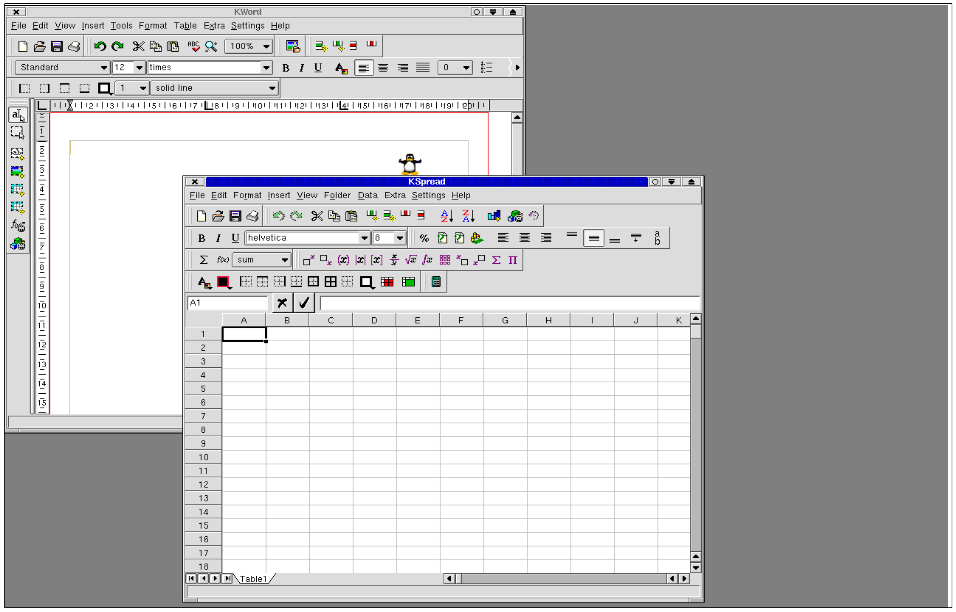

Figure 4-8 KOffice sample 1

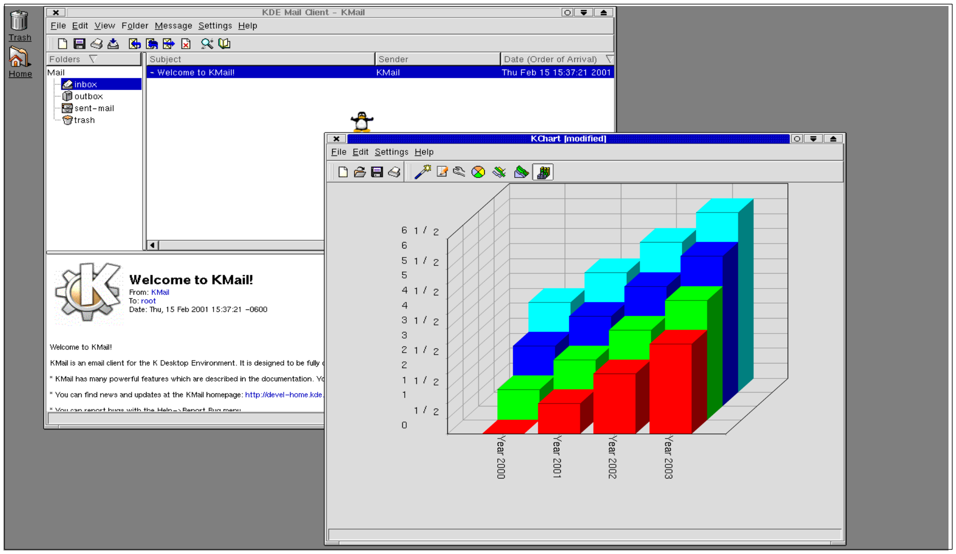

Figure 4-9 KOffice sample 2

To customize a frequently used application, add it to the application Starter menu by selecting **Application starter** → **Configure Panel** → **Add** → **Button** → **System** → **File Manager (Super User Mode)**.

Figure 4-10 shows the sequence in a graphical manner. Figure 4-11 on page 99 shows the result when the File Manager is finally added to the main panel.

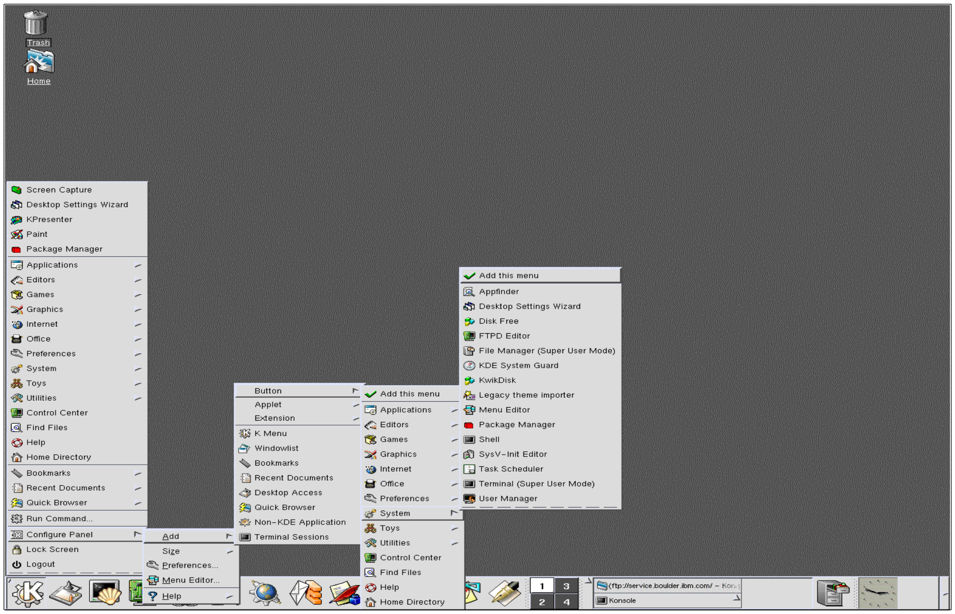

Figure 4-10 Adding an application to the KDE starter menu

We can now launch the File Manager application from the Starter menu by clicking the **Add** button. The result is shown in Figure 4-11 on page 99.

Since we have used the application KDE File Manager throughout our example, it is important to notice one of the great features the KDE File Manager application provides, which is the capability of accessing a Uniform Resource Locator (URL) from the Internet directly, and, by drag and drop, copy the remote file or complete directory to a local destination. This type of technology is used throughout the KDE desktop and is called network transparency. It allows KDE applications to drag and drop an icon from the File Manager to an editor or folder.

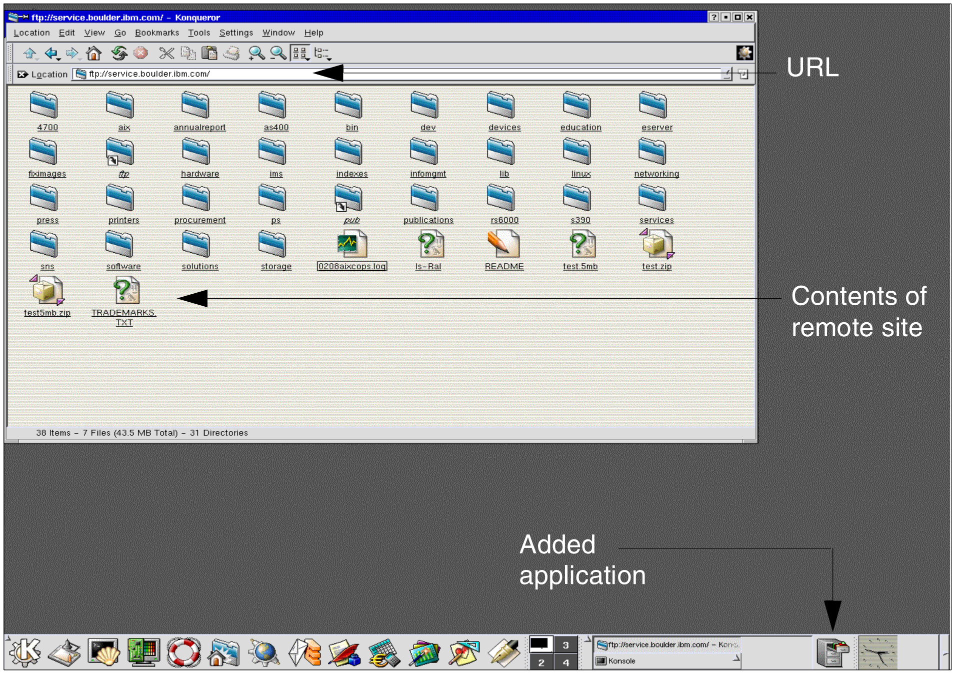

Figure 4-11 KDE File Manager main window

4.6 The Gnome desktop

The GNU Network Object Model Environment (Gnome) is an Open Source graphical desktop environment and a set of desktop applications. Gnome is highly configurable and can be adapted to look and feel like other graphical interfaces.

Gnome was originally developed to allow non-technical users to be able to use the power of UNIX systems; “Gnome is aimed at regular users—for home users or kids or secretary-people. It's for people who are not trained to be programmers", Miguel de Icaza (founder of the Gnome project).

The user interface part of Gnome is built on top of the X Window System and consists of the following application groups:

Gnome desktop A set of tools that provides a powerful desktop interface to users, plus various utility applications for day-to-day work

Gnome applications Various productivity applications such as word processor (Go), spreadsheet (Gnumeric), calendar (Gnomecal), and

	a photo retouching, image composition and image authoring tool (Gimp).
Gnome libraries	A set of libraries that ensures that Gnome applications look and behave properly.

For more information about Gnome, please refer to the Gnome Web site:

`http://www.gnome.org`

4.6.1 Gnome desktop layout

The Gnome desktop includes a set of standard desktop tools and applications, and a panel for starting applications and displaying status. An example of the Gnome desktop is shown in Figure 4-12.

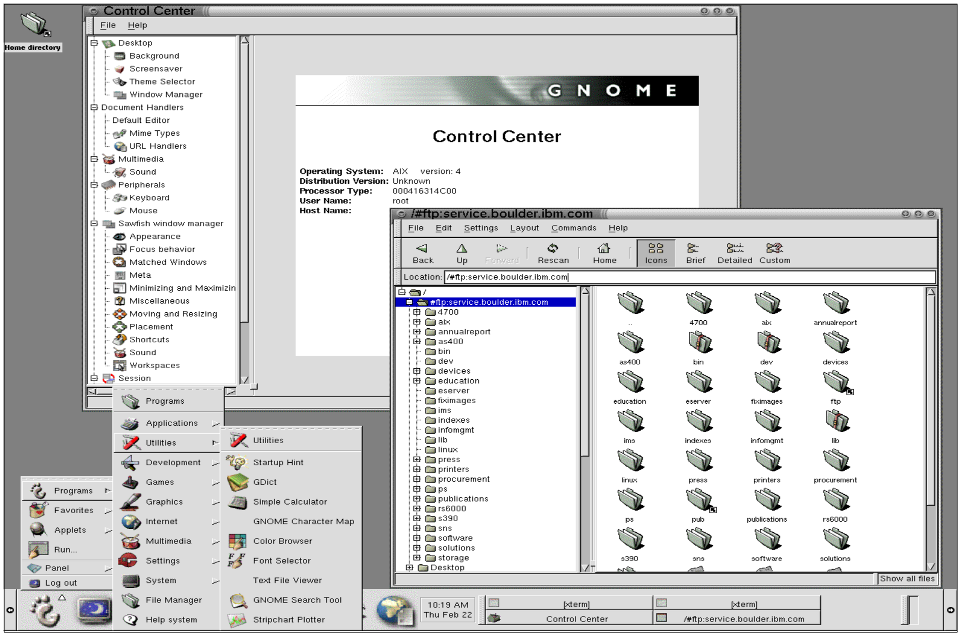

Figure 4-12 Gnome desktop

The Gnome *footprint*, shown in Figure 4-13 on page 101, allows you to launch all of Gnome's features, such as applications, configuration tools, the command line prompt, and logout and lock screen commands.

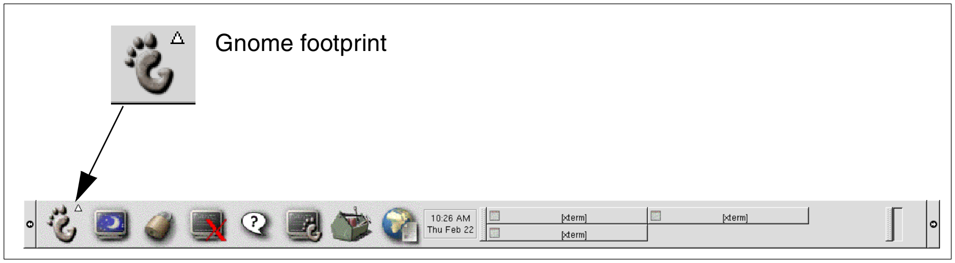

Figure 4-13 Gnome panel

4.6.2 GnomeRPM

The Toolbox includes the gnorpm tool, which is a graphical user interface for the RPM Package Manager that can be used to manage RPM packages much like the **rpm** command.

Figure 4-14 displays the main package window. It has a tree of the different package groups on the left and a list of packages in the selected group on the right. The package list on the right can be configured to display as icons or as a list. From this window, you can manipulate the packages that are currently installed in the system.

Figure 4-14 GnomeRPM (gnorpm) main window

After selecting a package, you can uninstall, query, or verify it. Any of these operations can be performed either from the menu items or from the toolbar buttons. The gnorpm Install window is used to install new packages on the system, as shown in Figure 4-15.

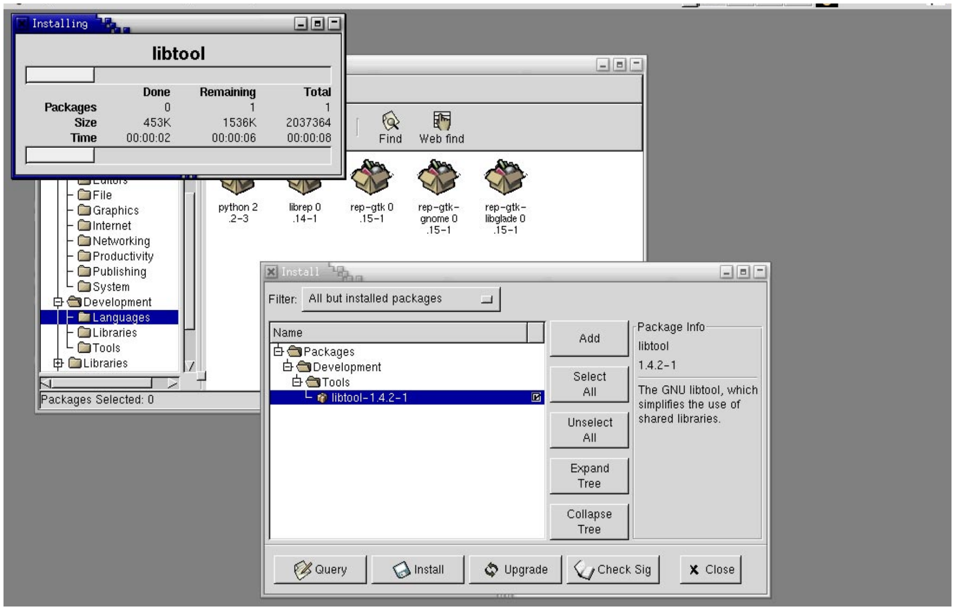

Figure 4-15 GnomeRPM (gnorpm) Install window

4.7 Installing KDE and Gnome

Installing KDE or Gnome on AIX can be done either during the AIX base operating system install, or later from the AIX Toolbox for Linux CD-ROM or Web site.

Note: The current versions of the Gnome and KDE desktops in the Toolbox are Version 1.4 and 2.2.1, respectively.

4.7.1 How to install a desktop during AIX 5.2 base installation

To install a desktop during installation of the AIX base operating system, you can select from NONE, CDE, GNOME, or KDE. The installation options are available by typing 3 in the More Options field on the Installation and Settings screen, shown in Figure 4-16 on page 103.

```
                 Install Options

 1.  Desktop.......................................................... NONE, CDE, GNOME, KDE
 2.  Enable Trusted Computing Base.................................. No
 3.  Graphics Software.............................................. Yes
 4.  Documentation Services Software................................ Yes
 5.  Enable System Backups to install any system..................... Yes
     (Installs all devices and kernels)

>>> 6.  Install More Software

    0  Install with the current settings listed above.

    88  Help ?
    99  Previous Menu

>>> Choice [6]:
```

Figure 4-16 The AIX 5.2 overwrite installation menu, on 32-bit systems

The default is CDE for new and complete overwrite installations. If you select NONE, a minimal configuration is installed including X11, Java, Perl, SMIT, and the Web-based System Manager (if Graphics Software is selected).

If you select GNOME or KDE, the BOS installation process prompts you for the AIX Toolbox for Linux Applications CD. If this CD-ROM is not available, you can type `q` to continue the installation without it.

How to disable CDE auto-start

To disable CDE from being used as the graphical login environment, use the **dtconfig** command (or **smit dtconfig**):

```
/usr/dt/bin/dtconfig -d
```

Note that **dtlogin** will be disabled and will not auto-start after the next system restart. If you want to terminate the CDE graphical login environment immediately, send the software termination signal, SIGTERM (15), to the currently running **dtlogin** daemon:

```
cat /var/dt/Xpid | xargs kill -term
```

How to enable CDE auto-start

To enable CDE to be used as the graphical login environment, use the **dtconfig** command (or **smit dtconfig**):

```
/usr/dt/bin/dtconfig -e
```

Note that **dtlogin** will auto-start after the next system restart. If you want to activate the CDE graphical login environment immediately, use the **dtlogin** command:

```
/usr/dt/bin/dtlogin -daemon
```

4.7.2 How to install and customize KDE from the Toolbox

The installation of KDE on AIX can be divided into two main steps:

- Installing the KDE software
- Configuring the KDE environment

> **Note:** If you have outdated KDE or dependent RPM packages installed already, either uninstall them with the -e option before installing the KDE software, install using the -U option, or add the --force option during installation.

How to install the KDE software

To install the KDE software you can either:

- Install the KDE packages from the AIX Toolbox for Linux Applications CD-ROM, with the **rpm** command.
- Download the KDE packages from the Toolbox FTP site to your system and then use the **rpm** command to install them.
- Use the **rpm** command to install the KDE packages directly from the Toolbox FTP site.

> **Note:** Remember to set /usr/linux/bin first in the PATH variable and that the /var file system (/var/tmp) has enough free space (> 150 MB). Also make sure that the /opt or /opt/freeware file system has enough free space.

Download and install KDE software from the Toolbox FTP site

To download the KDE and dependent packages from the Toolbox FTP site, you can use the following *easy install* Korn shell scripts:

getdesktop.base.sh Downloads common support programs for Gnome and KDE to the current directory

getkde2.all.sh Downloads the KDE Desktop Base and Applications to the current directory

getkde2.base.sh Downloads the KDE Desktop Base to the current directory

getkde2.opt.sh Downloads the KDE Desktop Applications to the current directory

The getkde2.all.sh script performs the same download as the getkde2.base.sh and getkde2.opt.sh scripts together.

All of the scripts use the **wget** command to download the KDE packages to your system. The scripts are available from the Toolbox FTP site ezinstall directory:

```
ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/ezinstall
```

To download and install the KDE Desktop Base from the Toolbox FTP site, use the following procedure:

1. **mkdir /opt/freeware/src/tools**
2. **cd /opt/freeware/src/tools**
3. **ncftpget ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/ezinstall/ppc/getdesktop.base.sh**
4. **ncftpget ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/ezinstall/ppc/getkde2.base.sh**
5. **cd /opt/freeware/src/packages/RPMS/ppc**
6. **ksh /opt/freeware/src/tools/getdesktop.base.sh**
7. **ksh /opt/freeware/src/tools/getkde2.base.sh**
8. **cd /opt/freeware/src/packages/RPMS/ppc/desktop.base && rpm -iv ***
9. **cd /opt/freeware/src/packages/RPMS/ppc/kde2.base && rpm -iv ***

You can also download KDE and dependent packages using the getkde2.all.sh or getkde2.opt.sh scripts in a similar way.

Note: If your **rpm -iv *** command, in steps 7 and 8 above, does not work properly, retry the following *for* loop until all packages are installed (and displayed as *already installed*):

```
for f in *.rpm;do rpm -iv $f;done
```

Install KDE software directly from the Toolbox FTP site

To use the **rpm** command to install the KDE packages directly from the Toolbox FTP site, prefix the package name using the following FTP URL as a template:

ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/*package directory*/*package*.rpm

Please refer to Chapter 2, “AIX Toolbox for Linux Applications” on page 15, for more information on installation options.

To install the KDE Desktop Base with the **rpm** command directly from the Toolbox FTP site, use the following procedure:

1. **mkdir /opt/freeware/src/tools**
2. **cd /opt/freeware/src/tools**
3. **ncftpget ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/ezinstall/ppc/getdesktop.base.sh**
4. **ncftpget ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/ezinstall/ppc/getkde2.base.sh**
5. **cd /opt/freeware/src/packages/RPMS/ppc**
6. **rpm -ivh $(cat $OLDPWD/getdesktop.base.sh $OLDPWD/getkde2.base.sh | grep ftp)**

You can also install KDE and dependent packages using the getkde2.all.sh or getkde2.opt.sh scripts in a similar way.

The next example shows how the **rpm** command would be executed (note that since this is a Internet-connected FTP server, the **rpm** command can fail using this method). The command used Figure 4-1 in was:

```
rpm -iv $(grep ftp getkde2.base.sh)
```

Example 4-1 Sample RPM installation of KDE from the Toolbox FTP server

```
rpm -iv ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/expat/
expat-1.95.2-1.aix4.3.ppc.rpm \
ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/kdebase/kdebase
-2.2.1-5.aix4.3.ppc.rpm \
ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/kdelibs/kdelibs
-2.2.1-3.aix4.3.ppc.rpm \
ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/kdelibs/kdelibs
-sound-2.2.1-3.aix4.3.ppc.rpm \
ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/libmng/libmng-1
.0.3-1.aix4.3.ppc.rpm \
ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/libxml2/libxml2
-2.4.19-1.aix4.3.ppc.rpm \
ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/pcre/pcre-3.7-1
.aix4.3.ppc.rpm \
ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/python/python-2
.2-3.aix4.3.ppc.rpm \
ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/qt/qt-2.3.1-1.a
ix4.3.ppc.rpm \
```

ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/xpm/**xpm-3.4k-2.aix4.3.ppc.rpm**

How to configure KDE with manual start

To configure your X Window System environment to use the KDE desktop after login using the ASCII Low Function Terminal (LFT), you can edit your personal X server client customization file (~/.xinitrc) and then start the X server with the **xinit** or **startx** command.

To customize your personal X server client customization file, copy part of the /usr/lpp/X11/defaults/xinitrc file to your home directory naming it .xinitrc:

```
head -124 /usr/lpp/X11/defaults/xinitrc > ~/.xinitrc
```

Add the following two lines to the newly created .xinitrc file:

```
export PATH=/opt/freeware/kde/bin:$PATH
exec /opt/freeware/kde/bin/startkde
```

Now we can start the X server in our LFT login session:

```
xinit
```

How to configure KDE with KDM graphical login

To configure your X Window System environment to use the KDE desktop by using KDE Display Manager (KDM) graphical login on your workstation. The **kdm** login manager will allow you to select desktops other than the KDE desktop, such as the Gnome and CDE desktops.

To enable KDM login on your workstation, run the following simple **kdmconf** script (similar to the /usr/lib/X11/xdm/xdmconf script, as the *root* user).

Example 4-2 kdmconf

```
#!/bin/ksh
# ...if the -d option is specified, just delete and stop kdm...
if [[ "$1" = "-d" ]];then
    stopsrc -s kdm -c
    rmssys -s kdm
    rmitab kdm
    exit
fi
# ...check if CDE is enabled, if so remove it...
/usr/dt/bin/dtconfig -d > /dev/null 2>&1
rmitab dt > /dev/null 2>&1
# ...add the kdm subsystem to the System Resource Controller...
if [[ -z "$(odmget -qsubsysname=kdm SRCsubsys)" ]]; then
    mkssys -s kdm -p /opt/freeware/kde/bin/kdm -a \
```

```
       "-nodaemon" -u 0 -S -n 15 -f 9 > /dev/null 2>&1
fi
# ...add the start of kdm to /etc/inittab...
lsitab kdm > /dev/null 2>&1
if [[ "$?" -eq 1 ]]; then
    mkitab "kdm:2:once:/usr/bin/startsrc -s kdm"
fi
```

If we place the kdmconf script in the /local/pbin directory (a file system of our own creation), we execute it with the following command:

```
/local/pbin/kdmconf
```

To customize your personal X server client customization file, copy part of the /usr/lpp/X11/defaults/xinitrc file to your home directory naming it .xinitrc or ~/.xsession:

```
head -124 /usr/lpp/X11/defaults/xinitrc > ~/.xinitrc
```

Add the lines shown in Figure 4-3 to the newly created .xinitrc or ~/.xsession file.

Example 4-3 Lines to add to the .xinitrc or .xsession file

```
[[ -z "$SESSION" ]]  && SESSION=failsafe
case $SESSION in
kde)
        export PATH=/opt/freeware/kde/bin:$PATH
        exec /opt/freeware/kde/bin/startkde ;;
gnome)
        exec /opt/freeware/bin/gnome-session ;;
CDE)
        exec /usr/dt/bin/Xsession ;;
failsafe|*)
        aixterm -geometry 80x35+0-0 -ls &
        exec mwm ;;
esac
```

You can also download and append the xinitrc.sample file from the Toolbox FTP site:

```
ncftpget -c
ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/contrib/xinitrc.samp
le >> ~/.xinitrc
```

Now we can start the kdm login manager for our workstation LFT:

```
startsrc -s xdm
```

After the next Initial Program Load (IPL or system boot), the init program will start the kdm login manager. To remove the kdm IPL configuration, use the **-d** option of the **kdmconf** command:

```
/local/pbin/kdmconf -d
```

How to configure KDE with XDM graphical login

To configure your X Window System environment to use the KDE desktop by using X Display Manager (XDM) graphical login on your workstation, edit your personal X server client customization file (~/.xinitrc or ~/.xsession) and then start the xdm graphical manager with the **startsrc** command.

To enable XDM login on your workstation, run the **/usr/lib/X11/xdm/xdmconf** command (as the *root* user):

```
/usr/lib/X11/xdm/xdmconf
```

To customize your personal X server client customization file, copy part of the /usr/lpp/X11/defaults/xinitrc file to your home directory naming it ~/.xinitrc or ~/.xsession:

```
head -124 /usr/lpp/X11/defaults/xinitrc > ~/.xinitrc
```

Add the following two lines to the newly created .xinitrc file:

```
export PATH=/opt/freeware/kde/bin:$PATH
exec /opt/freeware/kde/bin/startkde
```

Now we can start the xdm login manager for our workstation LFT:

```
startsrc -s xdm
```

After the next Initial Program Load (IPL or system boot), the /etc/rc.tcpip script will start the xdm login manager. To remove the xdm IPL configuration, use the -d option to the **xdmconf** command:

```
/usr/lib/X11/xdm/xdmconf -d
```

4.7.3 How to install and customize Gnome from the Toolbox

The installation of Gnome on AIX can be divided into two main steps:

- Installing the Gnome software
- Configuring the Gnome environment

> **Note:** If you have Gnome or dependent RPM packages installed already, either uninstall them with the -e option before installing the KDE package, install using the -U option, or add the --force option during installation.

How to install the Gnome software

To install the Gnome software you can either:

- Install the Gnome packages from the AIX Toolbox for Linux Applications CD-ROM, with the **rpm** command.
- Download the Gnome packages from the Toolbox FTP site to your system, and then use the **rpm** command to install them.
- Use the **rpm** command to install the Gnome packages directly from the Toolbox FTP site.

> **Note:** Remember to set /usr/linux/bin first in the PATH variable and that the /var file system (/var/tmp) has enough free space (> 150 MB). Also make sure that the /opt or /opt/freeware file system has enough free space.

Download and install Gnome software from the Toolbox FTP site

To download the Gnome and dependent packages from the Toolbox FTP site, you can use the following *easy install* Korn shell scripts:

getdesktop.base.sh Downloads common support programs for Gnome and KDE to the current directory

getgnome.base.sh Downloads the Gnome Desktop Base to the current directory

getgnome.apps.sh Downloads the Gnome Desktop Applications to the current directory

All of the scripts use the **wget** command to download the Gnome packages to your system. The scripts are available from the Toolbox FTP site ezinstall directory:

`ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/ezinstall`

To download and install the Gnome Desktop base from the Toolbox FTP site, use the following procedure:

1. **`mkdir /opt/freeware/src/tools`**
2. **`cd /opt/freeware/src/tools`**
3. **`ncftpget ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/ezinstall/ppc/getdesktop.base.sh`**
4. **`ncftpget ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/ezinstall/ppc/getgnome.base.sh`**
5. **`cd /opt/freeware/src/packages/RPMS/ppc`**

6. **ksh /opt/freeware/src/tools/getdesktop.base.sh**
7. **ksh /opt/freeware/src/tools/getgnome.base.sh**
8. **cd /opt/freeware/src/packages/RPMS/ppc/desktop.base && rpm -iv ***
9. **cd /opt/freeware/src/packages/RPMS/ppc/gnome.base && rpm -iv ***

You can also download the Gnome applications, and dependent packages, with the getgnome.apps.sh script, in a similar way.

> **Note:** If your **rpm -iv *** command, in step seven (7) and eight (8) above, does not work properly. Retry the following `for` loop until all packages are installed (are displayed as *already installed*):
>
> ```
> for f in *.rpm;do rpm -iv $f;done
> ```

Install Gnome software directly from the Toolbox FTP site

To use the **rpm** command to install the Gnome packages directly from the Toolbox FTP site, prefix the package name using the following FTP URL as a template:

`ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/`*`package directory`*`/`*`package`*`.rpm`

Please refer to Chapter 2, “AIX Toolbox for Linux Applications” on page 15, for more information on installation options.

To install the Gnome Desktop Base with the **rpm** command directly from the Toolbox FTP site, use the following procedure:

1. **mkdir /opt/freeware/src/tools**
2. **cd /opt/freeware/src/tools**
3. **ncftpget ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/ezinstall/ppc/getdesktop.base.sh**
4. **ncftpget ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/ezinstall/ppc/getgenome.base.sh**
5. **cd /opt/freeware/src/packages/RPMS/ppc**
6. **rpm -ivh $(cat $OLDPWD/getdesktop.base.sh $OLDPWD/getgnome.base.sh | grep ftp)**

You can also install the Gnome applications and dependent packages with the getgnome.apps.sh script, in a similar way.

Example 4-4 shows how the **rpm** command would be executed (note that since this is an Internet-connected FTP server, the **rpm** command can fail using this method). In the example we use Korn/POSIX shell syntax.

Example 4-4 Sample RPM installation of Gnome from the Toolbox FTP server

```
root@fenris:/opt/freeware/src/tools: rpm -iv $(grep ftp getgnome.base.sh)
Retrieving
ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/gconf/GConf-1.0
.7-2.aix4.3.ppc.rpm
Retrieving
ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/ORBit/ORBit-0.5
.12-1.aix4.3.ppc.rpm
Retrieving
ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/control-center/
control-center-1.4.0.4-1.aix4.3.ppc.rpm
Retrieving
ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/esound/esound-0
.2.24-1.aix4.3.ppc.rpm
Retrieving
ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/gdk-pixbuf/gdk-
pixbuf-0.11.0-1.aix4.3.ppc.rpm
Retrieving
ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/glib/glib-1.2.1
0-2.aix4.3.ppc.rpm
...(lines omitted)...
```

How to configure Gnome with manual start

To configure your X Window System environment to use the Gnome desktop after login using the ASCII Low Function Terminal (LFT), you can edit your personal X server client customization file (~/.xinitrc) and then start the X server with the **xinit** or **startx** command.

To customize your personal X server client customization file, copy part of the /usr/lpp/X11/defaults/xinitrc file to your home directory naming it .xinitrc:

```
head -124 /usr/lpp/X11/defaults/xinitrc > ~/.xinitrc
```

Add the following lines to the newly created .xinitrc file:

```
exec /opt/freeware/bin/gnome-session
```

Now we can start the X server in our LFT login session:

```
xinit
```

How to configure Gnome with XDM graphical login

To configure your X Window System environment to log into your Gnome desktop by using X Display Manager (XDM) graphical login on your workstation, edit your personal X server client customization file (~/.xinitrc or ~/.xsession) and then start the xdm graphical manager with the **startsrc** command.

To enable XDM login on your workstation, run the **/usr/lib/X11/xdm/xdmconf** command (as the *root* user):

```
/usr/lib/X11/xdm/xdmconf
```

To customize your personal X server client customization file, copy part of the /usr/lpp/X11/defaults/xinitrc file to your home directory naming it ~/.xinitrc or ~/.xsession:

```
head -124 /usr/lpp/X11/defaults/xinitrc > ~/.xinitrc
```

Add the following two lines to the newly created .xinitrc file:

```
exec /opt/freeware/bin/gnome-session
```

Now we can start the xdm login manager for our workstation LFT:

```
startsrc -s xdm
```

After the next Initial Program Load (IPL or system boot), the /etc/rc.tcpip script will start the xdm login manager. To remove the xdm IPL configuration, use the -d option of the **xdmconf** command:

```
/usr/lib/X11/xdm/xdmconf -d
```

How to configure Gnome with KDM or GDM graphical login

Instead of using XDM for graphical login, you can use KDM. Please refer to “How to configure KDE with KDM graphical login” on page 107.

You can also use the Gnome Display Manager (GDM) graphical login manager on your workstation. GDM is basically a replacement for XDM. This software is not available from the Toolbox at this time, but for more information on the GDM, please refer to the following Web site:

http://www.gnu.org/directory/gdm.html

5

Native Linux on pSeries

Besides recompiling Linux applications using the AIX Toolbox for Linux Applications and thus creating native AIX applications from them, there is an alternative way to get Linux applications running on pSeries hardware: Installing the Linux operating system natively on pSeries.

This chapter describes the current status of native Linux on pSeries, what distributions are available and what pSeries models they do support. It also describes how to install Linux on various types of pSeries systems (deskside and rack p610 and high-end p690 LPAR), including boot configurations and installing various applications (IBM software as well as Open Source Software).

5.1 Introduction

While the AIX Toolbox for Linux Applications offers an excellent way to make Linux applications available on pSeries systems, there might be cases where it is preferred to have them running under a native Linux operating system instead of AIX. One example might be OpenOffice, which has not yet been provided with the Toolbox, but can easily be installed under Linux on pSeries, as shown in “OpenOffice” on page 142.

Linux on pSeries is a key element of the IBM eServer Linux strategy. Please see the following Web site for more information about Linux at IBM in general:

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

See the following Web site for Linux on IBM eServers:

`http://www.ibm.com/eserver/linux`

And especially the following for information about Linux on pSeries (including whitepapers, how-to documents, and a lot of additional information):

`http://www.ibm.com/eserver/pseries/linux`

Compatibility of Linux and AIX

Because Linux and AIX use different Application Binary Interfaces (ABIs) (like Linux on different hardware platforms uses different ABIs), there is in general no binary compatibility when changing operating systems or hardware architectures. For example:

- Linux applications that have been compiled under Linux on hardware other than IBM pSeries or IBM iSeries can in general *not run* under Linux for pSeries without recompilation.
- Linux applications that have been compiled under Linux for pSeries *cannot run* under AIX, including the AIX Toolbox for Linux Applications.
- Linux applications that have been compiled under AIX using the AIX Toolbox for Linux Applications *cannot run* under Linux for pSeries.

A somewhat special case are pSeries and iSeries systems, as they are based on the same processor architecture, and thus binaries compiled under Linux for pSeries can run under Linux for iSeries and vice versa. In fact, Linux for pSeries and Linux for iSeries are almost identical, except for some differences in installation procedures and hardware support.

Please note that these statements apply to *binary* compatibility and not *source code* compatibility. In all cases mentioned above, there is full source code compatibility, which means that for applications that have been compiled on one architecture it should be possible to recompile them on the other architectures without any changes to the source code. The term architecture in this case describes any hardware/operating system combination mentioned above. However, a prerequisite for utilizing source code compatibility is having access to the source code, which might not always be given.

Linux kernel for PowerPC processors

The sources for the Linux kernel, which can be found at the following Web site, contain all necessary code for execution on pSeries.

http://www.kernel.org/

A first port of the Linux kernel to PowerPC processors, which are the heart of pSeries systems, has been developed by the Open Source community a couple of years ago. Later, IBM took an active, leadership role in Linux for the PowerPC.

The terms *PowerPC* and *PowerPC processor* are used in this book as a synonym for all processors that are based on the PowerPC architecture (that is, PowerPC 601, PowerPC 603, PowerPC 604, PowerPC 604e, RS64 II, RS64 III, RS64 IV, Power2, Power3, and Power4).

Currently, there are two main projects related to the Linux kernel on the PowerPC processor architecture:

http://penguinppc.org/

This project provides a 32-bit Linux kernel for PowerPC-based systems. It can be used on a large variety of pSeries, iSeries, RS/6000 machines, Power Macintoshes, PowerBooks, iMacs, and also the Motorola 8xx and IBM 4xx embedded processors. Please see the following Web sites for a listing of supported systems:

http://penguinppc.org/projects/hw/
http://oss.software.ibm.com/developer/opensource/linux/projects/ppc/models.php

http://penguinppc64.org/

This project provides a 64-bit kernel for 64-bit PowerPC based hardware, including Power3- and Power4-based IBM pSeries and iSeries systems. While the kernel has had full 64-bit support, the application address spaces remained 32-bit for some time. A full 64-bit environment has been released recently.See the introductory statements in 5.2, “Installation of native Linux on pSeries” on page 120.

For information about the IBM participation in the two mentioned projects see the following Web site:

http://oss.software.ibm.com/developer/opensource/linux/projects/ppc/

Older Power, RS64 II and Microchannel (MCA) based systems are not supported by either of the projects mentioned above. For further information about Linux on these machines see the following Web site:

http://www.sjdjweis.com/linux/rs6k/

While the 32-bit PowerPC Linux kernel could run on 64-bit PowerPC hardware, this is not for all hardware tested and might not be supported by some of the distributors. IBM's focus will be the 64-bit kernel, while the 32-bit kernel will be put into maintenance mode by IBM (other parties might still enhance the 32-bit PowerPC Linux kernel).

Note: All binaries that have been compiled for the 32-bit kernel can be executed under the 64-bit kernel, as the exact same ABI is used for 32-bit applications in both kernels. Of course the ABI for 64-bit applications is unique to the 64-bit kernel.

To find out what processor type your system is based on, the "Facts and Features" brochures from IBM might be a good reference. See the following Web sites:

http://www.ibm.com/servers/eserver/pseries/hardware/factsfeatures.html
http://www.ibm.com/servers/eserver/pseries/hardware/workstations/facts_features.html

In addition to running Linux on a whole pSeries system, it is also possible to run Linux in an LPAR of an LPAR-capable system, like the p690 ("Regatta") and p670 models. AIX and Linux can be installed in the LPARs of such a system in any combination. This includes a case in which all of the LPARs are running Linux.

AIX is neither required to partition the machine nor to install and run Linux. Nevertheless, AIX provides much greater features for hardware analysis, intermittent fault detection, error logging, etc. than Linux currently does.

I/O device and adapter support

Fortunately, the variety of adapters that needs to be enabled for Linux for pSeries is much smaller than in the Intel world. In addition to the base device support (that is, SCSI, Ethernet, etc.) the following adapters are currently enabled:

- 2104 Expandable Storage Plus

- 2605 Diskette Drive
- 2624 CD-ROM 32x/40x
- 2830 GXT130P Graphics
- 2848 GXT135P Graphics
- 2968 Ethernet 10/100
- 2969 Gigabit Ethernet (Fiber)
- 2975 Gigabit Ethernet (UTP)
- 4961 4-port Ethernet 10/100
- 6158 Tape - 4 mm Internal
- 6203 Ultra3 SCSI Differential
- 6204 Ultra SCSI Differential

Drivers for additional adapters are available, but were not thoroughly tested at the time of writing this book. Please check the Linux for pSeries whitepaper at the following Web site and the two project home pages mentioned in the previous paragraph for the latest updates.

http://www.ibm.com/servers/eserver/pseries/linux/whitepapers/linux_pseries.html

Distributions

As with Linux on other hardware platforms, it is very common to use a Linux distribution instead of downloading and installing all components of the Linux operating system individually. There are several distributors who have invested a significant amount of work to ensure smooth installation and trouble-free usage of the Linux distributions they provide. Usually a distribution does not only contain the basic components of a Linux system (that is, kernel, compiler, installer, libraries, etc.), but also a large amount of software for various purposes.

The following Web sites refer to several Linux for pSeries distributions that are currently available.

http://penguinppc.org/
http://penguinppc64.org/

For an overview of what pSeries systems are currently supported by which of IBM's Linux distribution partners see:

http://www.ibm.com/servers/eserver/pseries/linux/guide.html

Additional information can be found on the Web sites of the individual distributors:

Debian http://www.debian.org/ports/powerpc/

Red Hat http://www.redhat.com/software/eserver/pseries/

SuSE http://www.suse.com/us/private/products/suse_linux/ppc/

	http://www.suse.com/us/business/products/sles/sles_iSeries_pSeries/
TurboLinux	http://www.turbolinux.com/products/pseries/
UnitedLinux	http://www.unitedlinux.com/

5.2 Installation of native Linux on pSeries

This section provides detailed instructions on how to install Linux on various pSeries systems.

We have chosen SuSE Linux Enterprise Server 8 (SLES 8) as the distribution for these demonstration purposes, as it currently has the broadest hardware support for pSeries systems. See:

http://www.ibm.com/servers/eserver/pseries/linux/guide.html

At the time of writing this book only beta 4 and a first release candidate of SLES 8 were available. As there might be some changes and improvements in the final version compared to the beta code used in the following examples, not all details shown might apply for the final version.

SLES 8 is based on UnitedLinux 1.0; comes with kernel 2.4.19, glibc 2.2.5 and has full 64-bit application support when running the 64-bit kernel on 64-bit; machines.

5.2.1 Linux installation on pSeries deskside models

In this section we describe how to install Linux on entry-level pSeries systems, with and without a graphics display.

Systems with a graphics adapter

We used a pSeries p610 (7028-6E1) deskside model as an example to describe the installation procedure for pSeries systems with a graphics adapter. It had three disks. On hdisk0 AIX 5L was installed, while the two other disks we unused prior to the Linux install.

