IBM

IBM Certified Specialist

IBM Certification Study Guide eserver p5 and pSeries Administration and Support for AIX 5L Version 5.3

Developed specifically for the purpose of preparing for AIX 5L certification Test 222

Makes an excellent companion to classroom education

For AIX 5L Version 5.3 professionals

José Eduardo Martínez Cordero
Shiv Dutta
Liviu Rosca
Gilvan Pereira da Silva
Kristof Stroobants

Redbooks

ibm.com/redbooks

International Technical Support Organization

IBM Certification Study Guide eServer p5 and pSeries Administration and Support for AIX 5L Version 5.3

April 2006

SG24-7199-00

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

First Edition (April 2006)

This edition applies to AIX 5L Version 5.3, maintenance level 5300-03, program number 5765-G03 and subsequent releases running on an RS/6000 or pSeries server and is based on information available in November 2005.

© Copyright International Business Machines Corporation 2006. All rights reserved.
Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

© Copyright IBM Corp. 2006. All rights reserved.

Figures

© Copyright IBM Corp. 2006. All rights reserved.

Tables

© Copyright IBM Corp. 2006. All rights reserved.

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.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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

© Copyright IBM Corp. 2006. All rights reserved.

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 5L™
AIX®
Chipkill™
Domino®
Enterprise Storage Server®
eserver®
eserver®
eServer™
Geographically Dispersed Parallel Sysplex™
GDPS®
HACMP™
i5/OS®
IBM®
Lotus®
Micro Channel®
Micro-Partitioning™
Parallel Sysplex®
PowerPC®
POWER™
POWER2™
POWER3™
POWER4™
POWER5™
pSeries®
PTX®
Redbooks™
Redbooks (logo) ™
RS/6000®
System p5™
System Storage™
Tivoli®
xSeries®

The following terms are trademarks of other companies:

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

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

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

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

Preface

The AIX® 5L™ and IBM® eserver pSeries® Certifications offered through the Professional Certification Program from IBM are designed to validate the skills required of technical professionals who work in the powerful and often complex environments of AIX 5L and eserver pSeries. A complete set of professional certifications are available. They include:

- IBM Certified Specialist - eserver p5 and pSeries Administration and Support for AIX 5L V5.3
- IBM Certified Specialist - AIX 5L Basic Operations V5
- IBM Certified Specialist - p5 and pSeries Technical Sales Support
- IBM Certified Specialist - eserver p5 Sales Solutions
- IBM Certified Systems Expert - eserver p5 and pSeries Enterprise Technical Support AIX 5L V5.3
- IBM Certified Systems Expert - IBM System p5™ Virtulization Technical Support AIX 5L V5.3
- IBM Certified Systems Expert - pSeries HACMP™ for AIX 5L
- IBM Certified Advanced Technical Export - IBM System p5 2006

Each certification is developed by following a thorough and rigorous process to ensure the exam is applicable to the job role, and is a meaningful and appropriate assessment of skill. Subject Matter Experts who successfully perform the job participate throughout the entire development process. These job incumbents bring a wealth of experience into the development process, thus making the exams much more meaningful than the typical test, which only captures classroom knowledge. These experienced Subject Matter Experts ensure the exams are relevant to the real world and that the test content is both useful and valid. The result is a certification of value, which appropriately measures the skills required to perform the job role.

This IBM Redbook is designed as a study guide for professionals wishing to prepare for the certification exam to achieve: IBM Certified Specialist - eserver p5 and pSeries Administration and Support for AIX 5L V5.3.

© Copyright IBM Corp. 2006. All rights reserved.

The certification validates a broad scope of AIX 5L administration and support skills and the ability to perform general AIX 5L software system maintenance, including installation. The certification is applicable to AIX 5L administration professionals who conduct the AIX 5L problem determination and resolution activities needed to successfully support customers, or clients, in an AIX 5L environment, to maintain system reliability and support daily management.

This redbook helps AIX 5L administrators seeking a comprehensive and task-oriented guide for developing the knowledge and skills required for the certification. It is designed to provide a combination of theory and practical experience needed for a general understanding of the subject matter.

This redbook does not replace practical experience you should have, but is an effective tool that, when combined with education activities and experience, should prove to be a very useful preparation guide for the exam. Due to the practical nature of the certification content, this redbook can also be used as a desk-side reference. So, whether you are planning to take the IBM Certified Specialist - eserver p5 and pSeries Administration and Support for AIX 5L V5.3 exam, or if you just want to validate your AIX 5L system administration and support skills, this redbook is for you.

For additional information about certification and instructions on How to Register for an exam, visit our Web site at:

http://www.ibm.com/certify

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.

José Eduardo Martínez Cordero, born in 1975, is an IT Specialist in Mexico City. He is a Computer Engineer graduate from Universidad Nacional Autonoma de Mexico, ENEP Aragon, and started his career at IBM on May 1998. He provides technical support for customers on e-Business Hosting Services and recommendations for new proposals. He also has been responsible for planning and performing data center migrations for Strategic Outsourcing customers. His areas of expertise include AIX 5L, Internet services, firewalls, Linux®, security, Tivoli® Storage Manager. He is an IBM Certified Specialist on AIX 5L Communications, IBM eserver® pSeries Administration and Support for AIX 5L Version 5.3 and CISSP.

Shiv Dutta works for the Systems and Technology Group, IBM US in Austin. He assists ISVs with the enablement of their solutions on the pSeries and AIX 5L platform. Shiv has worked with AIX from its inception and has published a number of papers on various aspects of the operating system. He was a co-author of the redbook *AIX 5L Differences Guide Version 5.3 Edition*, SG24-7463 in 2004. Shiv is a Certified IT Specialist, has a Ph.D. in Physics from Ohio University, and a Project Management Professional (PMP) Certification from the Project Management Institute.

Liviu Rosca is a pSeries Specialist at IBM Global Services, Romania. He has been working for three years with IBM Integrated Technology Services, providing client support for pSeries, AIX 5L, HACMP, and WVR. His areas of expertise include designing pSeries, AIX 5L, HACMP, networking, and telecommunication solutions. He is IBM Certified AIX 5L and HACMP System Administrator and CCNP. He teaches AIX 5L and HACMP classes.

Gilvan Pereira da Silva is a Systems Support Specialist at IBM Brasil, and has been working for IBM Global Services - Strategic Outsourcing since December 2000 in Hortolandia, SP - Brasil. His expertise includes AIX, Tivoli Storage Manager, IBM pSeries,disk and tape storage. He is a certified Tivoli Storage Manager Administrator as well as IBM Certified Specialist - pSeries Administration and Support for AIX. He has a degree in Systems Analysis from FUMEC, Belo Horizonte, MG, Brasil. He co-authored the publication *Deployment Guide Series: IBM Tivoli Storage Manager Express*, SG24-7033-01.

Kristof Stroobants works as an Business Continuity and Recovery Services Specialist for IBM Belgium. He has four years of experience in Disaster Recovery on pSeries and xSeries® within IBM Belgium and Luxemburg. His areas of expertise include coordination and execution of Disaster Recovery Plans and architectural design for pSeries and xSeries platform. He holds a degree in Civil Engineering from the Katholieke Universiteit Leuven, Belgium. He is also an IBM Certified Specialist - IBM @server pSeries Administration and Support for AIX 5L Version 5.3.

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

Scott Vetter - Project Leader
International Technical Support Organization, Austin Center

Roji R John, Paul B Finley, Andy Solomon, Patrick Laffey, Lakshmi Yerneni, Paul Wadehra, Frank L Nichols, Eduardo L Reyes
IBM Austin

Helpful redbook authors who stopped what they were writing to assist us:

Hans Mozes, Steven A Edwards, Bruno Blanchard, Brad Gough
IBM Germany - United Kingdom - France - Australia

Helpful customers who wrote in with a clear set of recommended corrections:

Ayotunde M. Itayemi

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 or customers.

Your efforts will help increase product acceptance and client 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 E-mail to:

  `redbook@us.ibm.com`

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. JN9B Building 905
  11501 Burnet Road
  Austin, Texas 78758-3493

1

Certification exam - Test 222

This chapter provides information that useful for successfully preparing for IBM certification Test 222. For more recent/updated information about the objectives, visit our Web site at:

http://www.ibm.com/certify

On the same Web site, you can find sample and assessment tests.

© Copyright IBM Corp. 2006. All rights reserved.

1.1 Overview of the certification exam

The certification exam covers topics related to basic support and administration tasks on AIX 5L Version 5.3.

Table 1-1 gives a summary of the topics and the corresponding chapter in this redbook.

Table 1-1 Overview of the topics in Test 222

Topic	Chapter
Section 1 - Planning	
Determine device needs (such as communication adapters, disk controllers, HBAs, tape drives, CD-ROM drives, and so on).	Chapter 2, "System planning" on page 11
Determine the AIX 5L OS level compatibility.	
Determine redundancy requirements for levels of availability (for example, multiple I/O controllers, multiple communication controllers, multiple processors, multiple power supplies, RAID-level protection, dynamic CPU deallocation, and dead gateway detection).	
Select appropriate IBM resources for system planning.	
Describe LPAR and dynamic LPAR features and benefits.	
Determine disk requirements.	

Topic	Chapter
Section 2 - AIX 5L Installation and Updates	
Install AIX 5L from media using CD-ROM, DVD-ROM, or tape.	Chapter 3, "System and software installation" on page 27
Describe the NIM installation process and configuration.	
Identify why an installation hangs or fails.	
Identify installation LED codes.	
Describe the purpose of the Trusted Computing Base (TCB).	
Describe the alternate disk installation process and its advantages.	
Describe the process to clone a system from a **mksysb** command backup.	
Describe the differences between migration, preservation, and new and complete overwrite installation.	
Install software updates (PTFs and so on).	
Describe the options available for adding and removing filesets (for example, commit, apply, remove, or reject).	
Describe the AIX 5L installation packaging.	
Given an operating system or application requirement, select the appropriate AIX 5L installation method.	
Manage filesets (for example, use the **bffcreate** and **inutoc** commands).	
Identify the operating system maintenance level and installed filesets.	
Verify integrity of OS (for example, use the **lppchk** command and its flags).	
Describe the differences between the **installp** and **rpm** commands.	

Topic	Chapter
Section 3 - Boot Process	
Describe the boot process (**cfgmgr**, file system mounts, application startup, rootvg varyon, BIST, POST, boot sector load, and login enablement).	Chapter 4, “Boot process” on page 93
Describe the use of the /etc/inittab file.	
Describe LED codes (121, 223, 229, 551, 552, 553, 581, OC31, and OC32).	
Describe how to recover from a hung boot process.	
Describe run levels 0 through 9.	
Describe the use of the following rc.* files (rc.net, rc.tcpip, rc.boot, rc., rc.local).	
Show, modify, and reset bootlist	
Section 4 - Configuration	
Describe the function of the ODM and the locations of its files.	Chapter 5, “Configuration” on page 121
Query ODM.	
Describe the function of SMIT.	
Describe the process to run a Linux operating system application under the AIX 5L operating system.	
Configure the network using the following: TCP/IP, /etc/hosts, host name, **ifconfig**, **chdev**, **route**, /etc/resolv.conf, etc/netsvc.conf, **no**, and /etc/ntpd.conf.	
Configure an Ethernet adapter (speed, duplex, and virtual Ethernet).	
Use NFS to import or export file systems (for example, /etc/exports, biod, or nfsd).	
Configure, list, and modify paging space.	
Determine and modify existing device configuration.	
Reconfigure system to move from 32-bit to 64-bit mode and vice versa.	
Check for possible exposure in system access files (for example, /etc/hosts.equiv, .rhosts, .netrc, or .forward).	

Topic	Chapter
Section 5 - Storage and LVM	
Describe the relationship between physical and logical volumes and volume groups.	Chapter 6, "Disk storage management" on page 211 and Chapter 7, "File systems" on page 271
Describe the relationship between logical volumes and file systems.	
Describe the differences between JFS compared with JFS2.	
Manage volume groups (use the **varyonvg**, **varoffvg**, **mkvg**, **importvg**, **exportvg**, and **lsvg** commands).	
Manage logical volumes.	
Manage file systems (**mount**, **fsck**, **chfs**, **mkfs**, and **defragfs**).	
Describe the relationship between physical partition size and physical disk size.	
List the attributes of volume groups, logical volumes, and file systems that cannot be changed.	
Migrate data from one physical volume to another.	

Topic	Chapter
Section 6 - Monitoring and Tuning	
Monitor root mail for error messages.	Chapter 9, "Problem determination and resolution" on page 375
Monitor file systems' growth to prevent a full file system.	Chapter 8, "Monitoring and performance tuning" on page 287
Monitor the error log for hardware and software errors	
Use and interpret output from the **vmstat** and **sar** commands to determine performance problems.	
Use /etc/tunables for customization of system parameters.	
Given system performance problems, select the appropriate tool(s) (for example, **vmstat**, **iostat**, **topas**, **netstat**, **sar**, or **/proc**).	
Use and interpret output from the **netstat** and **iostat** commands to determine performance problems.	
Use and interpret the output of **/proc** commands for problem determination.	
Configure and monitor syslog for error conditions.	
Describe the system commands that document current system configuration.	
Use the Resource System Controller (RSCT) to set up resource and system monitoring.	
Describe the advantages of using Workload Manager (WLM) for system resource allocation.	
Describe Partition Load Manager features.	
Describe how to determine and recover from a full file system.	Chapter 9, "Problem determination and resolution" on page 375

Topic	Chapter
Section 7 - Problem Determination and Resolution	
Given a system problem, determine which tool (for example, **errpt**, **diag**, **netstat**, **ifconfig**, **ping**, **filemon**, or **traceroute**) is appropriate for problem determination.	Chapter 9, "Problem determination and resolution" on page 375
Monitor, interpret, and manage the error log using the **errpt** or **errclear** commands.	
Identify reasons to monitor the root user's mail.	
Identify when to use PerfPMRs.	
Manage a support call with IBM (for example, open, severity levels, escalation, and closing).	
Determine how to size and configure dump devices.	
Describe hot plug processes for replacing devices.	
Given user problems, determine the appropriate actions (for example, **usrck**, **grpck**, file permissions, resetting failed login attempts, or user limits).	
Given a hardware related problem, identify the probable cause.	
Given failed disk situations, describe the procedures for replacing the bad disk.	
Access rootvg in maintenance mode.	
Use the **alog** command to examine boot messages to locate startup problems.	
Troubleshoot system hangs when trying to start desktop.	
Troubleshoot error unable to open display.	

Topic	Chapter
Section 8 - Backup and Recovery / Disaster Recovery	
Describe methods of backing up and restoring the operating system (for example, using the **mksysb** command).	Chapter 10, “Backup and recovery” on page 459
Describe methods of managing and positioning tape backup media for backup and restore purposes.	
Describe methods of backing up and restoring data and data structures (**savevg** and **restvg**, **backup** and **restore**, **tar**, **cpio**, **pax**, **mkcd**, **gzip**, and **gunzip**).	
Describe methods to list or verify contents of the backup media.	
Describe methods of selectively altering or excluding data during the backup and restore process.	
Identify the elements of disaster recovery plans based on business needs.	
Describe the differences between full, differential, and incremental backups and their effect on backup and restore windows.	
Section 9 - Daily Management	
Add, delete, or alter user IDs and passwords.	Chapter 11, “Daily manageme nt” on page 491
Alter default user profiles and limits (for example, command shell profiles and /etc/security).	
Describe methods to monitor and manage processes (for example, **ps**, **kill**, **nice**, **fuser**, **topas**, or **svmon**).	
Identify and correct errors in shell script examples.	
Define AIX 5L file and directory permissions (for example, user, group, or other) and ownership.	
Define or modify file permissions and ownership (for example, **chmod**, **chown**, or **chgrp**).	
Describe the differences between foreground and background processes.	
Describe how to check and alter the priority of a process.	
Describe the differences between local and global variables and how to set, un-set, and show them.	

<table>
<tr><th>Topic</th><th>Chapter</th></tr>
<tr><td>Describe how to enable System V printing.</td><td rowspan="2">Chapter 14, “Printing” on page 619</td></tr>
<tr><td>Manage the print queuing subsystem.</td></tr>
<tr><td>Edit a file using the vi command.</td><td>Chapter 13, “Editing files using vi” on page 601</td></tr>
<tr><td>Describe methods to read, create, delete, or configure e-mail.</td><td>Chapter 12, “Electronic mail and sendmail” on page 581</td></tr>
<tr><td>Describe use of metacharacters and wild cards within the shell.</td><td rowspan="4">Chapter 11, “Daily manageme nt” on page 491</td></tr>
<tr><td>Describe how to redirect stdin, stdout, and stderr.</td></tr>
<tr><td>Use the cron command to schedule tasks.</td></tr>
<tr><td>Control system processes using System Resource Controller (SRC).</td></tr>
</table>

2

System planning

This chapter provides topics from the certification exam used to size a new AIX 5L system. System pre-planning is important, it is not only CPU, memory and DASD capacity that counts. By introducing new technologies, such as POWER5™, new additional hardware may be required for some features.