The procedure to install SuSE SLES 8 is as follows:

1. Boot from the SuSE CD.
2. Choose the Linux kernel to be used.
3. Do some pre-install configurations.
4. Start the installation (a reboot might occur during the install).
5. Do some post-install configurations (network, graphics display, etc.).
6. Define boot configuration and reboot.

Step 1

The first step is to insert the first SLES 8 CD and boot the system to Systems Management Services (SMS). To enter the graphical System Management Services, you must press the F1 key after the keyboard icon displays during startup and before the tone. If you have pressed the F1 key, the System Management Services menu displays after the initialization and power-on self-test (POST) are complete.

Choose the **Multiboot** option, then **Install From** and select the CD. Click **Install** to boot from the CD, and confirm the CD-ROM Identstring **SuSE SLES-8 (PPC)** by clicking it.

See your hardware user's guide for a detailed description of SMS including screenshots. In our case the *eServer pSeries 610 Model 6C1 and Model 6E1 User's Guide*, SA38-0598, can be found at:

```
http://www.ibm.com/servers/eserver/pseries/library/hardware_docs/p610.html
```

Step 2

The second step starts as the yaboot screen now appears. Yaboot is an OpenFirmware bootloader for OpenFirmware based machines and somewhat comparable to the lilo bootloader for Intel PCs. Because the p610 used is a 64-bit system (based on a Power 3-II processor) we chose to use the 64-bit kernel to be used. Reply with **Install** and then select the kernel. The system then loads the selected kernel and a initial system gets loaded.

Step 3

The third step begins as the YaST2, the SuSE installation and administration program, starts. Choose the desired language during install and then change the settings as necessary.

- Installation mode should be New installation.
- Default System is a good start for Software Selection.
- Most important are Partitioning and Booting. If acceptable, go with the suggested defaults. In our case this was not desired, as the proposal was to use the first disk /dev/sda, while we wanted to preserve the AIX 5L installation on that first disk. A custom partition setup was chosen with all three partitions (boot, swap, and root) on the second disk, /dev/sdb. Make sure the booting configuration is such that yaboot gets written to the same disk, /dev/sdb1.

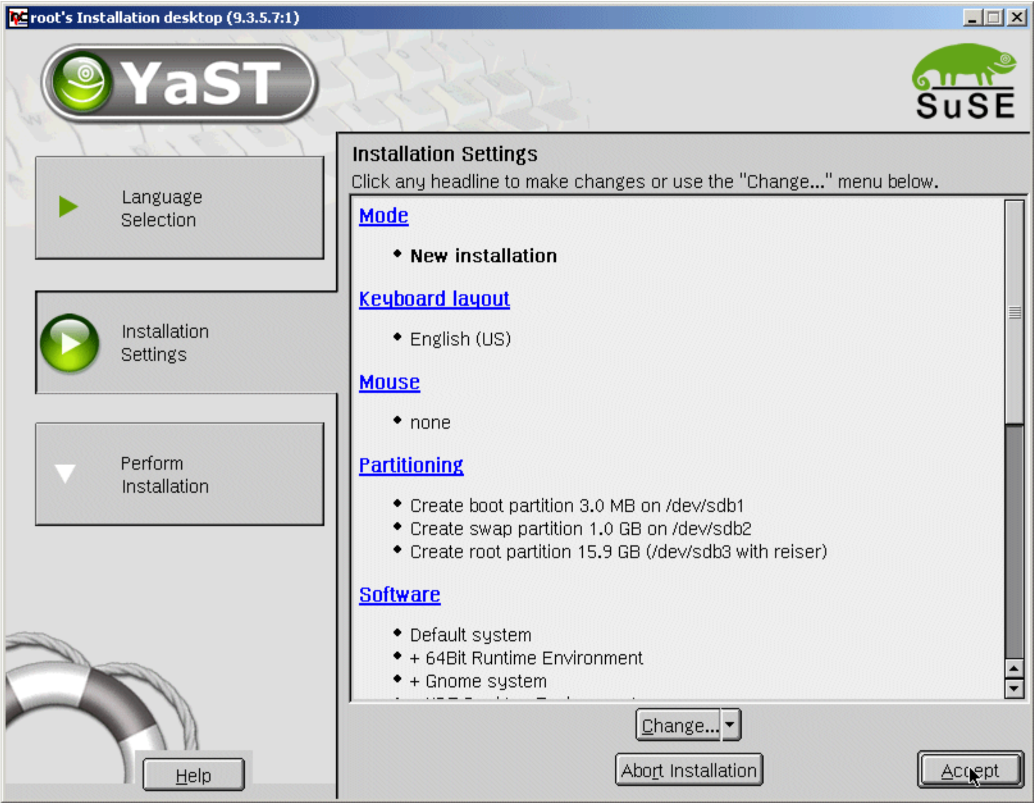

Figure 5-1 Disk partition definition

Step 4

Now, as the fourth step, start the installation by confirming the checkbox as shown in Figure 5-2 on page 123. Wait until the selected software is installed (total installation time in our case was about 15 minutes; see Figure 5-3 on page 124), after which an automatic reboot is initiated. Depending on the software selected, the system might now prompt for CD 2.

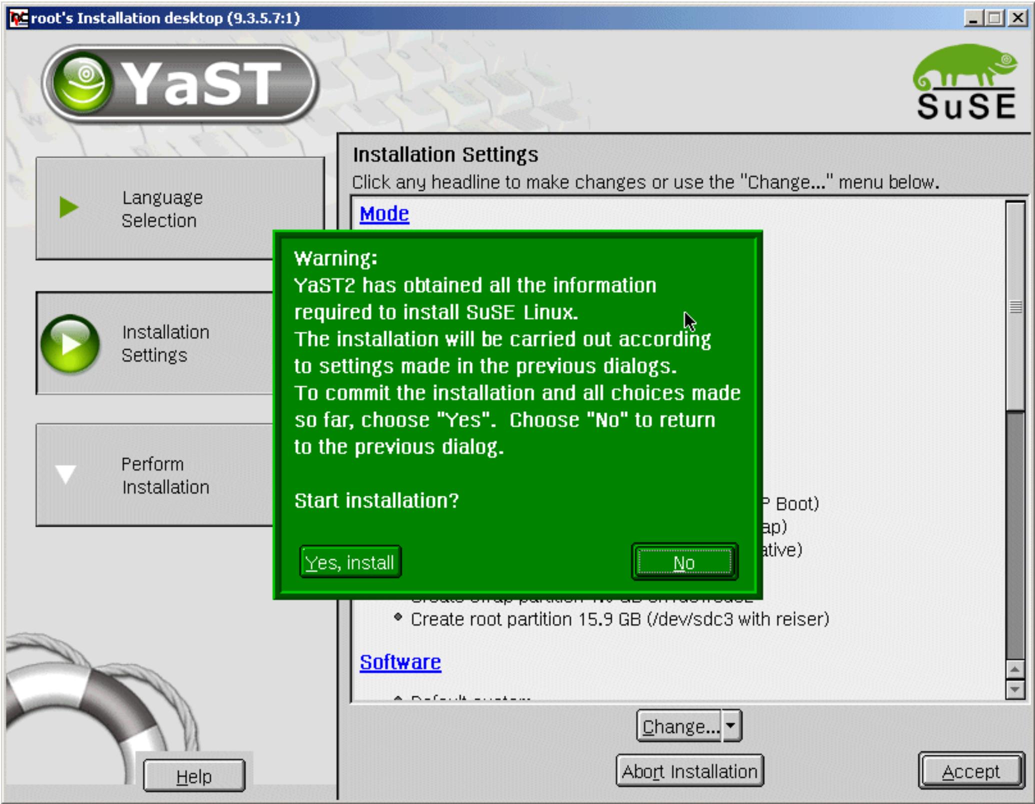

Figure 5-2   Start of software installation process

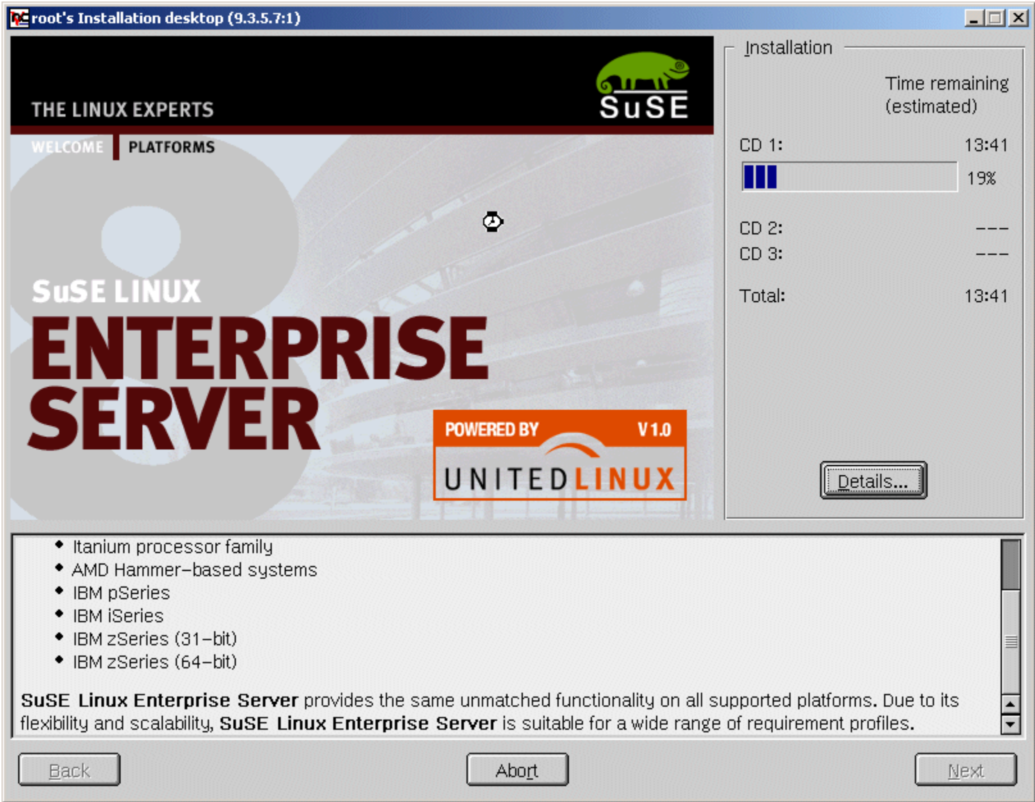

Figure 5-3 Software installation

Step 5

In the fifth step, some final configuration has to be done.

Choose a root password, create users as desired, and wait until the configuration scripts have finished.

At the end, the network interfaces can be configured. In our system a Token Ring adapter, two onboard 10/100 MBit ethernet adapters, and an additional gigabit Ethernet interface were installed. We chose to configure the first Ethernet adapter, eth0, to be used with DHCP, as shown in Figure 5-4 on page 125. If no DHCP server is available in your network, this option should be omitted and the network should be manually configured instead.

The network configuration can later be changed by running yast2 and choosing **Network/Basics** and then **Network card configuration**.

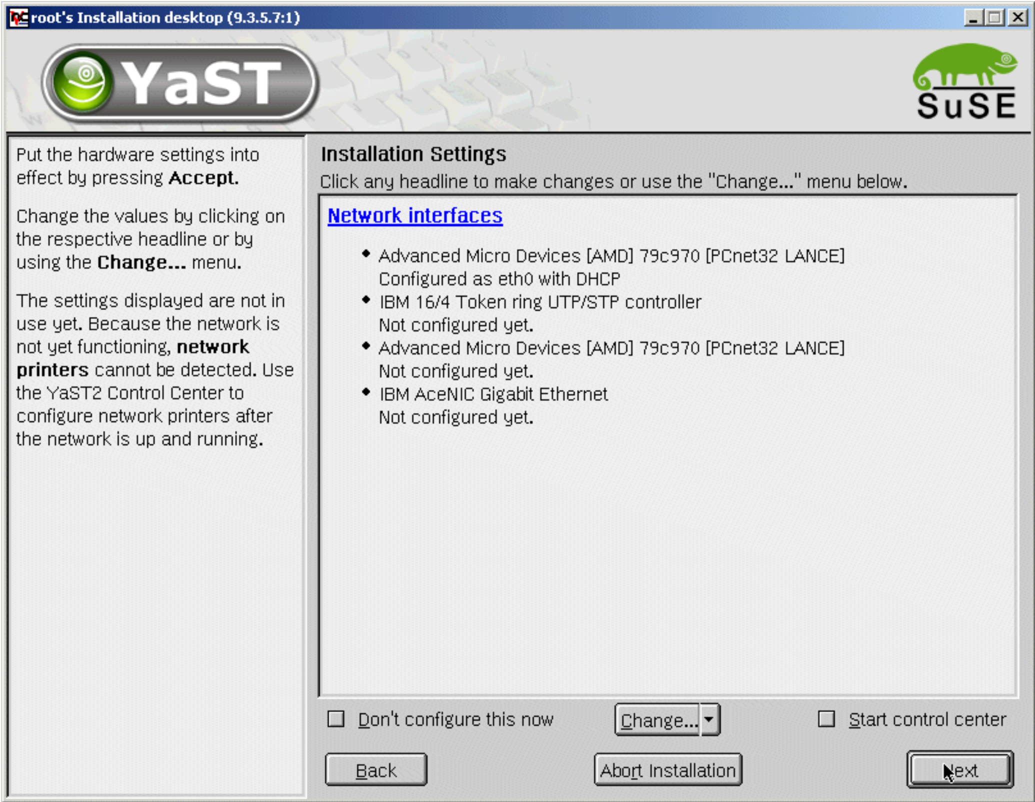

Figure 5-4 Network configuration

You can now do some optional configuration tasks, like configuring the graphics display. Our system had a standard graphics adapter and an IBM P201 monitor. Issue sax2 in a console window to start configuration. We chose **Manual setup** of the monitor and graphics card. Selecting **Properties** and then the IBM P200 monitor, 16-bit color, and 1280x1024 resolution, while not changing the graphics card definition, led to good results. Issuing `init 3` in a command window and then `init 5` on the ascii terminal restarts the graphics subsystem.

If desired, inetd can be started from the Network/Basics menu in yast2.

Step 6

The final step is to adapt the boot configuration if necessary.

The boot configuration is defined in the file /etc/lilo.conf (see Example 5-1 on page 126) and is written into the boot record by running the **`lilo`** command.

See 5.2.5, “Boot configuration and kernel recompile” on page 137, for an in-depth example. Unless in a more complex environment, there are usually no changes necessary to the boot configuration at this point.

Example 5-1 Boot configuration in /etc/lilo.conf

```
# Generated by YaST2

default=linux
timeout=100
boot=/dev/sdb1
activate

image = /boot/vmlinuz
        label = linux
        root = /dev/sdb3
        append = ""
```

After removing the CD, you can now reboot the system with:

```
shutdown -r now
```

Boot to SMS by pressing F1 and select the **Boot Sequence** in the Multiboot you prefer. Save the new configuration and continue the boot process by clicking **Exit** several times.

5.2.2 Using VNC and putty for easier installation

In this section we want to briefly introduce two tools that make installation of Linux on pSeries systems easier, especially on such that do not have a graphics display.

PuTTY: A free Win32 Telnet/SSH client

PuTTY is a free implementation of Telnet and SSH for Win32 platforms, along with an xterm terminal emulator. For further information and for a download of the executable needed, putty.exe, please see its home page:

http://www.chiark.greenend.org.uk/~sgtatham/putty/

For more information about configuration and usage of PuTTY see Section 4.4 in the *Managing AIX Server Farms*, SG24-6606.

SuSE Linux on pSeries provides ssh access already at a very early stage of the installation process, but for security reasons no telnet access. PuTTY runs on Windows-based platforms and can be used as an ssh (or telnet) client.

In case you have Linux running on your desktop instead of Windows, PuTTY is not needed, as most Linux systems come with a working ssh client installed (see `http://www.openssh.org/` for the most common implementation.)

Virtual desktops with VNC

VNC stands for Virtual Network Computing. It is, in essence, a remote display system that allows you to view a desktop environment not only on the machine where it is running, but from anywhere on the Internet and from a wide variety of machine architectures. For further information and to download the appropriate client for your desktop system, see:

http://www.uk.research.att.com/vnc/

All newer VNC servers provide access via HTTP. A small Web server runs on port 58*XX*, where *XX* is the display number of the virtual display. This means that you are able to connect to your VNC session by pointing any Java-capable browser to `http://`*`hostname`*`:5801/`. As the VNC server comes preinstalled on SuSE Linux for pSeries, no download or installation on your client system is necessary.

5.2.3 Systems without a graphics adapter

To simulate a system without a graphics adapter, such as the entry rack model p610 6C1, we simply disconnect the display of our deskside p610 6E1 and switch to a serial console using the procedure described in this section.

A serial cable has to be connected to a serial port of the pSeries and a terminal attached. Using a null modem cable, a normal PC can run a terminal emulation program, such as HyperTerminal (which is included in Windows) or minicom under Linux. To use HyperTerminal, make sure that the correct COM port, terminal type VT100, and 9600 Bits per second are selected.

Next switch the pSeries console by using the SMS Utilities menu. When selecting **Console** in this menu, the terminal or terminal emulation has to be physically connected already. Now the console has to be selected by pressing a number, as indicated on the new console. After a successful switch to the new serial console, the graphics display can be disconnected. An analogous procedure can be used to switch back to the graphical console.

Installing SuSE Linux

To install Linux on a system with a serial console, only a few steps are different from what was described in “Systems with a graphics adapter” on page 120.

Step 1

The first step is to insert the SuSE CD into the system and boot to SMS Multiboot by pressing 1 during the initialization phase (similar to the procedure for systems with a graphical console, but pressing 1 instead of F1). Select the CD drive as an install device, as shown in Example 5-2, and confirm the Identstring **SuSE** SLES-8 (PPC).

Example 5-2 SMS Multiboot - Select install device

```
pSeries Firmware
Version CLT02121
(c) Copyright IBM Corp. 2000, 2002  All rights reserved.
-------------------------------------------------------------------------------
Install Operating System From:
Device
Number     Name
1        Diskette
2        SCSI CD/DVD id=@2,0 (Integrated/Internal)
3        SCSI Tape id=@1,0 (Integrated/Internal)
4        Token-Ring  (slot=5/T1)
5        Port E1 - 100/10 Ethernet Adapter  (Integrated/E1)
6        Port E2 - 100/10 Ethernet Adapter  (Integrated/E2)
7        IBM Gigabit UTP PCI Adapter  (slot=1/E1)
8                 None
-------------------------------------------------------------------------------
Navigation keys:
M = return to Main Menu
ESC key = return to previous screen         X = eXit System Management Services
-------------------------------------------------------------------------------

Type the number of the menu item and press Enter or Select a Navigation Key: 2
```

Step 2

As we are already in step two, the appropriate kernel now needs to be selected on the yaboot menu, as shown in Example 5-3. In addition to the kernel selection with install, we also specify the option vnc=1 to have a virtual graphical display.

Example 5-3 Kernel selection with yaboot

```
Welcome to SuSE Linux!

  Use  "install"     to boot the ppc64 kernel
  Use  "install32"   to boot the 32bit RS/6000 kernel

  You can pass the option "noinitrd"  to skip the installer.
  Example: install noinitrd root=/dev/sda4

Welcome to yaboot version 1.3.6.SuSE
```

```
Enter "help" to get some basic usage information
boot: install vnc=1 vnc_password=susesles8 dhcp=1
```

The optional parameters given to kernel have the following meanings:

vnc=1 Starts a VNC server in the install system to make installation more convenient.

vnc_password Sets the VNC password (which has to comply to some standard password guidelines, that is, must not be too short).

dhcp=1 Tries to automatically configure a network interface using DHCP. Might not work when multiple network adapters are installed. (We had to choose eth0 manually.)

usessh=1 Allows log into the install system using ssh; may not be used together with vnc=1.

Next, the system prompts for the terminal type of the console, which is VT100 in this example. This is shown in Example 5-4 together with the message that the system is ready for a VNC connection.

Example 5-4 Terminal type selection

```
What type of terminal do you have ?
1) VT100
2) VT102
3) VT220
4) X Terminal Emulator (xterm)
5) X Terminal Emulator (xterm-vt220)
6) X Terminal Emulator (xterm-sco)
7) X Terminal Emulator (xterm-sun)
8) Linux VGA or Framebuffer Console
9) Other
Type the number of your choice and press Return: 1

Please wait while YaST2 will be started
OK

starting VNC server...
a log can be found in /tmp/vncserver.log ...
***
***           You can connect to 9.3.5.7, display :1 now with vncviewer
***           Or use a Java capable browser on  http://9.3.5.7:5801/
***
(When YaST2 is finished, close your VNC viewer and return to this window.)
```

Now connect with a browser to the indicated URL, or use the vncviewer application to connect to the VNC server running under the Linux for pSeries install system, as shown in Figure 5-5 on page 130.

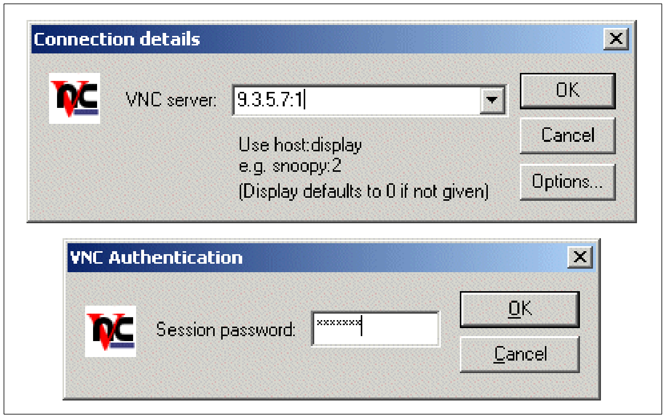

Figure 5-5 Start vncviewer

Steps 3 and 4

After entering the chosen VNC password, you might continue with steps three and four as in “Systems with a graphics adapter” on page 120. Although, planning ahead for 5.2.5, “Boot configuration and kernel recompile” on page 137, we installed this system on the third disk, /dev/sdc, instead of the second disk, /dev/sdb. In this case make sure that the boot information gets written to the same disk, /dev/sdc.

Step 5

In contrast to systems with a graphics adapter, no display has to be configured in step five in this case.

Also, as a network adapter has already been defined, no changes to the networking configuration are mandatory. But, because of the networking configuration that was done during step two, the following changes to the networking configuration might be advisable to adjust the configuration. In our case, the described installation procedure resulted in a configuration as shown in Figure 5-6 on page 131. It shows a total of five interfaces, while only four physically exist in the machine.

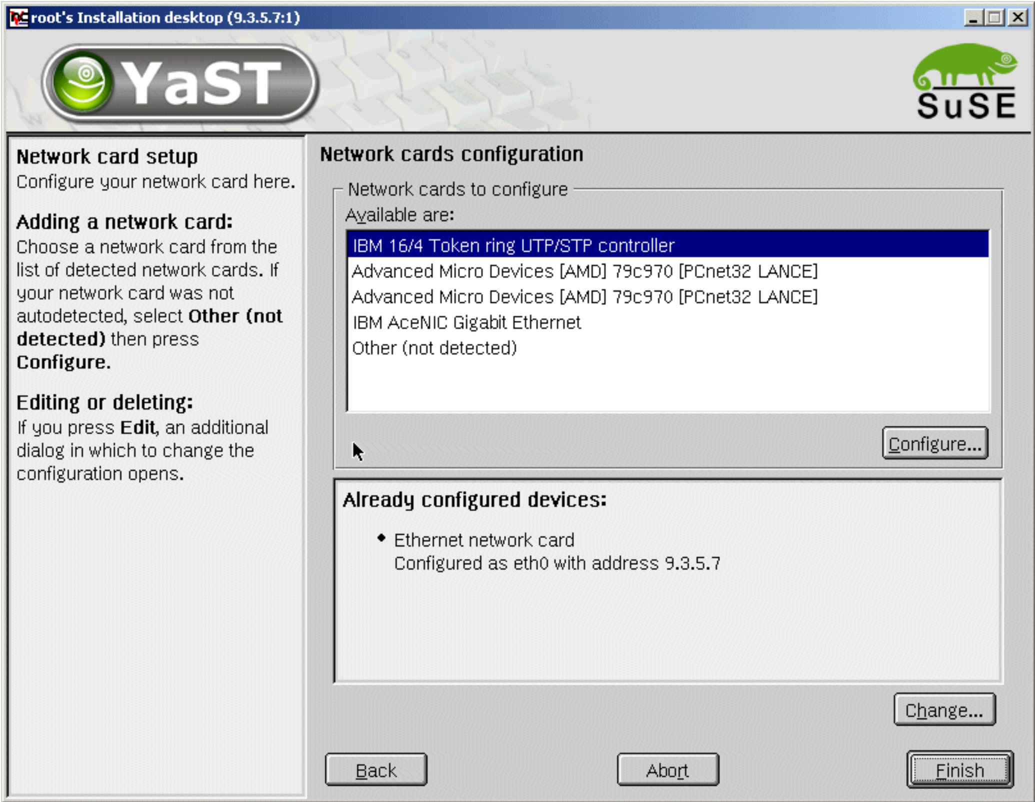

Figure 5-6 Originated networking configuration

This could be adjusted by deleting the already configured interface eth0 and then configuring the appropriate network card (shown in the upper window of Figure 5-6). The result is shown in Figure 5-7 on page 132. Be sure to not commit the deletion of the eth0 before having configured a new interface (thus, use **Back** after deleting eth0 and not **Finish**). Otherwise the system would not be reachable over the net anymore and recovery procedures would have be to taken.

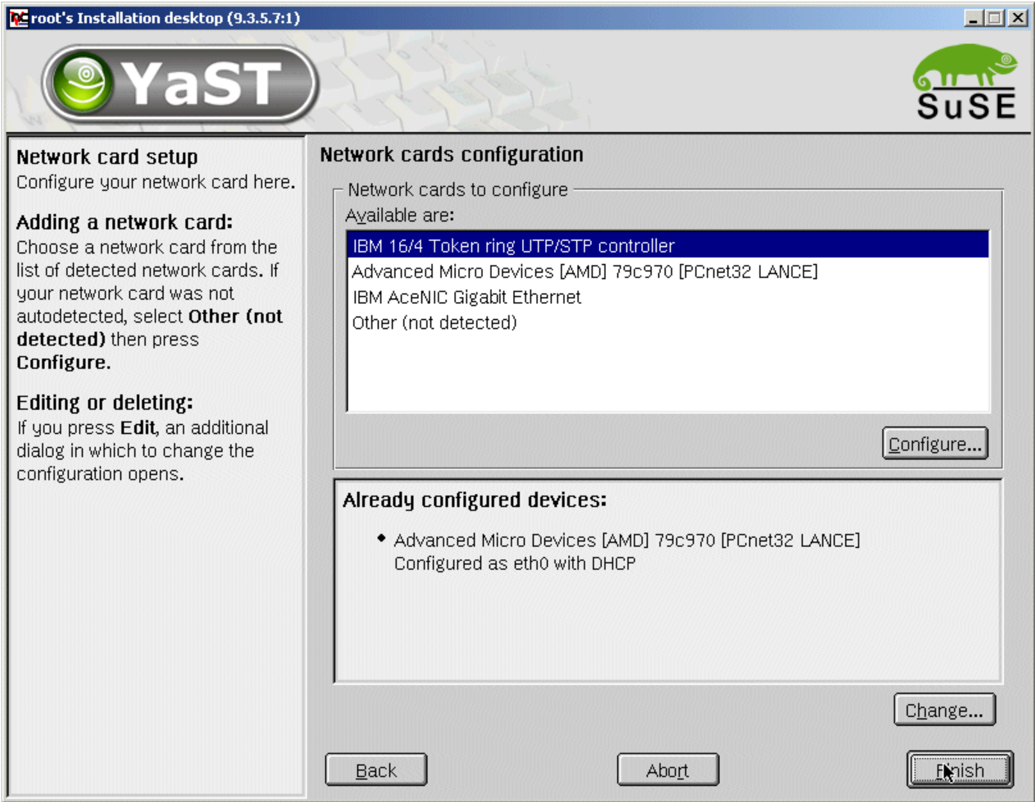

Figure 5-7 Adjusted networking configuration

At the end of this step, the system will change the run level and disconnect the VNC session.

Reconnecting via telnet is not possible, as telnet connections are not allowed by default, but ssh is enabled. Use any ssh client (which is standard on all Linux systems, or PuTTY on Windows) and connect to the pSeries system. Figure 5-8 on page 133 shows how to start PuTTY.

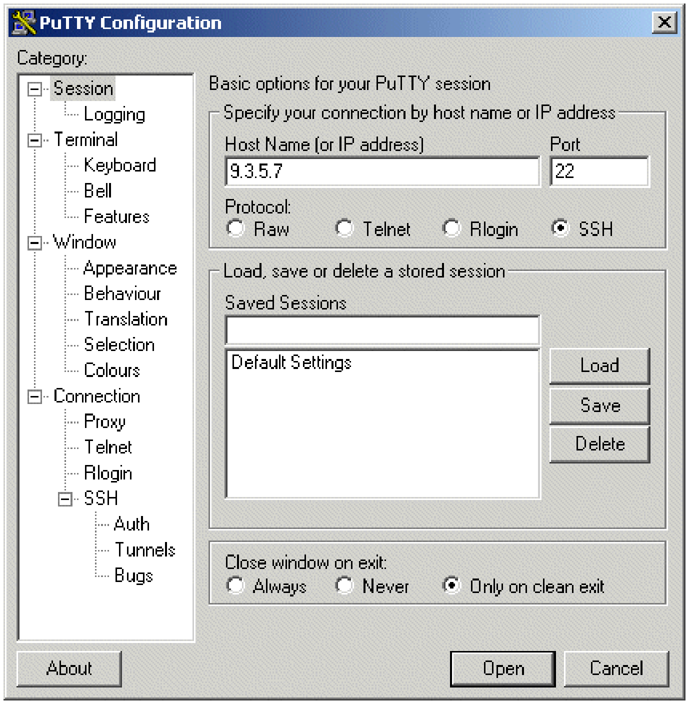

Figure 5-8 Connecting via ssh and PuTTY

Subsequently, some customizing might be done to VNC. Issue the following command to initialize, start, and immediately kill the VNC server:

```
vncserver; vncserver -kill :1
```

Then change to the directory ~/.vnc, which contains VNC-specific files. Edit the file ~/.vnc/xstartup as shown in Example 5-5 or to suit your needs. After starting the VNC server by issuing the following command, you can connect to the virtual graphical display as described before.

```
vncserver
```

Example 5-5 VNC ~/.vnc/xstartup file

```
#!/bin/sh

xrdb $HOME/.Xresources
xsetroot -solid grey
exec startkde
```

Connecting via ssh and setting up VNC cannot only be done by the root user, but by every user that is registered on the system.

Step 6

The final step is to verify the boot configuration and reboot the system. In some cases you will notice the additional kernel parameter console=ttyS1,9600 in /etc/lilo.conf. This allows logins on the serial console in case network connectivity gets lost.

5.2.4 Linux Installation on pSeries LPARs (p690 - Regatta)

Compared to the preceding description of how to install Linux on entry-level systems without a graphical display (see 5.2.3, “Systems without a graphics adapter” on page 127), there are only minor changes when installing Linux in a LPAR of a high-end Regatta system.

Linux on p670 and p690 Release Notes that describes the installation procedure in detail and also covers some important considerations regarding various service and support scenarios can be found on the pSeries Linux Web site:

`http://www.ibm.com/eserver/pseries/linux`

Please consult this documentation for updates and information regarding tested and supported hardware configurations. The only requirement we mention here is that the system’s firmware and HMC levels must be at GA2 or higher.

We assume in-depth knowledge of handling high-end pSeries systems running in a partitioned environment. Otherwise, the redbook *The Complete Partitioning Guide on IBM eServer pSeries Servers*, SG24-7039, and the LPAR section of the *IBM eServer pSeries 690 Availability Best Practices* whitepaper might be a good start. See:

`http://www.ibm.com/servers/eserver/pseries/hardware/whitepapers/p690_avail.html`

If not configured properly, any LPAR (including any that runs Linux) could impact the availability of the whole Regatta system transcending the LPAR itself.

The p690 system we used had eight partitions defined (see Figure 5-9 on page 135).

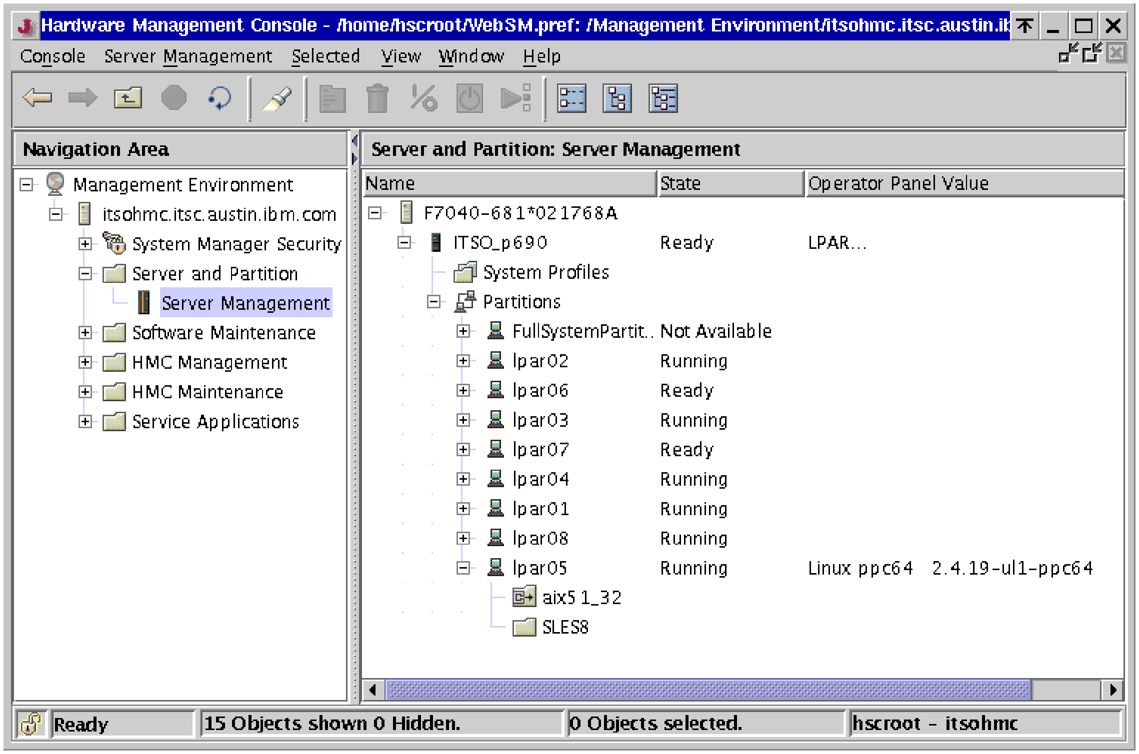

Figure 5-9 HMC partition setup

The Linux partition, LPAR05, had one processor minimum, two as desired, and four as maximum defined. Similarly, one GB of memory was set as the minimum, two GB as desired, and four GB as the maximum. The boot mode was set to SMS (see Figure 5-10 on page 136). The SCSI adapter controlling the CD drive was allocated to the Linux partition.

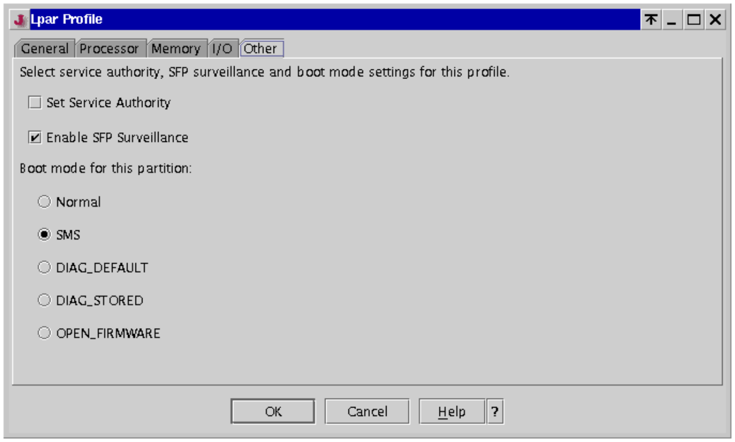

Figure 5-10 LPAR boot mode

Step 1

Analog to the procedure described in 5.2, “Installation of native Linux on pSeries” on page 120, the first step is to insert the first SuSE SLES 8 CD and activate the partition (do not forget to click the box to open the virtual terminal on the HMC). Boot to SMS Multiboot and select the CD as the install device, just as in 5.2.3, “Systems without a graphics adapter” on page 127.

Steps 2 to 5

Deviating from the prior procedure, in step two option **9) Other** and then **VT320** have to be selected as the terminal type. The remainder of step two and the steps three, four, and five are identical to the procedure described previously.

Step 6

In step six, the boot configuration needs to be updated. Select the appropriate boot device in SMS Multiboot and change the LPAR boot mode (see Figure 5-10) to Normal.

Figure 5-11 on page 137 shows a sample virtual desktop on a Regatta system. Please note the Linux version and processor information shown in the upper left corner.

Figure 5-11 The look of SuSE SLES 8 on Regatta

5.2.5 Boot configuration and kernel recompile

This section describes how to define a suitable boot configuration for yaboot and discusses the recompilation of the Linux kernel.

Compiling the kernel is an advanced topic and requires in-depth knowledge of the Linux operating system. Also, be aware that recompiling the kernel and changing kernel configuration parameters might have consequences on the support contract that might exist with a support partner such as IBM, SuSE, or others. We will only give a few hints here, which should be sufficient for an experienced Linux administrator to compile the kernel. Please see the SuSE manual for additional instructions.

Compiling the kernel requires several additional packages to be installed, such as compilers, header files, and kernel sources. The kernel sources can then be found in the directory /usr/src/linux.

The best way to start is using a kernel configuration that is known to be working. Such a configuration can be found in the file /usr/src/linux/arch/ppc64/configs/config.pseries64.suse.mod. Copy this file to /usr/src/linux as .config and see the comments in the script make_ppc64.sh about how to recompile the kernel with the previous kernel settings.

Boot configuration

We assume here that in addition to the standard kernel, /boot/vmlinuz, a second kernel, named /boot/myvmlinuz, exists and should be included in the boot configuration. In case the second kernel does not exist, the configuration can be simplified accordingly, or even the default accepted without any changes.

Because we are using yaboot for booting the system, the yaboot configuration, which is defined in /etc/lilo.conf, has to be changed. See Example 5-6 for the old /etc/lilo.conf file.

Example 5-6 Standard /etc/lilo.conf file

```
# Generated by YaST2

default=linux
timeout=100
boot=/dev/sdc1
activate

image = /boot/vmlinuz
        label = linux
        root = /dev/sdc3
        append = ""
```

Example 5-7 shows the /etc/lilo.conf file after our changes.

Example 5-7 Changed /etc/lilo.conf file

```
# Generated by YaST2

default=linux
timeout=100
boot=/dev/sdc1
activate

image = /boot/vmlinuz
        label = linux
```

```
        root = /dev/sdc3
        append = ""

image = /boot/myvmlinuz
        label = mykernel
        root = /dev/sdc3
        append = ""

image = /boot/vmlinuzSLES7
        label = SLES_7
        root = /dev/sdb3
        append = ""
```

The last four lines are not necessary at this point. We added them to demonstrate the possibility of installing several Linux systems on the same hardware and how to configure yaboot in this case. On the second disk, /dev/sdb, the older version SuSE SLES 7 was installed, while SLES 8 was installed on the third disk, /dev/sdc.

Because yaboot expects all kernel images to be on the same (boot-) disk, the SLES 7 kernel has to be copied to /dev/sdc in this case:

```
linux:/etc # mkdir /mnt/sdb3
linux:/etc # mount /dev/sdb3 /mnt/sdb3
linux:/etc # cp /mnt/sdb3/boot/vmlinuz /boot/vmlinuzSLES7
```

Note: At the time of writing this book, no solution for booting AIX and Linux from the same bootloader was available. Please check the Web for updates.

The final step before rebooting the system is to call lilo to write the yaboot configuration to disk. See Example 5-8.

Example 5-8 lilo writing the yaboot configuration

```
linux:/etc # lilo
running on chrp
install on /dev/sdc
Installing /boot/yaboot.chrp onto /dev/sdc1
519+1 records in
519+1 records out
check the image files
linux *
mykernel
SLES_7
`/tmp/ppc_lilo/yaboot.conf' -> `/etc/yaboot.conf'
```

When booting the SLES 8 and SLES 7 kernels one after the other, we get the information shown in Example 5-9 and Example 5-10.

Example 5-9 Choosing to boot linux (SLES 8) in yaboot

```
linux:~ # cat /proc/cpuinfo
processor       : 0
cpu             : POWER3 (630+)
clock           : 375MHz
revision        : 1.4

processor       : 1
cpu             : POWER3 (630+)
clock           : 375MHz
revision        : 1.4

timebase        : 93749159
machine         : CHRP IBM,7028-6E1

linux:~ # cat /proc/version
Linux version 2.4.19-ul1-ppc64-SMP (root@PowerPC64-pSeries.suse.de) (gcc
version 3.2) #1 SMP Thu Sep 26 13:16:28 UTC 2002

linux:~ # df
Filesystem           1K-blocks      Used Available Use% Mounted on
/dev/sdc3             16721404   4418140  12303264  27% /
shmfs                  1001316         0   1001316   0% /dev/shm
```

Example 5-10 Choosing to boot SLES_7 in yaboot

```
linux:~ # cat /proc/version
Linux version 2.4.13-ppc64-SMP (root@PowerPC64-pSeries.suse.de) (gcc version
3.0.2 20011104 (3.0.3 prerelease)) #1 SMP Mon Jan 14 16:38:51 GMT 2002
linux:~ # df
Filesystem           1k-blocks      Used Available Use% Mounted on
/dev/sdb3             16720380   1531768  15188612  10% /
shmfs                   991932         0    991932   0% /dev/shm
```

5.3 Available software for Linux on pSeries

At the time of writing this redbook, IBM currently offers a Java2 JDK 1.3.1 for Linux on pSeries. See the following Web sites:

http://www.ibm.com/java
http://www.ibm.com/developerworks/java/jdk/index.html
http://www.ibm.com/developerworks/java/jdk/linux140/

Java2 JDK 1.3.1 for Linux on pSeries can be downloaded from the following Web site (at no charge):

https://www6.software.ibm.com/dl/lxdk/lxdk-p

This JDK is also available on the SuSE SLES 8 CDs and gets automatically installed when choosing the default installation.