Topics discussed in this chapter are:

- Determining device needs for an AIX 5L system
- AIX 5L OS level compatibility
- System availability
- Appropriate IBM resources for system planning
- The LPAR Validation Tool
- Basic disk requirements

© Copyright IBM Corp. 2006. All rights reserved.

2.1 Determining device needs for an AIX 5L system

To get the most out of an AIX 5L system, the hardware used is important. It is always possible to add, upgrade, and configure new devices later, but an initial architecture planning step cannot be skipped. For example, a system may require an additional server named the Hardware Management Console (HMC) to provide advanced management functions.

During this planning phase, various documentation is available. Especially when designing a complex environment with virtualization on POWER5 processor-based systems, an LPAR Validation Tool can be downloaded. More information about this can be found in 2.5, "The LPAR Validation Tool" on page 23.

Table 2-1 shows a sample hardware checklist to guide you through the process of identifying required devices. For the current list of I/O features, visit:

http://www.ibm.com/systems/p/hardware/factsfeatures.html

and choose **I/O features.**

Table 2-1 A sample hardware checklist

Category	Requirement	Device needed
Communication	Token-ring network connection	Token-ring adapter
	Ethernet (10/100/1000 Mbps) connection	Ethernet adapter (copper/fiber)
Media I/O	Read from DVD-ROM disc	DVD-ROM drive
	Read/write DVD-RAM disc	DVD-RAM drive
	Read/write tape cartridge	Compatible tape streamer (SCSI/fiber)
Hard drives	Internal disks	Designated SCSI controller
	Disks with hardware RAID protection	Special RAID adapter
	External disks in a Storage Area Network	Fibre Channel Host Bus Adapter
	External disks in SSA tower	SSA adapter
Expansion	Extra slots to install adapters	Expansion drawer (including cables)
	Extra CPU / memory	Additional r riser cards
	Virtualization	APV feature, HMC, or other software

2.2 AIX 5L OS level compatibility

This section describes how binary compatibility is ensured for systems running mixed levels of the operating system.

After a migration installation, you may notice filesets on the system in the OBSOLETE state. Obsolete filesets were installed by earlier versions of the operating system, but they remain on the current system because the migration only replaced some, but not all, of the files they contain. These filesets remain necessary for systems running mixed levels of the operating system.

During a migration installation, the following filesets are automatically included:

- Base operating system commands
- Base operating system libraries
- Base operating system curses/termcap
- Base operating system networking
- Base operating system directories/files (symlinks)
- Messages
- X11R3
- X11R4
- X11 fonts

A system using AIX Version 4.2 can operate as a server system for client machines using AIX 5L Version 5.3 with the following exceptions:

- Network installation of AIX 5L Version 5.3 clients
- Service SNA or X.25 to AIX 5L Version 5.3 clients
- Service HCON to AIX 5L Version 5.3 clients
- Service CGE extensions of PEX and PEX-PHIGS
- Use of AIX 5L Version 5.3 client installation formats

Font servers might be required on the AIX 5L Version 5.3 clients to reliably handle AIX windows between server and client.

A system using AIX 5L Version 5.3 might operate as a server system for client machines using AIX Version 4.2 or later versions as long as the necessary compatibility options are installed. All statements about binary compatibility apply in this case. AIX 5L Version 5.3 applications might not execute reliably on AIX Version 4.2 systems using remote network mounts from an AIX 5L Version 5.3 file system.

2.2.1 Applications from earlier AIX 5L Version 5 releases

AIX 5L Version 5.1 and Version 5.2 applications written for RS/6000® POWER3™-, POWER4™-, POWER5-, and PowerPC®-based models can be executed on AIX 5L Version 5.3 without recompilation for the same and newer models in that processor family (POWER3, POWER4, POWER5, or PowerPC). Exceptions are applications compiled using POWER3-, POWER4-, POWER5-, or PowerPC-specific compiler options, but executed on models other than POWER3, POWER4, POWER5, or PowerPC, respectively, or applications using:

- Non-shared compiles of AIX-shared libraries
- Features explicitly described as non-portable by IBM in the AIX Version 4 or AIX 5L reference manuals
- Undocumented AIX 5L internal features
- X11R5 Server Extensions
- Locales based on IBM-850 codesets
- Legacy security library interfaces executing on AIX 5L Version 5 systems with long user names enabled

Any program that must run in all environments—POWER3, POWER4, POWER5, and PowerPC (601 and newer PowerPC processors)—must be compiled using the common mode or PowerPC option of the compiler. Programs compiled to exploit POWER5 technology must be run on POWER5 processor types. Programs compiled to exploit POWER4 technology must be run on POWER4 or POWER5 processor types. Programs compiled to exploit POWER3 technology may be run on POWER3, POWER4, or POWER5 processor types, though there may be some performance penalty when running on POWER4 or POWER5. Programs compiled to exploit PowerPC-based technology must be run on PowerPC-based processors. Existing binaries need not be recompiled to operate on the target processors

2.2.2 Applications on AIX 5L Version 5.3 with long user names

AIX 5L Version 5.3 systems can be configured to accommodate user and group names exceeding eight characters. Applications that have not been specifically structured to handle long user and group names and that use older security library interfaces with eight character name limits or depend on user and group names not exceeding eight characters in length may not work correctly on systems that have been enabled for long user and group names. AIX 5L Version 5.3 commands that display user and group names will truncate user and group names to eight characters to accommodate existing use unless command-specific options are utilized to display long user and group names.

The older security interfaces are provided in Table 2-2 on page 15.

Table 2-2 Previous security interfaces

Previous Security Library Interface	Long User name-Enabled Alternative
ckuserID()	authenticatex()
cuserid()	getpwuid()
getuinfo()	getuinfox()
getuinfo_r()	getuinfox()
getuserpw()	getuserpwx()
newpass()	newpassx()
putuserpw()	putuserpwx()
putuserwhist()	putuserpwxhist()

AIX 5L Version 5.3 systems running applications using security library interfaces should not be configured for long user names unless the applications have been tested successfully for long user name support.

2.2.3 32-bit applications from AIX Version 4 releases

AIX Version 4.1, 4.2, or 4.3 applications written for RS/6000 POWER3-, POWER4-, and PowerPC-based models can be executed on AIX 5L without recompilation for same and newer models in that processor family (POWER3, POWER4, POWER5, or PowerPC). Exceptions are applications compiled using POWER3-, POWER4-, or PowerPC-specific compiler options executed on models other than POWER3, POWER4, POWER5, or PowerPC, respectively, or applications using:

- Non-shared compiles of AIX shared libraries
- Features explicitly described as non-portable by IBM in the AIX Version 4 or AIX 5L reference manuals
- Undocumented AIX internal features
- X11R5 Server Extensions
- Locales based on IBM-850 codesets
- Legacy security interfaces executing on AIX 5L systems with long user names enabled

Programs compiled to exploit POWER5 technology must be run on POWER5 processor types. Programs compiled to exploit POWER4 technology must be run on POWER4 or POWER5 processor types. Programs compiled to exploit POWER3 technology may be run on POWER3, POWER4, or POWER5 processor types, though there may be some performance penalty when running on POWER4 or POWER5. Programs compiled to exploit PowerPC-based technology must be run on PowerPC-based processors. Existing binaries need not be recompiled to operate on the target processors.

2.2.4 64-bit applications from AIX Version 4 releases

Any 64-bit applications produced using AIX Version 4 will not execute on AIX 5L. These applications need to be recompiled from the source on AIX 5L to execute on this version of AIX. The 64-bit applications produced using AIX 5L on any of the 32-bit or 64-bit processor models will execute without recompilation on the 64-bit processor models. The 32-bit applications produced using AIX 5L on either 32-bit or 64-bit processor models will execute without recompilation on both models.

2.2.5 X11R5 and X11R6 compatibility on AIX 5L Version 5

The AIX 5L X-server uses the X-Consortium release 6 of X (commonly known as X11R6). The libraries shipped by IBM with X11R6 are backward compatible and the client applications that access these libraries work as on AIX Version 4. As on AIX Version 4, IBM will also ship X11R3, X11R4, and X11R5 compatibility installation options for maximum flexibility.

The majority of applications using X fall into this category and will not cause any difficulty. However, a small number of X-applications use the loadable extension facility provided by the X-server.

The X-server allows for the addition of new functionality through its extension mechanism. For each extension, part of the extension is loaded into the X-server before it can be executed. X11R6 has modified how this mechanism works in the course of improvements to X, and it is this part of the extension that must be made compatible with X11R6 to execute properly. All extensions supplied by IBM have been made compatible. In some circumstances, you may have an extension that does not work with X11R6; for example:

- Sample extension downloaded from the X-Consortium FTP site
- Customer-developed extension
- Third-party extension

In these cases, the extension needs to be made compatible with X11R6 before it executes properly. Customer-developed extensions and sample X consortium extensions need to be recompiled with the X11R6 environment. For third-party extensions, contact the vendor for a X11R6-compatible update.

If you use non-IBM display adapters, you may also be using vendor supplied software specific to those devices that uses X11R6 server capabilities. If so, this software must be compatible with X11R6 to operate properly. Contact the vendor of the display adapter for this software.

2.2.6 Binary differences between AIX Version 3 and AIX 5L

All AIX applications using AIX V3R3.2 or later, for POWER™-, POWER2™-, and PowerPC-based models that are written in accordance with the guidelines in this announcement and other AIX announcements, run on AIX 5L without recompilation for those same models. The exceptions to this statement would be applications compiled using POWER2- or PowerPC- specific compiler options, but run on models other than POWER2 or PowerPC, or applications using the following:

- Their own loadable kernel extensions
- Certain high-function terminal control interfaces
- X11R3 input device interfaces
- The CIO LAN device driver interface
- SCSI device configuration methods (IHVs)
- The nlist() interface
- DCE threads
- Previous security interfaces executing on AIX 5L systems with long user names enabled.

> **Note:** Any program that must run in all environments (POWER family, POWER2, and POWER-based models 601 and higher) must be compiled using the common mode of the compiler. Programs compiled to exploit POWER2 technology must be run on POWER2 processors. Existing code need not be recompiled to run.

Applications must have been created using the AIX shared libraries for these binary compatibility statements to apply.

2.3 System availability

In order to increase system availability, care must be taken when the system is designed and configured.

From a hardware point of view, you should include in your system configuration redundant components to avoid single points of failure (SPOF). This normally comes with a cost. The final configuration of your system will always be a trade-off between the level of availability and the price of those features.

The time spent to replace or upgrade a component also affects your system availability. When planning your system, keep in mind the placement of devices that are more likely to be replaced, such as adapters or disks. Serviceability of your system improves overall availability.

You should configure your system to be resilient to one or more device failures.

When planning and configuring the system, you should also take into account external factors, such as the network, that might affect the overall availability of the services provided by your system.

2.3.1 Availability of hardware components

You can increase the availability of your system using redundant or fault tolerant hardware components, such as:

- Hardware Management Console (HMC)

  If you are using advanced POWER5 processor-based systems that require the presence of an HMC, you should use dual HMCs to provide redundancy of this critical management interface.

- Redundant Service Processor (SP)

  If a service processor is unavailable, its function will dynamically be taken over by the redundant service processor.

- System Power Control Network (SPCN) cables

  SPCN cables provide communication between the server power system and drawer or tower power system. They are connected in a loop. Should any segment of the loop fails, the system and external drawers or towers will remain connected.

- I/O connectors

  The server design should use dual I/O connections between the Central Electronics Complex (CEC) and I/O drawers or I/O towers. Should an I/O cable become damaged, the system can automatically reconfigure to use the second cable.

- Power supplies

  To increase system availability, the power supplies should be redundant, powered from separate power circuits that use different fuses.

- Cooling

  The server should include multiple variable speed fans. Should any of them fail, the remaining fans will automatically increase their rotational speed to maintain the thermal conditions within limits.

2.3.2 Dynamic reconfiguration - LPAR features

On POWER5 processor-based systems, you can take advantage of the Advanced POWER Virtualization feature (if active) and dynamic LPAR. If you are using a system that can accommodate multiple partitions, it is possible to dynamically move system resources, physical processors, virtual processors, memory, and I/O slots, between partitions without rebooting. Design the profiles for your partitions with availability in mind.

- Processors

  If you are using POWER5 processor-based systems, you can take advantage of dynamic CPU deallocation. The service processor monitors all CPUs. When a CPU reaches a predefined error threshold, the operating system is informed to drain the run-queue for that CPU, redistribute the work to the remaining CPUs, deallocate the failing CPU, and continue normal operation. If there is an inactive CPU available, it will be used to replace the deallocated processor.

- Memory

  POWER5 processor-based systems have the ability to monitor memory modules and replace the failing memory with an unused memory card.

- Adapter sparing

  Adapter sparing can be achieved by maintaining a set of PCI adapters as global spares that are available for DR operations.

- Ethernet Link Aggregation

  Ethernet Link Aggregation (LA) is a network port aggregation technology that allows several Ethernet adapters to be aggregated together to form a single pseudo Ethernet adapter. Using Ethernet LA, you can avoid having a physical Ethernet adapter as a SPOF.

- Multipath I/O

  Multipath I/O (MPIO) allows supported MPIO devices to be accessed through one or more physical I/O adapters. By implementing MPIO and locating different physical adapters in different slots or different CECs, you can remove the loss of an physical adapter or an I/O bus from the SPOF list.

- Redundant Virtual I/O Servers

  A partition can be a client of one or more Virtual I/O Servers at a time. To increase the availability for your partition, you can connect it to two Virtual I/O Servers.

2.3.3 Configure the operating system for high availability

The following list provides an overview of operating specific topics that may be part of the certification exam.

- Mirroring at LVM layer

  To increase data availability, you can implement strict mirroring at the LVM layer for all logical volumes. If possible, copies of the same logical partition should be located on different physical volumes that are using different SCSI controllers, different SCSI Enclosure Services (SES) backplanes, and even different CECs. Try to use disk drives of the same size and keep the mirroring policy as simple as possible.

  Special care has to be taken for rootvg because failure of the rootvg will render the system unusable.

- Quorum of volume groups

  You should ensure that the quorum of volume groups, including rootvg is properly set.

- Boot list

  You should include in the bootlist every physical volume that contains a valid BLV.

- Copies of logical volumes

  When you create a mirror copy of a logical volume, ensure that you synchronize LV copies.

- Using RAID

  Depending on the level of the availability and hardware configuration, you can use different RAID level implementations to increase your data availability as follows:

  RAID 0 Also known as striping. Data is split into blocks of equal size and stored on different disks. There is no redundancy. If one disk fails, your data is lost.

  RAID 1 Also known as mirroring. Data is split into blocks of equal size. Duplicate copies are kept on separate physical disks. If one disk fails, the second copy is used and data is still available.

  RAID 5 Also known as striping with parity. Data is split into blocks of equal size. An additional data block containing parity information is generated. Both data blocks and the parity block are written onto separate physical disks. If one disk storing a block of data fails, the system will reconstruct the data from the failed drive using the parity information and the remaining data from the other drives.

  RAID 10 Also known as RAID 0+1 or enhanced RAID. It is a combination of mirroring (RAID 1) and striping (RAID 0).

- Software and microcode management

  Try to take advantage as much as possible of the ability to install, update, reject, and remove software and firmware without interrupting the system.

- Network interface physical connections

  When you connect your network interfaces, ensure that you are compliant with external network availability policies, such as:

  – Connecting different physical network adapters to different physical Ethernet switches

  – Configuring different physical network adapters to belong to a single VLAN

- Network interface configuration

  You should configure network interfaces so that your system will still be able to provide the services it is supposed to provide even if an interface fails or network is experiencing problems. You should consider using some of the following:

  – Connecting your system to multiple networks

  – Connecting multiple interfaces to the same network

  – Defining multiple routes, having different weights to the same network

  – Configuring dead gateway detection

  – Defining aliases for network interfaces

2.4 Appropriate IBM resources for system planning

To design a new system, you may be required to research hardware and compatibility information. This may also be required when you want to upgrade an elder machine.