IBM is developing plans to support WebSphere Application Server, DB2 Universal Database, several compilers, and many Tivoli products on Linux for pSeries. Please see the following Web site for updates on the availability of these products.

http://www.ibm.com/eserver/pseries/linux/

On the same Web site and in the *Linux for IBM eServer pSeries* whitepaper at http://www.ibm.com/servers/eserver/pseries/linux/whitepapers/linux_pseries.html you will also find information about what commercial software from other vendors is available on Linux for pSeries. Currently, the number of such applications is not very large, but it is increasing rapidly as a large number of independent software vendors (ISVs) have made Linux their preferred development platform. Once an application is available for Linux on any hardware platform, very little effort is required to recompile it for Linux on pSeries hardware.

A huge foundation for applications on Linux for pSeries is the Open Source community. For almost any purpose software is available and many times even included in the Linux distributions. This shows how easy it is to recompile Linux software for usage on different hardware platforms, such as pSeries.

In the following section we like to show how the Open Source Tomcat servlet engine can be installed and used together with the IBM JDK.

The section thereafter will talk about OpenOffice. OpenOffice is one of the largest Open Source projects currently running, and provides an office suite comparable to Microsoft Office or Lotus Smartsuite. It is currently not available with the AIX Toolbox for Linux Applications, so the only way to get it running on pSeries hardware currently is to install it under native Linux for pSeries.

IBM JDK 1.3.1 and the Tomcat servlet engine

The default install of SuSE SLES 8 installs two different JDKs at level 1.3.1 (try the command **`rpm -qa | grep ava`** to verify) and the Tomcat servlet engine. The Tomcat version on the SuSE SLES 8 CDs is 4.0.4, which implements the Servlet 2.3 and JSP 1.2 specifications.

See the Tomcat Web site (http://jakarta.apache.org/tomcat/) and the document RUNNING.txt at http://jakarta.apache.org/tomcat/tomcat-4.0-doc/ for instructions on how to download, install, and run Tomcat. For this section we have chosen the IBM JDK and de-installed JDK to make Tomcat work:

```
rpm -e java2-jre-1.3.1-22 java2-1.3.1-22
cd /usr/lib
rm java java2;
ln -s IBMJava2 java2; ln -s java2 java
```

Starting Tomcat is done on SuSE Linux systems with the command:

```
rctomcat start
```

Tomcat can then immediately be used in stand-alone mode, that is, without integration to a Web server like Apache. Connect a Web browser to port 8080 of your Linux system `http://`*`hostname`*`:8080/` and try the examples.

Integration of Tomcat into Apache is described on every SuSE system in the file:

```
/usr/share/doc/packages/jakarta-tomcat/README.SuSE
```

It involves the following steps:

1. Edit /etc/sysconfig/apache and set the variable HTTPD_SEC_MOD_TOMCAT to yes.
2. Run the following commands:

   ```
   SuSEconfig
   rctomcat restart
   rcapache start
   ```

3. After this setup, all http requests on the standard port 80 to the /examples folder will be forwarded from Apache to Tomcat.

   ```
   http://hostname:80/examples/
   ```

OpenOffice

In this section we describe the installation of the binary version of OpenOffice on Linux for pSeries. For the most current information on OpenOffice running under Linux for pSeries, see the following Web sites:

http://www.openoffice.org/
ftp://ftp.penguinppc.org/projects/openoffice/
ftp://ftp.yellowdoglinux.com/pub/yellowdog/software/openoffice/

First, download the installer image OOo_1.0.1c_LinuxPPC_installer.tar.gz for OpenOffice Version 1.0.1 from:

ftp://ftp.suse.com/pub/projects/powerpc/openoffice/release-1.0.1/

Or any Web site listed at:

http://whiteboard.openoffice.org/mirrors/ppclinks.html

The file OpenOffice.README_ppclinux in this directory contains installation instructions.

Actual installation is done using the following commands while using a graphics display:

```
tar -zxf OOo_1.0.1c_LinuxPPC_installer.tar.gz
cd OOo_1.0.1c_LinuxPPC_installer/
chown -R root:root .
./install --prefix=/usr/local
```

As a one-time setup, every user who wants to use OpenOffice has to run:

```
/usr/local/OpenOffice.org1.0.1/setup
```

And choose **Workstation Install**.

Afterwards, OpenOffice can be started and enjoyed with:

```
~/OpenOffice.org1.0.1/soffice
```

6

Tools in the Toolbox

This chapter describes a selected number of tools that are included in the Toolbox. We have included tools from the following functional groups:

- Networking and e-mail tools
- File handling tools
- Software development tools
- Miscellaneous tools
- Login shells

6.1 Toolbox tools

The AIX Toolbox for Linux Applications contains a collection of Open Source and GNU software built for AIX 4.3.3 and AIX 5L for IBM pSeries systems and IBM RS/6000. All the tools are packaged using the RPM installation format.

To acquire the same type of information as we have provided for you in this chapter, please use:

- The respective developer's Web site to find the online man pages for the tool.
- The online man pages (/opt/freeware/man), the online info pages (/opt/freeware/info), or the online HTML files (/opt/freeware/doc) of the installed tool's RPM package.
- The **rpm -qpl *package filename*** to list all files in the specified *uninstalled* package. Use **rpm -ql *package name*** to list all files in the specified *installed* package.
- The **rpm -qpi *package filename*** to list all information from the specified *uninstalled* package. Use **rpm -qi *package name*** to list information from the specified *installed* package.
- The ***tool* --help**, ***tool* -h**, or ***tool* -?** to get the usage information from the installed tool.

> **Note:** For version-specific documentation and usage instructions, please refer to each tool's development Web site.

6.2 Networking and e-mail tools

The networking packages described here are a sample of different tools from the Toolbox, which in some form use network resources to transfer data between different systems or their usages are in some way network dependant, such as e-mail client tools. We cover the following tools in this section:

- **ethereal**
- **rsync**
- **ftpcopy**
- **ncftp**
- **wget**
- **rdist**
- **lynx**
- **curl**
- **elm**
- **fetchmail**

- **mutt**
- **proftpd**
- **pine**
- **wu-ftpd**

6.2.1 ethereal

The Ethereal package provides the **ethereal** utility, which is a graphical network protocol analyzer. It allows you to examine data from a live network or from a capture file on disk. You can interactively browse the capture data, viewing summary and detail information for each packet. Ethereal has several powerful features, including a rich display filter language and the ability to view the reconstructed stream of a TCP session.

Synopsis

```
ethereal [ -B byte view height ] [ -c count ] [ -f filter expression ] [ -h ] [
-i interface ] [ -k ] [ -m font ] [ -n ] [ -o preference setting ] ... [ -p ] [
-P packet list height ] [ -Q ] [ -r infile ] [ -R filter expression ] [ -S ] [
-s snaplen ] [ -T tree view height ] [ -t time stamp format ] [ -v ] [ -w
savefile]
```

The following list describes options for the **ethereal** command:

-B	Sets the initial height of the byte view (bottom) pane.
-c count	Sets the default number of packets to read when capturing live data.
-f filter-expr	Sets the capture filter expression.
-h	Prints the version and options and exits.
-i interface	Sets the name of the network interface or pipe to use for live packet capture.
-k	Starts the capture session immediately.
-m font	Sets the name of the font used by **ethereal** for most text.
-n	Disables network object name resolution (such as hostname, TCP, and UDP port names).
-o prename:value	Sets a preference value, overriding the default value and any value read from a preference file.
-p	Do not put the interface into promiscuous mode.
-P packet list	Sets the initial height of the packet list (top) pane.
-Q	Causes **ethereal** to exit after the end of the capture session (useful in batch mode with the -c option, for instance); this option requires the -i and -w parameters.
-r file	Reads packet data from file.
-R filter expr	When reading a capture file specified with the -r flag, causes the specified filter (display filter format) to be

	applied to all packets read from the capture file; packets not matching the filter are discarded.
-S	Specifies that the live packet capture will be performed in a separate process, and that the packet display will automatically be updated as packets are seen.
-s snaplen	Sets the default snapshot length to use when capturing live data.
-T three view	Sets the initial height of the tree view (middle) pane.
-t format	Sets the format of the packet time stamp displayed in the packet list window. The format can be r (relative), a (absolute), ad (absolute with date), or d (delta).
-v	Prints the version and exits.
-w file	Sets the default capture file name.

Toolbox FTP archive Web site

ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/ethereal

Tool developer Web site

http://www.ethereal.com

Tool FTP archive Web site

ftp://ftp.ethereal.com/pub/ethereal

Files

```
/opt/freeware/bin/editcap
/opt/freeware/bin/ethereal
/opt/freeware/bin/tethereal
/opt/freeware/etc/manuf
/opt/freeware/lib/ethereal/plugins/
```

...(files omitted)...

```
/opt/freeware/man/man1/editcap.1
/opt/freeware/man/man1/ethereal.1
/opt/freeware/man/man1/tethereal.1
/usr/bin/editcap
/usr/bin/ethereal
/usr/bin/tethereal
/usr/lib/ethereal
```

Prereqs

The Ethereal package requires the ZLIB, REP-GTK, and GTK+ packages.

Examples

Start the **`ethereal`** graphical network protocol analyzer by typing `ethereal` in the command line:

```
ethereal
```

To start capturing data, select the **Capture** → **Start** menu option and in the Capture Preference pop-up window, select your preferences. If you select **Promiscuous mode** and then confirm by clicking the **OK** button, the Capture pop-up window will show the amount of traffic that is captured on the selected network interface. When you select the **Stop** button in the Capture pop-up window, capturing is halted and the information is processed and displayed, packet by packet, in the main Ethereal window, as shown in Figure 6-1below.

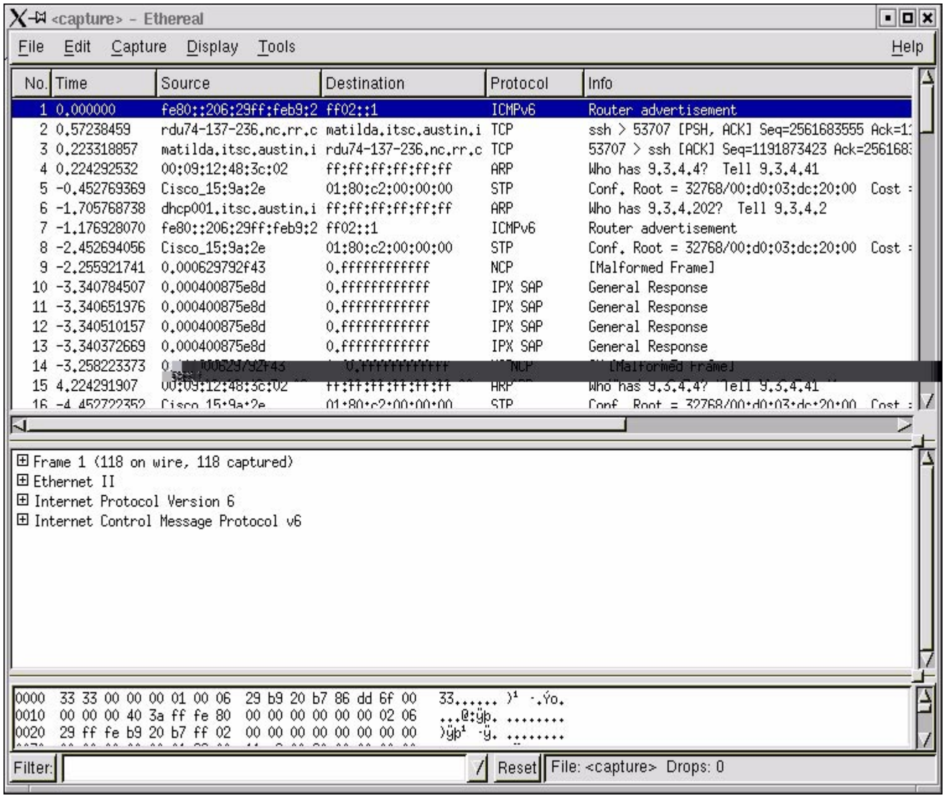

Figure 6-1 Ethereal default window

Note the expand (+) boxes in the middle pane. If you select them, the item content will be expanded in the pane.

6.2.2 rsync

The RSYNC package provides a command for synchronizing files over a network. The **rsync** command uses a quick and reliable algorithm to very quickly bring remote and host files into sync. The **rsync** command is fast because it just sends the differences in the files over the network (instead of sending the complete files). The **rsync** command is often used as a very powerful mirroring process or just as a more capable replacement for the **rcp** command.

Synopsis

```
rsync [OPTION]... SRC [SRC]... [USER@]HOST:DEST
rsync [OPTION]... [USER@]HOST:SRC DEST
rsync [OPTION]... SRC [SRC]... DEST
rsync [OPTION]... [USER@]HOST::SRC [DEST]
rsync [OPTION]... SRC [SRC]... [USER@]HOST::DEST
rsync [OPTION]... rsync://[USER@]HOST[:PORT]/SRC [DEST]
```

The following list describes options for the **rsync** command:

-v, --verbose	Increase verbosity.
-q, --quiet	Decrease verbosity.
-c, --checksum	Always checksum.
-a, --archive	Archive mode.
-r, --recursive	Recurse into directories.
-R, --relative	Use relative path names.
-b, --backup	Make backups (default ~ suffix).
--backup-dir	Make backups into this directory.
--suffix=SUFFIX	Override backup suffix.
-u, --update	Update only (do not overwrite newer files).
-l, --links	Copy symlinks as symlinks.
-L, --copy-links	Copy the referent of symlinks.
--copy-unsafe-links	Copy links outside the source tree.
--safe-links	Ignore links outside the destination tree.
-H, --hard-links	Preserve hard links.
-p, --perms	Preserve permissions.
-o, --owner	Preserve owner (root only).
-g, --group	Preserve group.
-D, --devices	Preserve devices (root only).
-t, --times	Preserve times.
-S, --sparse	Handle sparse files efficiently.
-n, --dry-run	Show what would have been transferred.
-W, --whole-file	Copy whole files, no incremental checks.
--no-whole-file	Turn off --whole-file.
-x, --one-file-system	Do not cross file system boundaries.
-B, --block-size=SIZE	Checksum blocking size (default 700).
-e, --rsh=COMMAND	Specify rsh replacement.

--rsync-path=PATH	Specify path to `rsync` on the remote machine.
-C, --cvs-exclude	Auto-ignore files in the same way CVS does.
--existing	Only update files that already exist.
--ignore-existing	Ignore files that already exist on the receiving side.
--delete	Delete files that do not exist on the sending side.
--delete-excluded	Also delete excluded files on the receiving side.
--delete-after	Delete after transferring, not before.
--ignore-errors	Delete even if there are I/O errors.
--max-delete=NUM	Do not delete more than NUM files.
--partial	Keep partially transferred files.
--force	Force deletion of directories even if not empty.
--numeric-ids	Do not map UID/GID values by user/group name.
--timeout=TIME	Set I/O time out in seconds.
-I, --ignore-times	Do not exclude files that match length and time.
--size-only	Only use file size when determining if a file should be transferred.
--modify-window=NUM	Time stamp window (seconds) for file match (default=0).
-T --temp-dir=DIR	Create temporary files in directory DIR.
--compare-dest=DIR	Also compare destination files relative to DIR.
-P	Equivalent to --partial --progress.
-z, --compress	Compress file data.
--exclude=PATTERN	Exclude files matching PATTERN.
--exclude-from=FILE	Exclude patterns listed in FILE.
--include=PATTERN	Do not exclude files matching PATTERN.
--include-from=FILE	Do not exclude patterns listed in FILE.
--version	Print version number.
--daemon	Run as a `rsync` daemon.
--no-detach	Do not detach from the parent.
--address=ADDRESS	Bind to the specified address.
--config=FILE	Specify alternate rsyncd.conf file.
--port=PORT	Specify alternate `rsyncd` port number.
--blocking-io	Use blocking IO for the remote shell.
--no-blocking-io	Turn off --blocking-io.
--stats	Give some file transfer stats.
--progress	Show progress during transfer.
--log-format=FORMAT	Log file transfers using specified format.
--password-file=FILE	Get password from FILE.
--bwlimit=KBPS	Limit I/O bandwidth, KBytes per second.
--write-batch=PREFIX	Write batch fileset starting with PREFIX.
--read-batch=PREFIX	Read batch fileset starting with PREFIX.
-h, --help	Usage information.

Toolbox FTP archive Web site

ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/rsync

Tool developer Web site

http://www.samba.org/rsync

Tool FTP archive Web site

ftp://rsync.samba.org/pub/rsync

Files

```
/opt/freeware/bin/rsync
/opt/freeware/doc/rsync-2.5.4
/opt/freeware/doc/rsync-2.5.4/COPYING
/opt/freeware/doc/rsync-2.5.4/README
/opt/freeware/doc/rsync-2.5.4/tech_report.tex
/opt/freeware/man/man1/rsync.1
/opt/freeware/man/man5/rsyncd.conf.5
/usr/bin/rsync
```

Prereqs

The RSYNC package does not have any prerequisites unless you wish to use SSL encryption, in which case the SSH package is required.

Examples

The following example will copy all files and directories, in archive mode, from /local/data on the host w2 to the /w2/local directory on the system where the command was issued from:

```
rsync -avz w2:/local/data /w2/local
```

Archive mode ensures that symbolic links, devices, attributes, and other file information is preserved in the transfer.

The next example illustrates how to mirror a directory between one master server and a slave server (w2), using SSH for encrypted transfer and deleting all files on the slave server, that is not transferred from the master server:

```
rsync -avz -e ssh --delete /www/servers/w2/ w2:/www/servers/w2
```

6.2.3 ftpcopy

The FTPCOPY packages provide commands to copy a FTP site recursively. It afterwards deletes all files in the local directory tree that were not found on the remote site. The primary purpose of the **ftpcopy** command is to mirror FTP sites that support the EPLF directory listing format, but it can also be used to mirror other sites.

Synopsis

```
ftpcopy [options] host[:port] remotedir [localdir]
ftpcopy [options] ftp://host[:port]/remotedir [localdir]
```

The following list describes options for the **`ftpcopy`** command:

--account=ACCOUNT	Send ACCOUNT as account name during login phase.
-d, --directories-only	Only create the directory hierarchy.
--dry-run	Do not do anything.
-l, --loglevel=NUMBER	Control the amount of logging done.
--bps	Log transfer rates.
-L, --list-options=OPTS	Add OPTS to LIST command.
-m, --max-days=MAX	Restrict on modification time.
-n, --no-delete	Do not delete files.
-p, --pass=PASSWORD	Use PASS as password to log into the FTP server.
-s, --symlink-hack	Deal with symbolic links.
-T, --timeout=NUMBER	Time out to use for read/write (sec.).
--tries=NUMBER	Number of tries to connect and log in.
-u, --user=NAME	Use `NAME` to log into the FTP server.
-x, --exclude=WILDCARD	Exclude paths matching WILDCARD.
-i, --include=WILDCARD	Include paths matching WILDCARD.
--force-select	Use select, not poll.
--interactive	Read directories from stdin.
--tolower	Change local names to lowercase.
--version	Show version information.
--help	Show a list of options or the long help on one.
--longhelp	Show longer help texts for all or one variable.

localdir defaults to the current working directory. If it is not given, the -n option must be used.

Toolbox FTP archive Web site

ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/ftpcopy

Tool developer Web site

http://www.ohse.de/uwe/ftpcopy/ftpcopy.html

Tool FTP archive Web site

ftp://www.ohse.de/uwe/ftpcopy

Files

```
/opt/freeware/bin/ftpcopy
/opt/freeware/bin/ftpcp
/opt/freeware/bin/ftpls
/opt/freeware/doc/ftpcopy-0.3.9
/opt/freeware/doc/ftpcopy-0.3.9/ChangeLog
/opt/freeware/doc/ftpcopy-0.3.9/NEWS
/opt/freeware/doc/ftpcopy-0.3.9/README
/opt/freeware/man/man1/ftpcopy.1
/opt/freeware/man/man1/ftpcp.1
/opt/freeware/man/man1/ftpls.1
/usr/bin/ftpcopy
/usr/bin/ftpcp
/usr/bin/ftpls
```

Prereqs

The FTPCOPY package does not have any prerequisites.

Examples

The following example will copy all files and directories, logging in as user `root` with the password `root`, from `/local/data` on the host `w2` to the `/w2/local/data` directory on the system where the command was issued from:

```
ftpcopy -u root -p root ftp://w2/local/data /w2/local/data
```

The next example will copy all files and directories, logging in as user `root` with the password `root`, from `/local/data` on the host `w2` to the `/w2/local/data` directory on the system where the command was issued from. It will also exclude all files with the name passwd and it will display the byte transfer rates (`bps`) for each file that is transferred:

```
ftpcopy -u root -p root --bps --exclude '*/passwd' ftp://w2/local/data
/w2/local/data
```

6.2.4 ncftp

The NCFTP package provides an improved FTP client. The **`ncftp`** command improvements include support for command line editing, command histories, automatic anonymous logins, progress meters, file name completion, background processing, auto-resume downloads, bookmarking, cached directory listings, host redialing, working with firewalls and proxies, and downloading entire directory trees.

Synopsis

`ncftp [flags] [`*`host | directory URL to browse`*`]`

The following list describes options for the **ncftp** command:

-u *XX*	Use user name *XX* instead of anonymous.
-p *XX*	Use password *XX* with the user name.
-P *XX*	Use port number *XX* instead of the default FTP service port (21).
-j *XX*	Use account *XX* with the user name (rarely needed).
-F	Dump a sample $HOME/.ncftp/firewall prefs file to stdout and exit.

Toolbox FTP archive Web site

ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/ncftp

Tool developer Web site

http://www.ncftp.com/ncftp

Tool FTP archive Web site

ftp://ftp.ncftp.com/ncftp

Files

```
/etc/X11/applnk/Internet/ncftp.desktop
/opt/freeware/bin/ncftp
/opt/freeware/bin/ncftpbatch
/opt/freeware/bin/ncftpbookmarks
/opt/freeware/bin/ncftpget
/opt/freeware/bin/ncftpls
/opt/freeware/bin/ncftpput
/opt/freeware/doc/ncftp-3.1.1
/opt/freeware/doc/ncftp-3.1.1/README.txt
/opt/freeware/doc/ncftp-3.1.1/doc/
...(files omitted)...
/opt/freeware/man/man1/ncftp.1
/opt/freeware/man/man1/ncftpbatch.1
/opt/freeware/man/man1/ncftpget.1
/opt/freeware/man/man1/ncftpls.1
/opt/freeware/man/man1/ncftpput.1
/usr/bin/ncftp
/usr/bin/ncftpbatch
/usr/bin/ncftpbookmarks
/usr/bin/ncftpget
/usr/bin/ncftpls
/usr/bin/ncftpput
```

Prereqs

The NCFTP package does not have any prerequisites.

Examples

The following example will copy all files and directories, logging in as user (-u) root with the password (-p) root, from /local/data on the host w2 to the /w2/local/data directory on the system where the command was issued from:

```
ncftpget -R -u root -p root w2 /w2/local /local/data
```

Note that the local directory is between the remote host (w2) and the remote directory specification (/local/data) in the example above. The next example uses another form to specify the same remote host and directory, but will download to the current directory.

```
cd /w2/local && ncftpget -R -u root -p root ftp://w2/local/data
```

6.2.5 wget

The WGET package provides a file retrieval utility which can use either the HTTP or FTP protocols. The **wget** command features include the ability to work in the background while you're logged out, recursive retrieval of directories, file name wildcard matching, remote file time stamp storage and comparison, support for proxy servers, and configurability.

Synopsis

`wget [OPTION]... [URL]...`

The following list describes options for the **wget** command:

-V, --version	Display the version of **wget** and exit.
-h, --help	Print this help.
-b, --background	Go to background after startup.
-e, --execute=COMMAND	Execute a .wgetrc style command.
-o, --output-file=FILE	Log messages to FILE.
-a, --append-output=FILE	Append messages to FILE.
-d, --debug	Print debug output.
-q, --quiet	Quiet (no output).
-v, --verbose	Be verbose (this is the default).
-nv, --non-verbose	Turn off verboseness, without being quiet.
-i, --input-file=FILE	Download URLs found in FILE.
-F, --force-html	Treat input file as HTML.
-B, --base=URL	Prepends URL to relative links in -F -i file.
--sslcertfile=FILE	Optional client certificate.
--sslcertkey=KEYFILE	Optional keyfile for this certificate.
--egd-file=FILE	File name of the EGD socket.
--bind-address=ADDRESS	Bind to ADDRESS (host name or IP) on local host.

-t, --tries=NUMBER	Set number of retries to NUMBER (0 unlimits).
-O --output-document=FILE	Write documents to FILE.
-nc, --no-clobber	Do not clobber existing files or use .# suffixes.
-c, --continue	Resume getting a partially downloaded file.
--progress=TYPE	Select progress gauge type.
-N, --timestamping	Do not re-retrieve files unless newer than local.
-S, --server-response	Print server response.
--spider	Do not download anything.
-T, --timeout=SECONDS	Set the read timeout to SECONDS.
-w, --wait=SECONDS	Wait SECONDS between retrievals.
--waitretry=SECONDS	Wait 1...SECONDS between retries of a retrieval.
--random-wait	Wait from 0...2*WAIT secs between retrievals.
-Y, --proxy=on/off	Turn proxy on or off.
-Q, --quota=NUMBER	Set retrieval quota to NUMBER.
--limit-rate=RATE	Limit download rate to RATE.
-nd --no-directories	Do not create directories.
-x, --force-directories	Force creation of directories.
-nH, --no-host-directories	Do not create host directories.
-P, --directory-prefix=PREFIX	Save files to PREFIX/....
--cut-dirs=NUMBER	Ignore `NUMBER` remote directory components.
--http-user=USER	Set http user to USER.
--http-passwd=PASS	Set http password to PASS.
-C, --cache=on/off	(Dis)allow server-cached data (normally allowed).
-E, --html-extension	Save all text/html documents with .html extension.
--ignore-length	Ignore Content-Length header field.
--header=STRING	Insert STRING among the headers.
--proxy-user=USER	`USER` as proxy user name.
--proxy-passwd=PASS	Set PASS as proxy password.
--referer=URL	Include Referer: URL header in HTTP request.
-s, --save-headers	Save the HTTP headers to file.
-U, --user-agent=AGENT	Identify as AGENT instead of **`wget`**/VERSION.
--no-http-keep-alive	Disable HTTP keep-alive (persistent connections).
--cookies=off	Do not use cookies.
--load-cookies=FILE	Load cookies from FILE before session.
--save-cookies=FILE	Save cookies to FILE after session.

-nr, --dont-remove-listing	Do not remove .listing files.
-g, --glob=on/off	Turn file name globbing on or off.
--passive-ftp	Use the passive transfer mode.
--retr-symlinks	When recursing, get linked-to files (not dirs).
-r, --recursive	Recursive Web-suck—use with care.
-l, --level=NUMBER	Maximum recursion depth (inf or 0 for infinite).
--delete-after	Delete files locally after downloading them.
-k, --convert-links	Convert non-relative links to relative.
-K, --backup-converted	Before converting file X, back up as X.orig.
-m, --mirror	Shortcut option equivalent to -r -N -l inf -nr.
-p, --page-requisites	Get all images, etc. needed to display HTML page.
-A, --accept=LIST	Comma-separated list of accepted extensions.
-R, --reject=LIST	Comma-separated list of rejected extensions.
-D, --domains=LIST	Comma-separated list of accepted domains.
--exclude-domains=LIST	Comma-separated list of rejected domains.
--follow-ftp	Follow FTP links from HTML documents.
--follow-tags=LIST	Comma-separated list of followed HTML tags.
-G, --ignore-tags=LIST	Comma-separated list of ignored HTML tags.
-H, --span-hosts	Go to foreign hosts when recursive.
-L, --relative	Follow relative links only.
-I, --include-directories=LIST	List of allowed directories.
-X, --exclude-directories=LIST	List of excluded directories.
-np, --no-parent	Do not ascend to the parent directory.

Toolbox FTP archive Web site

ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/wget

Tool developer Web site

http://www.gnu.org/software/wget/wget.html

Tool FTP archive Web site

ftp://ftp.gnu.org/gnu/wget

Files

```
/etc/wgetrc
/opt/freeware/bin/wget
/opt/freeware/doc/wget-1.8.1
/opt/freeware/doc/wget-1.8.1/AUTHORS
```

```
/opt/freeware/doc/wget-1.8.1/ChangeLog
/opt/freeware/doc/wget-1.8.1/INSTALL
/opt/freeware/doc/wget-1.8.1/MAILING-LIST
/opt/freeware/doc/wget-1.8.1/NEWS
/opt/freeware/doc/wget-1.8.1/README
/opt/freeware/info/wget.info-1.gz
/opt/freeware/info/wget.info-2.gz
/opt/freeware/info/wget.info-3.gz
/opt/freeware/info/wget.info-4.gz
/opt/freeware/info/wget.info.gz
/opt/freeware/share/locale/
...(files ommited)...
/usr/bin/wget
```

Prereqs

The WGET package requires the GETTEXT and INFO packages.

Examples

The following example will copy all files and directories, logging in as user root with the password root, from /local/data on the host w2 to the current directory (it will create the /w2/local/data hierarchy) on the system where the command was issued from. Note the usage of the *user:password@host* syntax in the FTP URL. It is required in this case since we do not allow *anonymous* FTP access to the /local/data directory on the `w2` host.

```
cd / && wget -r ftp://root:root@w2/local/data
```

The following example performs the same operation as in the previous example, but this host (wwwtest) allows *anonymous* FTP:

```
cd / && wget -r ftp://wwwtest/local/data
```

The next example will user the --mirror option instead of only the -r (recursive) option. The --mirror option turns on recursion (-r) and time-stamping (-N), sets infinite recursion depth (-l inf) and keeps FTP directory listings (-nr):

```
cd / && wget --mirror ftp://root:root@w2/local/data
```

6.2.6 rdist

The RDIST package provides the **`rdist`** command that maintains identical copies of files on multiple hosts. If possible, **`rdist`** will preserve the owner, group, mode, and mtime of files and it can update programs that are executing.

Synopsis

```
rdist [-nqbRhivwyD] [-f DistFile] [-d var=value] [-m Host] [File ...]
rdist [-nqbRhivwyD] -c Source [...] [Login@]Machine[:Dest]
```

The following list describes options for the **rdist** command:

-A num	Set the minimum number of free files (i-nodes) on a file system that must exist for **rdist** to update or install a file.
-a num	Set the minimum amount of free space (in bytes) on a file system that must exist for **rdist** to update or install a file.
-D	Enable copious debugging messages.
-d var=value	Define var to have value. This option is used to define or override variable definitions in the distfile. Value can be the empty string, one name, or a list of names surrounded by parentheses and separated by tabs and/or spaces.
-F	Do not fork any child **rdist** processes. All clients are updated sequentially.
-f distfile	Set the name of the distfile to use to be distfile. If distfile is specified as a dash (-), then read from standard input (*stdin*).
-l logopts	Set local logging options.
-L logopts	Set remote logging options. logopts are the same as for local logging except the values are passed to the remote server (rdistd).
-M num	Set the maximum number of simultaneously running child **rdist** processes to num. The default is 4.
-m machine	Limit which machines are to be updated. Multiple -m arguments can be given to limit updates to a subset of the hosts listed in the distfile.
-n	Print the commands without executing them. This option is useful for debugging distfile.
-odistopts	Specify the dist options to enable; distopts is a comma-separated list of options.
-p *rdistd-path*	Set the path where the rdistd server is searched for on the target host.
-P *transport-path*	Set the path to the transport command to be used.
-t timeout	Set the timeout period (in seconds) for waiting for responses from the remote **rdist** server. The default is 900 seconds.
-V	Print version information and exit.

Distopts

The following dist options can be used with the -o option:

verify	Verify that the files are up to date on all the hosts. Any files that are out of date will be displayed but no files will be changed nor any mail sent.
whole	Whole mode. The whole file name is appended to the destination directory name. Normally, only the last component of a name is used when renaming files. This

	will preserve the directory structure of the files being copied instead of flattening the directory structure.
noexec	Automatically exclude executable files that are in *a.out* format from being checked or updated.
younger	Younger mode. Files are normally updated if their mtime and size disagree. This option causes `rdist` to not update files that are younger than the master copy. This can be used to prevent newer copies on other hosts from being replaced. A warning message is printed for files that are newer than the master copy.
compare	Binary comparison. Perform a binary comparison and update files if they differ rather than comparing dates and sizes.
follow	Follow symbolic links. Copy the file that the link points to rather than the link itself.
ignlnks	Ignore unresolved links. `rdist` will normally try to maintain the link structure of files being transferred and warn the user if all the links cannot be found.
chknfs	Do not check or update files on target host that reside on NFS file systems.
chkreadonly	Enable check on target host to see if a file resides on a read-only file system. If a file does, then no checking or updating of the file is attempted.
chksym	If the target on the remote host is a symbolic link, but is not on the master host, the remote target will be left a symbolic link. This behavior is generally considered a bug in the original version of `rdist`, but is present to allow compatibility with older versions.
quiet	Quiet mode. Files that are being modified are normally printed on standard output. This option suppresses this.
remove	Remove extraneous files. If a directory is being updated, any files that exist on the remote host that do not exist in the master directory are removed. This is useful for maintaining truly identical copies of directories.
nochkowner	Do not check user ownership of files that already exist. The file ownership is only set when the file is updated.
nochkgroup	Do not check group ownership of files that already exist. The file ownership is only set when the file is updated.
nochkmode	Do not check file and directory permission modes. The permission mode is only set when the file is updated.
nodescend	Do not descend into a directory. Normally `rdist` will recursively check directories. If this option is enabled, then any files listed in the file list in the distfile that are directories are not recursively scanned. Only the

	existence, ownership, and mode of the directory are checked.
numchkgroup	Use the numeric group ID (GID) to check group ownership instead of the group name.
numchkowner	Use the numeric user ID (UID) to check user ownership instead of the user name.
savetargets	Save files that are updated instead of removing them. Any target is renamed to *filename*.OLD.
sparse	Enable checking for *sparse* files. One of the most common types of sparse files are those produced by ndbm. This option adds some additional processing overhead so it should only be enabled for targets likely to contain sparse files.

Toolbox FTP archive Web site

`ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/rdist`

Tool developer Web site

`http://www.magnicomp.com/rdist`

Tool FTP archive Web site

`http://www.magnicomp.com/cgi-bin/mcdownload.cgi`

Files

```
/opt/freeware/bin/rdist
/opt/freeware/bin/rdistd
/opt/freeware/doc/rdist-6.1.5
/opt/freeware/doc/rdist-6.1.5/README
/opt/freeware/man/man1/rdist.1
/opt/freeware/man/man8/rdistd.8
/usr/linux/bin/rdist
/usr/linux/bin/rdistd
```

Prereqs

The RDIST package does not have any prerequisites.

Examples

The first example uses ssh as the transport program. To use a transport program other than rsh we use the -P option. Note that the transport program that is used must be compatible with the syntax for rsh.

```
rdist -P /usr/bin/ssh -f do.rdist
```

The following is a sample of the do.**rdist** file. The mirror_dns is the stanza name. The variable dnsprimary is set to contain three host names: ns1, ns2 and ns3. The files named.conf and resolv.conf and the directory /etc/local/dns should all be installed on the three hosts.

```
mirror_dns:
    dnsprimary         = { ns1 ns2 ns3 }
    /etc/named.conf    -> { $dnsprimary }
    /etc/resolv.conf   -> { $dnsprimary }
    /etc/local/dns     -> { $dnsprimary }
        install;
```

The **install** command is used to copy out-of-date files and/or directories. Each source file and/or directory is copied to each host in the destination list.

The following is another sample of another do.**rdist** file, that can be used to install security files on systems with the same level of AIX:

```
mirror_pwd:
    wwwhosts       = { w1 w2 w3 }
    /etc/passwd    -> { $wwwhosts }
    /etc/group     -> { $wwwhosts }
    /etc/security  -> { $wwwhosts }
        install;
```

Example 6-1 uses the command line option (-c) of **rdist** to execute without using a *distfile*.

Example 6-1 Using rdist without a distfile

```
root@fenris:/: rdist -c /etc/motd root@w2:/tmp/motd
rdist: Updating the host w2.
rdist: installing: /etc/motd
```

The equivalent distfile for Example 6-1 would be the following:

```
( /etc/motd ) -> root@w2
        install /tmp/motd;
```

6.2.7 lynx

The LYNX package provides the **lynx** command that is a text-based Web browser and can be used on ASCII terminals. The **lynx** command does not display any images, but it does support frames, tables, and most other HTML tags. The **lynx** command starts and exits quickly, and swiftly displays Web pages.

Synopsis

```
lynx [options] [file]
```

The following list describes options for the `lynx` command:

-	Receive options and arguments from stdin.
-curses_pads	Use curses pad feature to support left/right shifting (on).
-dump	Dump the first file to stdout and exit.
-editor=EDITOR	Enable edit mode with specified editor.
-help	Print this usage message.
-nobold	Disable bold video-attribute.
-nobrowse	Disable directory browsing.
-nocolor	Turn off color support.
-nopause	Disable forced pauses for status line messages.
-noredir	Do not follow Location: redirection (off).
-noreferer	Disable transmission of Referer headers (off).
-noreverse	Disable reverse video-attribute.
-nostatus	Disable the miscellaneous information messages (off).
-nounderline	Disable underline video-attribute.
-reload	Flush the cache on a proxy server (only the first document affected) (off).
-show_cursor	Toggle hiding of the cursor in the lower right corner (on.
-show_rate	Toggle display of transfer rate (on).
-soft_dquotes	Toggle emulation of the old Netscape and Mosaic bug, which treated > as a co-terminator for double-quotes and tags (off).
-source	Dump the source of the first file to stdout and exit.
-startfile_ok	Allow non-http startfile and home page with `-validate` (off).
-stdin	Read startfile from standard input (off).
-term=TERM	Set terminal type to TERM.
-tlog	Toggle use of a Lynx Trace Log for the current session (on).
-trace	Turns on `lynx` trace mode (off).
-trace_mask	Customize `lynx` trace mode (0).
-traversal	Traverse all http links derived from startfile.
-verbose	Toggle [LINK], [IMAGE], and [INLINE] comments with file names of these images (on).
-version	Print `lynx` version information.
-vikeys	Enable vi-like key movement (off).
-width=NUMBER	Screen width for formatting of dumps (default is 80).

...(options omitted)...

Toolbox FTP archive Web site

ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/lynx

Tool developer Web site

http://lynx.isc.org

Tool FTP archive Web site

http://lynx.isc.org/current

Files

```
/opt/freeware/bin/lynx
/opt/freeware/doc/lynx-2.8.4
/opt/freeware/doc/lynx-2.8.4/INSTALLATION
/opt/freeware/doc/lynx-2.8.4/README
/opt/freeware/doc/lynx-2.8.4/docs/
```

...(files omitted)...

```
/opt/freeware/etc/lynx.cfg
/opt/freeware/man/man1/lynx.1
/usr/bin/lynx
```

Prereqs

The LYNX package does not have any prerequisites.

Examples

The first example dumps the first page from a URL, with **lynx**, into a file. In the following example we use www.ibm.com:

```
lynx -source www.ibm.com > ibmcom.html
```

The next example shows how to use **lynx** from a simple networked terminal session to the AIX system, such as **telnet** from a Windows system, and then using **lynx** to browse a URL. In this example we use www.ibm.com:

```
lynx -vikeys -term=vt100 www
```

Example 6-2 shows how to use **lynx** to display the files and structure of the main Toolbox directory by using the -dump option.

Example 6-2 Using lynx

```
root@fenris:/: lynx -dump ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox

Current directory is /aix/freeSoftware/aixtoolbox

   [1]Up to freeSoftware
  _______________________________________________________________________________
```

```
Please read the file README.txt
  it was last modified on Tue Aug 20 10:58:17 2002 - 50 days ago
  ___________________________________________________________________________

Oct  8 17:03  text/plain       [2]CONTENTS  25Kb
May 16 17:02  text/plain       [3]FAQ  10Kb
May 16 17:02  text/html        [4]FAQ.html  12Kb
May 16 17:02  text/plain       [5]FAQ.txt  10Kb
Apr 16  2001  Directory        [6]INSTALLP
Aug  6 16:55  Directory        [7]LICENSES
Aug 20 10:58  text/plain       [8]README.txt  35Kb
Apr 16  2001  Directory        [9]RPMS
Aug 30 10:54  Directory        [10]SRPMS
Sep 13 14:53  Directory        [11]contrib
Aug 27 12:54  Directory        [12]data
Apr 16  2001  Directory        [13]docs
Apr  5  2002  Directory        [14]ezinstall
Sep 17 15:03  text/plain       [15]toolreq.txt  2Kb

References

   1. ftp://ftp.software.ibm.com/aix/freeSoftware
   2. ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/CONTENTS
   3. ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/FAQ
   4. ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/FAQ.html
   5. ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/FAQ.txt
   6. ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/INSTALLP
   7. ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/LICENSES
   8. ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/README.txt
   9. ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS
  10. ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/SRPMS
  11. ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/contrib
  12. ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/data
  13. ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/docs
  14. ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/ezinstall
  15. ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/toolreq.txt
```

6.2.8 curl

The CURL package provides the **curl** command that is a client to get documents/files from or send documents to a server, using any of the supported protocols (HTTP, HTTPS, FTP, GOPHER, DICT, TELNET, LDAP, or FILE). The command is designed to work without user interaction.

Synopsis

`curl [options...] url`

The following list describes options for the `curl` command:

-a/--append	Append to target file when uploading (F).
-A/--user-agent *string*	User-agent to send to server (H).
-b/--cookie *name=string/file*	Cookie string or file to read cookies from (H).
-B/--use-ascii	Use ASCII/text transfer.
-c/--cookie-jar *file*	Write all cookies to this file after operation (H).
-C/--continue-at *offset*	Specify absolute resume offset.
-d/--data *data*	HTTP POST data (H).
--data-ascii *data*	HTTP POST ASCII data (H).
--data-binary *data*	HTTP POST binary data (H).
--disable-epsv	Prevent `curl` from using EPSV (F).
-D/--dump-header *file*	Write the headers to this file.
--egd-file *file*	EGD socket path for random data (SSL).
-e/--referer	Referer page (H).
-E/--cert *cert[:passwd]*	Specify your certificate file and password (HTTPS).
--cert-type *type*	Specify your certificate file type (DER/PEM/ENG) (HTTPS.
--key *key*	Specify your private key file (HTTPS).
--key-type *type*	Specify your private key file type (DER/PEM/ENG) (HTTPS).
--pass *pass*	Specify your pass phrase for the private key (HTTPS).
--engine *eng*	Specifies the crypto engine to use (HTTPS).
--cacert *file*	CA certificate to verify peer against (SSL).
--ciphers *list*	What SSL ciphers to use (SSL).
--connect-timeout *seconds*	Maximum time in *seconds* allowed for connection.
-f/--fail	Fail silently (no output at all) on errors (H).
-F/--form *name=content*	Specify HTTP POST data (H).
-g/--globoff	Disable URL sequences and ranges using {} and [].
-G/--get	Send the -d data with a HTTP GET (H).
-h/--help	This help text.
-H/--header *line*	Custom header to pass to server (H).
-i/--include	Include the HTTP-header in the output (H).
-I/--head	Fetch document info only (HTTP HEAD/FTP SIZE).
--interface *interface*	Specify the interface to be used.

--krb4 *level*	Enable *krb4* with specified security level (F).
-K/--config	Specify which configuration file to read.
-l/--list-only	List only names of an FTP directory (F).
-L/--location	Follow Location: hints (H).
-m/--max-time *seconds*	Maximum time allowed for the transfer.
-M/--manual	Display huge help text.
-n/--netrc	Read .netrc for user name and password.
-N/--no-buffer	Disables the buffering of the output stream.
-o/--output *file*	Write output to *file* instead of stdout.
-O/--remote-name	Write output to a file named as the remote file.
-p/--proxytunnel	Perform non-HTTP services through a HTTP proxy.
-P/--ftpport *address*	Use PORT with address instead of PASV when FTPing.
-q	When used as the first parameter disables .curlrc.
Q/--quote *cmd*	Send QUOTE command to FTP before file transfer (F).
-r/--range *range*	Retrieve a byte range from a HTTP/1.1 or FTP server.
-R/--remote-time	Set the remote file's time on the local output.
-s/--silent	Silent mode. Do not output anything.
-S/--show-error	Show error. With -s, make `curl` show errors when they occur.
--stderr *file*	Where to redirect stderr - means stdout.
-t/--telnet-option *OPT=val*	Set `telnet` option.
-T/--upload-file *file*	Transfer/upload *file* to remote site.
--url *URL*	Another way to specify URL to work with.
-u/--user *user[:password]*	Specify user and password to use.
-U/--proxy-user *user[:password]*	Specify proxy authentication.
-v/--verbose	Make the operation more talkative.
-V/--version	Output version number then quit.
-w/--write-out [format]	What to output after completion.
-x/--proxy *host[:port]*	Use proxy (default port is 1080).
--random-file *file*	File to use for reading random data from (SSL).
-X/--request *command*	Specific request command to use.
-y/--speed-time	Time needed to trig speed-limit abort. Default to 30.
-Y/--speed-limit	Stop transfer if below speed-limit for 'speed-time' secs.

-z/--time-cond ***time***	Include a time condition to the server (H).
-Z/--max-redirs ***num***	Set maximum number of redirections allowed (H).
-0/--http1.0	Force usage of HTTP 1.0 (H).
-1/--tlsv1	Force usage of TLSv1 (H).
-2/--sslv2	Force usage of SSLv2 (H).
-3/--sslv3	Force usage of SSLv3 (H).
-#/--progress-bar	Display transfer progress as a progress bar.
--crlf	Convert LF to CRLF in upload. Useful for MVS (OS/390).

Note: Options: (H) means HTTP/HTTPS only, (F) means FTP only.

Toolbox FTP archive Web site

`ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/curl`

Tool developer Web site

`http://curl.haxx.se`

Tool FTP archive Web site

`http://curl.haxx.se/download`

Files

```
/opt/freeware/include/curl/curl.h
/opt/freeware/include/curl/easy.h
/opt/freeware/include/curl/mprintf.h
/opt/freeware/include/curl/stdcheaders.h
/opt/freeware/include/curl/types.h
/opt/freeware/lib/libcurl.la
/opt/freeware/man/man3/
```

...(files omitted)...

```
/usr/include/curl
```

Prereqs

The CURL package does not have any prerequisites.

Examples

The following example will copy the first file provided by the remote Web server from the `http://curl.haxx.se` Web site and store it in the index.html file in the current directory.

Example 6-3 Using curl to copy a HTTP requested file

```
root@fenris:/tmp: curl -o index.html http://curl.haxx.se
  % Total    % Received % Xferd  Average Speed          Time             Curr.
                                 Dload  Upload Total    Current  Left    Speed
100  7521    0  7521    0     0  6787      0 --:--:--  0:00:01 --:--:-- 10918
```

The next example uses the --range/-r option to download different parts of a large file from different or the same FTP server. In our example we use the same FTP server. The total file size is 675315712 bytes and the file is the first CD-ROM image in ISO format for Version 8 of the Red Hat Linux distribution. Notice that we run the commands “simultaneously” in the background and save the progress report from **curl** in a separate log file (iso-#.out).

Example 6-4 Downloading parts of a large file using curl simultaneously

```
root@fenris:/tmp: curl -r 0-199999999 -o psyche-i386-disc1.iso-1
ftp://ftp.redhat.com/pub/redhat/linux/8.0/en/iso/i386/psyche-i386-disc1.iso >
iso-1.out &

root@fenris:/tmp: curl -r 200000000-399999999 -o psyche-i386-disc1.iso-2
ftp://ftp.redhat.com/pub/redhat/linux/8.0/en/iso/i386/psyche-i386-disc1.iso >
iso-2.out &

root@fenris:/tmp: curl -r 400000000- -o psyche-i386-disc1.iso-3
ftp://ftp.redhat.com/pub/redhat/linux/8.0/en/iso/i386/psyche-i386-disc1.iso >
iso-3.out &
```

The **curl** command will report the progress for each of the downloads. Example 6-5 shows a sample of the output.

Example 6-5 Sample progress report from a curl command using the --range/-r option

```
% Total    % Received % Xferd  Average Speed          Time             Curr.
                                 Dload  Upload Total    Current  Left    Speed
  0  644M    0   512    0     0     18      0 9895:37:02  0:00:27 9895:36:35
  0  644M    0  1024    0     0     36      0 5131:10:58  0:00:28 5131:10:30    5
  0  644M    0  1024    0     0     35      0 5314:44:25  0:00:29 5314:43:56    2
  0  644M    0  2560    0     0     86      0 2173:35:57  0:00:29 2173:35:28    7
  0  644M    0  2560    0     0     83      0 2247:01:19  0:00:30 2247:00:49    5
  0  644M    0  2560    0     0     80      0 2320:22:18  0:00:31 2320:21:47    4
  0  644M    0  3072    0     0     94      0 1975:17:19  0:00:32 1975:16:47    4
  0  644M    0  4608    0     0    138      0 1357:38:58  0:00:33 1357:38:25    8
  0  644M    0  4608    0     0    134      0 1398:26:24  0:00:34 1398:25:50    4
  0  644M    0  7168    0     0    198      0 942:54:38  0:00:36 942:54:02   858
  0  644M    0  7168    0     0    193      0 969:07:59  0:00:37 969:07:22   858
  0  644M    0  8704    0     0    230      0 813:23:24  0:00:37 813:22:47  1044
  0  644M    0 10752    0     0    277      0 675:56:25  0:00:38 675:55:47  1139
```

..(lines omitted)...

After the downloads are complete we can concatenate the files together with the **cat** command (or similar):

```
cat psyche-i386-disc1.iso-1 psyche-i386-disc1.iso-2 psyche-i386-disc1.iso-3
> psyche-i386-disc1.iso
```

6.2.9 elm

The ELM package provides the text-based **elm** e-mail user agent. The **elm** command offers all the features of **/bin/mail**, Berkeley Mail, **uumail**, and the AT&T Mail family of UNIX mailers in a unified and intuitive fashion.

Synopsis

```
elm [ -achKMmStVvz] [ -i file ] [ -r file ] [ -f alternate-folder ] [ -d
debug-level ] [ -s subject ] list of aliases or addresses
```

The following list describes options for the **elm** command:

-a Arrow - force the arrow cursor (instead of the inverse bar)

-c Checkalias - expand the following aliases and return.

-d debug-level Debug - set specified debug level.

-f alternative-folder Folder - read folder (specified) rather than the incoming mailbox.

-h Help - give this list of options

-i file Include - include prepared file in edit buffer for sending.

-k Keypad - force knowledge of HP terminal keyboard.

-K Keypad + softkeys - enable use of softkeys on HP terminals only.

-M Magic mode - treat all folders as spool files.

-m Menu - Turn off menu, using more of the screen

-r file Rcfile - use file rather than the default elmrc.

-S Send-only mode

-s subject Subject - specify subject for message to mail.

-t TiTe - do not use termcap/terminfo ti/te entries.

-V Enable sendmail voyeur mode.

-v Print out **elm** version information.

-z Zero - do not enter **elm** if no mail is pending.

Toolbox FTP archive Web site

ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/elm

Tool developer Web site

http://www.instinct.org/elm

Tool FTP archive Web site

ftp://ftp.virginia.edu/pub/elm

Files

```
/etc/X11/applnk/Internet/elm.desktop
/opt/freeware/bin/answer
/opt/freeware/bin/checkalias
/opt/freeware/bin/elm
/opt/freeware/bin/elmalias
/opt/freeware/bin/fastmail
/opt/freeware/bin/frm
/opt/freeware/bin/listalias
/opt/freeware/bin/messages
/opt/freeware/bin/newalias
/opt/freeware/bin/newmail
/opt/freeware/bin/nfrm
/opt/freeware/bin/printmail
/opt/freeware/bin/prlong
/opt/freeware/bin/readmsg
/opt/freeware/bin/wnewmail
/opt/freeware/lib/elm
```

...(files omitted)...

```
/opt/freeware/man/man1/1
```

...(files omitted)...

```
/usr/bin/answer
/usr/bin/checkalias
/usr/bin/elm
/usr/bin/elmalias
/usr/bin/fastmail
/usr/bin/frm
/usr/bin/listalias
/usr/bin/messages
/usr/bin/newalias
/usr/bin/newmail
/usr/bin/nfrm
/usr/bin/printmail
/usr/bin/prlong
/usr/bin/readmsg
/usr/bin/wnewmail
/usr/lib/elm
```

Prereqs

The ELM package does not have any prerequisites.

Examples

Start the **elm** e-mail user agent by typing `elm` on the command line:

```
elm
```

Figure 6-2 shows the default **elm** full-screen display in a telnet.exe window log i to the AIX system from a PC client.

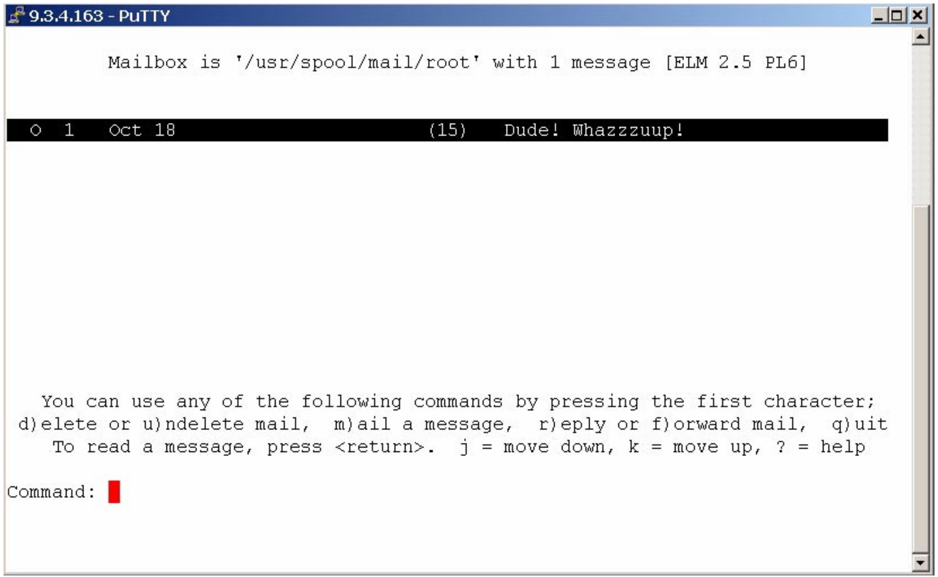

Figure 6-2 ELM screen display

6.2.10 fetchmail

The FETCHMAIL package provides the **fetchmail** command, which is a remote mail retrieval and forwarding utility intended to be used over on-demand TCP/IP links (such as SLIP or PPP connections). It retrieves mail from remote mail servers and forwards it to the local client system's delivery system, so that the mail can then be read by normal mail user agents such as **mutt**, **elm**, **pine**, or **mailx**. The **fetchmailconf** command is an interactive GUI configurator and is part of the FETCHMAILCONF package.

As each message is retrieved, **fetchmail** normally delivers it via SMTP to port 25 on the machine it is running on (localhost), just as though it were being passed in over a normal TCP/IP network connection. The mail will then be delivered locally via the installed Mail Delivery Agent (MDA), usually **sendmail**, that is listening on the SMTP port.

Synopsis

```
fetchmail [options] [server ...]
```

The following list describes options for the **fetchmail** command:

-?, --help	Display this option help.
-V, --version	Display version info.
-c, --check	Check for messages without fetching.
-s, --silent	Work silently.
-v, --verbose	Work noisily (diagnostic output).
-d, --daemon	Run as a daemon once per n seconds.
-N, --nodetach	Do not detach daemon process.
-q, --quit	Kill daemon process.
-L, --logfile	Specify log file name.
--syslog	Use syslog(3) for most messages when running as a daemon.
--invisible	Do not write Received & enable host spoofing.
-f, --fetchmailrc	Specify alternate run control file.
-i, --idfile	Specify alternate UIDs file.
--postmaster	Specify recipient of last resort.
--nobounce	Redirect bounces from user to postmaster.
--plugin	Specify external command to open connection.
--plugout	Specify external command to open SMTP connection.
-p, --protocol	Specify retrieval protocol.
-U, --uidl	Force the use of UIDLs (pop3 only).
-P, --port	TCP/IP service port to connect to.
--auth	Authentication type (password/kerberos/ssh).
-t, --timeout	Server non-response timeout.
-E, --envelope	Envelope address header.
-Q, --qvirtual	Prefix to remove from local user ID.
--principal	Mail service principal.
--tracepolls	Add poll-tracing information to Received header.
-u, --username	Specify user's login on server.
-a, --all	Retrieve old and new messages.
-K, --nokeep	Delete new messages after retrieval.
-k, --keep	Save new messages after retrieval.
-F, --flush	Delete old messages from server.
-n, --norewrite	Do not rewrite header addresses.
-l, --limit	Do not fetch messages over given size.
-w, --warnings	Interval between warning mail notification.
-S, --smtphost	Set SMTP forwarding host.
--fetchdomains	Fetch mail for specified domains.
-D, --smtpaddress	Set SMTP delivery domain to use.
--smtpname	Set SMTP full name username@domain.
-Z, --antispam	Set antispam response values.

-b, --batchlimit	Set batch limit for SMTP connections.
-B, --fetchlimit	Set fetch limit for server connections.
-e, --expunge	Set max deletions between expunges.
-m, --mda	Set MDA to use for forwarding.
--bsmtp	Set output BSMTP file.
--lmtp	Use LMTP (RFC2033) for delivery.
-r, --folder	Specify remote folder name.
--showdots	Show progress dots even in log files.

Toolbox FTP archive Web site

`ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/fetchmail`

Tool developer Web site

`http://tuxedo.org/~esr/fetchmail`

Tool FTP archive Web site

`http://tuxedo.org/~esr/fetchmail/fetchmail-*.tar.gz`

Files

```
/opt/freeware/bin/fetchmail
/opt/freeware/doc/fetchmail-5.9.6
/opt/freeware/doc/fetchmail-5.9.6/COPYING
/opt/freeware/doc/fetchmail-5.9.6/FAQ
/opt/freeware/doc/fetchmail-5.9.6/FEATURES
/opt/freeware/doc/fetchmail-5.9.6/NEWS
/opt/freeware/doc/fetchmail-5.9.6/NOTES
/opt/freeware/doc/fetchmail-5.9.6/README
/opt/freeware/doc/fetchmail-5.9.6/contrib/
```

...(files omitted)...

```
/opt/freeware/doc/fetchmail-5.9.6/design-notes.html
/opt/freeware/doc/fetchmail-5.9.6/fetchmail-FAQ.html
/opt/freeware/doc/fetchmail-5.9.6/fetchmail-features.html
/opt/freeware/lib/locale/
```

...(files omitted)...

```
/opt/freeware/man/man1/fetchmail.1
/opt/freeware/man/man1/fetchmailconf.1
/usr/bin/fetchmail
```

Prereqs

The FETCHMAIL package does not have any prerequisites.

Examples

The first **`fetchmail`** example (Example 6-6 on page 176) uses command line options with the **`fetchmail`** command. In the example we specify that the messages should not be deleted from the mailbox on the POP3 server mbox1, but should be kept there with the -k option. We also specify that the POP3 protocol should be used (default is the IMAP protocol), and we are prompted to supply the password interactively.

Example 6-6 Using fetchmail with command line options

```
root@fenris:/: fetchmail -u root -k -p POP3 mbox1
Enter password for root@mbox1:
1 message (1 seen) for root at mbox1 (324 octets).
skipping message root@mbox1:1 (324 octets) not flushed
```

Before you use **fetchmail**, you could create a .fetchmailrc file in your home directory. The ~/.fetchmailrc file can be extensively customized, so please refer to the man pages to set it up appropriately for your environment. Note that the file ~/.fetchmailrc must have no more than -rwx--x--- (0710) permissions.

The following is the syntax for the poll statement in the ~/.fetchmailrc file:

```
poll SERVERNAME protocol PROTOCOL username NAME password PASSWORD
```

The following is a very simple example of using a POP3 server.

If the POP3 server is an AIX system, check that inetd is listening on the POP3 port (normally 110) with the **lssrc -ls inetd** command. If it is not, remove the prefixed # sign in /etc/inetd.conf and execute **refresh -s inetd** after saving the file. Now the POP3 service should be available.

The ~/.fetchmailrc file (Example 6-7) specifies mail polling of the mbox1 host for user root with password nobody.

```
poll mbox1 protocol pop3 username root password nobody
```

Example 6-7 Using fetchmail

```
root@fenris:/: fetchmail
1 message for root at mbox1 (344 octets).
reading message root@mbox1:1 of 1 (344 octets)  flushed
```

6.2.11 mutt

The MUTT package provides the **mutt** command, which is a text mode e-mail user agent. The **mutt** mail user agent supports color, threading, arbitrary key remapping, and a lot of customization.

Synopsis

mutt [-nRzZ] [-e *cmd*] [-F *file*] [-m *type*] [-f *file*]

```
mutt [ -nx ] [ -e cmd ] [ -a file ] [ -F file ] [ -H file ] [ -i file ] [ -s
subj ] [ -b addr ] [ -c addr ] addr [ ... ]
mutt [ -n ] [ -e cmd ] [ -F file ] -p mutt -v[v]
```

The following list describes options for the **mutt** command:

-a *file*	Attach a file to the message.
-b *address*	Specify a blind carbon-copy (BCC) address.
-c *address*	Specify a carbon-copy (CC) address.
-e *command*	Specify a command to be executed after initialization.
-f *file*	Specify which mailbox to read.
-F *file*	Specify an alternate muttrc file.
-H *file*	Specify a draft file to read header from.
-i *file*>	Specify a file that **mutt** should include in the reply.
-m *type*	Specify a default mailbox type.
-n	Causes **mutt** to not read the system Muttrc.
-p	Recall a postponed message.
-R	Open mailbox in read-only mode.
-s *subj*	Specify a subject (must be in quotes if it has spaces).
-v	Show version and compile-time definitions.
-x	Simulate the **mailx** send mode.
-y	Select a mailbox specified in your *mailboxes* list.
-z	Exit immediately if there are no messages in the mailbox.
-Z	Open the first folder with new message, exit immediately if none.
-h	Help message.

Toolbox FTP archive Web site

ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/mutt

Tool developer Web site

http://www.mutt.org

Tool FTP archive Web site

ftp://ftp.mutt.org/mutt

Files

```
/etc/Muttrc
/opt/freeware/bin/mutt
/opt/freeware/bin/mutt_dotlock
/opt/freeware/bin/muttbug
/opt/freeware/bin/pgpewrap
/opt/freeware/bin/pgpring
/opt/freeware/doc
/opt/freeware/doc/mutt-1.2.5i/
```

...(files omitted)...
```
/opt/freeware/etc/Muttrc
/opt/freeware/man/man1/mutt.1
/opt/freeware/man/man1/mutt_dotlock.1
/opt/freeware/man/man5/muttrc.5
/opt/freeware/share/locale/
```
...(files omitted)...
```
/usr/bin/mutt
/usr/bin/mutt_dotlock
/usr/bin/muttbug
/usr/bin/pgpewrap
/usr/bin/pgpring
```

Prereqs

The MUTT package does not have any prerequisites.

Examples

Start the **mutt** e-mail user agent by typing `mutt` in the command line:

```
mutt
```

Figure 6-3 shows the default **mutt** full screen display in a telnet.exe window log into the AIX system from a PC client.

Figure 6-3 MUTT screen display

6.2.12 proftpd

The PROFTPD package provides an enhanced File Transfer Protocol (FTP) server with a focus toward simplicity, security, and ease of configuration. It features a very Apache-like configuration syntax, and a highly customizable server infrastructure, including support for multiple *virtual* FTP servers, anonymous FTP, and permission-based directory visibility.

Synopsis

`proftpd [options]`

The following list describes options for the **proftpd** command:

-h	Display **proftpd** usage.
-n	Disable background daemon mode (all output goes to tty, instead of syslog).
-d [level]	Set debugging level (0–5; 5 = most debugging).
-c [config-file]	Specify alternate configuration file.
-p [0|1]	Enable/disable default persistent passwd support.
-l	List all compiled-in modules.
-t	Test the syntax of the specified config.
-v	Print version number and exit.
-vv	Print extended version information and exit.

Toolbox FTP archive Web site

ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/proftpd

Tool developer Web site

http://www.proftpd.org

Tool FTP archive Web site

ftp://ftp.proftpd.org/distrib/source

Files

```
/etc/proftpd.conf
/home/ftp
/opt/freeware/bin/ftpcount
/opt/freeware/bin/ftpwho
/opt/freeware/doc/proftpd-1.2.4
/opt/freeware/doc/proftpd-1.2.4/README
/opt/freeware/doc/proftpd-1.2.4/README.AIX
...(files omitted)...
/opt/freeware/etc/proftpd.conf
/opt/freeware/man/man1/ftpcount.1
/opt/freeware/man/man1/ftpwho.1
```

```
/opt/freeware/man/man5/xferlog.5
/opt/freeware/man/man8/ftpshut.8
/opt/freeware/man/man8/proftpd.8
/opt/freeware/sbin/ftpshut
/opt/freeware/sbin/in.proftpd
/opt/freeware/sbin/proftpd
/usr/bin/ftpcount
/usr/bin/ftpwho
/usr/sbin/ftpshut
/usr/sbin/in.proftpd
/usr/sbin/proftpd
/var/run/proftpd
```

Prereqs

The PROFTPD package does not have any prerequisites.

Remarks

We recommend that you administer PROFTPD by using **webmin**. Connect a browser to the **webmin** server on the systems where you have installed PROFTPD.

Please refer to the WEBMIN package (6.5.1, “webmin” on page 254) for more information about **webmin**.

6.2.13 pine

The PINE package provides the **pine** command, which is a full-featured e-mail user agent that includes a simple text editor called **pico**. The **pine** command supports MIME extensions and can also be used to read news. It also supports IMAP, **mail**, and MH style folders.

Synopsis

`pine [ options ] [ address , address ]`

The following list describes options for the **pine** command:

address	Send mail to address.
-attach file	Send mail with the listed file as an attachment.
-attachlist file-list	Send mail with the listed filelist as an attachment.
-attach_and_delete file	Send mail with the listed file as an attachment, and remove the file after the message is sent.
-bail	Exit if the pinerc file does not exist.
-c context-number	context-number is the number corresponding to the folder-collection to which the -f command line argument should be applied.

-conf	Produce a sample/fresh copy of the system-wide configuration file, pine.conf, on the standard output.
-convert_sigs -p pinerc	Convert signature files into literal signatures.
-copy_abook *local_ abook remote_abook*	Copy the local address book file to a remote address book folder.
-copy_pinerc *local_ pinerc remote_pinerc*	Copy the local pinerc file to a remote pinerc folder.
-create_lu addrbook sort-order	Create auxiliary index (look-up) file for addrbook and sorts addrbook in sort-order.
-d debug-level	Output diagnostic info at debug-level (0-9) to the current .pine-debug[1-4] file. A value of 0 turns debugging off and suppresses the .pine-debug file.
-d key[=val]	Fine tuned output of diagnostic messages where "**`flush`**" causes debug file writing without buffering, "**`timestamp`**" appends each message with a time stamp, "**`imap=`*`n`***" where *n* is between 0 and 4 representing none to verbose IMAP telemetry reporting, "**`numfiles=`*`n`***" where *n* is between 0 and 31 corresponding to the number of debug files to maintain, and "**`verbose=`*`n`***" where *n* is between 0 and 9 indicating an inverse threshold for message output.
-f folder	Open folder (in first defined folder collection, use -c n to specify another collection) instead of INBOX.
-F file	Open named text file and view with **`pine`**'s browser.
-h	Help: list valid command-line options.
-i	Start up in the FOLDER INDEX screen.
-I keystrokes	Initial (comma-separated list of) keystrokes which **`pine`** should execute on startup.
-k	Use function keys for commands. This is the same as running the command **`pinef`**.
-n number	Start up with current message-number set to `number`.
-o	Open first folder read-only.
-p config-file	Use config-file as the personal configuration file instead of the default .pinerc.
-P config-file	Use config-file as the configuration file instead of default system-wide configuration file pine.conf.
-pinerc file	Output fresh pinerc configuration to file, preserving the settings of variables. Use file set to “-” to make output go to standard out.
-r	Use restricted/demo mode.
-sort order	Sort the FOLDER INDEX display in one of the following orders: arrival, date, subject, orderedsubj, thread, from, or reverse.

-supported	Some options may or may not be supported depending on how **pine** was compiled.
-url url	Open the given URL. Cannot be used with -f, -F, or -attach options.
-v, -version	Print version information.
-x config	Use configuration exceptions in config.
-z	Enable ^Z and SIGTSTP so **pine** may be suspended.
-option=*value*	Assign *value* to the config option.

Toolbox FTP archive Web site

ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/pine

Tool developer Web site

http://www.washington.edu/pine

Tool FTP archive Web site

ftp://ftp.cac.washington.edu/pine

Files

```
/etc/pine.conf
/etc/pine.conf.fixed
/opt/freeware/bin/pico
/opt/freeware/bin/pilot
/opt/freeware/bin/pine
/opt/freeware/doc/pine-4.44/
...(files omitted)...
/opt/freeware/man/man1/pico.1
/opt/freeware/man/man1/pilot.1
/opt/freeware/man/man1/pine.1
/usr/bin/pico
/usr/bin/pilot
/usr/bin/pine
```

Prereqs

The PINE package requires **libldap** from the OpenLDAP package.

Examples

Start the **pine** e-mail user agent by typing `pine` on the command line:

```
pine
```

Figure 6-4 on page 183 shows the **pine** folder view full-screen display in a telnet.exe window login in to the AIX system from a PC client.

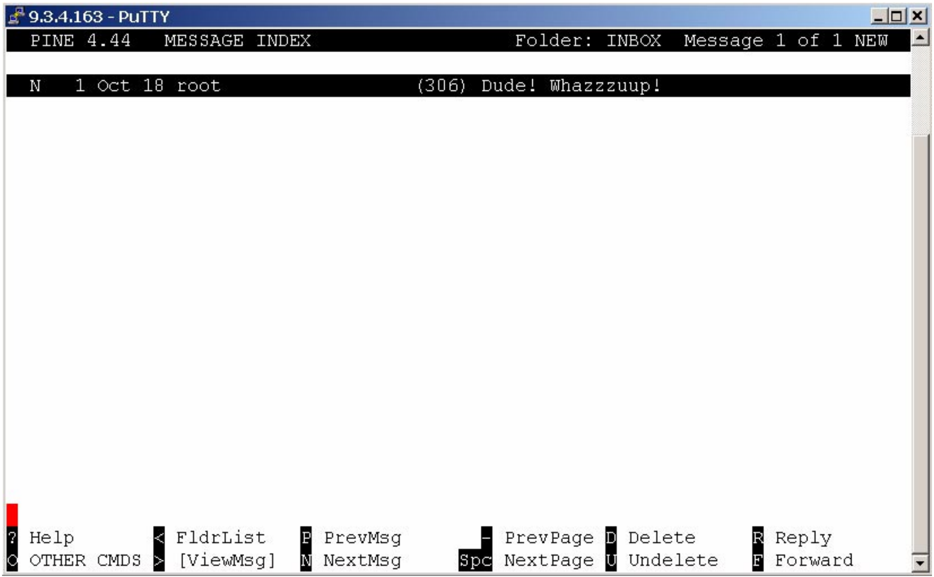

Figure 6-4 PINE screen display

6.2.14 wu-ftpd

The WU-FTPD package contains the **wu-ftpd** File Transfer Protocol server daemon. The FTP protocol is a method of transferring files between machines on a network and/or over the Internet. **wu-ftpd** features include logging of transfers, logging of commands, on the fly compression and archiving, classification of users' type and location, per class limits, per directory upload permissions, restricted guest accounts, system-wide and per-directory messages, directory alias, cdpath, file name filter, and virtual host support.

Synopsis

```
ftpd [ -d ] [ -v ] [ -l ] [ -t timeout ] [ -T maxtimeout ] [ -a ] [ -A ] [ -L ]
[ -i ] [ -I ] [ -o ] [ -p ctrlport ] [ -P dataport ] [ -q ] [ -Q ] [ -r rootdir
] [ -s ] [ -S ] [ -u umask ] [ -V ] [ -w ] [ -W ] [ -X ]
```

The following list describes options for the **wu-ftpd** command:

-d, -v If the -d or -v option is specified, debugging information is written to the **syslog**.

-l If the -l option is specified, each FTP session is logged in the **syslog**.

-t timeout If the -t option is specified, the inactivity timeout period will be set to timeout seconds.

-a	If the -a option is specified, the use of the ftpaccess configuration file is enabled.
-A	If the -A option is specified, use of the ftpaccess configuration file is disabled. This is the default.
-L	If the -L option is specified, commands sent to the **`ftpd`** server will be logged to the **`syslog`**.
-i	If the -i option is specified, files received by the **`ftpd`** server will be logged to the xferlog.
-o	If the -o option is specified, files transmitted by the **`ftpd`** server will be logged to the xferlog.
-X	If the -X option is specified, the output created by the -i and -o options is not saved to the xferlog file but saved via syslog so you can collect output from several hosts on one central loghost.
-u umask	If the -u option is specified, the default umask is set to `umask`.
-W	If the -W option is specified, user logins are not recorded in the wtmp file. The default (-w) is to record every login and logout.
-s, -S	The -s and -S options place the daemon in stand-alone operation mode.
-p ctrlport **-P dataport**	The -p and -P options override the port numbers used by the daemon.
-q, -Q	The -q and -Q options determine whether the daemon uses the PID files.
-r rootdir	The -r option instructs the daemon to **`chroot`** to the specified **`rootdir`** immediately upon loading.

Toolbox FTP archive Web site

`ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/wu-ftpd`

Tool developer Web site

`http://www.wu-ftpd.org`

Tool FTP archive Web site

`ftp://ftp.wu-ftpd.org/pub`

Files

```
/etc/ftpaccess
/etc/ftpconversions
/opt/freeware/bin/ftpcount
/opt/freeware/bin/ftpwho
/opt/freeware/doc/wu-ftpd-2.6.2/
```

```
...(files omitted)...
/opt/freeware/man/man1/ftpcount.1
/opt/freeware/man/man1/ftpwho.1
/opt/freeware/man/man5/ftpaccess.5
/opt/freeware/man/man5/ftpconversions.5
/opt/freeware/man/man5/ftphosts.5
/opt/freeware/man/man5/ftpservers.5
/opt/freeware/man/man5/xferlog.5
/opt/freeware/man/man8/ftpd.8
/opt/freeware/man/man8/ftprestart.8
/opt/freeware/man/man8/ftpshut.8
/opt/freeware/man/man8/privatepw.8
/opt/freeware/sbin/ckconfig
/opt/freeware/sbin/ftprestart
/opt/freeware/sbin/ftpshut
/opt/freeware/sbin/in.ftpd
/opt/freeware/sbin/in.wuftpd
/opt/freeware/sbin/privatepw
/opt/freeware/sbin/wu.ftpd
/opt/freeware/sbin/xferstats
/usr/bin/ftpcount
/usr/bin/ftpwho
/usr/sbin/ckconfig
/usr/sbin/ftprestart
/usr/sbin/ftpshut
/usr/sbin/in.ftpd
/usr/sbin/in.wuftpd
/usr/sbin/privatepw
/usr/sbin/wu.ftpd
/usr/sbin/xferstats
```

Prereqs

The WU-FTPD package does not have any prerequisites.

Remarks

We recommend that you administer WU-FTPD by using **webmin**. Connect a browser to the **webmin** server on the systems where you have installed WU-FTPD. Please refer to the WEBMIN package (6.5.1, “webmin” on page 254) for more information about **webmin**.

6.3 File handling tools

The file handling packages described here are a sample of different tools from the Toolbox that are mainly used for file handling or file manipulation.

We cover the following tools in this section:

- **zip**
- **unzip**
- **gzip / gunzip**
- **bzip2 / bunzip2**
- **zoo**
- **lsof**
- **less**
- **fileutils**
- **findutils**
- **diffutils**
- **hexedit**
- **sh-utils**

6.3.1 zip

The ZIP package (from the RPMS/ppc/zip directory) is a collection of compression and file packaging utilities. The **zip** command is compatible with PKZIP from PKWARE (http://www.pkware.com).

Synopsis

```
zip [-options] [-b path] [-t mmddyyyy] [-n suffixes] [zipfile list] [-xi list]
```

The following list describes options for the **zip** command:

-f	Freshen - only changed files.
-d	Delete entries in zipfile.
-r	Recurse into directories.
-0	Store only.
-1	Compress faster.
-q	Quiet operation.
-c	Add one-line comments.
-@	Read names from stdin.
-x	Exclude the following names.
-F	Fix zipfile (-FF try harder).
-A	Adjust self-extracting exe.
-T	Test zipfile integrity.
-y	Store symbolic links as the link instead of the referenced file.
-R	PKZIP recursion (see manual).
-h	Show this help.
-u	Update - only changed or new files.
-m	Move into zipfile (delete files).
-j	Junk (do not record) directory names.

-l	Convert LF to CR LF.
-ll	Convert CR LF to LF.
-9	Compress better.
-v	Verbose operation/print version info.
-z	Add zipfile comment.
-o	Make zipfile as old as latest entry.
-i	Include only the following names.
-D	Do not add directory entries.
-J	Junk zipfile prefix (unzipsfx).
-X	Exclude extra file attributes.
-n	Do not compress these suffixes.

Toolbox FTP archive Web site

ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/zip

Tool developer Web site

http://www.info-zip.org

Tool FTP archive Web site

ftp://ftp.uu.net/pub/archiving/zip/src

Files

```
/opt/freeware/bin/zip
/opt/freeware/bin/zipcloak
/opt/freeware/bin/zipnote
/opt/freeware/bin/zipsplit
/opt/freeware/doc/zip-2.3
/opt/freeware/doc/zip-2.3/BUGS
/opt/freeware/doc/zip-2.3/CHANGES
/opt/freeware/doc/zip-2.3/MANUAL
/opt/freeware/doc/zip-2.3/README
/opt/freeware/doc/zip-2.3/TODO
/opt/freeware/doc/zip-2.3/WHATSNEW
/opt/freeware/doc/zip-2.3/WHERE
/opt/freeware/doc/zip-2.3/algorith.txt
/opt/freeware/man/man1/zip.1
/usr/bin/zip
/usr/bin/zipcloak
/usr/bin/zipnote
/usr/bin/zipsplit
```

Prereqs

The ZIP package does not have any prerequisites.

Examples

Example 6-8 shows how to compress several files in a directory into a compressed archive file, /tmp/zippo.zip.

Example 6-8 Using zip

```
root@fenris:/: zip /tmp/zippo .kshrc .profile .twmrc .xinitrc
updating: .kshrc (deflated 33%)
updating: .profile (stored 0%)
updating: .twmrc (deflated 65%)
updating: .xinitrc (deflated 66%)

root@fenris:/: ls -l /tmp/zippo*
-rw-r--r--   1 root     system        3533 Oct 09 16:28 /tmp/zippo.zip
```

6.3.2 unzip

The UNIZIP package (from the RPMS/ppc/unzip directory) is used to list, test, or extract files from a **zip** archive. The **unzip** command is compatible with PKUNZIP from PKWARE (http://www.pkware.com).

Synopsis

```
unzip [-Z] [-opts[modifiers]] file[.zip] [list] [-x xlist] [-d exdir]
```

The following list describes options for the **unzip** command:

-p Extract files to pipe, no messages.
-f Freshen existing files, create none.
-u Update files, create if necessary.
-x Exclude files that follow (in xlist).
-l List files (short format).
-t Test compressed archive data.
-z Display archive comment.
-d Extract files into exdir.

The following list describes modifiers for the **unzip** command:

-q Quiet mode.
-qq Quieter than -q mode.
-a Auto-convert any text files.
-aa Treat ALL files as text.
-v Be verbose/print version info.
-L Make (some) names lowercase.
-V Retain VMS version numbers.
-M Pipe through the **more** command.
-n Never overwrite existing files.

-o	Overwrite files without prompting.
-j	Junk paths (do not make directories).
-C	Match file names' case-insensitively.
-X	Restore UID/GID info.

Toolbox FTP archive Web site

ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/unzip

Tool developer Web site

http://www.info-zip.org

Tool FTP archive Web site

ftp://ftp.uu.net/pub/archiving/zip/src

Files

```
/opt/freeware/bin/funzip
/opt/freeware/bin/unzip
/opt/freeware/bin/unzipsfx
/opt/freeware/bin/zipgrep
/opt/freeware/bin/zipinfo
/opt/freeware/doc/unzip-5.41/
```

...(files omitted)...

```
/opt/freeware/man/man1/funzip.1
/opt/freeware/man/man1/unzip.1
/opt/freeware/man/man1/unzipsfx.1
/opt/freeware/man/man1/zipgrep.1
/opt/freeware/man/man1/zipinfo.1
/usr/bin/funzip
/usr/bin/unzip
/usr/bin/unzipsfx
/usr/bin/zipgrep
/usr/bin/zipinfo
```

Prereqs

The UNZIP package does not have any prerequisites.

Examples

The following example shows how to uncompress files from a compressed archive file, /tmp/zippo.zip.

Example 6-9 Using unzip

```
root@fenris:/: unzip /tmp/zippo.zip
Archive:  /tmp/zippo.zip
  inflating: motd
```

```
  inflating: .kshrc
 extracting: .profile
  inflating: .twmrc
  inflating: .xinitrc
```

6.3.3 gzip/gunzip

The GZIP package (from the RPMS/ppc/gzip directory) is a collection of compression and file packaging utilities. The **gzip** command reduces the size of the named files using Lempel-Ziv coding (LZ77). Whenever possible, each file is replaced by one with the extension .gz, while keeping the same ownership modes, access, and modification times. The **gzip** and **gunzip** commands are the same tool and both can be used to either **zip** or **unzip** archives, depending on the command line options used.

Synopsis

```
gzip [-cdfhlLnNrtvV19] [-S suffix] [file ...]
```

The following list describes options for the **gzip** command:

-c --stdout	Write on standard output, keep original files unchanged.
-d --decompress	Decompress.
-f --force	Force overwrite of output file and compress links.
-h --help	Give this help.
-l --list	List compressed file contents.
-L --license	Display software license.
-n --no-name	Do not save or restore the original name and time stamp.
-N --name	Save or restore the original name and time stamp.
-q --quiet	Suppress all warnings.
-r --recursive	Operate recursively on directories.
-S .suf --suffix .suf	Use suffix .suf on compressed files.
-t --test	Test compressed file integrity.
-v --verbose	Verbose mode.
-V --version	Display version number.
-1 --fast	Compress faster.
-9 --best	Compress better.
file...	Files to (de)compress. If none given, use standard input.

```
gunzip [-cdfhlLnNrtvV19] [-S suffix] [file ...]
```

The following list describes options for the **gunzip** command:

-c --stdout	Write on standard output, keep original files unchanged.
-d --decompress	Decompress.
-f --force	Force overwrite of output file and compress links.
-h --help	Give this help.

-l --list	List compressed file contents.
-L --license	Display software license.
-n --no-name	Do not save or restore the original name and time stamp.
-N --name	Save or restore the original name and time stamp.
-q --quiet	Suppress all warnings.
-r --recursive	Operate recursively on directories.
-S .suf --suffix .suf	Use suffix .suf on compressed files.
-t --test	Test compressed file integrity.
-v --verbose	Verbose mode.
-V --version	Display version number.
-1 --fast	Compress faster.
-9 --best	Compress better.
file...	Files to (de)compress. If none given, use standard input.

Toolbox FTP archive Web site

ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/gzip

Tool developer Web site

http://www.gzip.org

Tool FTP archive Web site

ftp://gatekeeper.dec.com/pub/GNU/gzip

Files

```
/opt/freeware/bin/gunzip
/opt/freeware/bin/gzexe
/opt/freeware/bin/gzip
/opt/freeware/bin/zcat
/opt/freeware/bin/zcmp
/opt/freeware/bin/zdiff
/opt/freeware/bin/zforce
/opt/freeware/bin/zgrep
/opt/freeware/bin/zless
/opt/freeware/bin/zmore
/opt/freeware/bin/znew
/opt/freeware/doc/gzip-1.2.4a
/opt/freeware/doc/gzip-1.2.4a/NEWS
/opt/freeware/doc/gzip-1.2.4a/README
/opt/freeware/info/gzip.info.gz
/opt/freeware/man/man1/
```

...(files omitted)...

```
/usr/bin/gunzip
/usr/bin/gzexe
/usr/bin/gzip
/usr/bin/zcmp
```

```
/usr/bin/zdiff
/usr/bin/zforce
/usr/bin/zgrep
/usr/bin/zless
/usr/bin/zmore
/usr/bin/znew
/usr/linux/bin/gunzip
/usr/linux/bin/gzexe
/usr/linux/bin/gzip
/usr/linux/bin/zcat
/usr/linux/bin/zcmp
/usr/linux/bin/zdiff
/usr/linux/bin/zforce
/usr/linux/bin/zgrep
/usr/linux/bin/zless
/usr/linux/bin/zmore
/usr/linux/bin/znew
```

Prereqs

The GZIP package requires the **/sbin/install-info** command from the INFO package.