You can find a quick reference for the current IBM @server pSeries at:

http://www.ibm.com/systems/p/hardware/factsfeatures.html

where the latest *Facts and Features* document resides. Within this document, various hardware descriptions can be found, and some examples follow:

- Physical system packaging (desk-side or rack drawer) and dimensions
- Microprocessor type and quantity
- Clock rates available
- System memory (standard - maximum)
- Data - instruction (L1) cache - Level 2 (L2) cache - Level 3 (L3) cache
- Chipkill™ memory
- Service processor
- Hot-swappable disks (internal and external)
- Dynamic Processor Deallocation
- Dynamic deallocation: PCI-X bus slots
- Hot-plug slots
- Blind-swap slots
- Redundant hot-plug power
- Redundant hot-plug cooling
- Capacity and expandability
- Capacity on Demand (CoD) functions
- Maximum logical partitions/micro-partitions
- Maximum available PCI-X slots

- Maximum PCI-X bus speed
- Disk | media bays
- Communication adapters
- pSeries High Performance Switch
- Display adapter

An important factor in sizing your system, the benchmarks, are also shown in that document, for different benchmarking tests.

If you want to know more about the capabilities of adapters, and where to install them, you can consult the *RS/6000 & eServer pSeries Adapter Placement Reference for AIX*, SA38-0538. You can find this publication by searching on "PCI adapter placement" from within the InfoCenter.

2.5 The LPAR Validation Tool

The LPAR Validation Tool (LVT) is available to assist the user in the design of a system configured to use LPARs and to provide an LPAR validation report that reflects the user's system requirements. This is also a helpful tool to give an inventory of your system and it provides a picture of slot layout and usage.

The LVT is not a marketing configurator. It does not automatically add hardware features. It will not prevent inefficient system design so long as the design itself meets manufacturing card placement rules and minimum LPAR recommendations.

The LVT is a PC based tool intended to be run as a stand-alone Java™ application.

> **Note:** Keep this tool updated to ensure it has all the latest models, features, and rules.

Download the tool on the Web at:

```
http://www.ibm.com/servers/eserver/iseries/lpar/systemdesign.html
```

Figure 2-1 shows an example of the LVT. At the left side, you can choose among the optional parts (disk drives, tapes, adapters, and so on). In the example, the RAID enabler card is selected. At the right side, the LVT will automatically highlight the positions where the card can be installed. Clicking the **Add** button will insert the card into that position and attaching it to the specified LPAR.

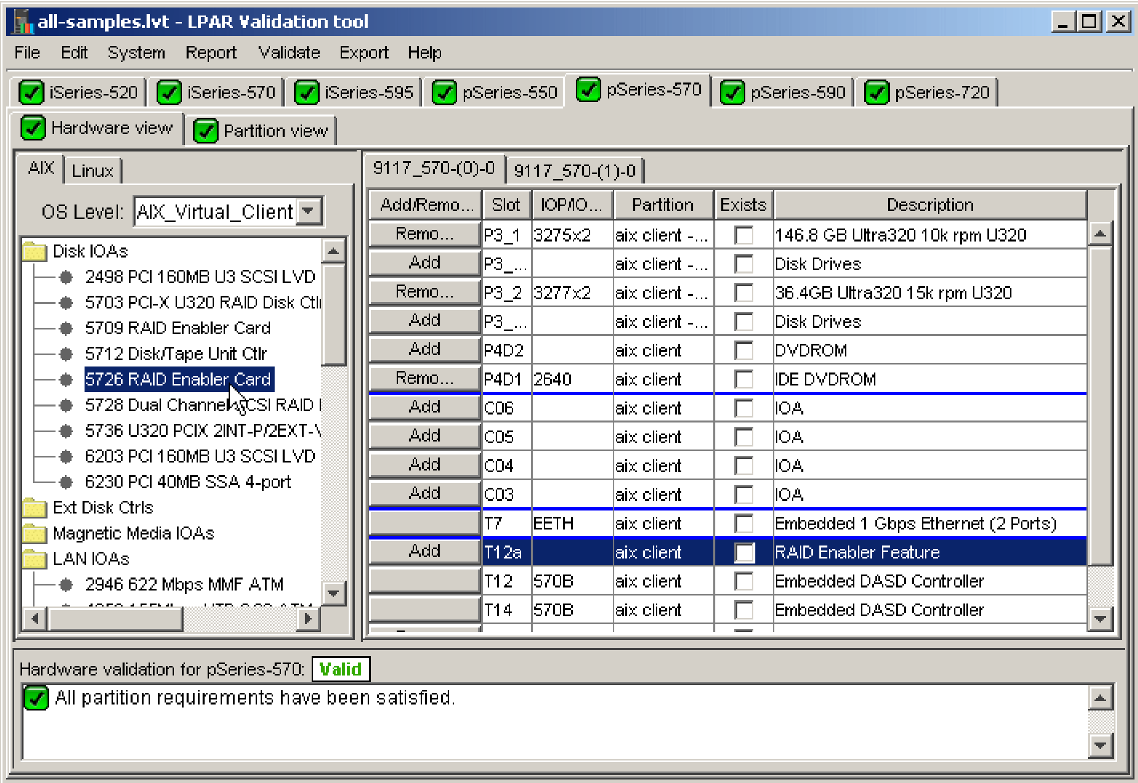


Figure 2-1 Sample LVT configuration

> **Tip:** the LVT tool can be used to configure both IBM *@server* i5 and p5 systems. It will adapt the hardware requirements to the appropriate LPAR operating system.

> **Important:** Consult your hardware manual before installing a new device into your system. Some adapters from previous generation machines may not work.

2.6 Basic disk requirements

AIX 5L Version 5.3 requires at least 2.2 GB of physical disk space for Base Operating System (BOS) in a stand-alone system or an LPAR. To increase the level of data redundancy, you need more physical volumes to create RAID protection or disk mirroring. For disk requirements for other (non-system) applications, consult the documentation of that application.

Note: It is recommended that application data (for example, database files) is stored in separate volume groups other than rootvg. Data can be placed on local disk or Storage Area Network.

3

System and software installation

This chapter describes the installation process, the common commands that are used with the installation process, and the different methods available for installing software onto a system.

You can find more information about the following topics in this chapter:

- Base Operating System (BOS) installation options
- Installation of the optional software
- Maintenance levels
- Software requirements
- BOS integrity

© Copyright IBM Corp. 2006. All rights reserved.

3.1 AIX 5L installation methods

When you install the BOS, you have a choice between three primary methods:

- New and Complete Overwrite Installation
- Migration installation
- Preservation installation

These methods are discussed in the following sections.

3.1.1 New and Complete Overwrite installation

Generally, the New and Complete Overwrite method is used when:

- You have a new machine without a prior or useful system installation. In this case, the hard disk or disks on which you are installing the BOS are empty. This is the only possible installation method for a new machine.
- You want to install onto a hard disk that contains an existing root volume group that you wish to completely overwrite. For example, this might occur if your root volume group has become corrupted.
- You want to reassign your hard disks, that is, to make your rootvg smaller and assign less disk space to it.

Note: The New and Complete Overwrite installation overwrites all data on the selected destination disk. This means that, after the installation is complete, you will have to manually configure your system using the Configuration Assistant application, SMIT, or the command line. If you want to preserve your system configuration, and you do not need to completely overwrite your root volume group, do not use the New and Complete Overwrite option.

3.1.2 Migration installation

Use the migration installation method to upgrade AIX Version 4.2, 4.3, AIX 5L Version 5.1, or Version 5.2 to AIX 5L Version 5.3 while preserving the existing root volume group (see the release notes for restrictions). The installation process determines which optional software products must be installed. With the exception of /tmp, this method preserves most file systems, including the root volume group, logical volumes, and system configuration files.

In most cases, the user configuration files from the previous version of a product are saved when the new version is installed during a migration installation.

3.1.3 Preservation installation

Use the preservation installation method when a version of the BOS is installed on your system, and you want to preserve the user data in the root volume group. However, this method overwrites the /usr, /tmp, /var, and / (root) file systems by default, so any user data in these directories is lost. These file systems are removed and recreated, so any other LPPs or filesets that you installed on the system will also be lost. System configuration must be done after doing a preservation installation.

The /etc/preserve.list file contains a list of system files to be copied and saved during a preservation BOS installation. The /etc/filesystems file is listed by default. You can add the full path names of any additional files that you want to save during the Preservation installation to the preserve.list file. For example, you can alter the /etc/preserve.list file to tell your installation process that you want to preserve your /var file system.