Examples

Example 6-10 and Example 6-11 show how to compress several files in a directory into a compressed archive file, /tmp/zippo.gz. Note that we use the **tar** command to create the archive before compressing it.

Example 6-10 Using gzip

```
root@fenris:/: tar cvf - .kshrc .profile .twmrc .xinitrc | gzip >/tmp/zippo.gz
a .kshrc 1 blocks.
a .profile 1 blocks.
a .twmrc 8 blocks.
a .xinitrc 8 blocks.

root@fenris:/: ls -l /tmp/zippo*
-rw-r--r--   1 root     system         2827 Oct 09 16:39 /tmp/zippo.gz
```

The next example shows how to uncompress files from the compressed TAR archive file, /tmp/zippo.gz.

Example 6-11 Using gunzip

```
root@fenris:/: gunzip < /tmp/zippo.gz | tar xvf -
x .kshrc, 220 bytes, 1 media blocks.
x .profile, 23 bytes, 1 media blocks.
x .twmrc, 3774 bytes, 8 media blocks.
```

```
x .xinitrc, 3649 bytes, 8 media blocks.
```

6.3.4 bzip2/bunzip2

The BZIP2 package (from the RPMS/ppc/bzip2 directory) is a collection of compression and file packaging utilities. **bzip2** compresses files using the Burrows-Wheeler block sorting text compression algorithm, and Huffman coding. This type of compression is *generally* considerably better than that achieved by more conventional LZ77/LZ78-based compressors such as from the GZIP package.

Synopsis

```
bzip2 [ -cfkqstvVL123456789 ] [ filenames ... ]
```

The following list describes options for the **bzip2** command:

-h --help	Print this message.
-k --keep	Keep (do not delete) input files.
-f --force	Overwrite existing output files.
-t --test	Test compressed file integrity.
-c --stdout	Output to standard out.
-q --quiet	Suppress noncritical error messages.
-v --verbose	Be verbose (a second -v gives more).
-L --license	Display software version and license.
-V --version	Display software version and license.
-s --small	Use less memory (at most 2500 k).
-1 .. -9	Set block size to 100–900 k.
--fast	Alias for -1.
--best	Alias for -9.

```
bunzip2 [ -cfkqstvVL ] [ filenames ... ]
```

The following list describes options for the **bunzip2** command:

-h --help	Print this message.
-k --keep	Keep (do not delete) input files.
-f --force	Overwrite existing output files.
-t --test	Test compressed file integrity.
-c --stdout	Output to standard out.
-q --quiet	Suppress noncritical error messages.
-v --verbose	Be verbose (a second -v gives more).
-L --license	Display software version & license.
-V --version	Display software version & license.
-s --small	Use less memory (at most 2500 k).

Toolbox FTP archive Web site

ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/bzip2

Tool developer Web site

http://sources.redhat.com/bzip2

Tool FTP archive Web site

ftp://sources.redhat.com/pub/bzip2

Files

```
/opt/freeware/bin/bunzip2
/opt/freeware/bin/bzcat
/opt/freeware/bin/bzcmp
/opt/freeware/bin/bzdiff
/opt/freeware/bin/bzegrep
/opt/freeware/bin/bzfgrep
/opt/freeware/bin/bzgrep
/opt/freeware/bin/bzip2
/opt/freeware/bin/bzip2recover
/opt/freeware/bin/bzless
/opt/freeware/bin/bzmore
/opt/freeware/doc/bzip2-1.0.2/
...(files omitted)...
/opt/freeware/include/bzlib.h
/opt/freeware/lib/libbz2.a
/opt/freeware/man/man1/bzcmp.1
/opt/freeware/man/man1/bzdiff.1
/opt/freeware/man/man1/bzegrep.1
/opt/freeware/man/man1/bzfgrep.1
/opt/freeware/man/man1/bzgrep.1
/opt/freeware/man/man1/bzip2.1
/opt/freeware/man/man1/bzless.1
/opt/freeware/man/man1/bzmore.1
/usr/bin/bunzip2
/usr/bin/bzcat
/usr/bin/bzcmp
/usr/bin/bzdiff
/usr/bin/bzegrep
/usr/bin/bzfgrep
/usr/bin/bzgrep
/usr/bin/bzip2
/usr/bin/bzip2recover
/usr/bin/bzless
/usr/bin/bzmore
/usr/include/bzlib.h
/usr/lib/libbz2.a
```

Prereqs

The BZIP2 package does not have any prerequisites.

Examples

Example 6-12 and Example 6-13 show how to compress several files in a directory into a compressed archive file, /tmp/zippo.bz2. Note that we use the **tar** command to create the archive before compressing it.

Example 6-12 Using bzip2

```
root@fenris:/: tar cvf - .kshrc .profile .twmrc .xinitrc | bzip2 >
/tmp/zippo.bz2
a .kshrc 1 blocks.
a .profile 1 blocks.
a .twmrc 8 blocks.
a .xinitrc 8 blocks.

root@fenris:/: ls -l /tmp/zippo*
-rw-r--r--   1 root     system        3026 Oct 09 16:45 /tmp/zippo.bz2
```

Example 6-13 shows how to uncompress files from the compressed TAR archive file, /tmp/zippo.bz2.

Example 6-13 Using bunzip2

```
root@fenris:/: bunzip2 < /tmp/zippo.bz2 | tar xvf -
x .kshrc, 220 bytes, 1 media blocks.
x .profile, 23 bytes, 1 media blocks.
x .twmrc, 3774 bytes, 8 media blocks.
x .xinitrc, 3649 bytes, 8 media blocks.
```

6.3.5 zoo

The ZOO package (from the RPMS/ppc/zoo* directory) is a compression and file archiving utility for maintaining collections of files, and can store and selectively extract multiple generations of the same file. Data can be recovered from damaged archives by skipping the damaged portion and locating undamaged data. The **zoo** command uses Lempel-Ziv compression.

Synopsis

`zoo {command}[options] archive files(s)`

The following list describes options for the **zoo** command:

-add Add the specified files to the archive.

-extract	Extract the specified files from the archive. If no file is specified all files are extracted.
-move	Add the specified files to the archive and delete the source files after addition.
-test	Equivalent to **`-extract`** except that the extracted data is not saved, but any errors encountered are reported.
-print	Extract the specified files from the archive to standard output. If no file is specified all files are extracted.
-delete	Delete the specified files from the archive.
-list	Give information about the specified archived files, including any attached comments, if no files are specified all files are listed.
-update	Add a specified file to the archive if either an older file by the same name already exists in the archive or a file by the same name does not already exist in the archive.
-freshen	Add a specified file to the archive if and only if an older file by the same name already exists in the archive.
-comment	Allow the user to add or update comments attached to archived files.

```
zoo {acDeglLPTuUvx}[aAcCdEfInmMNoOpPqu1:/.@n] archive file
```

a	Add files.
e, x	Extract files.
l	List archive info.
L	List info for multiple archives.
v	List verbose file info and archive comment.
V	List verbose file info, archive, and file comments.
c	Comment changes, change or add comments to listed files (changes all file comments if no files given).
P	Pack archive, remove deleted or overwritten files.
D	Delete files by name.
T	Time stamp adjust, make archive age of newest file.
g	Generation commands.
f	Filter, copy stdin to stdout with (de)compression.

Toolbox FTP archive Web site

`ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/zoo`

Tool developer Web site

`N/A`

Tool FTP archive Web site

`ftp://sunsite.unc.edu/pub/Linux/utils/compress`

Files

```
/opt/freeware/bin/fiz
/opt/freeware/bin/zoo
/opt/freeware/doc/zoo-2.10
/opt/freeware/doc/zoo-2.10/Copyright
/opt/freeware/man/man1/fiz.1
/opt/freeware/man/man1/zoo.1
/usr/bin/fiz
/usr/bin/zoo
```

Prereqs

The ZOO package does not have any prerequisites.

Examples

Example 6-14 shows how to compress several files in a directory into a compressed archive file, /tmp/zippo.zip.

Example 6-14 Using zoo to compress

```
root@fenris:/: zoo a /tmp/zippo.zip .kshrc .profile .twmrc .xinitrc
Zoo:  .kshrc   --  (17%) added
Zoo:  .profile --  ( 0%) added
Zoo:  .twmrc   --  (42%) added
Zoo:  .xinitrc --  (53%) added

root@fenris:/: ls -l /tmp/zippo*
-rw-r--r--   1 root     system        4525 Oct 09 16:27 /tmp/zippo.zip
```

Example 6-15 shows how to uncompress files from the compressed archive file, /tmp/zippo.zip.

Example 6-15 Using zoo to uncompress

```
root@fenris:/: zoo x /tmp/zippo.zip
Zoo:  .kshrc         -- extracted
Zoo:  .profile       -- extracted
Zoo:  .twmrc         -- extracted
Zoo:  .xinitrc       -- extracted
```

6.3.6 lsof

The LSOF package contains the **lsof** command, which lists open files open by processes running on a Linux/UNIX system (LSOF stands for LiSt Open Files).

Synopsis

```
lsof [-?abhlnNoOPRstUvVX] [-c c] [+|-d s] [+|-D D] [+|-f[cfgGn]] [-F [f]] [-g
[s]] [-i [i]] [+|-L [l]] [-m m] [+|-M] [-o [o]] [-p s] [+|-r [t]] [-S [t]] [-T
[t]] [-u s] [+|-w] [--] [names]
```

The following list describes options for the **lsof** command:

-? -h	These two equivalent options select a usage (help) output list.
-a	This option causes list selection options to be ANDed.
-b	This option causes **lsof** to avoid kernel functions that might block - lstat(), readlink(), and stat().
-c c	This option selects the listing of files for processes executing the command that begins with the characters of c. Multiple commands may be specified, using multiple -c options. They are joined in a single ORed set before participating in AND option selection.
-C	This option disables the reporting of any path name components from the kernel's name cache.
+d s	This option causes lsof to search for all open instances of directory s and the files and directories it contains at its top level.
-d s	This option specifies a list of file descriptors (FDs) to exclude from or include in the output listing.
+D D	This option causes **lsof** to search for all open instances of directory D and all the files and directories it contains to its complete depth.
-D D	This option directs **lsof**'s use of the device cache file.
+l-f [cfgGn]	f by itself clarifies how path name arguments are to be interpreted. When followed by c, f, g, G, or n in any combination it specifies that the listing of kernel file structure information is to be enabled (+) or inhibited (-)
-F f	This option specifies a character list, f, that selects the fields to be output for processing by another program, and the character that terminates each output field. Each field to be output is specified with a single character in f.
-g [s]	This option selects the listing of files for the processes whose optional process group identification (PGID) numbers are in the comma separated set.
-i [i]	This option selects the listing of files, any of whose Internet address matches the address specified in i.
-l	This option inhibits the conversion of user ID numbers to login names. It is also useful when login name lookup is working improperly or slowly.

+|-L [l]	This option enables (+) or disables (-) the listing of file link counts, where they are available.
-m m	This option specifies a kernel memory file, c, in place of /dev/kmem or /dev/mem.
+|-M	Enables (+) or disables (-) the reporting of portmapper registrations for local TCP and UDP ports.
-n	This option inhibits the conversion of network numbers to host names for network files.
-N	This option selects the listing of NFS files.
-o	This option directs `lsof` to display file offset at all times. It causes the SIZE/OFF output column title to be changed to OFFSET. The -o and -s options are mutually exclusive.
-o o	This option defines the number of decimal digits (o) to be printed after the 0t for a file offset before the form is switched to 0x....
-O	This option directs `lsof` to bypass the strategy it uses to avoid being blocked by some kernel operations.
-p s	This option selects the listing of files for the processes whose ID numbers are in the comma-separated set s. Multiple process ID numbers are joined in a single ORed set before participating in AND option selection.
-P	This option inhibits the conversion of port numbers to port names for network files.
+|-r [t]	This option puts `lsof` in repeat mode. There `lsof` lists open files as selected by other options, delays t seconds (default fifteen), then repeats the listing, delaying and listing repetitively until stopped by a condition defined by the prefix to the option.
-R	This option directs `lsof` to list the parent process identification number in the PPID column.
-s	This option directs `lsof` to display file size at all times. It causes the SIZE/OFF output column title to be changed to SIZE. If the file does not have a size, nothing is displayed.
-S [t]	This option specifies an optional time-out seconds value for kernel functions - lstat(), readlink(), and stat(), which might otherwise deadlock. The minimum for t is two; the default, fifteen; when no value is specified, the default is used.
-T [t]	This option controls the reporting of some TCP/TPI information, also reported by `netstat`, following the network addresses. In normal output the information appears in parentheses, each item except state identified by a keyword, followed by =, separated from others by a single space.

-t	This option specifies that **lsof** should produce terse output with process identifiers only and no header (for example, so that the output may be piped to **kill**). This option selects the -w option.
-u s	This option selects the listing of files for the user whose login names or user ID numbers are in the comma-separated set s.
-U	This option selects the listing of UNIX domain socket files.
-v	This option selects the listing of **lsof** version information.
-V	This option directs **lsof** to indicate the items it was asked to list and failed to find (command names, file names, Internet addresses or files, login names, NFS files, PIDs, PGIDs, and UIDs).
+|-w	Enables (+) or disables (-) the suppression of warning messages.
-X	This is a dialect-specific option.

Note: The use of the -X option on a busy AIX system might cause an application process to hang in kernel mode.

The AIX -X option directs **lsof** to use the kernel **readx**() function, by default use of **readx**() is disabled. On AIX 5L and above **lsof** may need **setuid**-root permission to perform the actions this option requests. The **lsof** builder may specify that the -X option be restricted to processes whose real UID is root. If that has been done, the -X option will not appear in the -h or -? help output unless the real UID of the **lsof** process is root. The default **lsof** distribution allows any UID to specify -X, so by default it will appear in the help output. When AIX **readx**() use is disabled, **lsof** may not be able to report information for all text and loader file references. When **readx**() is enabled, **lsof** will attempt to report information on the text file being executed by each process and the shared libraries it uses.

Toolbox FTP archive Web site

ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/lsof

Tool developer Web site

ftp://vic.cc.purdue.edu/pub/tools/unix/lsof/README

Tool FTP archive Web site

ftp://vic.cc.purdue.edu/pub/tools/unix/lsof

Files

/opt/freeware/doc/lsof-4.61

```
/opt/freeware/doc/lsof-4.61/00.README.FIRST
/opt/freeware/doc/lsof-4.61/00CREDITS
/opt/freeware/doc/lsof-4.61/00DCACHE
/opt/freeware/doc/lsof-4.61/00DIALECTS
/opt/freeware/doc/lsof-4.61/00DIST
/opt/freeware/doc/lsof-4.61/00FAQ
/opt/freeware/doc/lsof-4.61/00LSOF-L
/opt/freeware/doc/lsof-4.61/00MANIFEST
/opt/freeware/doc/lsof-4.61/00PORTING
/opt/freeware/doc/lsof-4.61/00QUICKSTART
/opt/freeware/doc/lsof-4.61/00README
/opt/freeware/doc/lsof-4.61/00XCONFIG
/opt/freeware/man/man8/lsof.8
/opt/freeware/sbin/lsof
/opt/freeware/sbin/lsof64
/usr/man/man8/lsof.8
/usr/sbin/lsof
/usr/sbin/lsof64
```

Prereqs

The LSOF package does not have any prerequisites.

Examples

The **lsof** command can help identify processes that use any type of file that is used in AIX. The first example shows how to use **lsof** to list all open Internet (-i) and UNIX (-U) domain files (sockets):

```
lsof -i -U
```

The **lsof** command can also report on all processes that have files open in file systems. This can be especially important if you have a file sytem that grows, but the files in the file system do not grow and no process using any file can be found using the **fuser** command (base AIX command). The file name could have been deleted, but since the running process is still using the file, the *i-node* for this file is not removed from the kernel file table.

Example 6-16 on page 202 illustrates how this could be done and how the **lsof** command can find out which process is holding a file, without a name, open. To illustrate our point, we use the **cat** command to create a file, with no input, in the background. Then we remove the file name in the foreground with the **rm** command.

Normally the file's *inode* would have been removed if no other process was referencing this file. Nor is there any link to the *inode* from any file name anywhere else on the system. In our case the **cat** command is still executing but there is no file name in the directory nor any link to the inode.

Even though the **fuser -d** command is supposed to find files that have been deleted but are still in use, it does not provide the desired functionality. However, the **lsof** command still finds the process that has open files in the file system. The **kill %%** command terminates the cat process (since it is the only process running in our shell background), and the final **lsof** command does not report anything, as it should.

Example 6-16 Examining open files in a file system with lsof

```
root@fenris:/: cat > /waste_of_space/NAUGHTYNAUGHTYNAUGHTY &
root@fenris:/: rm /waste_of_space/NAUGHTYNAUGHTYNAUGHTY
root@fenris:/: fuser -d /waste_of_space
/waste_of_space:
root@fenris:/: lsof /waste_of_space
COMMAND   PID USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
cat     21616 root    1w  VREG  10,10        0   17 /waste_of_space (/dev/lv00)
root@fenris:/: kill %%
root@fenris:/: lsof /waste_of_space
```

6.3.7 less

The LESS packages provides the **less** command, which is a pager program that displays text files. Other pagers commonly in use are **more** and **pg**. Pagers are often used in command-line environments like the UNIX shell. The **less** command allows you to move backwards in the file as well as forwards (much like the **more** pager).

Synopsis

```
less [-[+]aBcCdeEfFgGiIJmMnNqQrRsSuUVwWX] [-?|--help]
[-b space] [-h lines] [-j line] [-k keyfile]
[-{oO} logfile] [-p pattern] [-P prompt] [-t tag]
[-T tagsfile] [-x tab,...] [-y lines] [-[z] lines]
[+[+]cmd] [--] [filename]...
```

The following list describes options for the **less** command:

-?, --help	Display help (from command line).
-a, --search-skip-screen	Forward search skips current screen.
-b [N], --buffers=[N]	Number of buffers.
-B, --auto-buffers	Do not automatically allocate buffers for pipes.
-c, -C	Repaint by scrolling/clearing.
-d, --dumb	Dumb terminal.
-e, -E	Quit at end of file.
-f, --force	Force open non-regular files.
-F, --quit-if-one-screen	Quit if entire file fits on first screen.

-g, --hilite-search	Highlight only last match for searches.
-G, --HILITE-SEARCH	Do not highlight any matches for searches.
-h [N], --max-back-scroll=[N]	Backward scroll limit.
-i, --ignore-case	Ignore case in searches that do not contain uppercase.
-I, --IGNORE-CASE	Ignore case in all searches.
-j [N], --jump-target=[N]	Screen position of target lines.
-J, --status-column	Display a status column at left edge of screen.
-k [file], --lesskey-file=[file]	Use a lesskey file.
-m, -M, --long-prompt	Set prompt style.
-n, -N, --line-number	Use line numbers.
-o [file], --log-file=[file]	Copy to log file (standard input only).
-O [file], --LOG-FILE=[file]	Copy to log file (unconditionally overwrite).
-p [pattern], --pattern=[pattern]	Start at pattern (from command line).
-P [prompt], --prompt=[prompt]	Define new prompt.
-q, -Q, --quiet, --silent	Quiet the terminal bell.
-r, -R, --raw-control-chars	Output "raw" control characters.
-s, --squeeze-blank-lines	Squeeze multiple blank lines.
-S, --chop-long-lines	Chop long lines.
-t [tag], --tag=[tag]	Find a tag.
-T [tagsfile], --tag-file=[tagsfile]	Use an alternate tags file.
-u, -U, --underline-special	Change handling of backspaces.
-V, --version	Display the version number of `less`.
-w, --hilite-unread	Highlight first new line after forward-screen.
-W, --HILITE-UNREAD	Highlight first new line after any forward movement.
-x [N[,...]], --tabs=[N[,...]]	Set tab stops.
-X, --no-init	Do not use termcap init/deinit strings.
--no-keypad	Do not use termcap keypad. init/deinit strings.
-y [N], --max-forw-scroll=[N]	Forward scroll limit.
-z [N], --window=[N]	Set size of window.
-" [c[c]], --quotes=[c[c]]	Set shell quote characters.
-~, --tilde	Do not display tildes after end of file.
-# [N], --shift=[N]	Horizontal scroll amount (0 = one half screen width).

Toolbox FTP archive Web site

ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/less

Tool developer Web site

http://www.greenwoodsoftware.com/less

Tool FTP archive Web site

http://www.greenwoodsoftware.com/less/less*version*-tar.gz

Files

```
/opt/freeware/bin/less
/opt/freeware/bin/lessecho
/opt/freeware/bin/lesskey
/opt/freeware/man/man1/less.1
/opt/freeware/man/man1/lesskey.1
/usr/bin/less
/usr/bin/lessecho
/usr/bin/lesskey
```

Prereqs

The LESS package does not have any prerequisites.

Examples

To start the **less** pager, suffix the file name of the file you wish to view after the **less** command itself:

```
less filename
```

6.3.8 findutils

The FINDUTILS package contains programs that will help you locate files on your system. The **find** utility searches through a hierarchy of directories looking for files that match a certain set of criteria (such as a file name pattern). The **xargs** utility builds and executes command lines from standard input arguments (usually lists of file names generated by the **find** command).

Synopsis

```
find [path...] [expression]

xargs [-0prtx] [-e[eof-str]] [-i[replace-str]] [-l[max-lines]]

locate [-d path] [--database=path] [--version] [--help] pattern...

updatedb [--localpaths='dir1 dir2...'] [--netpaths='dir1 dir2...']
[--prunepaths='dir1 dir2...'] [--output=dbfile] [--netuser=user] [--old-format]
[--version] [--help]
```

Toolbox FTP archive Web site

ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/findutils

Tool developer Web site

http://www.gnu.org/software/findutils/findutils.html

Tool FTP archive Web site

ftp://ftp.gnu.org/pub/gnu/findutils

Files

```
/opt/freeware/bin/find
/opt/freeware/bin/locate
/opt/freeware/bin/updatedb
/opt/freeware/bin/xargs
/opt/freeware/doc/findutils-4.1
/opt/freeware/doc/findutils-4.1/NEWS
/opt/freeware/doc/findutils-4.1/README
/opt/freeware/info/find.info-1.gz
/opt/freeware/info/find.info-2.gz
/opt/freeware/info/find.info.gz
/opt/freeware/libexec/bigram
/opt/freeware/libexec/code
/opt/freeware/libexec/frcode
/opt/freeware/man/man1/find.1
/opt/freeware/man/man1/xargs.1
/usr/bin/locate
/usr/bin/updatedb
/usr/linux/bin/find
/usr/linux/bin/locate
/usr/linux/bin/updatedb
/usr/linux/bin/xargs
```

Prereqs

The FINDUTILS package does not have any prerequisites.

Examples

The following example illustrates some of the many different options that the GNU **find** command has compared to the AIX version of the same command. The **-type f** command is the same as for the AIX **find** command and specifies that only files should be examined. The -daystart specifies that checking should start from midnight (when the current day started) and -empty specifies that we want to find files with *zero* size. Finall,y we want find to display detailed information about each file with the same -ls options that can be found for the AIX **find** command:

```
/usr/linux/bin/find / -type f -daystart -empty -ls
```

6.3.9 fileutils

The FILEUTILS package includes a number of GNU versions of the following file management commands:

chgrp	Changes a file's group ownership.
chown	Changes a file's ownership.
chmod	Changes a file's permissions.
cp	Copies files), dd (copies and converts files.
df	Shows a file system's disk usage.
dir	Gives a brief directory listing.
dircolors	The setup program for the color version of the ls command.
du	Shows disk usage.
install	Copies files and sets permissions.
ln	Creates file links.
ls	Lists directory contents.
mkdir	Creates directories.
mkfifo	Creates FIFOs or named pipes.
mknod	Creates special files.
mv	Renames files.
rm	Removes/deletes files.
rmdir	Removes empty directories.
sync	Synchronizes memory and disk.
touch	Changes file time stamps.
vdir	Provides long directory listings.

Toolbox FTP archive Web site

ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/fileutils

Tool developer Web site

http://www.gnu.org/software/fileutils/fileutils.html

Tool FTP archive Web site

ftp://ftp.gnu.org/pub/gnu/fileutils

Files

```
/opt/freeware/bin/chgrp
/opt/freeware/bin/chmod
/opt/freeware/bin/chown
/opt/freeware/bin/cp
/opt/freeware/bin/dd
/opt/freeware/bin/df
/opt/freeware/bin/dir
/opt/freeware/bin/dircolors
```

```
/opt/freeware/bin/du
/opt/freeware/bin/install
/opt/freeware/bin/ln
/opt/freeware/bin/ls
/opt/freeware/bin/mkdir
/opt/freeware/bin/mkfifo
/opt/freeware/bin/mknod
/opt/freeware/bin/mv
/opt/freeware/bin/rm
/opt/freeware/bin/rmdir
/opt/freeware/bin/shred
/opt/freeware/bin/sync
/opt/freeware/bin/touch
/opt/freeware/bin/vdir
/opt/freeware/doc/fileutils-4.1/
...(files omitted)...
/opt/freeware/man/man1/
...(files omitted)...
/opt/freeware/share/locale/cs/
...(files omitted)...
/usr/bin/dir
/usr/bin/dircolors
/usr/bin/vdir
/usr/linux/bin/chgrp
/usr/linux/bin/chmod
/usr/linux/bin/chown
/usr/linux/bin/cp
/usr/linux/bin/dd
/usr/linux/bin/df
/usr/linux/bin/du
/usr/linux/bin/install
/usr/linux/bin/ln
/usr/linux/bin/ls
/usr/linux/bin/mkdir
/usr/linux/bin/mkfifo
/usr/linux/bin/mknod
/usr/linux/bin/mv
/usr/linux/bin/rm
/usr/linux/bin/rmdir
/usr/linux/bin/shred
/usr/linux/bin/sync
/usr/linux/bin/touch
```

Prereqs

The FILEUTILS package does not have any prerequisites.

6.3.10 diffutils

The DIFFUTILS package includes four utilities: **`diff`**, **`cmp`**, **`diff3`**, and **`sdiff`**. The **`diff`** command compares two files and shows the differences, line by line. The **`cmp`** command shows the offset and line numbers where two files differ, or **`cmp`** can show the characters that differ between the two files. The **`diff3`** command shows the differences between three files, and can be used when two people have made independent changes to a common original; **`diff3`** can produce a merged file that contains both sets of changes and warnings about conflicts. The **`sdiff`** command can be used to merge two files interactively.

Synopsis

```
diff [OPTION]... FILE1 FILE2
```

The following list describes options for the **`diffutils`** command:

-i --ignore-case	Consider upper- and lowercase to be the same.
-w --ignore-all-space	Ignore all white space.
-b --ignore-space-change	Ignore changes in the amount of white space.
-B --ignore-blank-lines	Ignore changes whose lines are all blank.
-I RE --ignore-matching-lines=RE	Ignore changes whose lines all match RE.
-a --text	Treat all files as text
-c -C NUM --context[=NUM]	Output `NUM` (default 2) lines of copied context.
-u -U NUM --unified[=NUM]	Output NUM (default 2) lines of unified context.
-NUM	Use NUM context lines.
-L LABEL --label LABEL	Use LABEL instead of file name.
-p --show-c-function	Show which C function each change is in.
-F RE --show-function-line=RE	Show the most recent line matching RE.
-q --brief	Output only whether files differ.
-e --ed	Output an ed script.
-n --rcs	Output an RCS format **`diff`**.
-y --side-by-side	Output in two columns.
-w NUM --width=NUM	Output at most NUM (default 130) characters per line.
--left-column	Output only the left column of common lines.
--suppress-common-lines	Do not output common lines.
-DNAME --ifdef=NAME	Output merged file to show #ifdef NAME diffs.
--GTYPE-group-format=GFMT	Similar, but format GTYPE input groups with GFMT.
--line-format=LFMT	Similar, but format all input lines with LFMT.

--LTYPE-line-format=LFMT	Similar, but format LTYPE input lines with LFMT.
-l --paginate	Pass the output through `pr` to paginate it.
-t --expand-tabs	Expand tabs to spaces in output.
-T --initial-tab	Make tabs line up by prepending a tab.
-r --recursive	Recursively compare any subdirectories found.
-N --new-file	Treat absent files as empty.
-P --unidirectional-new-file	Treat absent first files as empty.
-s --report-identical-files	Report when two files are the same.
-x PAT --exclude=PAT	Exclude files that match PAT.
-X FILE --exclude-from=FILE	Exclude files that match any pattern in FILE.
-S FILE --starting-file=FILE	Start with FILE when comparing directories.
--horizon-lines=NUM	Keep NUM lines of the common prefix and suffix.
-d --minimal	Try hard to find a smaller set of changes.
-H --speed-large-files	Assume large files and many scattered small changes.
-v --version	Output version info.
--help	Output this help.

```
sdiff [OPTIONS]... FILE1 FILE2
```

-o FILE --output=FILE	Operate interactively, sending output to FILE
-i --ignore-case	Consider upper- and lowercase to be the same.
-W --ignore-all-space	Ignore all white space.
-b --ignore-space-change	Ignore changes in the amount of white space.
-B --ignore-blank-lines	Ignore changes whose lines are all blank.
-I RE --ignore-matching-lines=RE	Ignore changes whose lines all match RE.
-a --text	Treat all files as text.
-w NUM --width=NUM	Output at most NUM (default 130) characters per line.
-l --left-column	Output only the left column of common lines.
-s --suppress-common-lines	Do not output common lines.
-t --expand-tabs	Expand tabs to spaces in output.
-d --minimal	Try hard to find a smaller set of changes.
-H --speed-large-files	Assume large files and many scattered small changes.
-v --version	Output version info.
--help	Output this help.

```
diff3 [OPTION]... MYFILE OLDFILE YOURFILE
```

-e --ed	Output unmerged changes from OLDFILE to YOURFILE into MYFILE.
-E --show-overlap	Output unmerged changes, bracketing conflicts.
-A --show-all	Output all changes, bracketing conflicts.
-x --overlap-only	Output overlapping changes.
-X	Output overlapping changes, bracketing them.
-3 --easy-only	Output unmerged nonoverlapping changes.
-m --merge	Output merged file instead of ed script (default -A).
-L LABEL --label=LABEL	Use LABEL instead of file name.
-i	Append `w` and `q` commands to ed scripts.
-a --text	Treat all files as text.
-T --initial-tab	Make tabs line up by prepending a tab.
-v --version	Output version info.
--help	Output this help.

```
cmp [OPTION]... FILE1 [FILE2]
```

-c --print-chars	Output differing bytes as characters.
-i N --ignore-initial=N	Ignore differences in the first *N* bytes of input.
-l --verbose	Output offsets and codes of all differing bytes.
-s --quiet --silent	Output nothing; yield exit status only.
-v --version	Output version info.
--help	Output this help.

Toolbox FTP archive Web site

ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/diffutils

Tool developer Web site

http://www.gnu.org/software/diffutils/diffutils.html

Tool FTP archive Web site

ftp://ftp.gnu.org/pub/gnu/diffutils

Files

```
/opt/freeware/bin/cmp
/opt/freeware/bin/diff
/opt/freeware/bin/diff3
/opt/freeware/bin/sdiff
/opt/freeware/doc/diffutils-2.7
```

```
/opt/freeware/doc/diffutils-2.7/NEWS
/opt/freeware/doc/diffutils-2.7/README
/opt/freeware/info/diff.info*.gz
```
...(files omitted)...
```
/usr/linux/bin/cmp
/usr/linux/bin/diff
/usr/linux/bin/diff3
/usr/linux/bin/sdiff
```

Prereqs

The DIFFUTILS package does not have any prerequisites.

6.3.11 hexedit

The HEXEDIT package provides the **hexedit** command, which is a utility that allows you to view and edit hexadecimal or ASCII files and/or view binary files.

Synopsis

```
hexedit [-s | --sector] [-m | --maximize] [-h | --help] filename
```

The following list describes options for the **hexedit** command:

-s, --sector	Format the display to have entire sectors.
-m, --maximize	Try to maximize the display.
-h, --help	Show the usage.

Toolbox FTP archive Web site

ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/hexedit

Tool developer Web site

http://merd.net/pixel/hexedit.html

Tool FTP archive Web site

http://merd.net/pixel/hexedit.html

Files

```
/opt/freeware/bin/hexedit
/opt/freeware/doc/hexedit-1.2.2
/opt/freeware/doc/hexedit-1.2.2/COPYING
/opt/freeware/doc/hexedit-1.2.2/Changes
/opt/freeware/doc/hexedit-1.2.2/TODO
/opt/freeware/man/man1/hexedit.1
/usr/bin/hexedit
```

Prereqs

The HEXEDIT package does not have any prerequisites.

Examples

Start the **hexedit** editor by typing `hexedit` on the command line:

```
hexedit
```

The following figure shows the **hexedit** full-screen display in a telnet.exe window log into the AIX system from a PC client. To move within the window you can use the arrow keys, and when you type hexadecimal values in the HEX part of the window, you will see the equivalent ASCII representation in the ASCII part. To jump between ASCII mode editing and HEX editing, use the Tab key. To quit the program press CTRL+C, to get help press the F1 key.

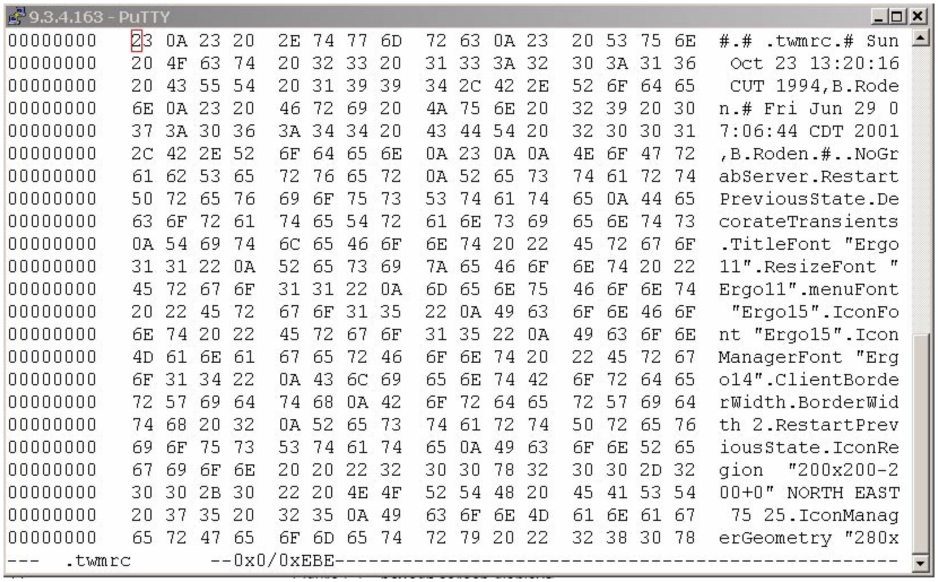

Figure 6-5 hexedit screen display

6.3.12 sh-utils

The SH-UTILS package includes a number of GNU versions of the following shell utility commands:

basename	To remove the path prefix from a specified path name.
chroot	To change the root directory.
date	To print/set the system time and date.

dirname	To remove the last level or the file name from a given path.
echo	To print a line of text.
env	To display/modify the environment.
expr	To evaluate expressions.
factor	To print prime factors.
false	To return an unsuccessful exit status.
groups	To print the groups a specified user is a member of.
id	To print the real/effective UID/GID.
logname	To print the current login name.
nice	To modify a scheduling priority.
nohup	To allow a command to continue running after logging out.
pathchk	To check a file name's portability.
printenv	To print environment variables.
printf	To format and print data.
pwd	To print the current directory.
seq	To print numeric sequences.
sleep	To suspend execution for a specified time.
stty	To print/change terminal settings.
su	To become another user or the superuser.
tee	To send output to multiple files.
test	To evaluate an expression.
true	To return a successful exit status.
tty	To print the terminal name.
uname	To print system information.
users	To print current users' names.
who	To print a list of the users who are currently logged in.
whoami	To print the effective user ID.
yes	To print a string indefinitely.

Toolbox FTP archive Web site

ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/sh-utils

Tool developer Web site

http://www.gnu.org/software/shellutils

Tool FTP archive Web site

ftp://ftp.gnu.org/pub/gnu/sh-utils

Files

```
/opt/freeware/bin/[
/opt/freeware/bin/basename
/opt/freeware/bin/chroot
/opt/freeware/bin/date
```

```
/opt/freeware/bin/dirname
/opt/freeware/bin/echo
/opt/freeware/bin/env
/opt/freeware/bin/expr
/opt/freeware/bin/factor
/opt/freeware/bin/false
/opt/freeware/bin/groups
/opt/freeware/bin/hostid
/opt/freeware/bin/hostname
/opt/freeware/bin/id
/opt/freeware/bin/logname
/opt/freeware/bin/nice
/opt/freeware/bin/nohup
/opt/freeware/bin/pathchk
/opt/freeware/bin/pinky
/opt/freeware/bin/printenv
/opt/freeware/bin/printf
/opt/freeware/bin/pwd
/opt/freeware/bin/seq
/opt/freeware/bin/sleep
/opt/freeware/bin/stty
/opt/freeware/bin/su
/opt/freeware/bin/tee
/opt/freeware/bin/test
/opt/freeware/bin/true
/opt/freeware/bin/tty
/opt/freeware/bin/uname
/opt/freeware/bin/uptime
/opt/freeware/bin/users
/opt/freeware/bin/who
/opt/freeware/bin/whoami
/opt/freeware/bin/yes
/opt/freeware/doc/sh-utils-2.0
/opt/freeware/doc/sh-utils-2.0/NEWS
/opt/freeware/doc/sh-utils-2.0/README
/opt/freeware/info/sh-utils.info.gz
/opt/freeware/man/man1/
...(files omitted)...
/opt/freeware/share/locale/
...(files omitted)...
/usr/bin/pinky
/usr/bin/seq
/usr/linux/bin/[
/usr/linux/bin/basename
/usr/linux/bin/chroot
/usr/linux/bin/date
/usr/linux/bin/dirname
/usr/linux/bin/echo
```

```
/usr/linux/bin/env
/usr/linux/bin/expr
/usr/linux/bin/factor
/usr/linux/bin/false
/usr/linux/bin/groups
/usr/linux/bin/hostid
/usr/linux/bin/hostname
/usr/linux/bin/id
/usr/linux/bin/logname
/usr/linux/bin/nice
/usr/linux/bin/nohup
/usr/linux/bin/pathchk
/usr/linux/bin/pinky
/usr/linux/bin/printenv
/usr/linux/bin/printf
/usr/linux/bin/pwd
/usr/linux/bin/seq
/usr/linux/bin/sleep
/usr/linux/bin/stty
/usr/linux/bin/su
/usr/linux/bin/tee
/usr/linux/bin/test
/usr/linux/bin/true
/usr/linux/bin/tty
/usr/linux/bin/uname
/usr/linux/bin/uptime
/usr/linux/bin/users
/usr/linux/bin/who
/usr/linux/bin/whoami
/usr/linux/bin/yes
```

Prereqs

The SH-UTILS package does not have any prerequisites.

6.4 Development tools

The development packages described here are a sample of different tools from the Toolbox that are mainly used for software development purposes. We cover the following tools in this section:

- **libtool**
- **autoconf** and **automake**
- **rcs**
- **patch**
- **gcc**
- **gdb**

- **binutils**
- **gmake**
- **m4**
- **cvs**
- **info**

6.4.1 libtool

The LIBTOOL package contains the GNU **libtool** command, a set of shell scripts that can be used to configure and build shared libraries. The **libtool** command provides a consistent, portable command interface that simplifies the process of using shared libraries.

Synopsis

```
libtool [option]... [mode-arg]...
```

The following list describes options for the **libtool** command:

--config Display **libtool** configuration variables and exit.

--debug Dump a trace of shell script execution to standard output. This produces a lot of output, so you may wish to pipe it to less (or more) or redirect to a file.

-n, --dry-run Do not create, modify, or delete any files, just show what commands would be executed by **libtool**.

--features Display basic configuration options. This provides a way for packages to determine whether shared or static libraries will be built.

--finish Same as --mode=finish.

--help Display a help message and exit. If --mode=mode is specified, then detailed help for mode is displayed.

--mode=mode Use mode as the operation mode. By default, the operation mode is inferred from the mode-args. If mode is specified, it must be one of the following:

- **compile** Compile a source file into a **libtool** object.
- **execute** Automatically set the library path so that another program can use uninstalled **libtool**-generated programs or libraries.
- **finish** Complete the installation of **libtool** libraries on the system.
- **install** Install libraries or executables.
- **link** Create a library or an executable.
- **uninstall** Delete installed libraries or executables.
- **clean** Delete uninstalled libraries or executables.

--version Print **libtool** version information and exit.

The mode-args are a variable number of arguments, depending on the selected operation mode. In general, each mode-arg is interpreted by programs **libtool** invokes, rather than **libtool** itself.

Toolbox FTP archive Web site

ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/libtool

Tool developer Web site

http://www.gnu.org/software/libtool/libtool.html

Tool FTP archive Web site

ftp://ftp.gnu.org/gnu/libtool

Files

```
/opt/freeware/bin/libtool
/opt/freeware/bin/libtoolize
/opt/freeware/doc/libtool-1.4.2
/opt/freeware/doc/libtool-1.4.2/AUTHORS
/opt/freeware/doc/libtool-1.4.2/COPYING
/opt/freeware/doc/libtool-1.4.2/ChangeLog
/opt/freeware/doc/libtool-1.4.2/INSTALL
/opt/freeware/doc/libtool-1.4.2/NEWS
/opt/freeware/doc/libtool-1.4.2/README
/opt/freeware/doc/libtool-1.4.2/THANKS
/opt/freeware/doc/libtool-1.4.2/TODO
/opt/freeware/doc/libtool-1.4.2/demo/
...(files omitted)...
/opt/freeware/include/ltdl.h
/opt/freeware/info/libtool.info*.gz
...(files omitted)...
/opt/freeware/lib/libltdl.a
/opt/freeware/lib/libltdl.la
/opt/freeware/share/aclocal/libtool.m4
/opt/freeware/share/libtool
/opt/freeware/share/libtool/config.guess
/opt/freeware/share/libtool/config.sub
/opt/freeware/share/libtool/libltdl/
...(files omitted)...
/opt/freeware/share/libtool/ltmain.sh
/usr/bin/libtool
/usr/bin/libtoolize
/usr/include/ltdl.h
/usr/lib/libltdl.a
/usr/lib/libltdl.la
```

Prereqs

The LIBTOOL package does not have any prerequisites.

6.4.2 autoconf/automake

The AUTOCONF and AUTOMAKE packages contain programs that will help you adapt configuration of your software source code packages to many kinds of UNIX-like systems, without manual user intervention.

The **`autoconf`** command creates a configuration script for a package from a template file that lists the operating system features that the package can use, in the form of m4 macro calls.

The **`automake`** command is a tool for automatically generating makefile.in files. A Makefile.am is basically a series of make macro definitions and rules. The generated Makefile.in files are compliant with the GNU Makefile standards and can be interpreted by the **`gmake`**, and in most cases, the **`make`** command.

For more information on porting software, please refer to Chapter 3, “Porting Open Source Software to AIX” on page 57.

Synopsis

```
autoconf [OPTION] ... [TEMPLATE-FILE]
```

The following list describes options for the **`autoconf`** command:

-h, --help	Print help.
-V, --version	Print version number.
-v, --verbose	Verbosely report processing.
-d, --debug	Do not remove temporary files.
-I, --include=DIR	Look for input files in `DIR` (cumulative).
-f, --force	Consider all files obsolete.
-o, --output=FILE	Save output in `FILE` (stdout is the default).
-W, --warnings=CATEGORY	Report the warnings falling in CATEGORY.

```
automake [OPTION] ... [Makefile]
```

The following list describes options for the **`automake`** command:

--help	Print help.
--version	Print version number, then exit.
-v, --verbose	Verbosely list files processed.
-o, --output-dir=DIR	Put generated Makefile.in into DIR.
--no-force	Only update Makefile.in that are out of date.
-i, --ignore-deps	Disable dependency tracking code.
--include-deps	Enable dependency tracking code.

--cygnus	Assume program is part of Cygnus-style tree.
--foreign	Set strictness to foreign.
--gnits	Set strictness to Gnits.
--gnu	Set strictness to GNU.
-a, --add-missing	Add missing standard files to package.
--libdir=DIR	Directory storing library files.
-c, --copy	With -a, copy missing files (default is symbolic link).
-f, --force-missing	Force update of standard files.

Toolbox FTP archive Web site

`ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/autoconf`
`ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/automake`

Tool developer Web site

`http://www.gnu.org/software/autoconf/autoconf.html`
`http://www.gnu.org/software/automake/automake.html`

Tool FTP archive Web site

`http://ftp.gnu.org/gnu/autoconf`
`http://ftp.gnu.org/gnu/automake`

Files

```
/opt/freeware/bin/autoconf
/opt/freeware/bin/autoheader
/opt/freeware/bin/autom4te
/opt/freeware/bin/autoreconf
/opt/freeware/bin/autoscan
/opt/freeware/bin/autoupdate
/opt/freeware/bin/ifnames
/opt/freeware/info/autoconf.info.gz
/opt/freeware/share/autoconf
```

...(files omitted)...

```
/usr/bin/autoconf
/usr/bin/autoheader
/usr/bin/autom4te
/usr/bin/autoreconf
/usr/bin/autoscan
/usr/bin/autoupdate
/usr/bin/ifnames
/usr/share/autoconf

/opt/freeware/bin/aclocal
/opt/freeware/bin/automake
/opt/freeware/doc/automake-1.5/
```

...(files omitted)...

```
/opt/freeware/info/automake.info.gz
```
...(files omitted)...
```
/opt/freeware/share/automake/install-sh
/opt/freeware/share/automake/mdate-sh
/opt/freeware/share/automake/missing
/opt/freeware/share/automake/mkinstalldirs
/usr/bin/aclocal
/usr/bin/automake
/usr/share/automake
```

Prereqs

The AUTOCONF package requires the M4 package.

Examples

To use the GNU **`autoconf`** and **`automake`** utilities to generate a makefile that can be used by the **`make`** program to create executable programs of our source code, we need to go through the following steps:

1. Create source code and additional support files.
2. Run **`autoscan`**.
3. Modify configure.in.
4. Run **`autoheader`**.
5. Run **`autoconf`**.
6. Run **`aclocal`**.
7. Run **`automake`**.
8. Run **`configure`**.
9. Run **`make`**.

For our examples, we use a directory that contains our source code (ourcode-1.0). The directory has three files: ourcode.c, ourcode.h, and ourcode.8. The ourcode.c is the C language source file; the ourcode.h file is the C language header file that will be included at compile time into ourcode.c; and the ourcode.8 file is the tagged man file for the ourcode program. To create a nice package we also add the following files to our source directory: INSTALL, NEWS, README, COPYING, AUTHORS, and ChangeLog, with the intent to keeping them up to date.

First we use **`autoscan`** to check our source files and build a template configure.scan file, in the current directory, which we can use to create our configure.in file in next step.

Example 6-17 Using autoscan

```
root@fenris:/home/work/rpm/ourcode-1.0: autoscan -v
autoscan: srcdir = .
cfiles: ourcode.c ourcode.h
makefiles:
shfiles:

functions:
main: ourcode.c:3
printf: ourcode.c:5

headers:

identifiers:
argc: ourcode.c:3
argv: ourcode.c:3
version: ourcode.c:5 ourcode.h:2

programs:
cc: ourcode.c: ourcode.h:5

makevars:

libraries:
```

Example 6-18 shows the produced configure.scan file.

Example 6-18 autoscan created configure.scan file

```
# Process this file with autoconf to produce a configure script.
AC_INIT(FULL-PACKAGE-NAME, VERSION, BUG-REPORT-ADDRESS)
AC_CONFIG_SRCDIR([ourcode.c])
AC_CONFIG_HEADER([config.h])

# Checks for programs.
AC_PROG_CC

# Checks for libraries.

# Checks for header files.

# Checks for typedefs, structures, and compiler characteristics.

# Checks for library functions.

AC_CONFIG_FILES([])
AC_OUTPUT
```

We can copy, or link, the configure.scan file to a configure.in file:

```
cp configure.scan configure.in
```

After creating our basic configure.in file, we need to check it and modify it to suite our purposes. We replace the following line from the configure.scan file:

```
AC_INIT(FULL-PACKAGE-NAME, VERSION, BUG-REPORT-ADDRESS)
```

With the following for our purposes:

```
AC_INIT(ourcode, 1.0)
```

Example 6-19 Sample configure.in configuration file

```
# Process this file with autoconf to produce a configure script.
AC_INIT(ourcode, 1.0)
AC_CONFIG_SRCDIR([ourcode.c])
AC_CONFIG_HEADER([config.h])

# Checks for programs.
AC_PROG_CC

# Checks for libraries.

# Checks for header files.

# Checks for typedefs, structures, and compiler characteristics.

# Checks for library functions.

AC_CONFIG_FILES([])
AC_OUTPUT
```

Next we could run the **autoheader** command to create the config.in.h file.

Example 6-20 Using the autoheader command

```
root@fenris:/home/work/rpm/ourcode-1.0: autoheader
autoheader: `config.h.in' is created
```

Example 6-21 shows the created config.in.h file.

Example 6-21 Sample config.in.h configuration file

```
/* config.h.in.  Generated from configure.in by autoheader.  */

/* Define to the address where bug reports for this package should be sent. */
#undef PACKAGE_BUGREPORT

/* Define to the full name of this package. */
```

```
#undef PACKAGE_NAME

/* Define to the full name and version of this package. */
#undef PACKAGE_STRING

/* Define to the one symbol short name of this package. */
#undef PACKAGE_TARNAME

/* Define to the version of this package. */
#undef PACKAGE_VERSION
```

If you use conditional defines, you could use the **ifnames** command to list those defines from your source code.

```
ifnames filenames
```

In our example we have one define in the ourcode.c program:

```
#ifdef ourcode
        #define SWEET "ourcode"
#endif
```

Example 6-22 shows how to use the **ifnames** command on the ourcode.c file.

Example 6-22 Using ifnames

```
root@fenris:/home/work/rpm/ourcode-1.0: ifnames ourcode.c
ourcode ourcode.c
```

If we do not have a Makefile.in previously created by **automake**, we need to go through a couple more steps.

First we update configure.in and add AM_INIT_AUTOMAKE after the AC_INIT line. Then we create the aclocal.m4 file (since we did not have one before) with the **aclocal** command. The AM_INIT_AUTOMAKE line should be similar to the AC_INIT line:

```
AM_INIT_AUTOMAKE(ourcode,1.0)
```

Then we run the **autoconf** command:

```
autoconf
```

The **autoconf** command will create the configure script in the current directory (it also creates a **m4** cache directory, autom4te.cache). Now we create our Makefile.am file.

Example 6-23 Sample Makefile.am file

```
bin_PROGRAMS = ourcode
ourcode_SOURCES = ourcode.c ourcode.h
```

Now we can run **automake** with the --add-missing option to create missing files and links.

Example 6-24 Using automake

```
root@fenris:/home/work/rpm/ourcode-1.0: automake --add-missing
automake: configure.in: installing `./install-sh'
automake: configure.in: installing `./mkinstalldirs'
automake: configure.in: installing `./missing'
automake: configure.in: installing `./depcomp'
```

Now we are now ready to use the configure script.

Example 6-25 Using ./configure

```
root@fenris:/home/work/rpm/ourcode-1.0: ./configure
configure: loading cache /dev/null
checking for a BSD-compatible install... /usr/linux/bin/install -c
checking whether build environment is sane... yes
checking for gawk... gawk
checking whether make sets ${MAKE}... yes
checking for gcc... gcc
checking for C compiler default output... a.out
checking whether the C compiler works... yes
checking whether we are cross compiling... no
checking for suffix of executables...
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for style of include used by make... GNU
checking dependency style of gcc... gcc
configure: creating ./config.status
config.status: creating Makefile
config.status: creating config.h
config.status: config.h is unchanged
config.status: executing default-1 commands
```

Finally we can run the **make** program (or **gmake**) to create our executable. However, the first time we run **make** it does some “housekeeping”.

Example 6-26 Using make for the first time after automake

```
root@fenris:/home/work/rpm/ourcode-1.0: make
        cd . && /bin/sh /home/work/rpm/ourcode-1.0/missing --run aclocal
```

```
        cd . && /bin/sh /home/work/rpm/ourcode-1.0/missing --run autoheader
autoheader: `config.h.in' is unchanged
        cd . && /bin/sh /home/work/rpm/ourcode-1.0/missing --run autoconf
        /bin/sh ./config.status --recheck
running /bin/sh ./configure   --no-create --no-recursion
configure: loading cache /dev/null
checking for a BSD-compatible install... /usr/linux/bin/install -c
checking whether build environment is sane... yes
checking for gawk... gawk
checking whether make sets ${MAKE}... yes
checking for gcc... gcc
checking for C compiler default output... a.out
checking whether the C compiler works... yes
checking whether we are cross compiling... no
checking for suffix of executables...
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for style of include used by make... GNU
checking dependency style of gcc... gcc
configure: creating ./config.status
        cd .  && CONFIG_FILES= CONFIG_HEADERS=config.h  /bin/sh ./config.status
config.status: creating config.h
config.status: config.h is unchanged
config.status: executing default-1 commands
        make  all-am
        cd . &&  /bin/sh /home/work/rpm/ourcode-1.0/missing --run automake
--gnu  Makefile
configure.in: 6: `automake requires `AM_CONFIG_HEADER', not `AC_CONFIG_HEADER'
WARNING: `automake' is missing on your system.  You should only need it if
         you modified `Makefile.am', `acinclude.m4' or `configure.in'.
         You might want to install the `Automake' and `Perl' packages.
         Grab them from any GNU archive site.
        cd . &&  CONFIG_HEADERS= CONFIG_LINKS=  CONFIG_FILES=Makefile /bin/sh
./
config.status
config.status: creating Makefile
config.status: executing default-1 commands
        source='ourcode.c' object='ourcode.o' libtool=no
depfile='.deps/ourcode.Po' tmpd
epfile='.deps/ourcode.TPo'  depmode=gcc /bin/sh ./depcomp  gcc -DHAVE_CONFIG_H
-I.
-I. -I.      -g -O2 -c `test -f ourcode.c || echo './'`ourcode.c
        gcc  -g -O2   -o ourcode  ourcode.o
        cd .  && CONFIG_FILES= CONFIG_HEADERS=config.h  /bin/sh ./config.status
config.status: creating config.h
config.status: config.h is unchanged
config.status: executing default-1 commands
Target "all-am" is up to date.
```

```
root@fenris:/home/work/rpm/ourcode-1.0:
```

The next example shows how the normal **make** process is performed.

Example 6-27 Using make/gmake

```
root@fenris:/home/work/rpm/ourcode-1.0: make
        make  all-am
        source='ourcode.c' object='ourcode.o' libtool=no
depfile='.deps/ourcode.Po' tmpd
epfile='.deps/ourcode.TPo'  depmode=gcc /bin/sh ./depcomp  gcc -DHAVE_CONFIG_H
-I.
-I. -I.      -g -O2 -c `test -f ourcode.c || echo './'`ourcode.c
        gcc  -g -O2   -o ourcode  ourcode.o
Target "all-am" is up to date.
```

6.4.3 rcs

The RCS package (Revision Control System) is a system for managing multiple versions of files. The **rcs** command automates the storage, retrieval, logging, identification, and merging of file revisions. Revision control is useful for text files that are revised frequently (for example, programs, documentation, graphics, papers, and form letters).

Synopsis

```
rcs -{ae}logins -Afile -{blu}[rev] -cstring -{iILqTU} -ksubst -mrev:msg
-{nN}name[:[rev]] -orange -sstate[:rev] -t[text] -Vn -xsuff -zzone file ...
```

The following list describes options for the **rcs** command:

-i	Create and initialize a new RCS file, but do not deposit any revision.
-alogins	Append the login names appearing in the comma-separated list logins to the access list of the RCS file.
-Aoldfile	Append the access list of oldfile to the access list of the RCS file.
-e[logins]	Erase the login names appearing in the comma-separated list logins from the access list of the RCS file.
-b[rev]	Set the default branch to rev.
-cstring	Set the comment leader to string.
-ksubst	Set the default keyword substitution to subst.
-l[rev]	Lock the revision with number rev.
-u[rev]	Unlock the revision with number rev.
-L	Set locking to strict.

-U	Set locking to non-strict.
-mrev:msg	Replace revision rev's log message with msg.
-M	Do not send mail when breaking somebody else's lock.
-nname[:[rev]]	Associate the symbolic name with the branch or revision rev.
-Nname[:[rev]]	Act like -n, except override any previous assignment of name.
-orange	Deletes (outdates) the revisions given by range.
-q	Run quietly; do not print diagnostics.
-I	Run interactively, even if the standard input is not a terminal.
-sstate[:rev]	Set the state attribute of the revision rev to state.
-t[file]	Write descriptive text from the contents of the named file into the RCS file, deleting the existing text.
-t-string	Write descriptive text from the string into the RCS file, deleting the existing text.
-T	Preserve the modification time on the RCS file unless a revision is removed.
-V	Print RCS's version number.
-Vn	Emulate RCS version `n`.
-xsuffixes	Use suffixes to characterize RCS files.
-zzone	Use `zone` as the default time zone.

Toolbox FTP archive Web site

ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/rcs

Tool developer Web site

http://www.fsf.org/software/rcs/rcs.html

Tool FTP archive Web site

ftp://ftp.cs.purdue.edu/pub/RCS

Files

```
/opt/freeware/bin/ci
/opt/freeware/bin/co
/opt/freeware/bin/ident
/opt/freeware/bin/merge
/opt/freeware/bin/rcs
/opt/freeware/bin/rcsclean
/opt/freeware/bin/rcsdiff
/opt/freeware/bin/rcsmerge
/opt/freeware/bin/rlog
/opt/freeware/doc/rcs-5.7
/opt/freeware/doc/rcs-5.7/NEWS
```

```
/opt/freeware/doc/rcs-5.7/REFS
/opt/freeware/man/man1/
```
...(files omitted)...
```
/opt/freeware/man/man5/rcsfile.5
/usr/bin/ci
/usr/bin/co
/usr/bin/ident
/usr/bin/merge
/usr/bin/rcs
/usr/bin/rcsclean
/usr/bin/rcsdiff
/usr/bin/rcsmerge
/usr/bin/rlog
```

Prereqs

The RCS package requires the DIFFUTILS package to be installed.

Examples

To use the RCS version control system, you need some source code to control. The following examples are based on the ourcode.c program. Please refer to 6.4.5, “gcc” on page 233.

To use RCS version control for the ourcode.c program, we need to create RCS version control information for the ourcode.c file with the -i option of the **rcs** command. The version control information will be created by default in the current directory, or in the RCS directory in the current directory, if this directory exists.

First we create the RCS directory:

```
mkdir RCS
```

Then we create the RCS file for the ourcode.c program:

```
rcs -i ourcode.c
```

In the RCS directory, **rcs** has now created the ourcode.c,v file. In the next example we use the RCS **ci** command to check in the source code.

Example 6-28 Using RCS ci command

```
root@fenris:/home/work: ci ourcode.c
RCS/ourcode.c,v  <--  ourcode.c
initial revision: 1.1
done
```

After checking in the ourcode.c program, the **ci** command has removed it from the current directory. To edit the source code again, use the **co** command to check out the source for editing:

```
co -l ourcode.c
```

The **co** command has now created the latest version of the ourcode.c program in the current directory, locked for other users so we can edit it again. If we only wanted to compile the ourcode.c program, we would not have to lock it with the -l option, but then it would not have been created writable by the **co** command. After changing the source of the ourcode.c program, please refer to 6.4.6, “gdb” on page 236. We update the version control information with our new source code.

Example 6-29 Checking in ourcode.c again with the ci command

```
root@fenris:/home/work: ci ourcode.c
RCS/ourcode.c,v  <--  ourcode.c
new revision: 1.2; previous revision: 1.1
enter log message, terminated with single '.' or end of file:
>> Updated the printf statement //BR
>> .
done
```

To check the progress of the ourcode.c program according to its RCS information, we can use the **rlog** command. The **rlog** command shows that we have two revisions, and some information respectively about Version 1.1 and Version 1.2.

Example 6-30 Using the rlog command

```
root@fenris:/home/work: rlog ourcode.c

RCS file: RCS/ourcode.c,v
Working file: ourcode.c
head: 1.2
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
total revisions: 2;     selected revisions: 2
description:
----------------------------
revision 1.2
date: 2002/10/10 00:16:51;  author: root;  state: Exp;  lines: +1 -1
Updated the printf statement //BR
----------------------------
revision 1.1
```

```
date: 2002/10/10 00:02:15;  author: root;  state: Exp;
Initial revision
=============================================================================
```

6.4.4 patch

The PATCH package provides the **patch** command, which applies GNU **diff** files to originals. The **diff** command is used to compare an original to a changed file. The **diff** command lists the changes made to the file. A person who has the original file can then use the **patch** command with the **diff** file to add the changes to their original file, patching the file.

> **Note:** The AIX **diff** command does not have the -u option that is frequently used to produce **diff** files for the GNU **patch** command.

Synopsis

```
patch [OPTION]... [ORIGFILE [PATCHFILE]]
```

The following list describes options for the **patch** command:

-p NUM --strip=NUM	Strip NUM leading components from file names.
-F LINES --fuzz LINES	Set the fuzz factor to LINES for inexact matching.
-l --ignore-whitespace	Ignore white space changes between patch and input.
-c --context	Interpret the patch as a context difference.
-e --ed	Interpret the patch as an ed script.
-n --normal	Interpret the patch as a normal difference.
-u --unified	Interpret the patch as a unified difference.
-N --forward	Ignore patches that appear to be reversed or already applied.
-R --reverse	Assume patches were created with old and new files swapped.
-i PATCHFILE **--input=PATCHFILE**	Read patch from PATCHFILE instead of stdin.
-o FILE --output=FILE	Output patched files to FILE.
-r FILE --reject-file=FILE	Output rejects to FILE.
-D NAME --ifdef=NAME	Make merged if-then-else output using NAME.
-E --remove-empty-files	Remove output files that are empty after patching.
-Z --set-utc	Set times of patched files, assuming **diff** uses UTC (GMT).
-T --set-time	Likewise, assuming local time.

--quoting-style=WORD	Output file names using quoting style WORD. Valid WORDs are: literal, shell, shell-always, c, escape. Default is taken from QUOTING_STYLE environment variable, or 'shell' if unset.
-b --backup	Back up the original contents of each file.
--backup-if-mismatch	Back up if the patch does not match exactly.
--no-backup-if-mismatch	Back up mismatches only if otherwise requested.
-V STYLE	
--version-control=STYLE	Use STYLE version control. STYLE is either 'simple', 'numbered', or 'existing'.
-B PREFIX	
--prefix=PREFIX	Prepend PREFIX to back up file names.
-Y PREFIX	
--basename-prefix=PREFIX	Prepend PREFIX to back up file base names.
-z SUFFIX --suffix=SUFFIX	Append SUFFIX to back up file names.
-g NUM --get=NUM	Get files from RCS, etc., if positive; ask if negative.
-t --batch	Ask no questions; skip bad-prereq patches; assume reversed.
-f --force	Like -t, but ignore bad-prereq patches, and assume unreversed.
-s --quiet --silent	Work silently unless an error occurs.
--verbose	Output extra information about the work being done.
--dry-run	Do not actually change any files; just print what would happen.
--posix	Conform to the POSIX standard.
-d DIR --directory=DIR	Change the working directory to DIR first.
--binary	Read and write data in binary mode (no effect on this platform).
-v --version	Output version info.
--help	Output this help.

Toolbox FTP archive Web site

ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/patch

Tool developer Web site

http://www.fsf.org/software/patch/patch.html

Tool FTP archive Web site

ftp://ftp.gnu.org/pub/gnu/patch

Files

```
/opt/freeware/bin/patch
/opt/freeware/doc/patch-2.5.4
/opt/freeware/doc/patch-2.5.4/NEWS
/opt/freeware/doc/patch-2.5.4/README
/opt/freeware/man/man1/patch.1
/usr/linux/bin/patch
```

Prereqs

The PATCH package does not have any prerequisites.

Examples

The GNU **patch** command uses output from the GNU **diff** command to change one source code file based on the changes made to the other. In the following example we create a **diff** output file for two versions of the ourcode.c program. Please refer to 6.4.5, “gcc” on page 233, for details about this source code example.

First we have two subdirectories, one directory with the original code (1), and one directory with our modified version (2). We use the **tee** command to save the output from the **diff -u** command in the ourcode.diff file, while at the same time displaying the changes to *stdout*.

Example 6-31 Using GNU diff

```
root@fenris:/work: /usr/linux/bin/diff -u 1/ourcode.c 2/ourcode.c | tee
ourcode.diff
--- 1/ourcode.c    Wed Oct  9 20:05:59 2002
+++ 2/ourcode.c    Wed Oct  9 20:05:37 2002
@@ -1 +1,6 @@
-main () { printf("ourcode!\n"); }
+main () {
+char *apointer;
+apointer = (char *)malloc(100000);
+strncpy(apointer,"SWEET!",100000);
+printf("ourcode! %s EXCELLENT\n",apointer);
+}
```

In the **diff** output above, you see the first two lines describing the files that were compared. Lines after the @@ signs that start with a minus sign (-) are to be removed, and lines staring with a plus sign (+) are to be added.

Other GNU **diff** command options can be used to create patch files is the **-Naur** combination. The -N option lets the patch create and remove files; -a lets the patch update non-text files; -u generates useful time stamps and enough context; and -r lets the patch update subdirectories.

To patch the original ourcode.c program, we use the following **patch** command:

```
/usr/linux/bin/patch -i ourcode.diff -p0
```

Example 6-32 Using patch

```
root@fenris:/work: /usr/linux/bin/patch -i ourcode.diff -p0
patching file 1/ourcode.c
```

The reason we use the -p0 option with the **patch** command, in Example 6-32 is that **patch**, by default, strips all directories from the file name to **patch**, from the **diff** file. Since our current directory was the same, as when we created the **diff** file, we used zero with the -p option, which instructs the **patch** command to not strip any slashes (directories) from the path of the file to apply the patch to.

6.4.5 gcc

The **gcc** command is part of the GNUPro package for AIX. The **gcc** is a compiler. Simplified it transforms source code programs written in a programming language into machine-executable form. It is possible to use **gcc** as a front end to language translators for other programming languages. The compilation of files goes through one or more of four stages: *Preprocessing*, *compilation*, *assembly*, and *linking*.

Synopsis

```
gcc [options] file...
```

The following list describes options for the **gcc** command:

-pass-exit-codes	Exit with highest error code from a phase.
--help	Display this information. (Use -v --help to display command-line options of sub-processes.)
-dumpspecs	Display all of the built-in spec strings.
-dumpversion	Display the version of the compiler.
-dumpmachine	Display the compiler's target processor.
-print-search-dirs	Display the directories in the compiler's search path.
-print-libgcc-file-name	Display the name of the compiler's companion library.
-print-file-name=*lib*	Display the full path to library *lib.*
-print-prog-name=*prog*	Display the full path to compiler component *prog*
-print-multi-directory	Display the root directory for versions of libgcc.
-print-multi-lib	Display the mapping between command-line options and multiple library search directories.
-Wa,*options*	Pass comma-separated *options* on to the assembler.

-Wp,*options*	Pass comma-separated *options* on to the preprocessor.
-Wl,*options*	Pass comma-separated *options* on to the linker.
-Xlinker *arg*	Pass *arg* on to the linker.
-save-temps	Do not delete intermediate files.
-pipe	Use pipes rather than intermediate files.
-time	Time the execution of each subprocess.
-specs=*file*	Override built-in specs with the contents of *file.*
-std=*standard*	Assume that the input sources are for *standard.*
-B *directory*	Add *directory* to the compiler's search paths.
-b *machine*	Run **gcc** for target *machine*, if installed.
-V *version*	Run **gcc** version number *version*, if installed.
-v	Display the programs invoked by the compiler.
-E	Preprocess only; do not compile, assemble, or link.
-S	Compile only; do not assemble or link.
-c	Compile and assemble, but do not link.
-o *file*	Place the output into *file.*
-x *language*	Specify the language of the following input files. Permissible languages include: c, c++, assembler, or `none` (`none` means revert to the default behavior of guessing the language based on the file's extension).

Options starting with -g, -f, -m, -O, or -W are automatically passed on to the various subprocesses invoked by **gcc**. In order to pass other options on to these processes the ***-Wletter*** options must be used.

Toolbox FTP archive Web site

ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/GNUPro

Tool developer Web site

http://www.gnu.org/software/gcc/gcc.html

Tool FTP archive Web site

ftp://ftp.gnu.org/pub/gnu/gcc

Files

```
/opt/freeware/GNUPro/COPYING
/opt/freeware/GNUPro/COPYING.LIB
/opt/freeware/GNUPro/COPYING.NEWLIB
/opt/freeware/GNUPro/CYGNUS
/opt/freeware/GNUPro/GNUPro.pdf
/opt/freeware/GNUPro/Install
/opt/freeware/GNUPro/bin/cpp
```

```
/opt/freeware/GNUPro/bin/gcc
/opt/freeware/GNUPro/bin/gcov
/opt/freeware/GNUPro/lib/
...(files omitted)...
/opt/freeware/GNUPro/release-notes.txt
/opt/freeware/info/cpp.info*
...(files omitted)...
/opt/freeware/info/gcc.info*
...(files omitted)...
/opt/freeware/info/standards.info
/opt/freeware/man/man1/cccp.1
/opt/freeware/man/man1/cpp.1
/opt/freeware/man/man1/gcc.1
/usr/bin/gcc
/usr/bin/gcov
/usr/bin/powerpc-ibm-aix5.1.0.0-gcc
/usr/lib/gcc-lib
/usr/linux/bin/cpp
```

Prereqs

The **gcc** part of the GNUPro package does not have any prerequisites.

Examples

To compile a C program with the **gcc** compiler, first you must have some source code written in the C programming language. Let us start by creating a simple example (end the input to the **cat** command by typing the CTRL-D command sequence):

```
cat > ourcode.c
main () { printf("ourcode!\n"); }
CTRL-D
```

Now we are ready to compile our one-line C program. Just type **gcc *filename*** like in the example below for the ourcode.c program:

```
gcc ourcode.c
```

If the compile was ok, which it was, **gcc** has now created a binary and executable file in the current directory named a.out. You can run the compiled ourcode.c program by typing `a.out` on the command line. If you do, you will see the following text displayed:

```
ourcode!
```

To create an executable file named ourcode, and not the default a.out, use the -o option with the **gcc** compiler:

```
gcc -o ourcode ourcode.c
```

6.4.6 gdb

The **gdb** command is part of the GNUPro package for AIX. The **gdb** command is a debugging tool for monitoring how a program executes, or what a program was doing at the moment it crashed. Programs to be debugged can be written in C, C++, Pascal, Modula, and several other languages. The programs are normally executed on the same machine as the **gdb** command (*native*) or on another machine (*remote*).

Synopsis

```
gdb [options] [executable-file [core-file or process-id]]
```

The following list describes options for the **gdb** command:

--[no]async	Enable (disable) asynchronous version of CLI.
-b BAUDRATE	Set serial port baud rate used for remote debugging.
--batch	Exit after processing options.
--cd=DIR	Change current directory to DIR.
--command=FILE	Execute **gdb** commands from FILE.
--core=COREFILE	Analyze the core dump COREFILE.
--dbx	**dbx** compatibility mode.
--directory=DIR	Search for source files in DIR.
--epoch	Output information used by epoch emacs-gdb interface.
--exec=EXECFILE	Use EXECFILE as the executable.
--fullname	Output information used by emacs-gdb interface.
--help	Print this message.
--interpreter=INTERP	Select a specific interpreter/user interface.
--mapped	Use mapped symbol files if supported on this system.
--nw	Do not use a window interface.
--nx	Do not read .gdbinit file.
--quiet	Do not print version number on startup.
--readnow	Fully read symbol files on first access.
--se=FILE	Use FILE as symbol file and executable file.
--symbols=SYMFILE	Read symbols from SYMFILE.
--tty=TTY	Use TTY for input/output by the program being debugged.
--version	Print version information and then exit.
-w	Use a window interface.
--write	Set writing into executable and core files.
--xdb	**xdb** compatibility mode.

Toolbox FTP archive Web site

ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/GNUPro

Tool developer Web site

http://www.gnu.org/software/gdb/gdb.html

Tool FTP archive Web site

ftp://ftp.gnu.org/gnu/gdb

Files

```
/opt/freeware/GNUPro/bin/gdb
/opt/freeware/GNUPro/
```
...(files omitted)...
```
/opt/freeware/info/gdb.info*
```
...(files omitted)...
```
/opt/freeware/info/gdbint.info*
```
...(files omitted)...
```
/opt/freeware/info/stabs.info*
```
...(files omitted)...
```
/opt/freeware/man/man1/gdb.1
/usr/bin/gdb
/usr/share/cygnus
```

Prereqs

The **gdb** part of the GNUPro package does not have any prerequisites.

Examples

To use the **gdb** binary debugger you need a program to debug. The following examples are based on the ourcode.c program. Please refer to 6.4.5, "gcc" on page 233.

First we start the **gdb** in quiet mode (Example 6-33). By default it displays information about GNU software and the lack of warranties for the usage of the debugger. The **r** command starts the program and the **q** command quits the debugging session. The `ourcode!` text displayed is the output from the executing program itself.

Example 6-33 Using gdb

```
root@fenris:/home/work: gdb -q ourcode
(gdb) r
Starting program: /home/work/ourcode
ourcode!

Program exited with code 06.
(gdb) q
```

Example 6-34 on page 238 shows how to use **gdb** in a post mortem (after the executable crashed and dumped), to find where it went wrong.

Example 6-34 Using gdb with SIGSEGV

```
root@fenris:/home/work: gdb -q ourcode
(gdb) r
Starting program: /home/work/ourcode

Program received signal SIGSEGV, Segmentation fault.
0x10000544 in strncpy ()
(gdb) where
#0  0x10000544 in strncpy ()
#1  0x10000380 in main ()
#2  0x100001dc in __start ()
(gdb) q
A debugging session is active.
Do you still want to close the debugger?(y or n) y
```

Example 6-34 shows that the process (the running executable), received a SIGSEGV signal due to a segmentation fault. This is of course a memory allocation error of some kind. We then issued the **where** command to find out where the error occurred, and **gdb** displayed the stack trace. The stack trace ended with the strncpy() subroutine, so we can take a closer look at that part of the code and how it works. Looking at the source in Example 6-35 we see that we do not allocate any memory space for the apointer pointer, but we copy 100000 bytes from a string that is only 6 bytes in size. This is what caused the segmentation violation.

Example 6-35 The SIGSEGV failing source code

```
main () {
char *apointer;
strncpy(apointer,"SWEET!",100000);
printf("ourcode! %s\n",apointer);
}
```

For the next **gdb** example, we will step through the execution based on our C source code. For this purpose we have rectified our source, and it now looks like Example 6-36. The memory allocation part is highlighted (and we do not use the free() subroutine since it is just a short example and the allocated memory will be released when the process exits).

Example 6-36 The corrected source code

```
main () {
char *apointer;
apointer = (char *)malloc(100000);
strncpy(apointer,"SWEET!",100000);
printf("ourcode! %s\n",apointer);
```

```
}
```

The **gdb** example (Example 6-37), with the corrected and recompiled source code above, starts by setting a breakpoint at the printf() subroutine (the **b printf** command). Then we start the execution of the program, which stops at the printf() breakpoint. Next we display the stack backtrace by issuing the **bt** command, and **gdb** displays that we are at the printf() subroutine. Finally we continue by using the **c** command and quit (q) the debugging session after the process has terminated with the return code 015.

Example 6-37 Stepping though the executable

```
root@fenris:/home/work: gdb -q ourcode
(gdb) b printf
Breakpoint 1 at 0x100005b8
(gdb) r
Starting program: /home/work/ourcode
Breakpoint 1 at 0x1e108
Breakpoint 1 at 0xd01ebe50

Breakpoint 1, 0xd01ebe50 in printf ()
(gdb) bt
#0  0xd01ebe50 in printf ()
#1  0x100003a8 in main ()
#2  0x100001dc in __start ()
(gdb) c
Continuing.
ourcode! SWEET!

Program exited with code 015.
(gdb) q
```

6.4.7 binutils

The commands in the BINUTILS package are a collection of program development commands that are part of the GNUPro package for AIX. The *binutil* commands part of the GNUPro package is:

addr2line Converts addresses to file and line.
ar Creates, modifies, and extracts from archives.
as The portable GNU assembler.
gasp GNU Assembler Macro Preprocessor.
gld The GNU linker.
gperf Generates a perfect hash function from a key set.
gstrip Discards symbols.
nm Lists symbols from object files.

objcopy	Copies and translates object files.
objdump	Displays information from object files.
ranlib	Indexes the contents of an archive.
readelf	Displays the contents of ELF format files.
send-pr	Sends problem report (PR) to a central support site.
size	Lists the section sizes of an object or archive file.
strings	Lists printable strings from files.

Synopsis

addr2line [-CfsHV] [-b bfdname] [--target=bfdname] [-e executable] [--exe=executable] [--demangle] [--basenames] [--functions] [addr addr ...]
ar [-]{dmpqrstx}[abcilosSuvV] [member-name] archive-file file...
ar -M [<mri-script]
as [option...] [asmfile...]
gasp [option...] [in-file...]
gld [options] file...
gperf [OPTION]... [INPUT-FILE]
gstrip *switches* in-file(s)
nm [-aABCDglnopPrsuvV] [-t radix] [--radix=radix] [--target=bfdname]
[--debug-syms] [--extern-only] [--print-armap] [--print-file-name]
[--numeric-sort] [--no-sort] [--reverse-sort] [--size-sort]
[--undefined-only] [--portability] [-f {bsd,sysv,posix}]
[--format={bsd,sysv,posix}] [--demangle] [--no-demangle] [--dynamic]
[--defined-only] [--line-numbers]
[--version] [--help]
[file...]
objcopy *switches* in-file [out-file]
objdump *switches* file(s)
ranlib [-vV] archive
readelf {options} elf-file(s)
send-pr [-PVL] [-t address] [-f filename] [-s severity] [-c address]
[--request-id] [--version]
size [-ABdoxV] [--format=berkeley|sysv] [--radix=8|10|16] [--target=bfdname]
[--version] [--help] [file...]
strings [-afov] [-n min-len] [-min-len] [-t {o,x,d}] [-] [--all]
[--print-file-name] [--bytes=min-len] [--radix={o,x,d}] [--target=bfdname]
[--help] [--version] file...

Toolbox FTP archive Web site

ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/GNUPro

Tool developer Web site

http://sources.redhat.com/binutils

Tool FTP archive Web site

ftp://ftp.gnu.org/gnu/binutils

Files

```
/opt/freeware/GNUPro/bin/addr2line
/opt/freeware/GNUPro/bin/ar
/opt/freeware/GNUPro/bin/as
/opt/freeware/GNUPro/bin/gasp
/opt/freeware/GNUPro/bin/gld
/opt/freeware/GNUPro/bin/gperf
/opt/freeware/GNUPro/bin/gprof
/opt/freeware/GNUPro/bin/gstrip
/opt/freeware/GNUPro/bin/nm
/opt/freeware/GNUPro/bin/objcopy
/opt/freeware/GNUPro/bin/objdump
/opt/freeware/GNUPro/bin/ranlib
/opt/freeware/GNUPro/bin/readelf
/opt/freeware/GNUPro/bin/send-pr
/opt/freeware/GNUPro/bin/size
/opt/freeware/GNUPro/bin/strings
/opt/freeware/GNUPro/include/ansidecl.h
/opt/freeware/GNUPro/include/bfd.h
/opt/freeware/GNUPro/include/bfdlink.h
/opt/freeware/GNUPro/lib/aix64/libiberty.a
```
...(files omitted)...
```
/opt/freeware/info/as.info*
```
...(files omitted)...
```
/opt/freeware/man/dvi/gperf.dvi
/opt/freeware/man/html/gperf.html
/opt/freeware/man/man1/
```
...(files omitted)...
```
/usr/bin/addr2line
/usr/bin/gasp
/usr/bin/gperf
/usr/bin/objcopy
/usr/bin/objdump
/usr/bin/readelf
/usr/bin/send-pr
/usr/linux/bin/ar
/usr/linux/bin/as
/usr/linux/bin/gprof
/usr/linux/bin/nm
/usr/linux/bin/ranlib
/usr/linux/bin/size
/usr/linux/bin/strings
```

Prereqs

The BINUTILS part of the GNUPro package does not have any prerequisites.

6.4.8 m4

The M4 package provides the implementation of the traditional UNIX macro processor. The **m4** command is useful for writing text files that can be logically parsed, and is used by many programs as part of their build processes. The **m4** command has built-in functions for including files, running shell commands, doing arithmetic, etc. The **autoconf** program needs **m4** for generating configure scripts, but not for running configure scripts.

Synopsis

```
m4 [OPTION...] [MACRO-DEFINITIONS...] [INPUT-FILE...]
```

The following list describes options for the **m4** command:

--version	Print the version number of the program on standard output.
--help	Print a help summary on standard output.
-G, --traditional	Suppress all the extensions made in this implementation, compared to the System V version.
-E, --fatal-warnings	Stop execution and exit **m4** once the first warning has been issued, considering all warnings to be fatal.
-dFLAGS,--debug=FLAGS	Set the debug level according to the flags FLAGS.
-lNUM,--arglength=NUM	Restrict the size of the output generated by macro tracing.
-oFILE,--error-output=FILE	Redirect debug and trace output to the named FILE.
-IDIR,--include=DIR	Make **m4** search DIR for included files that are not found in the current working directory.
-e,--interactive	Makes this invocation of **m4** interactive.
-s, --synclines	Generate synchronization lines for use by the C preprocessor or other similar tools.
-P,--prefix-builtins	Internally modify *all* built-in macro names so they all start with the prefix m4_.
-HN,--hashsize=*N*	Make the internal hash table for symbol lookup be *N* entries big.
-LN,--nesting-limit=*N*	Artificially limit the nesting of macro calls to *N* levels, stopping program execution if this limit is ever exceeded. When not specified, nesting is limited to 250 levels.
-Q,--quiet, --silent	Suppress warnings about missing or superfluous arguments in macro calls.

-B,-S,-T	These options are present for compatibility with System V m4, but do nothing in this implementation.
-NN,--diversions=N	These options are present only for compatibility with previous versions of GNU **m4**.
-DNAME,-DNAME=VALUE, --define=NAME, --define=NAME=VALUE	This enters `NAME` into the symbol table, before any input files are read. If =VALUE is missing, the value is taken to be the empty string. The VALUE can be any string, and the macro can be defined to take arguments, just as if it were defined from within the input.
-UNAME,--undefine=NAME	This deletes any predefined meaning NAME might have.
-tNAME,--trace=NAME	This enters NAME into the symbol table, as undefined but traced.
-FFILE, --freeze-state FILE	Once execution is finished, write out the frozen state on the specified FILE.
-RFILE, --reload-state FILE	Before execution starts, recover the internal state from the specified frozen FILE.

Toolbox FTP archive Web site

ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/m4

Tool developer Web site

http://www.gnu.org/software/m4

Tool FTP archive Web site

ftp://ftp.gnu.org/pub/gnu/m4

Files

```
/opt/freeware/bin/m4
/opt/freeware/doc/m4-1.4
/opt/freeware/doc/m4-1.4/NEWS
/opt/freeware/doc/m4-1.4/README
/opt/freeware/info/m4.info-1.gz
/opt/freeware/info/m4.info-2.gz
/opt/freeware/info/m4.info-3.gz
/opt/freeware/info/m4.info.gz
/usr/linux/bin/m4
```

Prereqs

The M4 package does not have any prerequisites.

Examples

To exemplify the usage of the **m4** command, we use a simple example with a template Korn shell file.

The script is intended to have certain information statically assigned to variables in the script. Another reason could be to make sure that any licensing or usage documentation in the script is added by the **m4** command, before packaging the shell script for further distribution (such as delivery to a customer or the Open Source community). The next example is the template Korn shell file, named kornshelltemplate.

Example 6-38 Template Korn shell script

```
#!/bin/ksh
# kornshelltemplate
include(m4macrodefinitionfile)

integer fatloop=M4_fatloop
integer i=0
while ((i < fatloop));do
   print ".\c"
   ((i=$i+1))
done
print
exit
```

Notice the include() macro in the top part of the Korn shell template file. We will use the **m4** command to replace the M4-fatloop text with other information. The macro definitions file also contain inclusion of other files containing fixed text, in our case. Example 6-39 is the **m4** macro definition file.