3.1.4 Summary

Table 3-1 lists the differences in the installation steps among the installation methods.

Table 3-1 AIX 5L BOS installation methods

Installation steps	New and complete overwrite	Preservation	Migration
Create rootvg	Yes	No	No
Create file system /, /usr, /var	Yes	Yes	No
Create file system /home	Yes	No	No
Save configuration	No	No	Yes
Restore BOS	Yes	Yes	Yes
Install additional filesets	Yes	Yes	Yes
Restore configuration	No	No	Yes

3.2 Trusted Computing Base

The Trusted Computing Base (TCB) is the part of the system that is responsible for enforcing the information security policies of the system. By installing a system with the TCB option, you enable the trusted path, trusted shell, trusted processes, and system-integrity checking. Because every device is part of the TCB, every file in the /dev directory is monitored by the TCB. In addition, the TCB automatically monitors over 600 additional files, storing critical information about these files in the /etc/security/sysck.cfg file. It is recommended to back up this file on removable media, such as tape or CD, immediately after installing the BOS.

To demonstrate one of the features of TCB, we take the example of the trusted path. A trusted communication path is established by pressing the Secure Attention Key (SAK) reserved by the key sequence (Ctrl-X and then Ctrl-R). This will *kill* all processes that attempt to access the terminal you are working on, and any links to it (for example, /dev/console can be linked to /dev/tty0). If a new login screen displays, you have a secure path. If the trusted shell prompt displays, the initial login screen was an unauthorized program that might have been trying to capture your password. If this is the case, you can determine who is currently using this terminal by using the **who** command and then log off.

It is important to note that you can enable TCB only at installation time. If you set the attribute to YES, the install process installs the bos.rte.security fileset and you can configure TCB. If you decide not to install TCB, you will have to reinstall the operating system to enable TCB at a later stage. TCB can be removed by removing the bos.rte.security fileset from the system.

To check if TCB is enabled on your system, issue the **/usr/bin/tcbck** command. If a usage statement is returned, TCB is enabled:

```
# /usr/bin/tcbck

Usage: tcbck -a <filename> [ [ <attr> | <attr>=[<value> | ] ] ... ]
             -a -f <filename>
             -a sysck [ treeck_novfs=<dir> ] [ checksum=<program> ]
             -d <filename> | <class> [ <filename> | <class> ] ...
             -d -f <filename>
                 -l /dev/<filename> [ /dev/<filename> ] ...
             -(p|y|n|t) [-i] [ [<filename>|<class>] ... | ALL | tree ]
```

Otherwise, the following message is displayed:

```
# /usr/bin/tcbck
3001-101 The Trusted Computing Base is not enabled on this machine.
        To enable the Trusted Computing Base, you must reinstall and
        set the 'Install Trusted Computing Base' option to YES.
        No checking is being performed.
```

To learn more about the **tcbck** command and TCB features, consult the product information pages.

3.3 Base Operating System installation

In this section, we describe a sample BOS system installation on an IBM @server p5-550. Installation on your server can differ, depending on your hardware configuration. For more information about the installation of AIX 5L, consult *AIX 5L Version 5.3 Installation Guide and Reference*, SC23-4887.

For each step in the installation process, the server shows LED codes on the operator panel; all installation LED codes are described in 3.7.4, "Installation LED codes" on page 62.

3.3.1 Preparation

Before installing the BOS, make sure your hardware is installed properly. Have useful information ready, such as the host name, network configuration settings, and so on, that you will be using.

For this example, we use an HMC connected to the server. You need to set up the partitions on the HMC if you plan to use them. In this redbook, we assume that the partition is already configured with at least:

- One CD-ROM or DVD-ROM drive for CD installation.
- Enough disk capacity (locally or externally).
- One network adapter.
- The minimum CPU and memory requirements are met.

> **Note:** Make sure you know the type of CD-ROM or DVD-ROM; during the procedure, you have to choose the right device, whether it is IDE or SCSI.

The HMC in our setup is used as the console; if you are not using an HMC, make sure you have a console (serial or graphic) attached to the system.

3.3.2 Sample AIX 5L installation procedure

Figure 3-1 shows a flow chart of the steps for installing a system.

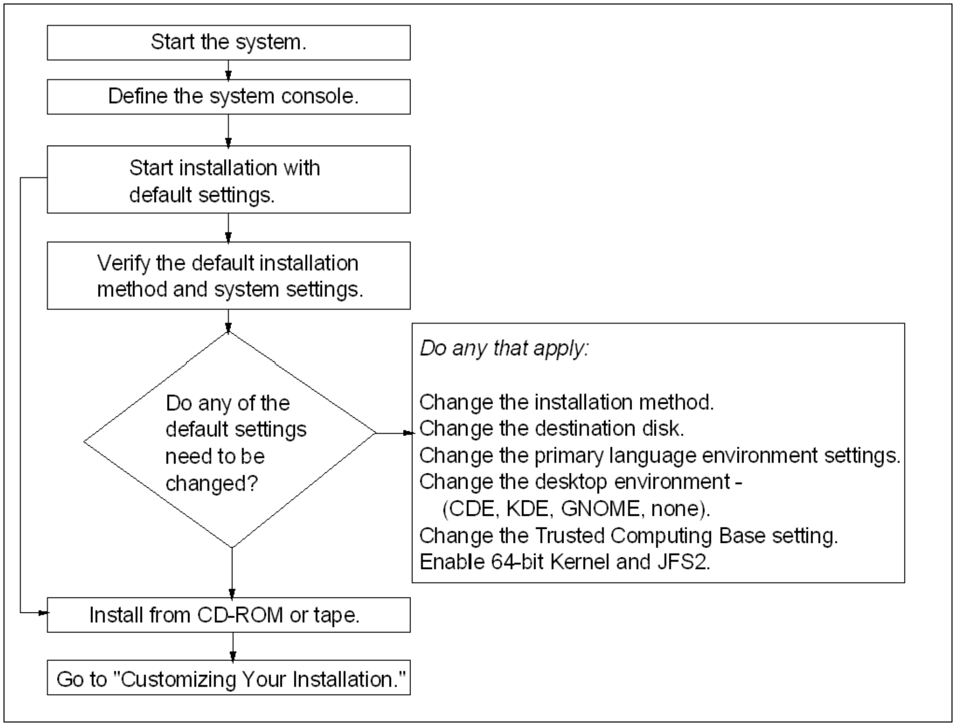


Figure 3-1 Flow chart for AIX 5L Version 5.3 system installation

Perform the following steps:

1. Boot the server into SMS mode:
   a. With an HMC, choose **Activate**, the partition, and override the boot mode to SMS.
   b. Without an HMC, wait until the POST finishes and you hear the two beeps, then press the 1 key.
2. Insert the AIX 5L Version 5.3 Volume 1 CD into the managed CD device.

3. A screen, similar to Figure 3-2, is shown. Enter option 5 (Select Boot Options).

```
PowerPC Firmware
Version SF235_160
SMS 1.6 (c) Copyright IBM Corp. 2000,2005 All rights reserved.
-------------------------------------------------------------------------------
Main Menu
1.   Select Language
2.   Setup Remote IPL (Initial Program Load)
3.   Change SCSI Settings
4.   Select Console
5.   Select Boot Options










-------------------------------------------------------------------------------
Navigation Keys:

                                           X = eXit System Management Services
-------------------------------------------------------------------------------
Type menu item number and press Enter or select Navigation key:
```

Figure 3-2 SMS menu

4. In the following menu, as shown in Figure 3-3, choose Option 1 (Select Install/Boot Device) and press Enter.

```
PowerPC Firmware
Version SF235_160
SMS 1.6 (c) Copyright IBM Corp. 2000,2005 All rights reserved.
-------------------------------------------------------------------------------
Multiboot
1.   Select Install/Boot Device
2.   Configure Boot Device Order
3.   Multiboot Startup <OFF>











-------------------------------------------------------------------------------
Navigation keys:
M = return to Main Menu
ESC key = return to previous screen         X = eXit System Management Services
-------------------------------------------------------------------------------
Type menu item number and press Enter or select Navigation key:
```

Figure 3-3 SMS menu - Boot options

5. The installation device of AIX 5L in this example is a CD-ROM. Select 3 (CD/DVD), as shown in Figure 3-4.

```
PowerPC Firmware
Version SF235_160
SMS 1.6 (c) Copyright IBM Corp. 2000,2005 All rights reserved.
-------------------------------------------------------------------------------
Select Device Type
1.   Diskette
2.   Tape
3.   CD/DVD
4.   IDE
5.   Hard Drive
6.   Network
7.   List all Devices







-------------------------------------------------------------------------------
Navigation keys:
M = return to Main Menu
ESC key = return to previous screen         X = eXit System Management Services
-------------------------------------------------------------------------------
Type menu item number and press Enter or select Navigation key:
```

Figure 3-4 SMS menu - Install/boot device options

6. Figure 3-5 shows the possibilities regarding the Media Type of the installation media. If your CD-ROM device is an internal IDE device, select 4 (IDE). If your CD-ROM device is SCSI, then select 1 (SCSI). Press Enter.

```
PowerPC Firmware
Version SF235_160
SMS 1.6 (c) Copyright IBM Corp. 2000,2005 All rights reserved.
-------------------------------------------------------------------------------
Select Media Type
1.   SCSI
2.   SSA
3.   SAN
4.   IDE
5.   ISA
6.   List All Devices









-------------------------------------------------------------------------------
Navigation keys:
M = return to Main Menu
ESC key = return to previous screen         X = eXit System Management Services
-------------------------------------------------------------------------------
Type menu item number and press Enter or select Navigation key:
```

Figure 3-5   SMS menu - Select Media Type

7. Figure 3-6 is an example of the output of all IDE CD-ROM devices. As this is the only one present in the system, there is no confusion over which one to choose. If there are more choices presented on your system, you have to analyze the device descriptions. The location code of every device indicates the exact location of the connected card. Based on that information, you have to identify the required device number.

   In this example, type option 1 and press Enter.

```
PowerPC Firmware
Version SF235_160
SMS 1.6 (c) Copyright IBM Corp. 2000,2005 All rights reserved.
-------------------------------------------------------------------------------
Select Device
Device  Current  Device
Number  Position  Name
1.        -      IDE CD-ROM
                 ( loc=U787B.001.DNW0974-P4-D2 )









-------------------------------------------------------------------------------
Navigation keys:
M = return to Main Menu
ESC key = return to previous screen         X = eXit System Management Services
-------------------------------------------------------------------------------
Type menu item number and press Enter or select Navigation key:
```

Figure 3-6 SMS menu - Select Device

8. Select the Normal Mode Boot by entering 2, as shown in Figure 3-7 on page 37.

```
Version SF235_160
SMS 1.6 (c) Copyright IBM Corp. 2000,2005 All rights reserved.
-------------------------------------------------------------------------------
Select Task

IDE CD-ROM
    ( loc=U787B.001.DNW0974-P4-D2 )

1.   Information
2.   Normal Mode Boot
3.   Service Mode Boot









-------------------------------------------------------------------------------
Navigation keys:
M = return to Main Menu
ESC key = return to previous screen         X = eXit System Management Services
-------------------------------------------------------------------------------
Type menu item number and press Enter or select Navigation key:
```

Figure 3-7 SMS menu - Select boot mode

9. Confirm to exit SMS by typing 1 and press Enter, as shown in Table 3-8 on page 91.

```
PowerPC Firmware
Version SF235_160
SMS 1.6 (c) Copyright IBM Corp. 2000,2005 All rights reserved.
-------------------------------------------------------------------------------
Are you sure you want to exit System Management Services?
1.   Yes
2.   No












-------------------------------------------------------------------------------
Navigation Keys:

                                            X = eXit System Management Services
-------------------------------------------------------------------------------
Type menu item number and press Enter or select Navigation key:
```

Figure 3-8 SMS menu - Confirm exit

10. The system boots from the media. You will see a result similar to Figure 3-9.

```
IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM

Elapsed time since release of system processors: 169 mins 19 secs







-------------------------------------------------------------------------------
                                Welcome to AIX.
                       boot image timestamp: 04:50 08/26
                 The current time and date: 00:44:30 11/18/2005
               number of processors: 2    size of memory: 3792MB
 boot device: /pci@800000020000002/pci@2,3/ide@1/disk@0:\ppc\chrp\bootfile.exe
                     kernel size: 10941966; 32 bit kernel
                       kernel debugger setting: enabled
-------------------------------------------------------------------------------

AIX Version 5.3
Starting NODE#000 physical CPU#001 as logical CPU#001... done.
Starting NODE#000 physical CPU#002 as logical CPU#002... done.
Starting NODE#000 physical CPU#003 as logical CPU#003... done.
```

Figure 3-9 System boot from CD media

11. The text shown in Figure 3-10 appears on all console devices. Type the number seen on your display and press Enter.

```
Starting NODE#000 physical CPU#003 as logical CPU#003... done.
Preserving 19558 bytes of symbol table [lft_loadpin32]


 ******* Please define the System Console. *******

Type a 2 and press Enter to use this terminal as the
  system console.
Pour definir ce terminal comme console systeme, appuyez
  sur 2 puis sur Entree.
Taste 2 und anschliessend die Eingabetaste druecken, um
  diese Datenstation als Systemkonsole zu verwenden.
Premere il tasto 2 ed Invio per usare questo terminal
  come console.
Escriba 2 y pulse Intro para utilizar esta terminal como
  consola del sistema.
Escriviu 1 2 i premeu Intro per utilitzar aquest
  terminal com a consola del sistema.
Digite um 2 e pressione Enter para utilizar este terminal
  como console do sistema.
```

Figure 3-10 Define the System Console

12. Figure 3-11 shows the different options regarding the language during the installation. Choose your language and press Enter.

```
>>>  1 Type 1 and press Enter to have English during install.
     2 Entreu 2 i premeu Intro per veure la instal·lació en català.
     3 Entrez 3 pour effectuer l'installation en français.
     4 Für Installation in deutscher Sprache 4 eingeben
        und die Eingabetaste drücken.
     5 Immettere 5 e premere Invio per l'installazione in Italiano.
     6 Digite 6 e pressione Enter para usar Português na instalação.
     7 Escriba 7 y pulse Intro para la instalación en español.









    88  Help ?


>>> Choice [1]:
```

Figure 3-11 Choose the menu language

13. The main menu of the BOS installation and maintenance is shown as in Figure 3-12.

Attention: The pointer “>>>” indicates the selected menu. Pressing Enter with the pointer on a wrong option or menu can change parameters or initiate the installation.

```
                      Welcome to Base Operating System
                      Installation and Maintenance

Type the number of your choice and press Enter.  Choice is indicated by >>>.

>>> 1 Start Install Now with Default Settings

    2 Change/Show Installation Settings and Install

    3 Start Maintenance Mode for System Recovery










    88  Help ?
    99  Previous Menu

>>> Choice [1]:
```

Figure 3-12 BOS installation main menu

14. Select option 2 (Change/Show Installation Settings and Install), which will display the screen shown in Figure 3-13 on page 41.

```
                    Installation and Settings

Either type 0 and press Enter to install with current settings, or type the
number of the setting you want to change and press Enter.

    1  System Settings:
         Method of Installation.............New and Complete Overwrite
         Disk Where You Want to Install.....hdisk0...

    2  Primary Language Environment Settings (AFTER Install):
         Cultural Convention................English (United States)
         Language ..........................English (United States)
         Keyboard ..........................English (United States)
         Keyboard Type......................Default

    3  More Options  (Desktop, Security, Kernel, Software, ...)

>>> 0  Install with the current settings listed above.

                       +-----------------------------------------------------
    88  Help ?         |    WARNING: Base Operating System Installation will
    99  Previous Menu  |    destroy or impair recovery of ALL data on the
                       |    destination disk hdisk0.
>>> Choice [0]:
```

Figure 3-13 Change/Show Installation Settings

15. Press 1 and Enter to change the Method of Installation. Some different methods are shown as in Figure 3-14. More information about the differences between methods of installation is covered in 3.1, “AIX 5L installation methods” on page 28. In this example, a New And Complete Overwrite method is chosen. Press 1 and Enter.

```
                    Change Method of Installation

  Type the number of the installation method and press Enter.

>>> 1 New and Complete Overwrite
      Overwrites EVERYTHING on the disk selected for installation.
      Warning: Only use this method if the disk is totally empty or if there
      is nothing on the disk you want to preserve.

    2 Preservation Install
      Preserves SOME of the existing data on the disk selected for
      installation.  Warning: This method overwrites the usr (/usr),
      variable (/var), temporary (/tmp), and root (/) file systems.  Other
      product (applications) files and configuration data will be destroyed.




    88  Help ?
    99  Previous Menu

>>> Choice [1]:
```