Example 6-39 m4 macro definition file

```
define(M4_fatloop, 100)
syscmd(cat ../template/OSS_LICENSE)
include(../template/OSS_SUPPORT)
```

The define() macro replaces all occurrences of the text M4_fatloop with the text 100. The syscmd() macro executes the **cat** command of the ../template/OSS_LICENSE file, and the include() macro includes the ../template/OSS_SUPPORT file.

The following is the content of our sample OSS_LICENSE file:

```
# This is the LICENSE description for this software...
```

The following is the content of our sample OSS_SUPPORT file:

```
# This is the description of the support procedure for this software...
```

Now all we have to do is execute the **m4** command, specifying the template Korn shell file and redirecting the output from the **m4** command to the Korn shell script to be distributed (amazing_loop.ksh):

```
m4 kornshelltemplate > amazing_loop.ksh
```

The resulting Korn shell script, the amazing_loop.ksh, will then look like inExample 6-40.

Example 6-40 Output Korn shell script from the m4 macro processing

```
#!/bin/ksh

# This is the LICENSE description for this software...

# This is the description of the support procedure for this software...



integer fatloop=100
integer i=0
while ((i < fatloop));do
        print ".\c"
        ((i=$i+1))
done
print
```

6.4.9 gmake

The MAKE package provides a tool for controlling the generation of executables and other non-source files of a program from the program's source files. The **gmake** command allows users to build and install packages without any significant knowledge about the details of the build process. The details about how the program should be built are provided for **gmake** in the program's makefile.

Synopsis

```
gmake [-einqrst] [-k|-S] [-d[A|adg[1|2]mstv]] [-D variable] [-f makefile ]
[variable=value ...] [target ...]
```

The following list describes options for the **gmake** command:

-C dir Change to directory dir before reading the makefiles or doing anything else. If multiple -C options are specified, each is interpreted relative to the previous one: -C / -C etc is equivalent to -C /etc. This is typically used with recursive invocations of `gmake`.

-d	Print debugging information in addition to normal processing. The debugging information says which files are being considered for remaking, which file times are being compared and with what results, which files actually need to be remade, which implicit rules are considered, and which are applied.
-e	Give variables taken from the environment precedence over variables from makefiles.
-f file	Use file as a makefile.
-i	Ignore all errors in commands executed to remake files.
-I dir	Specify a directory dir to search for included makefiles. If several -I options are used to specify several directories, the directories are searched in the order specified. Unlike the arguments to other flags of `gmake`, directories given with -I flags may come directly after the flag; -Idir is allowed, as well as -I dir. This syntax is allowed for compatibility with the C preprocessor's -I flag.
-j jobs	Specify the number of jobs (commands) to run simultaneously. If there is more than one -j option, the last one is effective. If the -j option is given without an argument, make will not limit the number of jobs that can run simultaneously.
-k	Continue as much as possible after an error. While the target that failed, and those that depend on it, cannot be remade, the other dependencies of these targets can be processed all the same.
-l, -l load	Specify that no new jobs (commands) should be started if there are others jobs running and the load average is at least load (a floating-point number). With no argument, removes a previous load limit.
-n	Print the commands that would be executed, but do not execute them.
-o file	Do not remake the file even if it is older than its dependencies, and do not remake anything on account of changes in file. Essentially the file is treated as very old and its rules are ignored.
-p	Print the database (rules and variable values) that results from reading the makefiles; then execute as usual or as otherwise specified. This also prints the version information given by the -v switch. To print the database without trying to remake any files, use `gmake -p -f/dev/null`.

-q Query mode, do not run any commands or print anything; just return an exit status that is zero if the specified targets are already up to date; nonzero otherwise.

-r Eliminate use of the built-in implicit rules. Also clear out the default list of suffixes for suffix rules.

-s Silent operation; do not print the commands as they are executed.

-S Cancel the effect of the -k option. This is never necessary except in a recursive make where -k might be inherited from the top-level make via MAKEFLAGS or if you set -k in MAKEFLAGS in your environment.

-t Touch files (mark them up to date without really changing them) instead of running their commands. This is used to pretend that the commands were done, in order to fool future invocations of **`gmake`**.

-v Print the version information about the program.

-w Print a message containing the working directory before and after other processing. This may be useful for tracking down errors from complicated nests of recursive **`gmake`** commands.

-W file Pretend that the target file has just been modified. When used with the -n flag, this shows what would happen if you were to modify that file.

Toolbox FTP archive Web site

ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/make

Tool developer Web site

http://www.gnu.org/software/make/make.html

Tool FTP archive Web site

ftp://ftp.gnu.org/pub/gnu/make

Files

```
/opt/freeware/bin/gmake
/opt/freeware/bin/make
/opt/freeware/doc/make-3.79.1
/opt/freeware/doc/make-3.79.1/NEWS
/opt/freeware/doc/make-3.79.1/README
/opt/freeware/info/make.info*.gz
```

...(files omitted)...

```
/opt/freeware/man/man1/make.1
/usr/bin/gmake
/usr/linux/bin/make
```

Prereqs

The MAKE package does not have any prerequisites.

Examples

To use gmake we need a makefile. The makefile can either be named makefile, Makefile, or something else. If you name it something else, you must use the -f *makefilename* option of the **gmake** command.

The following is a hardcoded example of a makefile for the ourcode.c program. Please refer to 6.4.5, “gcc” on page 233. Notice that the second line is indented. You *must* indent with Tab only.

```
ourcode:ourcode.c
        gcc -o ourcode ourcode.c
```

To compile the **ourcode** program from the ourcode.c C source file, just type `gmake`, like in Example 6-41. The second line is the output from the **make** command.

Example 6-41 Using make

```
root@fenris:/home/work: gmake
        gcc -o ourcode ourcode.c

root@fenris:/home/work: ls -l ourcode
-rwxr-xr-x   1 root     system       240660 Oct 09 18:11 ourcode

root@fenris:/home/work: ourcode
ourcode!
```

The **gmake** command’s makefile can be extensively customized, so please refer to the man pages to utilize its capabilities and set it up appropriately for your development environment.

6.4.10 cvs

The Concurrent Version System (CVS) package provides a version control system that allows you to keep old versions of files (usually source code); and keep a log of who, when, and why changes occurred, etc., like RCS or SCCS. The **cvs** command only stores the differences between versions, instead of every version of every file you have ever created. The **cvs** command also keeps a log of the changes that occur.

The **cvs** command is very helpful for managing releases and controlling the concurrent editing of source files among multiple authors. Instead of providing version control for a collection of files in a single directory, the **cvs** command provides version control for a hierarchical collection of directories consisting of revision controlled files. These directories and files can then be combined together to form a software release.

Synopsis

```
cvs [cvs-options] command [command-options-and-arguments]
cvs update ...
cvs add ...
cvs remove ...
cvs commit ...
```

The following list describes options for the **cvs** command:

-H [--help]	Display usage information about the specified **cvs** command.
-Q	Cause the command to be really quiet; the command will generate output only for serious problems.
-q	Cause the command to be somewhat quiet
-b bindir	Use bindir as the directory where RCS programs are located.
-d CVS_root_directory	Use CVS_root_directory as the root directory path name of the master source repository.
-e editor	Use editor to enter revision log information.
-f	Do not read the **cvs** startup file (~/.cvsrc).
-l	Do not log the **cvs** command in the command history
-n	Do not change any files.
-t	Trace program execution; display messages showing the steps of **cvs** activity.
-r	Make new working files read-only.
-v [--version]	Display version and copyright information for **cvs**.
-w	Make new working files read-write (default).
-x	Encrypt all communication between the client and the server (Kerberos).
-z compression-level	When transferring files across the network use **gzip** with compression level compression-level to compress and decompress data as it is transferred. Requires the presence of the **gzip** program in the current search path at both ends of the link.

Toolbox FTP archive Web site

ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/cvs

Tool developer Web site

http://www.cvshome.org

Tool FTP archive Web site

http://ccvs.cvshome.org/servlets/ProjectDownloadList

Files

/opt/freeware/bin/cvs
/opt/freeware/bin/cvsbug
/opt/freeware/bin/rcs2log
/opt/freeware/doc/cvs-1.11.1p1/
...(files omitted)...
/opt/freeware/info/cvs.info*.gz
...(files omitted)...
/opt/freeware/info/cvsclient.info*.gz
...(files omitted)...
/opt/freeware/man/man1/cvs.1
/opt/freeware/man/man5/cvs.5
/opt/freeware/man/man8/cvsbug.8
/opt/freeware/share/cvs
/opt/freeware/share/cvs/contrib/
...(files omitted)...
/usr/bin/cvs
/usr/bin/cvsbug
/usr/bin/rcs2log

Prereqs

The CVS package does not have any prerequisites.

Examples

When using **cvs** to connect to a CVS server, you need to create the $HOME/.cvspass file first, if it is the very first time you run **cvs**:

```
touch ~/.cvspass
```

The following example connects us to the www.rpm.org CVS server:

```
cvs -d :pserver:anonymous@cvs.rpm.org:/cvs/devel login
```

Now we can examine the content of the .cvspass file. We just use the **cat** command, as in **cat ~/.cvspass**. The output from our example above could then be:

```
/1 :pserver:anonymous@cvs.rpm.org:2401/cvs/devel A
```

On the rpm.org site, we use **cvs** to acquire the Maximum RPM book (max-rpm) in SGML format. In Example 6-42 on page 251, the **get** command is used and the max-rpm directory (in the current directory) will be created.

Example 6-42 Using cvs to download the Maximum RPM book (max-rpm)

```
root@fenris:/cvs: cvs -d :pserver:anonymous@cvs.rpm.org:/cvs/devel get max-rpm
cvs server: Updating max-rpm
U max-rpm/.cvsignore
U max-rpm/Makefile.am
...(files omitted)...
U max-rpm/rpmrc-file.sgml
U max-rpm/xref.sgml
```

Our next example shows how to use **cvs** to connect to samba.org:

```
cvs -d :pserver:cvs@pserver.samba.org:/cvsroot login
```

This added the following entry to our ~/.cvspass file (**cat ~/.cvspass**):

```
/1 :pserver:cvs@pserver.samba.org:2401/cvsroot Ah<Z
```

Note the string Ah<Z; this is the encrypted password, since the site maintainers want users to use the public password is **cvs**. Now we do a *check out* of the complete samba source tree:

```
cvs -z5 -d :pserver:cvs@pserver.samba.org:/cvsroot co samba
```

Note the usage of level five (5) compression with the -z5 option. The CVS downloaded source is now available in the `samba` directory (in our current directory). We could have used a different target directory, such as /local/oss/samba, and then the source would be stored there instead.

If the CVSROOT environment variable is not set and the -d *CVSROOT* option is not used, the current directory is used as the *root* directory for **cvs** operations on our local host.

Many Open Source sites give instructions on how to use **cvs** to keep your own source code up to date. The `http://samba.org/samba/cvs.html` Web page gives the instructions for samba.org.

6.4.11 info

The INFO package provides the **info** command for reading the texinfo file format that is used in most GNU documentation.

Synopsis

```
info [OPTION]... [MENU-ITEM...]
```

The following list describes options for the **info** command:

--apropos=SUBJECT Look up SUBJECT in all indices of all manuals.

--directory=DIR	Add DIR to INFOPATH.
--dribble=FILENAME	Remember user keystrokes in FILENAME.
--file=FILENAME	Specify info file to visit.
--help	Display this help and exit.
--index-search=STRING	Go to node pointed to by index entry STRING.
--node=NODENAME	Specify nodes in first visited info file.
--output=FILENAME	Output selected nodes to FILENAME.
--restore=FILENAME	Read initial keystrokes from FILENAME.
--show-options, --usage	Go to command-line options node.
--subnodes	Recursively output menu items.
--vi-keys	Use **vi**-like and **less**-like key bindings.
--version	Display version information and exit.

Toolbox FTP archive Web site

ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/texinfo

Tool developer Web site

http://www.fsf.org/software/texinfo/texinfo.html

Tool FTP archive Web site

ftp://ftp.gnu.org/pub/gnu/texinfo

Files

```
/etc/info-dir
/opt/freeware/bin/info
/opt/freeware/bin/install-info
/opt/freeware/info/dir
/opt/freeware/info/info-stnd.info-1.gz
/opt/freeware/info/info-stnd.info-2.gz
/opt/freeware/info/info-stnd.info.gz
/opt/freeware/info/info.info.gz
/sbin/install-info
/usr/info
```

Prereqs

The INFO package does not have any prerequisites. The source for **info** is in the TEXINFO SRPM package.

Examples

Note that **info** is a browser for information, much like a Web browser, in text mode.

You can use the arrow keys to go around in the screen panel, and to select a topic, position the cursor over the topic, and press the Enter key.

To end your info session, just press the q key. Type the h key to get help on how to use the interactive **info** session. Start info with the **info *topic*** command (where topic usually is the name of a command).

The following example shows how to start **info** to read more about the **gzip** command:

```
info gzip
```

The next example shows how to start **info** reading about the **gzip** command with vi keystroke emulation. Note that the arrow keys work differently when you use the vi keystroke emulation mode:

```
info --vi-keys gzip
```

Figure 6-6 info screen

6.5 Miscellaneous tools

The miscellaneous packages described here are additional tools from the Toolbox that does not fit the other sections, but can be of great use. We cover the following tools in this section:

- **webmin**
- **samba**

- **vnc**
- **php**
- **sudo**

6.5.1 webmin

The WEBMIN package is a Web-based administration interface for Linux/UNIX systems. Using **webmin** you can configure DNS, DHCP, Samba, NFS, local/remote file systems, FTP servers, databases, users, groups, running commands, and more, using your Web browser. WEBMIN can also be integrated with SSH/SSL.

Synopsis

`/etc/rc.d/init.d/webmin [ start | stop | status | restart ]`

Toolbox FTP archive Web site

ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/webmin

Tool developer Web site

http://www.webmin.com

Tool FTP archive Web site

http://dl.sourceforge.net/sourceforge/webadmin

Files

```
/etc/rc.d/init.d/webmin
/etc/rc.d/rc2.d/S99webmin
/etc/rc.d/rc3.d/S99webmin
/etc/rc.d/rc5.d/S99webmin
/etc/sysconfig/daemons/webmin
/opt/freeware/libexec/webmin
```

...(omitted 6620 filenames)...

Prereqs

The WEBMIN package does not have any prerequisites.

Examples

If the Toolbox WEBMIN services are out of date, download the latest TAR archive from the http://www.webmin.com Web site. You usually find a link to the download page in the upper right corner of the first Web page on this site, or you can look for the latest TAR archive in the http://dl.sourceforge.net/sourceforge/webadmin directory.

The following example shows how to use the **wget** command to download Version 1.020:

```
wget http://dl.sourceforge.net/sourceforge/webadmin/webmin-1.020.tar.gz
```

For updating, building, and installing the WEBMIN RPM package from the downloaded source code package, please refer to Chapter 3, “Porting Open Source Software to AIX” on page 57.

If you decide to install the **webmin** tools directly from the downloaded and unpacked TAR archive, run the **setup.sh** command. In Example 6-43 we use the default values; all values and keyboard keys used are highlighted.

Example 6-43 Configuring webmin from the unpacked TAR archive directory

```
root@fenris:/images/src/webmin-1.020: setup.sh
***********************************************************************
*            Welcome to the Webmin setup script, version 1.020        *
***********************************************************************
Webmin is a web-based interface that allows Unix-like operating
systems and common Unix services to be easily administered.

Installing Webmin in /usr/sys/inst.images/src/webmin-1.020 ...

***********************************************************************
Webmin uses separate directories for configuration files and log files.
Unless you want to run multiple versions of Webmin at the same time
you can just accept the defaults.

Config file directory [/etc/webmin]: <ENTER>
Log file directory [/var/webmin]: <ENTER>

***********************************************************************
Webmin is written entirely in Perl. Please enter the full path to the
Perl 5 interpreter on your system.

Full path to perl (default /usr/bin/perl): <ENTER>

Testing Perl ...
Perl seems to be installed ok

***********************************************************************
For Webmin to work properly, it needs to know which operating system
type and version you are running. Please select your system type by
entering the number next to it from the list below
---------------------------------------------------------------------------
  1) Sun Solaris           2) Caldera OpenLinux eS   3) Caldera OpenLinux
  4) Red Hat Linux          5) Slackware Linux        6) Debian Linux
  7) SuSE Linux            8) United Linux           9) Corel Linux
```

```
 10) TurboLinux           11) Cobalt Linux         12) Mandrake Linux
 13) Mandrake Linux Corpo 14) Delix DLD Linux      15) Conectiva Linux
 16) MSC Linux            17) MkLinux              18) LinuxPPC
 19) XLinux               20) LinuxPL              21) Trustix
 22) Cendio LBS Linux     23) Ute Linux            24) Lanthan Linux
 25) Yellow Dog Linux     26) Corvus Latinux       27) Immunix Linux
 28) Gentoo Linux         29) Generic Linux        30) FreeBSD
 31) OpenBSD              32) NetBSD               33) BSDI
 34) HP/UX                35) SGI Irix             36) DEC/Compaq OSF/1
 37) IBM AIX              38) SCO UnixWare         39) SCO OpenServer
 40) Darwin               41) Mac OS X             42) Mac OS X / OS X Serv
 43) Cygwin
----------------------------------------------------------------------------
Operating system: 37

Please choose which version of IBM AIX you are running, by entering
the number next to it from the list below
----------------------------------------------------------------------------
  1) IBM AIX 4.3
----------------------------------------------------------------------------
Version: 1

Operating system name:    IBM AIX
Operating system version: 4.3

***********************************************************************
Webmin uses its own password protected web server to provide access
to the administration programs. The setup script needs to know :
 - What port to run the web server on. There must not be another
   web server already using this port.
 - The login name required to access the web server.
 - The password required to access the web server.
 - The hostname of this system that the web server should use.
 - If the webserver should use SSL (if your system supports it).
 - Whether to start webmin at boot time.

Web server port (default 10000): <ENTER>
Login name (default admin): <ENTER>
Login password: admin<ENTER>
Password again: admin<ENTER>
Web server host name (default fenris): <ENTER>
The Perl SSLeay library is not installed. SSL not available.
Start Webmin at boot time (y/n): <ENTER>
***********************************************************************
Creating web server config files..
..done

Creating access control file..
..done
```

```
Inserting path to perl into scripts..

Creating start and stop scripts..
..done

Copying config files..
..done

Creating uninstall script /etc/webmin/uninstall.sh ..
..done

Changing ownership and permissions ..
..done

Running postinstall scripts ..
..done

Attempting to start Webmin mini web server..
Starting Webmin server in /usr/sys/inst.images/src/webmin-1.020
..done

***********************************************************************
Webmin has been installed and started successfully. Use your web
browser to go to

  http://fenris:10000/

and login with the name and password you entered previously.
```

Note that the user ID and password in the example above *are not* recommendations, but are shown here only for educational purposes.

How to start the webmin services

To use the WEBMIN services, you must install and then start the **webmin** server on each system that you will connect to and administrate from your Web browser. This is done with the **webmin start** command. The next example uses the default path to the **webmin** configuration directory (from Figure 6-43 on page 255):

```
/etc/webmin/start
```

How to use the webmin services

After the WEBMIN services are started, you can connect your Web browser to port 10000 (ten thousand is the default port number) on the host where the **webmin** server is running. Figure 6-7 on page 258 shows part of the login screen.

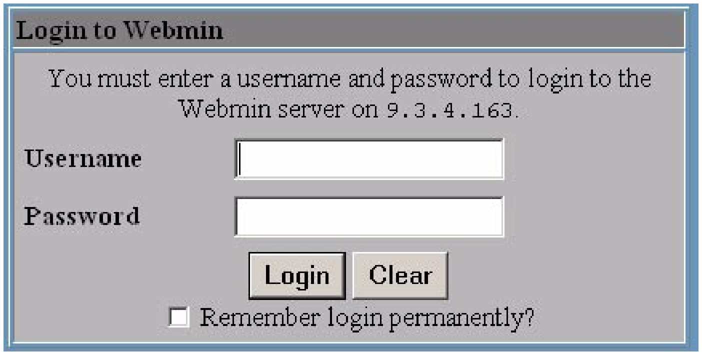

Figure 6-7 webmin login Web page

Figure 6-8 shows the Servers management panel to the **webmin** user interface (the first screen is by default the webmin management panel). Selectable menus in the upper part of the browser screen are webmin, System, Servers, Networking, Hardware, Cluster, and Others.

Figure 6-8 webmin Servers panel

The Servers panel contains subpanels for different system services. In the example above you can manage both the **proftpd** server and **wu-ftpd** server FTP file server tools. When you use the Samba Windows File Sharing panel to configure Samba's smb.conf file, **webmin** does not destroy your previous formatting as the Samba **swat** tool does. While on the subject, you can use the **webmin Others** → **File Manger** tool to export directories using both NFS and

Samba, by selecting a directory in the File Manger and then selecting the **Sharing** icon at the top in the File Manager toolbar.

The advantage of using the WEBMIN services to configure some of the available system services is that, apart from having a graphical user interface, **webmin** also supplys links to man and help pages for the current topic and it is non-intrusive when updating configuration files (it does not alter the current format of the configuration files).

How to stop the webmin services

To stop the **webmin** server, use the **webmin stop** command. The next example uses the default path to the **webmin** configuration directory (from Figure 6-43 on page 255):

```
/etc/webmin/start
```

How to customize the webmin services

Either you update the Toolbox WEBMIN package or customize the WEBMIN services by hand (for information on how to update the WEBMIN RPM from the downloaded source code package, please refer to Chapter 3, “Porting Open Source Software to AIX” on page 57).

To customize WEBMIN services by hand keep the following in mind:

- Unpack the TAR archive in a directory in a permanent file system since it will be used by the WEBMIN services (for example, the **/etc/webmin/start** program points to the miniserv.pl program in the directory where the TAR archive was unpacked).
- Customize the installation when running the **setup.sh** command.
- For each WEBMIN service there is a configuration directory in /etc/webmin (the default path). In this directory there is a config file that usually contains important **webmin** variables for the service.
- The part of the WEBMIN services that updates the specific tool configuration files is written in Perl, but the HTTP graphical user interface is mostly written in Java.

6.5.2 samba

The SAMBA (common, server, and client) Toolbox packages provide:

- A SMB server (**smbd**), to provide Windows NT and LAN Manager-style file and print services to SMB clients.
- A NetBIOS (**nmbd**) name server, which amongst other things gives browsing support.

- An FTP-like SMB file transfer client (**smbclient**) so you can access PC resources (disks and printers) from Linux/UNIX. There is also a tar extension to the client for backing up data on PCs.
- A command-line tool (**rpcclient**) that supports some of the NT administrative functionality, which can be used on Samba, NT workstation, and NT server.
- A Web administration tool (**swat**) that allows remote management of the SAMBA server using a Web browser. SAMBA can also be managed via the WEBMIN package.

smbd is the server daemon that provides file sharing and printing services to Windows clients. The server provides file space and printer services to clients using the SMB/CIFS protocol. This is compatible with the LanManager protocol, and can service LanManager clients. These include MSCLIENT 3.0 for DOS, Windows for Workgroups, Windows 95/98/ME, Windows NT, Windows 2000, OS/2, DAVE for Macintosh, and Server Message Block File System (SMBFS) for Linux or Common Internet File System (CIFS) for AIX .

nmbd is a server that understands and can reply to NetBIOS over IP name service requests, like those produced by SMB/CIFS clients. It also participates in the browsing protocols that make up the Windows Network Neighborhood view.

Synopsis

smbd [-D] [-a] [-i] [-o] [-P] [-h] [-V] [-d *debug level*] [-l *log directory*] [-p *portnumber*] [-O *socket option*] [-s *configuration file*]

The following list describes options for the **samba** command:

-D This parameter causes the server to operate as a daemon.

-a This parameter causes each new connection to append log messages to the log file. This is the default.

-i This parameter causes the server to run "interactively," not as a daemon.

-o This parameter causes the log files to be overwritten when opened. By default, **smbd** will append entries to the log files.

-P Passive option. Causes **smbd** to not send any network traffic out. Used for debugging by the developers only.

-h Prints the help information (usage) for **smbd**.

-v Prints the version number for **smbd**.

-d *debug level*	Debug level is an integer from 0 to 10. The default value if this parameter is not specified is zero.
-l *log directory*	The log directory specifies a log directory into which the "log.smbd" log file will be created for informational and debug messages from the running server.
-O *socket options*	See the socket options parameter in the smb.conf file for details.
-p *port number*	Port number is a positive integer value. The default value if this parameter is not specified is 139.
-s *config file*	The file specified contains the configuration details required by the server.

```
nmbd [ -D ]  [ -a ]  [ -i ]  [ -o ]  [ -P ]  [ -h ]  [ -V ] [ -d debug level ]
[ -H lmhosts file ]  [ -l log directory ]  [ -n primary netbios name ]  [ -p
port number ]  [ -s configuration file ]
```

-D	If specified, this parameter causes **nmbd** to operate as a daemon.
-a	If this parameter is specified, each new connection will append log messages to the log file. This is the default.
-i	If this parameter is specified it causes the server to run "interactively," not as a daemon.
-o	If this parameter is specified, the log files will be overwritten when opened.
-h	Prints the help information (usage) for **nmbd**.
-H *filename*	NetBIOS lmhosts file.
-V	Prints the version number for **nmbd**.
-d *debug level*	Debug level is an integer from 0 to 10.
-l *log directory*	The -l parameter specifies a directory into which the "log.nmbd" log file will be created for operational data from the running **nmbd** server.
-n *primary NetBIOS name*	This option allows you to override the NetBIOS name that Samba uses for itself.
-p *UDP port number*	UDP port number is a positive integer value. The default UDP port number is 137.
-s *configuration file*	The default configuration file name is set at build time.

Toolbox FTP archive Web site

ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/samba

Tool developer Web site

http://www.samba.org

Tool FTP archive Web site

http://www.samba.org/samba/ftp

Files

```
/etc/codepages
```
...(files omitted)...
```
/etc/lmhosts
/opt/freeware/bin/make_printerdef
/opt/freeware/bin/make_smbcodepage
/opt/freeware/bin/make_unicodemap
/opt/freeware/bin/smbcacls
/opt/freeware/bin/smbcontrol
/opt/freeware/bin/smbspool
/opt/freeware/bin/testparm
/opt/freeware/bin/testprns
/opt/freeware/doc/samba-2.2.3a/docs/
```
...(files omitted)...
```
/opt/freeware/doc/samba-2.2.3a/examples/
```
...(files omitted)...
```
/opt/freeware/man/
```
...(files omitted)...
```
/opt/freeware/sbin/nmbd
/opt/freeware/sbin/smbd
/opt/freeware/sbin/swat
/opt/freeware/share/swat/
```
...(files omitted)...
```
/usr/bin/make_printerdef
/usr/bin/make_smbcodepage
/usr/bin/make_unicodemap
/usr/bin/nmblookup
/usr/bin/rpcclient
/usr/bin/smbcacls
/usr/bin/smbclient
/usr/bin/smbcontrol
/usr/bin/smbpasswd
/usr/bin/smbspool
/usr/bin/smbstatus
/usr/bin/smbtar
/usr/bin/testparm
/usr/bin/testprns
/usr/sbin/nmbd
/usr/sbin/smbd
/usr/sbin/swat
```

```
/usr/share/swat
/var/locks/samba
/var/spool/samba
```

Prereqs

The SAMBA server and client packages require that the SAMBA common package is installed. SAMBA client also requires the READLINE package.

Examples

With AIX, and especially AIX 5.2, there are a few **configure** options that could benefit the usage of Samba in a mixed UNIX/Windows environment. This requires that you download the latest SRPM package from the Toolbox Web site:

ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/SRPMS/samba

For building and installing the RPM from the source code package please refer to Chapter 3, “Porting Open Source Software to AIX” on page 57.

You can also download the latest build in TAR archive format from the Samba Web site (http://www.samba.org) and rebuild the SRPM and then rebuild and install the new RPM. Usually the latest stable version of Samba can be downloaded by using the following URL:

http://www.samba.org/samba/ftp/samba-latest.tar.gz

To enable Access Control List (ACL) handling with Samba for AIX, use the --with-acl option of the **configure** command during the build process. By enabling ACL support in Samba, you can add multiple specific user and group permissions to files and directories. By using the ACL support in AIX you are not limited to the *user-group-other* level of permission settings for files and directories. Although Samba does not support all permission settings that are possible to use in a Windows environment, the ACL support enhances the security environment in a mixed UNIX/Windows environment.

To enable Pluggable Authentication Module (PAM) and **winbind** support with Samba for AIX (AIX 5.2 only), use the --with-winbind and --with-pam options of the **configure** command using the build process.

The next example executes **configure** with all the options above, before building the Samba software, from the source directory in the unpacked TAR archive:

```
./configure --with-acl-support --with-winbind --with-pam
```

Testing

With AIX 5.2 you can use the new **cifs** file system type to mount file systems from a SMB/CIFS-compliant file server. This can be used to test Samba file serving functionality without involving a Windows client. You can use the **mount** command to mount a **cifs** file system:

```
mount -v cifs -n samba1/user1/pass1 /home /home/user1
```

In the example above samba1 is the host, user1 is the user we authenticate with the pass1 password. The /home file system is the Windows share name and /home/user1 is the local mount point.

6.5.3 vnc

The Virtual Network Computing (VNC) package provides a remote display system that allows you to view a computing desktop environment not only on the machine where it is running, but from anywhere on the Internet and from a wide variety of machine architectures.

Synopsis

```
vncserver [:number] [-name desktop-name] [-depth depth] [-geometry
widthxheight] [-pixelformat rgbNNN|bgrNNN] Xvnc-options...

vncserver -kill :number
```

The following list describes options for the **vnc** command:

-name name	Every desktop has a name that may be displayed by the viewer. It defaults to X.
-geometry widthxheight	Specify the desktop size to be created (default is 1024x768).
-depth depth	Specify the pixel depth in bits for the desktop to be created (default is 8).
-pixelformat format	Specify the pixel format for server to use (BGRnnn or RGBnnn).
-inetd	When **Xvnc** is launched on demand by **inetd**.
-alwaysshared	Always treat new clients as shared.
-nevershared	Never treat new clients as shared.
-dontdisconnect	Do not disconnect existing clients when a new *non-shared* connection comes in. Instead the new connection is refused.
-localhost	Only allow connections from the same machine.
-cc n	Set the color Visual class used by the server.
-economictranslate	The server normally uses a lookup table for translating pixel values when the viewer requests a different format from the native one used by the server.
-deferupdate n	**Xvnc** now uses a *deferred update* mechanism, which enhances performance in many cases.

-kill :*number* Terminate the *number* vncserver.

Toolbox FTP archive Web site

ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/vnc

Tool developer Web site

http://www.uk.research.att.com/vnc

Tool FTP archive Web site

http://www.uk.research.att.com/vnc/download.html

Files

```
/opt/freeware/bin/Xvnc
/opt/freeware/bin/vncpasswd
/opt/freeware/bin/vncserver
/opt/freeware/bin/vncviewer
/opt/freeware/doc/vnc-3.3.3r1
/opt/freeware/doc/vnc-3.3.3r1/LICENCE.TXT
/opt/freeware/doc/vnc-3.3.3r1/README
/opt/freeware/share/vnc/classes
/opt/freeware/share/vnc/classes/DesCipher.class
/opt/freeware/share/vnc/classes/animatedMemoryImageSource.class
/opt/freeware/share/vnc/classes/authenticationPanel.class
/opt/freeware/share/vnc/classes/clipboardFrame.class
/opt/freeware/share/vnc/classes/index.vnc
/opt/freeware/share/vnc/classes/optionsFrame.class
/opt/freeware/share/vnc/classes/rfbProto.class
/opt/freeware/share/vnc/classes/shared.vnc
/opt/freeware/share/vnc/classes/vncCanvas.class
/opt/freeware/share/vnc/classes/vncviewer.class
/opt/freeware/share/vnc/classes/vncviewer.jar
/opt/freeware/vnc
/usr/lpp/X11/bin/Xvnc
/usr/lpp/X11/bin/vncpasswd
/usr/lpp/X11/bin/vncserver
/usr/lpp/X11/bin/vncviewer
```

Prereqs

The VNC package does not have any prerequisites.

Examples

To use the VNC tools, you must start a **Xvnc** server on the system that you will connect to from your Web browser. This is done with the **vncserver** command:

```
vncserver
```

The next example shows how to start **vncserver** manually on the command line. Note that the highlighted text is what we type, and the asterisks (*) are substituted for the password that we use; however, your password will not be echoed back to your screen after typing. Password entry is required the first time the **vncserver** command is executed. It also creates a ~/.vnc directory containing specific configuration files. Example 6-44 starts **Xvnc** on a UNIX/AIX system.

Example 6-44 Manually starting vncserver

```
root@fenris:/: vncserver

You will require a password to access your desktops.

Password: ******
Verify: ******

New 'X' desktop is fenris:1

Creating default startup script //.vnc/xstartup
Starting applications specified in //.vnc/xstartup
Log file is //.vnc/fenris:1.log
```

Now all we have to do is start our Java-capable Web browser and point to the `http://fenris:5801` Web site (or the IP address). In Example 6-44, the **vncserver** uses X desktop 1 on the host fenris. You add the X desktop number to the port number, by default 5800, and connect to that port with the Web browser. With the example above it will be 5801. Instead of a Web browser you could also use the **vncviewer** command that is available on many different operating systems.

To terminate the **vncserver** that was started in Figure 6-44, use the -kill :# option with the **vncserver** command (where # is the VNC X desktop number):

```
vncserver -kill :1
```

For additional examples on the usage of VNC, please refer to 5.2.2, “Using VNC and putty for easier installation” on page 126.

6.5.4 php

PHP provides an HTML-embedded scripting language. PHP offers built-in database integration for several commercial and non-commercial database management systems, so writing a database-enabled script with PHP is fairly simple.

The most common use of PHP coding is probably as a replacement for CGI scripts, but it is also possible to use the PHP language with the **php** interpreter.

The interpreter is created when the **configure** step of the RPM build process is run *without* specifying any target Web server. The **php** command can be used as a PHP script interpreter.

Toolbox FTP archive Web site

ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/php

Tool developer Web site

http://www.php.net

Tool FTP archive Web site

http://www.php.net/downloads.php

Files

```
/etc/opt/freeware/apache/php.ini
/opt/freeware/apache/libexec/libphp4.so
/opt/freeware/doc/php-4.0.6
/opt/freeware/doc/php-4.0.6/CODING_STANDARDS
/opt/freeware/doc/php-4.0.6/CREDITS
/opt/freeware/doc/php-4.0.6/INSTALL
/opt/freeware/doc/php-4.0.6/LICENSE
/opt/freeware/doc/php-4.0.6/NEWS
/opt/freeware/doc/php-4.0.6/README.CVS-RULES
/opt/freeware/doc/php-4.0.6/README.EXT_SKEL
/opt/freeware/doc/php-4.0.6/README.QNX
/opt/freeware/doc/php-4.0.6/README.SELF-CONTAINED-EXTENSIONS
/opt/freeware/doc/php-4.0.6/README.STREAMS
/opt/freeware/doc/php-4.0.6/README.Zeus
/opt/freeware/doc/php-4.0.6/ZEND_CHANGES
/opt/freeware/doc/php-4.0.6/ZEND_LICENSE
```

Prereqs

The PHP package requires that the *apache* Web server is installed and that the file /etc/opt/freeware/apache/httpd.conf exists.

6.5.5 sudo

The SUDO package provides the **sudo** (SUperuser DO) command that allows a system administrator to give certain users, or groups of users, the ability to run some (or all) commands as root while logging all commands and arguments. The **sudo** command operates on a per-command basis, but it is not a replacement for the shell. Some of the features are the ability to restrict what commands a user may run on a per-host basis, logging of each command (providing a clear audit trail of who did what), a configurable timeout of the **sudo** command, and the ability to use the same configuration file on many different machines.

Synopsis

```
sudo -V | -h | -l | -L | -v | -k | -K | -s | [ -H ] [-P ] [-S ] [ -b ] | [ -p
prompt ] [ -u username|#uid ] command
```

The following list describes options for the **sudo** command:

-V	The -V (version) option causes **sudo** to print the version number and exit.
-l	The -l (list) option will list out the allowed (and forbidden) commands for the user on the current host.
-L	The -L (list defaults) option will list out the parameters that may be set in a Defaults line along with a short description for each.
-h	The -h (help) option causes **sudo** to print a usage message and exit.
-v	If given the -v (validate) option, **sudo** will update the user's time stamp, prompting for the user's password if necessary. This extends the **sudo** timeout for another 5 minutes (or whatever the timeout is set to in sudoers) but does not run a command.
-k	The -k (kill) option to **sudo** invalidates the user's time stamp by setting the time on it to the epoch. The next time **sudo** is run a password will be required.
-K	The -K (sure kill) option to **sudo** removes the user's time stamp entirely. This option does not require a password.
-b	The -b (background) option tells **sudo** to run the given command in the background. Note that if you use the -b option you cannot use shell job control to manipulate the process.
-p prompt	The -p (prompt) option allows you to override the default password prompt and use a custom one. If the password prompt contains the %u escape, %u will be replaced with the user's login name. Similarly, %h will be replaced with the local host name.
-u usernamel#uid	The -u (user) option causes **sudo** to run the specified command as a user other than root. To specify a UID instead of a user name, use #uid.
-s	The -s (shell) option runs the shell specified by the SHELL environment variable if it is set or the shell as specified in /etc/passwd.
-H	The -H (HOME) option sets the HOME environment variable to the homedir of the target user (root by default)

	as specified in /etc/passwd. By default, **sudo** does not modify HOME.
-P	The -P (preserve group vector) option causes **sudo** to preserve the user's group vector unaltered. By default, **sudo** will initialize the group vector to the list of groups the target user is in. The real and effective group IDs, however, are still set to match the target user.
-S	The -S (stdin) option causes **sudo** to read the password from standard input instead of the terminal device.
--	The **--** flag indicates that **sudo** should stop processing command line arguments. It is most useful in conjunction with the -s flag.

Toolbox FTP archive Web site

ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/sudo

Tool developer Web site

http://www.courtesan.com/sudo

Tool FTP archive Web site

http://www.courtesan.com/sudo/dist/sudo*

Files

```
/etc/sudoers
/opt/freeware/bin/sudo
/opt/freeware/doc/sudo-1.6.5p2/
...(files omitted)...
/opt/freeware/man/man5/sudoers.5
/opt/freeware/man/man8/sudo.8
/opt/freeware/man/man8/visudo.8
/opt/freeware/sbin/visudo
/usr/bin/sudo
/usr/sbin/visudo
/var/run/sudo
```

Prereqs

The SUDO package does not have any prerequisites.

Examples

Before you use **sudo**, you should edit the /etc/sudoers file with the **visudo** editor. Just type `visudo` on the command line:

```
visudo
```

The /etc/sudoers file can be extensively customized, so please refer to the man pages to set it up appropriately for your environment. Note that the file /etc/sudoers should have the -r--r----- (0660) permissions, and be owned by the root user and the systems group.

The following is a very simple example of using **sudo**.

```
sudo ls -lR /etc/security
```

If the **sudo** authentication failed (the users executing **sudo** are asked to verify who they are with their own password), the following message will be displayed by default and a notification e-mail be sent to the root user (also by default). The user that was trying to use **sudo** was *nobody*:

```
nobody is not in the sudoers file.  This incident will be reported.
```

However, if the user nobody is allowed to perform some **sudo** operations, but not the one that the user tried to do, a similar output to what is shown below will be displayed, but no e-mail will be sent (by default). In this case the user nobody tried to display the content of the /etc/security/passwd file:

```
Sorry, user nobody is not allowed to execute '/usr/bin/cat
/etc/security/passwd' as root on fenris.
```

To enable the nobody user to use the **cat** command, everywhere, use the following line when editing the /etc/sudoer file with the **visudo** editor:

```
nobody  ALL = (root) /usr/bin/cat
```

6.6 Login shells

A shell is an interactive command interpreter and command programming language that understands and executes the commands a user enters.

We cover the following tools in this section:

- **bash**
- **tcsh**
- **zsh**

We also provide a general comparison between the AIX shells and the shells provided by the Toolbox. You will find examples of how to use the Toolbox shells in this section.

6.6.1 AIX shells

Shells provide a way for you to communicate with the operating system. This communication is carried out either interactively (input from the keyboard is acted upon immediately) or as a shell script. A shell script is a sequence of shell and operating system commands that is stored in a file.

Please refer to the *Shells* section of the *AIX 5L Version 5.2 System User's Guide: Operating System and Devices,* for more in-depth information on shells.

AIX provides the following shells to use both interactively and with shell scripts:

ksh, **ksh93, and psh** The Korn shell is an interactive command interpreter and command programming language. The Korn shell offers many of the same features as the Bourne and C shells, such as I/O redirection capabilities, variable substitution, and file name substitution. It also includes several additional command and programming language features. The `ksh` command invokes the Korn shell.

The `psh` command invokes the POSIX shell, which is the same as the Korn shell.

The enhanced version of the Korn shell is based on the 93 standard. The original `ksh` command is based on the 88 standard. Some of the enhancements are: Associative arrays, compound variables and assignments, variable name references, and additional parameter-expansion constructs. The `ksh93` command invokes the enhanced Korn shell.

bsh The Bourne shell is an interactive command interpreter and command programming language. It can be run as a login shell or as a subshell under the login shell. The `bsh` command invokes the Bourne shell.

csh The C shell is an interactive command interpreter and a command programming language. It uses a syntax that is similar to the C programming language. The `csh` command invokes the C shell.

Rsh The restricted shell is used to set up login names and execution environments whose capabilities must be more controlled than those of the regular Bourne shell. The behavior of the restricted shell is identical to the `bsh` command, except that some actions are *not* allowed. The `Rsh` or `bsh -r` command invokes the restricted shell.

tsh The trusted shell is a command interpreter that provides greater security than the Korn shell. Generally, a user calls the trusted shell by using the secure attention key (SAK) sequence, which is **CTRL-X** followed by **CTRL-R**, after a login. The trusted shell also can be invoked by defining it as the login shell in the /etc/passwd file. The trusted shell differs from the Korn shell in some ways. The `tsh` command invokes the trusted shell.

The default shell, `/usr/bin/sh` (or `/bin/sh`), is linked to `ksh` in AIX.

Table 6-1 AIX standard shells feature comparison

Feature	bsh	csh	ksh
Compatible with bsh	N/A	No	Yes
Job control	Yes	Yes	Yes
Command history	No	Yes	Yes
Command-line editing	No	Yes	Yes
Aliases	No	Yes	Yes
noclobber (protecting files from editing)	No	Yes	Yes
ignoreeof (ignore control-D)	No	Yes	Yes
Logout file	No	Yes	No

Environment setting during the login sequence

When we log in, the shell defines the user environment after reading the shell startup files. During the login process, the general characteristics of the user environment are defined by the values given to the environment variables; this environment is kept until the user logs off the system.

Login execution sequence

Regardless of what shell we are running, the /etc/environment and /etc/security/environ files are always read prior to the startup of the shell by the log-in process. Table 6-2 on page 273 and Table 6-48 on page 275 display the order in which the login execution sequence takes place.

The /etc/environment file sets up the user environment, such as the minimal search path, time zone, and language. This file is not a shell script type file and the only data format that it accepts is *variablename=value*.

The /etc/security/environ file is an ASCII file that can contain stanzas with environment attributes for each individual user. Each stanza is identified by a user name and accepts the format *attribute name*=*value*.

The user stanza in the /etc/security/environ file can have the following attributes:

usrenv Defines environment variables (separated by commas) to be placed in the user environment at login time.

sysenv Defines environment variables to be placed in the user-protected state environment at login time. These variables are protected from access by unprivileged programs.

Table 6-2 shows how the environment setting, specific to standard AIX shells, is performed.

Table 6-2 Login execution sequence for ksh, csh, and bsh

Korn shell	**C shell**	**Bourne shell**
/etc/environment	/etc/environment	/etc/environment
/etc/security/environ	/etc/security/environ	/etc/security/environ
/etc/profile	/etc/csh.cshrc	/etc/profile
$HOME/.profile	/etc/csh.login	$HOME/.profile
$HOME/.kshrc	$HOME/.cshrc	
	$HOME/.login	

The following are sample startup files for the Korn shell. The .profile points to .kshrc file. For Korn shell, only the file pointed to by the ENV variable will be read during each start of a new shell. Do not put variables that append the same variable to the end of the new value; it will accumulate since the current shell will inherit values from the parent shell.

In the sample .profile file (Example 6-45), we set the PATH and MANPATH search variables.

Example 6-45 Sample ksh .profile file

```
ENV=~/.kshrc
export ENV
export PATH=/usr/linux/bin:$PATH
export MANPATH=/opt/freeware/man:$MANPATH
```

The following sample .kshrc file (Example 6-46), will set both command line and shell editor to the powerful **vi** editing mode and the command-line prompt to:

```
user@host:/currentdirectory:
```

Example 6-46 Sample ksh .kshrc file

```
export PS1="$LOGNAME@$(/usr/bin/hostname -s):\$PWD: "
export EDITOR=vi
export VISUAL=$EDITOR
```

Emacs command-line editing in Korn shell

If you must use emacs style command-line editing and the arrow keys to navigate on the command line, you need to alias some key sequences. The Korn shell commands that enable this are shown in Figure 6-47.

You can append the aliases and export of the VISUAL variable in the file pointed to by your ENV variable (usually exported from the ~/.profile file to point to ~/.kshrc):

```
export ENV=$HOME/.kshrc
```

You can also load a script file containing the aliases and the VISUAL variable, with the load operator in the Korn shell (the dot):

```
. ~/emacs_keys
```

Example 6-47 emacs command line cursor keys in Korn shell

```
alias -x __A=`echo "\020"` # up arrow    ^p  (back a command)
alias -x __B=`echo "\016"` # down arrow  ^n  (down a command)
alias -x __C=`echo "\006"` # right arrow ^   (forward a character)
alias -x __D=`echo "\002"` # left arrow  ^b  (back a character)
alias -x __H=`echo "\001"` # home        ^a  (start of line)
export VISUAL=emacs
```

The Toolbox shells bash, tcsh, and zsh use emacs style command-line editing by default.

6.6.2 Toolbox shells

The AIX Toolbox for Linux Applications introduces new shells for AIX. These new shells are common in the Linux user community. The new shells are **bash**, **tcsh**, and **zsh**. All these shells have the following major features in common:

- Command history
- Command aliasing
- Shell scripting

- File name completion
- Command-line editing
- Job control

Example 6-48 shows how the environment setting, specific to Toolbox shells, is performed.

Example 6-48 Login execution sequence for bash, tcsh, and zsh

Bash shell	**Tcsh shell**	**Z shell**
/etc/environment	/etc/environment	/etc/environment
/etc/security/environ	/etc/security/environ	/etc/security/environ
/etc/profile	/etc/csh.cshrc	/etc/zshenv
$HOME/.bash_profile	/etc/csh.login	$HOME/.zshenv
$HOME/.bash_login	$HOME/.tcshrc	/etc/zprofile
$HOME/.profile	($HOME/.cshrc)	$HOME/.zprofile
($HOME/.bashrc)	$HOME/.history	/etc/zshrc
	$HOME/.login	$HOME/.zshrc
	$HOME/.cshdirs	/etc/zlogin
		$HOME/zlogin

If you want to use any of the Toolbox shells as your default login shell, you need to update the /etc/security/login.cfg and /etc/passwd.

To update /etc/security/login.cfg use the **chsec** and **lssec** commands. The next example shows how to do it, note that the command should be entered as one command line or save to a script file (Korn shell):

```
chsec -f /etc/security/login.cfg -s usw -a shells=$(lssec -f
/etc/security/login.cfg -s usw -a shells|cut -f2 -d=),/usr/bin/bash
```

In the example above we use the **lssec** command to extract the current setting for the shells attribute in the usw stanza from the /etc/security/login.cfg file. This information is needed since we want to add **/usr/bin/bash** to this list, *not replace* the list with **/usr/bin/bash**. In case you have weak nerves, copy the /etc/security/login.cfg file to /etc/security/login.cfg~1 before using the **chsec** command.

After you have executed the **chsec** command, use the **lssec** command to verify that the shells attribute is correct, and if so, then remove the /etc/security/login.cfg~1 file since it is not needed anymore:

```
lssec -f /etc/security/login.cfg -s usw -a shells
```

Since the list of allowed shells has been updated with the **/usr/bin/bash** command, we can now update the login shell for a user account in the /etc/passwd file; use the **chuser** command. In the following example we change the login shell to bash for the groda user:

```
chuser shell=/usr/bin/bash groda
```

If the list of allowed shells is not updated properly, you will get a similar error message to the one shown in Example 6-50 on page 278.

Example 6-49 chuser error message

```
root@fenris:/: chuser shell=/usr/bin/bash groda
3004-703 Check "/etc/security/login.cfg" file.
3004-692 Error changing "shell" to "/usr/bin/bash" : Value is invalid.
```

6.6.3 bash

The BASH package provides the Bourne Again SHell (**bash**), which is a shell or command language interpreter that is compatible with the Bourne shell (**sh**). The bash shell incorporates useful features from the Korn shell (**ksh**) and the C shell (**csh**). Most sh scripts can be run by **bash** without modification.

Synopsis

```
bash [options] [file]
```

The following list describes options for the **bash** command:

-c string If the -c option is present, then commands are read from string. If there are arguments after the string, they are assigned to the positional parameters, starting with $0.

-r If the -r option is present, the shell becomes restricted.

-i If the -i option is present, the shell is interactive.

-s If the -s option is present, or if no arguments remain after option processing, then commands are read from the standard input. This option allows the positional parameters to be set when invoking an interactive shell.

-D A list of all double-quoted strings preceded by $ is printed on the standard output. These are the strings that are subject to language translation when the current locale is not C or POSIX. This implies the -n option; no commands will be executed.

[-+]O [shopt_option] shopt_option is one of the shell options accepted by the shopt built-in. If shopt_option is present, -O sets the value of that option; +O unsets it. If shopt_option is not supplied, the names and values of the shell options accepted by shopt are printed on the standard output. If the invocation option is +O, the output is displayed in a format that may be reused as input.

-- A double dash (`--`) signals the end of options and disables further option processing. Any arguments after the `--` are treated as file names and arguments. An argument of `-` is equivalent to `--`.

Toolbox FTP archive Web site

`ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/bash`

Tool developer Web site

`http://www.gnu.org/software/bash/bash.html`

Tool FTP archive Web site

`ftp://ftp.gnu.org/pub/gnu/bash`

Files

```
/bin/bash
/bin/bash2
/opt/freeware/bin/bash
/opt/freeware/bin/bashbug
/opt/freeware/doc/bash2-2.05a/
```

...(files omitted)...

```
/opt/freeware/man/man1/bash.1
/opt/freeware/man/man1/bashbug.1
/usr/bin/bash2bug
/usr/bin/bashbug
```

Prereqs

The BASH package requires the FILEUTILS package.

Features and examples

The **bash** shell is a Bourne shell compatible command language interpreter that executes commands read from the standard input or from a file. **bash** also incorporates useful features from the Korn and C shells (**ksh** and **csh**).

The **bash** shell is intended to be an implementation that conforms to the IEEE POSIX Shell and Tools specification (IEEE Working Group 1003.2). It offers functional improvements over **sh** for both interactive and programming use. **bash** is portable, and currently runs on nearly every version of UNIX and a few other operating systems.

The **bash** shell provides:

Bourne shell style Looping and conditional constructs

C-shell style Job control, history expansion, protected redirection, C shell variables, and tilde expansion

Korn shell style Korn shell constructs, Korn shell built-in, Korn variables and alias built-in

Some unique **bash** built-in commands are:

bind *args* Binds a key sequence to a read line function, or to a macro.

builtin *args* Runs a shell built-in. This is useful when you wish to rename a shell built-in to be a function, but need the functionality of the built-in within the function itself.

command *args* Runs the command specified after the command operator, ignoring shell functions and aliases. If you have a shell command alias called **ls**, and you wish to call the command **ls**, you can use **command ls**, and the **ls** command will be used.

declare *args* Declares variables and/or gives them attributes.

enable *args* Enables and disables built-in shell commands.

help *args* Displays helpful information about built-in commands.

local *args* Creates a local variable. **local** can only be used within a function.

type *args* Checks the command specified after the type operator, and indicates how it would be interpreted if used as a command name. If you want to find out if **ls** is an alias or a file, you can use **type ls**, and **bash** will tell you.

Example 6-50 bash help command usage

```
root@fenris:/: help type
type: type [-apt] name [name ...]
    For each NAME, indicate how it would be interpreted if used as a
    command name.

    If the -t option is used, `type' outputs a single word which is one of
    `alias', `keyword', `function', `builtin', `file' or `', if NAME is an
```

```
alias, shell reserved word, shell function, shell builtin, disk file,
or unfound, respectively.

If the -p flag is used, `type' either returns the name of the disk
file that would be executed, or nothing if `type -t NAME' would not
return `file'.

If the -a flag is used, `type' displays all of the places that contain
an executable named `file'.  This includes aliases and functions, if
and only if the -p flag is not also used.
```

The following sample .bashrc file will set the command line prompt to:

```
user@host:/currentdirectory:
```

Then set the shell command-line mode to the powerful **vi** editing mode. Then we set the PATH and MANPATH search variables (Example 6-51).

Example 6-51 Sample bash .bashrc file

```
export PS1="$LOGNAME@$(/usr/bin/hostname -s):\$PWD: "
export EDITOR=vi
export VISUAL=vi
export PATH=/usr/linux/bin:$PATH
export MANPATH=/opt/freeware/man:$MANPATH
```

For a complete reference of the **bash** shell, please refer to:

http://www.gnu.org/software/bash/bash.html

6.6.4 tcsh

The TCSH package provides the **tcsh** shell, which is an enhanced but completely compatible version of **csh**, the C shell. The **tcsh** shell is a command language interpreter that can be used both as an interactive login shell and as a shell script command processor. The **tcsh** shell includes a command-line editor, programmable word completion, spelling correction, a history mechanism, job control, and a C language-like syntax.

Synopsis

```
tcsh [ -bcdefilmnqstvVxX ] [ argument ... ]
```

The following list describes options for the **tcsh** command:

-b Forces a “break” from option processing, causing any further shell arguments to be treated as non-option arguments. The remaining arguments will not be interpreted as shell options.

-c Commands are read from the following argument (which must be present, and must be a single argument), stored in the command shell variable for reference, and executed. Any remaining arguments are placed in the **argv** shell variable.

-d The shell loads the directory stack from ~/.cshdirs, as described under startup and shutdown, whether or not it is a login shell.

-e The shell exits if any invoked command terminates abnormally or yields a non-zero exit status.

-f The shell ignores ~/.tcshrc, and thus starts faster.

-i The shell is interactive and prompts for its top-level input, even if it appears to not be a terminal. Shells are interactive without this option if their inputs and outputs are terminals.

-l The shell is a login shell. Applicable only if -l is the only flag specified.

-m The shell loads ~/.tcshrc even if it does not belong to the effective user.

-n The shell parses commands but does not execute them.

-q The shell accepts SIGQUIT and behaves when it is used under a debugger. Job control is disabled.

-s Command input is taken from the standard input.

-t The shell reads and executes a single line of input. A backslash (\) may be used to escape the new line at the end of this line and continue onto another line.

-v Sets the **verbose** shell variable, so that command input is echoed after history substitution.

-x Sets the **echo** shell variable, so that commands are echoed immediately before execution.

-V Sets the **verbose** shell variable even before executing ~/.tcshrc.

-X Is to -x as -V is to -v.

Toolbox FTP archive Web site

ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/tcsh

Tool developer Web site

http://www.tcsh.org

Tool FTP archive Web site

ftp://ftp.astron.com/pub/tcsh [US]
ftp://ftp.gw.com/pub/unix/tcsh [US]
ftp://ftp.funet.fi/pub/unix/shells/tcsh [Finland]

Files

```
/opt/freeware/bin/csh
/opt/freeware/bin/tcsh
```

```
/opt/freeware/doc/tcsh-6.11
/opt/freeware/doc/tcsh-6.11/FAQ
/opt/freeware/doc/tcsh-6.11/Fixes
/opt/freeware/doc/tcsh-6.11/NewThings
/opt/freeware/doc/tcsh-6.11/README
/opt/freeware/doc/tcsh-6.11/WishList
/opt/freeware/doc/tcsh-6.11/Y2K
/opt/freeware/doc/tcsh-6.11/complete.tcsh
/opt/freeware/doc/tcsh-6.11/csh-mode.el
/opt/freeware/doc/tcsh-6.11/eight-bit.txt
/opt/freeware/doc/tcsh-6.11/tcsh.html
...(files omitted)...
/opt/freeware/man/man1/tcsh.1
/usr/bin/tcsh
/usr/linux/bin/csh
```

Prereqs

The TCSH package requires the FILEUTILS package.

Features and examples

tcsh is an enhanced but completely compatible version of the Berkeley UNIX C shell (**csh**). It is a command language interpreter usable both as an interactive login shell and a shell script command processor. It includes a command line editor, programmable word completion, spelling correction, a history mechanism, a job control, and a C-like syntax.

Key features of the **tcsh** shell are spelling correction, command completion, and command-line editing.

Spelling correction

The shell can correct the spelling of file names, commands, and variable names, as well as complete and list them.

Individual words can have their spellings corrected with the **spell-word editor** command (usually bound to CTRL-s and CTRL-S) and the entire input buffer can be corrected with **spell-line** (usually bound to CTRL-$). To learn how your keys are set up, run the command **bindkey -b**.

The correct shell variable can be set to *cmd* to correct the command name or to *all* to correct the entire line each time return is typed. Autocorrect can be set to correct the word to be completed before each completion attempt. Example 6-52 on page 282 shows how to set the spelling function and the output result. The highlighted text is what we type.

Example 6-52 Use of tcsh spelling correction capability

```
root@fenris: set correct=cmd

root@fenris: lx d* (Type ENTER key)

CORRECT>lex d* (y|n|e|a)? edit
root@fenris: ls d*
```

Completion and listing

The shell is often able to complete words when given a unique abbreviation. Type part of a word (for example, **`ls /usr/lost`**) and press the Tab key to run the **`complete-word editor`** command.

The shell completes the file name /usr/lost to /usr/lost+found/, replacing the incomplete word with the complete word in the input buffer. Note the terminal `/`; completion adds a `/` to the end of completed directories and a space to the end of other completed words to speed typing and provide a visual indicator of successful completion.

The **`addsuffix`** shell variable can be unset to prevent this. If no match is found (perhaps /usr/lost+found does not exist), the terminal bell rings. If the word is already complete (perhaps there is a /usr/lost on your system, or perhaps you were thinking too far ahead and typed the whole thing), a `/` or space is added to the end if it is not already there.

Completion works anywhere in the line, not just at the end; completed text pushes the rest of the line to the right. Completion in the middle of a word often results in leftover characters to the right of the cursor, which need to be deleted.

In Example 6-53 we use the CTRL-D key sequence to let **`tcsh`** do the word completion. After typing `CTRL-D`, **`tcsh`** returns with the same command on the command line, should we want to refine our file name search pattern and execute the command again. After the first **`ls`** command, and completion by using CTRL-D, we add a `u` and perform the completion a second time. The highlighted text is what we type.

Example 6-53 Using CTRL-D to complete a command line

```
root@fenris: ls -l d (Type CTRL-D)
depcomp@ ourcode*    ourcode.8*  ourcode.c*  ourcode.h*  ourcode.o
root@fenris: ls -l du (Type CTRL-D)
ourcode*   ourcode.8* ourcode.c* ourcode.h* ourcode.o
```

Example 6-54 on page 283 uses the Tab key for command completion. By typing `/usr/t` and then the Tab key, **`tcsh`** expands /usr/t to /usr/tmp and we can continue typing.

Note, however, that if there are several files or directories that match the typed amount of characters, **tcsh** will not perform command completion with the Tab key. To find out why we get stuck without completion, we can use the CTRL-D key sequence, and then continue typing. Example 6-54 shows the combination of using the Tab and CTRL-D key sequences; the highlighted text is what we type.

Example 6-54 Use of Tab and CTRL-D to complete a command line

```
root@fenris: ls d (Type TAB and nothing happens)
root@fenris: ls d (Type CTRL-D)
depcomp@ ourcode*    ourcode.8*  ourcode.c*  ourcode.h*  ourcode.o
root@fenris: ls de (Type TAB)
root@fenris: ls depcomp
```

Command-line editing

Command-line input can be edited using key sequences much like those used in GNU **emacs** or **vi**. The editor is active only when the edit shell variable is set, which it is by default in interactive shells. The **bindkey** built-in can display and change key bindings. Emacs-style key bindings are used by default (unless the shell was compiled otherwise; see the version shell variable), but **bindkey** can change the key bindings to **vi** style bindings all at once.

The following sample .tcshrc file, will set the command-line prompt to:

```
user@host:
```

Then set the shell command line mode to the powerful vi editing mode, then we set the PATH and MANPATH search variables.

Example 6-55 Sample tcsh .tcshrc file

```
set prompt = ( "$LOGNAME@`hostname -s`: " )
bindkey -v
setenv PATH /usr/linux/bin:${PATH}::
setenv MANPATH /opt/freeware/man:${MANPATH}
```

For a complete reference of the **tcsh** shell, please refer to:

http://howto.tucows.com/man/man1/tcsh.1.html

6.6.5 zsh

The ZSH package provides the **zsh** shell command interpreter usable as an interactive login shell and as a shell script command processor. The **zsh** shell resembles the **ksh** shell (the Korn shell), but includes many enhancements. The **zsh** shell supports command-line editing, built-in spelling correction, programmable command completion, shell functions (with autoloading), a history mechanism, and more.

Synopsis

```
zsh [options] [argument ...]
```

The following list describes options for the **zsh** command:

-c Take the first argument as a command to execute, rather than reading commands from a script or standard input. If any further arguments are given, the first one is assigned to $0, rather than be used as a positional parameter.

-i Force shell to be interactive.

-s Force shell to read commands from the standard input. If the -s flag is not present and an argument is given, the first argument is taken to be the path name of a script to execute.

Toolbox FTP archive Web site

ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/zsh

Tool developer Web site

http://www.zsh.org

Tool FTP archive Web site

ftp://ftp.zsh.org/pub

Files

/opt/freeware/bin/zsh
/opt/freeware/bin/zsh-4.0.4/
...(files omitted)...
/opt/freeware/info/zsh.info*.gz
...(files omitted)...
/opt/freeware/lib/zsh
/opt/freeware/lib/zsh/4.0.4
/opt/freeware/lib/zsh/4.0.4/zsh/
...(files omitted)...
/opt/freeware/man/man1/
...(files omitted)...
/opt/freeware/share/zsh
/opt/freeware/share/zsh/4.0.4
/opt/freeware/share/zsh/4.0.4/functions
...(files omitted)...
/opt/freeware/share/zsh/site-functions
/usr/bin/zsh
/usr/bin/zsh-4.0.4
/usr/lib/zsh
/usr/share/zsh

Prereqs

The ZSH package requires the FILEUTILS package.

Features and examples

The **zsh** is a UNIX command interpreter (shell), usable as an interactive login shell and as a shell script command processor. Of the standard shells, **zsh** most closely resembles **ksh**, but includes many enhancements. The **zsh** has command-line editing, built-in spelling correction, programmable command completion, shell functions, and a history mechanism.

Some of the key features of the **zsh** shell are:

Command-line editing Programmable completion, which incorporates the ability to use the power of **zsh** globbing. Multi-line commands editable as a single buffer, variable editing, command buffer stack, and in-line expansion of variables and history commands.

Globbing Globbing is a very powerful feature and includes recursive globbing, file attribute qualifiers, full alternation, and negation of patterns.

Redirections Handling of multiple redirections (simpler than **tee**).

Path expansion File name completion of complete words when given a unique abbreviation.

Spelling correction Correct the spelling of file names, commands, and variable names.

The following examples illustrate the use of globbing. First we use the **setopt extendedglob** to enable globbing. Experiment by using the Tab key in combination with part of a file name that you know exists. You can let **zsh** display the files it matches by continue to uses the Tab key until you find the file you were looking for. Example 6-56 lists all files in the current directory, except for files ending with .o.

Example 6-56 Using zsh with globbing

```
root@fenris: setopt extendedglob

root@fenris: ls ^*.o
AUTHORS       NEWS            config.h.in~    ourcode          stamp-h
COPYING       README          config.log      ourcode.8        stamp-h.in
ChangeLog     aclocal.m4      config.status   ourcode.c        stamp-h1.in
INSTALL       autom4te.cache  configure       ourcode.h        stamp-h2.in
Makefile      autoscan.log    configure.in    install-sh
Makefile.am   config.h        configure.scan  missing
Makefile.in   config.h.in     depcomp         mkinstalldirs
```

Example 6-57 lists only files ending with .c or .h.

Example 6-57 Using zsh with grouping

```
root@fenris: ls *(.c|.h)
config.h  ourcode.c    ourcode.h
```

Example 6-58 lists only files that have the SETUID bit set in the permissions for the file (s).

Example 6-58 Using zsh to find setuid files

```
root@fenris: ls -l *(s)
-rwsr-sr-x   1 root     system       242382 Oct 10 14:05 ourcode
```

Example 6-59 lists only files that have the SETGID bit set in the permissions for the file (S), and we did not have any file with the SETGID bit set.

Example 6-59 Using zsh to find setgid files

```
root@fenris: ls -l *(S)
zsh: no matches found: *(S)
```

The following sample .zshrc file will set the command line prompt to:

```
user@host:
```

Then set the shell command line mode to the powerful vi editing mode and set the PATH and MANPATH search variables.

Example 6-60 Sample zsh .zshrc file

```
PROMPT="$LOGNAME@$(/usr/bin/hostname -s): "
bindkey -v
export PATH=/usr/linux/bin:$PATH
export MANPATH=/opt/freeware/man:$MANPATH
```

For a complete reference for the zsh shell, please refer to:

http://www.zsh.org

Related publications

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

IBM Redbooks

For information on ordering these publications, see, “How to get IBM Redbooks” on page 299.

- *AIX 5L Differences Guide Version 5.2 Edition*, SG24-5765
- *AIX 5L Performance Tools Handbook,* SG24-6039
- *AIX 5L Porting Guide*, SG24-603
- *Building a Linux HPC Cluster with xCAT*, SG24-6623
- *Building Linux Systems Under IBM VM*, REDP0120
- *C and C++ Application Development on AIX*, SG24-5674
- *How to install Red Hat Linux on the xSeries 440*, TIPS0042
- *Implementing IBM LTO in Linux and Windows*, SG24-6268
- *Implementing Linux with IBM Disk Storage*, SG24-6261
- *Linux and Windows Integration for IBM eServer xSeries Servers*, REDP0412
- *Linux Clustering with CSM and GPFS*, SG24-6601
- *Linux for IBM eServer zSeries and S/390: Distributions*, SG24-6264
- *Linux on IBM eServer zSeries and S/390: Cloning Linux Images in z/VM*, REDP0301
- *Linux on IBM eServer zSeries and S/390: Large Scale Linux Deployment*, SG24-6824
- *Linux on IBM zSeries and S/390: Server Consolidation with Linux for zSeries*, REDP0222
- *Linux on IBM eServer zSeries and S/390: Systems Management*, SG24-6820
- *Linux on IBM eServer zSeries: Configuring gcc as a cross-compiler*, TIPS0005
- *Linux on IBM zSeries and S/390: High Availability for z/VM and Linux*, REDP0220

- *Linux System Administration and Backup Tools for IBM eServer xSeries and Netfinity*, SG24-6228
- *Managing AIX Server Farms*, SG24-6606
- *The Complete Partitioning Guide on IBM eServer pSeries Servers*, SG24-7039
- *Tivoli Storage Manager Version 5.1 Technical Guide*, SG24-6554

Other resources

These publications are also relevant as further information sources:

- Edward C. Bailey, *Maximum RPM*, July 1997, Red Hat Press, ISBN 0-67231-105-4. Also found at:

  `http://www.rpm.org/max-rpm`
- AIX Manuals, found at:

  `http://www.ibm.com/servers/aix/library`

Referenced Web sites

These Web sites are also relevant as further information sources:

- AIX Toolbox Crypto Web site

  `http://www6.software.ibm.com/dl/aixtbx/aixtbx-p`
- AIX Toolbox development Web site

  `http://oss.software.ibm.com/developerworks/projects/aixtoolbox`
- AIX Toolbox for Linux Applications Web site

  `http://www.ibm.com/servers/aix/products/aixos/linux`
- Binutils Web site

  `http://sources.redhat.com/binutils`
- Bzip2 Web site

  `http://sources.redhat.com/bzip2`
- Center for the Public Domain

  `ftp://sunsite.unc.edu/pub`
- Cryptographic Content (SSL) for certain Toolbox packages

  `http://www6.software.ibm.com/dl/aixtbx/aixtbx-p`

- CURL Web site

  http://curl.haxx.se
- CVS Web site

  http://www.cvshome.org
- Debian

  http://www.debian.org/ports/powerpc/
- Directory on Toolbox FTP server containing getapp-dev.sh

  ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/ezinstall/ppc
- **destroyRPMS** script from the contrib directory on the Toolbox Web site (to remove all installed RPM packages with their dependencies)

  ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/contrib/destroyRPMS
- Detailed information on the current content of the Toolbox

  http://www.ibm.com/servers/aix/products/aixos/linux/rpmgroups.html
- Differences between Toolbox commands and utilities compared to the same supplied with AIX

  ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/docs/index.html
- Elm Web site

  http://www.instinct.org/elm
- *eServer pSeries 610 Model 6C1 and Model 6E1 User's Guide*, SA38-0598

  http://www.ibm.com/servers/eserver/pseries/library/hardware_docs/p610.html
- Ethereal Web site

  http://www.ethereal.com
- Fetchmail Web site

  http://tuxedo.org/~esr/fetchmail
- Free Standards Group Web site

  http://www.freestandards.org
- Free Software Foundation Web site

  http://www.fsf.org
- Freshmeat

  http://www.freshmeat.net
- Freshrpms

  http://www.freshrpms.net

- FTPCOPY Web site

  `http://www.ohse.de/uwe/ftpcopy/ftpcopy.html`
- FVWM Web site

  `http://fvwm.org`
- GIMP Toolkit Web site

  `http://www.gtk.org`
- **gnugo** source code

  `http://www.gnu.org/software/gnugo/gnugo.html`
- Gnome Display Manager information

  `http://www.gnu.org/directory/gdm.html`
- Gnome Web site

  `http://www.gnome.org`
- GNU

  `http://www.gnu.org/directory`
- GNU Coding Standards Web site

  `http://www.gnu.org/prep/standards_toc.html`
- GNU Project Web site

  `http://www.gnu.org`
- GNU software Web sites

  `http://www.gnu.org/software/autoconf`
  `http://www.gnu.org/software/autoconf/autoconf.html`
  `http://www.gnu.org/software/automake/automake.html`
  `http://www.gnu.org/software/bash/bash.html`
  `http://www.gnu.org/software/diffutils/diffutils.html`
  `http://www.gnu.org/software/fileutils/fileutils.html`
  `http://www.gnu.org/software/findutils/findutils.html`
  `http://www.gnu.org/software/gcc/gcc.html`
  `http://www.gnu.org/software/gdb/gdb.html`
  `http://www.gnu.org/software/libtool`
  `http://www.gnu.org/software/libtool/libtool.html`
  `http://www.gnu.org/software/m4`
  `http://www.gnu.org/software/make/make.html`
  `http://www.gnu.org/software/shellutils`
  `http://www.gnu.org/software/wget/wget.html`
- Groupe Bulls free software Web site

  `http://www.bullfreeware.com`

- Gzip Web site

  http://www.gzip.org
- Hexedit Web site

  http://merd.net/pixel/hexedit.html
- ibiblio Web site

  http://www.ibiblio.org
- IBM AIX Library

  http://www.ibm.com/servers/aix/library
- IBM Developer Kit for Linux: Overview

  http://www.ibm.com/developerworks/java/jdk/linux140/
- IBM developer kit porting

  http://www.ibm.com/developerworks/java/jdk/index.html
- IBM Developer Works Web site for the Toolbox - information regarding the development of the Toolbox, Toolbox problem reporting, and Toolbox mailing list

  http://oss.software.ibm.com/developerworks/projects/aixtoolbox
- *IBM eServer pSeries 690 Availability Best Practices* whitepaper

  http://www.ibm.com/servers/eserver/pseries/hardware/whitepapers/p690_avail.html
- IBM - Java Technology Zone

  http://www.ibm.com/java
- IBM Linux Technology Center - Hardware Models

  http://oss.software.ibm.com/developer/opensource/linux/projects/ppc/models.php
- IBM Operating System Web site

  http://www.ibm.com/servers/aix/os/index.html
- Installer image OOo_1.0.1c_LinuxPPC_installer.tar.gz for OpenOffice Version 1.0.1 download

  ftp://ftp.suse.com/pub/projects/powerpc/openoffice/release-1.0.1/
- **installremoved.sh** script from the contrib directory on the Toolbox Web site (to reinstall the RPM packages that were removed by the **destroyRPMS** script)

  http://ftp.software.ibm.com
- IY15017 fix - search and download

  http://techsupport.services.ibm.com/server/support

- IY15017 fix - version is 27

  ftp://techsupport.services.ibm.com/aix/fixes/v4/X11/X11.base.lib.4.3.3.27.bff

- Java2 JDK 1.3.1 for Linux on pSeries download

  https://www6.software.ibm.com/dl/lxdk/lxdk-p

- KDE Web site

  http://www.kde.org

- Less Web site

  http://www.greenwoodsoftware.com/less

- **libtool** documentation

  http://www.gnu.org/software/libtool/manual.html

- Licenses associated with the various packages are available for viewing on the Toolbox CD and on the Toolbox Web site

  http://www.ibm.com/servers/aix/products/aixos/linux/altlic.html

- Linux at IBM general information

  http://www.ibm.com/linux

- Linux for IBM eServer pSeries

  http://www.ibm.com/eserver/pseries/linux

- Linux for IBM eServer pSeries - An overview for customers

  http://www.ibm.com/servers/eserver/pseries/linux/whitepapers/linux_pseries.html

- *Linux for IBM eServer pSeries* whitepaper

  http://www.ibm.com/servers/eserver/pseries/linux/whitepapers/linux_pseries.html

- Linux for pSeries system guide

  http://www.ibm.com/servers/eserver/pseries/linux/guide.html

- Linux kernel Web site

  http://www.kernel.org

- Linuxlinks

  http://www.linuxlinks.com/Software

- Linux on IBM eServers information

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

- Linux on Older Power, RS64 II, and MCA (Microchannel) based systems information

  `http://www.sjdjweis.com/linux/rs6k/`
- Linux on PowerPC Web site

  `http://linuxppc.org`
- Linux Standard Base Web site

  `http://www.linuxbase.org`
- LSB certification

  `http://www.opengroup.org/lsb/cert`
- Lynx Web site

  `http://lynx.isc.org`
- MUTT Web site

  `http://www.mutt.org`
- **`ncftp`** tool's corresponding RPM package download

  `ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/ncftp`
- NCFTP Web site

  `http://www.ncftp.com/ncftp`
- On AIX 5L the SSH packages are now part of the AIX Bonus Pack CD-ROM distribution - additional information

  `http://www.ibm.com/servers/aix/products/bonuspack/aix51/bpcontent.html`
- OpenOffice.org PPC Linux download sites

  `http://whiteboard.openoffice.org/mirrors/ppclinks.html`
- OpenOffice running under Linux for pSeries (most current information)

  `http://www.openoffice.org/`
- OSI Open Source Web site

  `http://www.opensource.org`
- Overview of what pSeries systems are currently supported by which of IBM's Linux distribution partners

  `http://www.ibm.com/servers/eserver/pseries/linux/guide.html`
- Penguin ppc - Hardware Compatability List

  `http://penguinppc.org/projects/hw/`
- Penguin ppc64 port

  `http://penguinppc64.org/`

- Penguin ppc64 Web site

  http://penguinppc.org/
- PHP Web site

  http://www.php.net
- Pine Web site

  http://www.washington.edu/pine
- PKWARE Web site

  http://www.pkware.com
- PROFTPD Web site

  http://www.proftpd.org
- pSeries Linux Web site

  http://www.ibm.com/eserver/pseries/linux
- putty.exe download

  http://www.chiark.greenend.org.uk/~sgtatham/putty/
- Red Hat

  http://www.redhat.com/software/eserver/pseries/
- Red Hat Package Manager Web site

  http://www.rpm.org
- References to several currently available Linux for pSeries distributions

  http://penguinppc.org/
  http://penguinppc64.org/
- RDIST Web site

  http://www.magnicomp.com/rdist
- Rpmfind

  http://www.rpmfind.net/linux/RPM
- RPMfind searchpage Web site

  http://rpmfind.net/linux/rpm2html/search.php
- RPMfind Web site

  http://rpmfind.net
- RPM HOWTO Web site

  http://www.rpm.org/RPM-HOWTO/index.html
- RSYNC Web site

  http://www.samba.org/rsync

- RUNNING.txt document for instructions how to download, install, and run Tomcat

  `http://jakarta.apache.org/tomcat/tomcat-4.0-doc/`
- Samba Web site

  `http://www.samba.org`
- SLES 7 for iSeries and pSeries

  `http://www.suse.com/us/business/products/sles/sles_iSeries_pSeries/`
- SourceForge

  `http://sourceforge.net/softwaremap`
- SUDO Web site

  `http://www.courtesan.com/sudo`
- SuSE

  `http://www.suse.com/us/private/products/suse_linux/ppc/`
- Tcsh Web site

  `http://www.tcsh.org`
- The Free Software Definition

  `http://www.gnu.org/philosophy/free-sw.html`
- The latest stable version of Samba

  `http://www.samba.org/samba/ftp/samba-latest.tar.gz`
- The Open Group Web site

  `http://www.opengroup.org`
- The Open Group - Consortia Services

  `http://www.opengroup.org/certification/index.htm`
- The Open Group - The Single UNIX Specification, Version 3

  `http://www.unix-systems.org/version3`
- The Open Source definition

  `http://www.opensource.org/docs/definition_plain.html`
- Tomcat Web site

  `http://jakarta.apache.org/tomcat/`
- Toolbox documentation on API differences

  `ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/docs/index.html`
- Toolbox FTP site

  `ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox`

- Toolbox FTP archive Web sites

  ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/ethereal
  ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/ftpcopy
  ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/ncftp
  ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/wget
  ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/rdist
  ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/lynx
  ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/curl
  ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/elm
  ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/fetchmail
  ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/elm
  ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/fetchmail
  ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/mutt
  ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/proftpd
  ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/pine
  ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/wu-ftpd
  ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/zip
  ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/unzip
  ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/gzip
  ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/bzip2
  ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/zoo
  ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/lsof
  ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/less
  ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/findutils
  ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/fileutils
  ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/diffutils
  ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/hexedit
  ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/sh-utils
  ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/libtool
  ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/autoconf
  ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/automake
  ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/rcs
  ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/patch
  ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/GNUPro
  ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/m4
  ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/make
  ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/cvs
  ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/texinfo
  ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/samba
  ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/webmin
  ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/vnc
  ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/php
  ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/sudo

ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/bash
ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/tcsh
ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/zsh

- Toolbox FTP site ezinstall directory

  ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/ezinstall

- Toolbox packages downloads

  http://www.ibm.com/servers/aix/products/aixos/linux/download.html

- Toolbox Web site

  http://www.ibm.com/servers/aix/products/aixos/linux

- Tool FTP archive Web sites

  ftp://ftp.ethereal.com/pub/ethereal
  ftp://rsync.samba.org/pub/rsync
  ftp://www.ohse.de/uwe/ftpcopy
  ftp://ftp.ncftp.com/ncftp
  ftp://ftp.gnu.org/gnu/wget
  http://www.magnicomp.com/cgi-bin/mcdownload.cgi
  http://lynx.isc.org/current
  http://curl.haxx.se/download
  ftp://ftp.virginia.edu/pub/elm
  http://tuxedo.org/~esr/fetchmail/fetchmail-*.tar.gz
  ftp://ftp.virginia.edu/pub/elm
  ftp://ftp.mutt.org/mutt
  ftp://ftp.proftpd.org/distrib/source
  ftp://ftp.cac.washington.edu/pine
  ftp://ftp.wu-ftpd.org/pub
  ftp://ftp.uu.net/pub/archiving/zip/src
  ftp://gatekeeper.dec.com/pub/GNU/gzip
  ftp://sources.redhat.com/pub/bzip2
  ftp://sunsite.unc.edu/pub/Linux/utils/compress
  ftp://vic.cc.purdue.edu/pub/tools/unix/lsof
  http://www.greenwoodsoftware.com/less/lessversion-tar.gz
  ftp://ftp.gnu.org/pub/gnu/findutils
  ftp://ftp.gnu.org/pub/gnu/fileutils
  ftp://ftp.gnu.org/pub/gnu/diffutils
  http://merd.net/pixel/hexedit.html
  ftp://ftp.gnu.org/pub/gnu/sh-utils
  ftp://ftp.gnu.org/gnu/libtool
  http://ftp.gnu.org/gnu/autoconf
  http://ftp.gnu.org/gnu/automake
  ftp://ftp.cs.purdue.edu/pub/RCS
  ftp://ftp.gnu.org/pub/gnu/patch
  ftp://ftp.gnu.org/pub/gnu/gcc
  ftp://ftp.gnu.org/gnu/gdb
  ftp://ftp.gnu.org/gnu/binutils
  ftp://ftp.gnu.org/pub/gnu/m4

ftp://ftp.gnu.org/pub/gnu/make
http://ccvs.cvshome.org/servlets/ProjectDownloadList
ftp://ftp.gnu.org/pub/gnu/texinfo
http://dl.sourceforge.net/sourceforge/webadmin
http://www.samba.org/samba/ftp
http://www.uk.research.att.com/vnc/download.html
http://www.php.net/downloads.php
http://www.courtesan.com/sudo/dist/sudo*
ftp://ftp.gnu.org/pub/gnu/bash
ftp://ftp.astron.com/pub/tcsh [US]
ftp://ftp.gw.com/pub/unix/tcsh [US]
ftp://ftp.zsh.org/pub

- TrollTech Web site

  http://www.trolltech.com

- Tucows Linux Programmers Guide Web site

  http://howto.tucows.com/LDP/LDP/lpg/node1.html

- TurboLinux

  http://www.turbolinux.com/products/pseries/

- Tuxfinder

  http://www.tuxfinder.org

- UCLA

  http://aixpdslib.seas.ucla.edu/aixpdslib.html

- UCLA Public Domain Software Library for AIX

  http://aixpdslib.seas.ucla.edu/aixpdslib.html

- UnitedLinux

  http://www.unitedlinux.com/

- University of Cambridge VNC Web site

  http://www.uk.research.att.com/vnc

- Updates on availability of WebSphere Application Server, DB2 Universal Database, several compilers, and many Tivoli products on Linux for pSeries, which IBM is devloping support for

  http://www.ibm.com/eserver/pseries/linux/

- Webmin Web site

  http://www.webmin.com

- **wget** source code from the GNU FTP server

  ftp://ftp.gnu.org/pub/gnu/wget

- **wget** tool's corresponding RPM package download

  ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/wget
- WGET Web site

  http://www.wget.org
- Wu-ftp Web site

  http://www.wu-ftpd.org
- X Window System Web site

  http://www.X.org
- Xwinman Web site

  http://xwinman.org
- Zip/Unzip Web site

  http://www.info-zip.org
- Zsh Web sites

  http://sunsite.dk/zsh
  http://www.zsh.org

How to get IBM Redbooks

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

ibm.com/redbooks

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

IBM Redbooks collections

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

Index

X

Y

Z

IBM

Redbooks

Linux Applications on pSeries

Linux Applications on pSeries

AIX Toolbox for Linux Applications

Recompiling Linux Applications on AIX

Running Linux natively on pSeries

The strengths of the AIX operating system are well known among the UNIX software community. Its reliability and great degree of scaling make AIX the perfect choice for hosting mission-critical applications. It is a robust and flexible operating system that meets all the requirements for the various demands of e-business environments. At the same time, Linux is maturing rapidly and generating excitement among software developers that has not been seen in many years.

This IBM Redbook will show you what you need to do to run Linux applications on pSeries. The AIX Toolbox for Linux Applications contains a large collection of Open Source and GNU software built for AIX 5L and AIX 4.3.3 for IBM eServer pSeries systems and IBM RS/6000. This publication will help you comprehend and install the Toolbox, understand the procedure to follow for porting Open Source Software utilizing the Toolbox, and give a detailed description of a large number of the tools delivered in the Toolbox. It will also describe the procedure to install a native Linux operating system on several pSeries hardware types in detail, including some common Linux applications.

SG24-6033-01 ISBN 0738427950