Figure 3-14 Different methods of installing AIX 5L

16. A list of available hard disks is shown in Figure 3-15. Type a number in front of the disk to include or exclude that disk. In this example, we use only one hard disk to install AIX 5L Version 5.3. In this case, we do not have to make any changes.

```
                 Change Disk(s) Where You Want to Install

Type one or more numbers for the disk(s) to be used for installation and press
Enter. To cancel a choice, type the corresponding number and Press Enter.
At least one bootable disk must be selected. The current choice is indicated
by >>>.

        Name      Location Code   Size(MB)  VG Status   Bootable

>>>  1  hdisk0   05-08-00-3,0     70006   rootvg          Yes    No
     2  hdisk1   05-08-00-4,0     70006   none            Yes    No
     3  hdisk2   05-08-00-5,0     70006   none            Yes    No
     4  hdisk3   05-08-00-8,0     70006   none            Yes    No
     5  hdisk4   09-08-00-3,0     70006   none            Yes    No
    06  MORE CHOICES...

>>>  0   Continue with choices indicated above
    55  More Disk Options
    66  Devices not known to Base Operating System Installation
    77  Display More Disk Information
    88  Help ?
    99  Previous Menu

>>> Choice [0]:
```

Figure 3-15 Sample disk configuration

17. If you do not want to erase your hard disks before installing the BOS, continue with step 18. Otherwise, choose option 55 and press Enter. 3.3.3, “Procedure to erase a hard disk” on page 48 explains more about the new capabilities of the erasure. Figure 3-16 on page 43 displays the different options you have.

```
                        Erasure Options for Disks

Select the number of times the disk(s) will be erased,
and select the corresponding pattern to use for each disk erasure.
If the number of patterns to write is 0 then no disk erasure will occur.
This will be a time consuming process.  Either type 0 and press Enter to
continue with the current settings, or type the number of the setting
you want to change and press Enter.

     1. Number of patterns to write..................................... 1
     2. Pattern #1...................................................... 00
     3. Pattern #2...................................................... ff
     4. Pattern #3...................................................... a5
     5. Pattern #4...................................................... 5a
     6. Pattern #5...................................................... 00
     7. Pattern #6...................................................... ff
     8. Pattern #7...................................................... a5
     9. Pattern #8...................................................... 5a

>>>  0   Continue with choices indicated above
    88  Help ?
    99  Previous Menu

>>> Choice [0]:
```

Figure 3-16 Erase Disk options

18. Press 0 and Enter to return to the Installation and Settings menu. You can alter the primary language settings. We continue with the default values, and enter the More Options menu by entering 3. For example, AIX 5L Version 5.3 installs JFS2 file system as the default. Figure 3-17 shows the Install Options screen.

```
                             Install Options

 1.  Enable Trusted Computing Base................................... No
 2.  Enable CAPP and EAL4+ Technology................................ No
     (English only, 64-bit kernel enablement, JFS2 file systems)
 3.  Enable 64-bit Kernel............................................ Yes
 4.  Create JFS2 File Systems........................................ Yes
 5.  Graphics Software................................................ Yes
 6.  Enable System Backups to install any system..................... Yes
     (Installs all devices and kernels)




>>> 7.  Install More Software

    0  Install with the current settings listed above.

    88  Help ?
    99  Previous Menu

>>> Choice [7]:
```

Figure 3-17 Install Options

19. Choose the different options by entering the number in front of them. Press 0 and Enter to confirm. A summary of your configuration will be shown. Confirm your choice by entering 1 and the installation begins, as shown in Figure 3-18.

```
                    Installing Base Operating System



        Please wait...









        Approximate     Elapsed time
      % tasks complete   (in minutes)


          7                0      Restoring base operating system
```

Figure 3-18 Installation of AIX 5L in progress

20. During installation, you are asked to change the volume number of the CD from one to another:

```
installp:  Please insert volume 2 into device /dev/cd0 and press Enter
           to continue or enter "q" to quit.
```

Repeat the previous step if a CD change is required.

21. After the installation, the system reboots automatically and the system starts from the new boot device, which is, in our example, `/pci@800000020000002/pci@2,4/pci1069,b166@1/scsi@0/sd@3:2`, as shown in Figure 3-19 on page 45. Notice the difference with the bootfile.exe in step 10.

```
IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM
IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM
IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM
IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM
IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM
IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM
IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM
-
Elapsed time since release of system processors: 1022 mins 14 secs







-------------------------------------------------------------------------------
                                Welcome to AIX.
                       boot image timestamp: 14:55 11/18
                 The current time and date: 14:57:25 11/18/2005
               number of processors: 2    size of memory: 3792MB
    boot device: /pci@800000020000002/pci@2,4/pci1069,b166@1/scsi@0/sd@3:2
                     kernel size: 13352175; 64 bit kernel
-------------------------------------------------------------------------------
```

Figure 3-19   AIX 5L startup screen

22. Choose the terminal type and press Enter, as shown in Figure 3-20. For a graphical console, choose LFT. If your terminal type is not listed, choose vt100.

```
                            Set Terminal Type
The terminal is not properly initialized.  Please enter a terminal type
and press Enter.  Some terminal types are not supported in
non-English languages.

     ibm3101          tvi912          vt330          aixterm
     ibm3151          tvi920          vt340          dtterm
     ibm3161          tvi925          wyse30         xterm
     ibm3162          tvi950          wyse50         lft
     ibm3163          vs100           wyse60         sun
     ibm3164          vt100           wyse100
     ibmpc            vt320           wyse350

                      +-----------------------Messages------------------------
                      | If the next screen is unreadable, press Break (Ctrl-c)
    88  Help ?        | to return to this screen.
                      |
>>> Choice []:
```

Figure 3-20   Terminal selection screen

23. In the screen shown in Figure 3-21, choose Accept License Agreements (after a careful reading) and press Enter. Change the value to yes and press Enter.

```
                          Software License Agreements

Move cursor to desired item and press Enter.

  Show Installed License Agreements
  Accept License Agreements










F1=Help              F2=Refresh           F3=Cancel            Esc+8=Image
Esc+9=Shell          Esc+0=Exit           Enter=Do
```

Figure 3-21   License agreement

24. Use F3 (or Escape - 3), to return back to the main Installation Assistant screen, as displayed in Figure 3-22 on page 47. This configuration assistant can be started again later using the SMIT tool **`smitty assist`**. More information about SMIT is in 5.2, “System Management Interface Tool” on page 130.

```
                         Installation Assistant

Move cursor to desired item and press Enter.

  Set Date and Time
  Set root Password
  Configure Network Communications
  Install Software Applications
  Using SMIT (information only)
  Tasks Completed - Exit to Login











F1=Help            F2=Refresh         F3=Cancel          Esc+8=Image
Esc+9=Shell        Esc+0=Exit         Enter=Do
```

Figure 3-22 Installation Assistant main menu

25. Next, the window displays a login prompt, as shown in Figure 3-23, and you are asked to log in to system. The first time you log in as the root user without a password, or with the password you set up in the installation assistant.

```
AIX Version 5
(C) Copyrights by IBM and by others 1982, 2005.
Console login: root
*******************************************************************************
*                                                                             *
*                                                                             *
*  Welcome to AIX Version 5.3!                                                *
*                                                                             *
*                                                                             *
*  Please see the README file in /usr/lpp/bos for information pertinent to    *
*  this release of the AIX Operating System.                                  *
*                                                                             *
*                                                                             *
*******************************************************************************
Last login: Fri Nov 18 09:42:38 CST 2005 on /dev/vty0

#
```

Figure 3-23 Console login prompt

At this time, the BOS installation procedure is finished. Now you have to configure the system on various domains:

- System environment: Paging space, volume groups, and so on. See 5.10, “Paging space” on page 189 and 6.3, “Volume groups” on page 226 for more information.
- Network configuration: Host name, network settings, and so on. See 5.5, “Network configuration” on page 154 for more information.

3.3.3 Procedure to erase a hard disk

If you need to process hard disks with sensitive data, you can use a new option that allows the data to be erased.

From within the maintenance menu and also together with the BOS installation, you have the choice to overwrite hard disks with selectable binary patterns. If you want your disks erased at installation time, the process erases your hard disks and continues installation afterwards. Otherwise, the erasure process reaches 100 percent completion and stops.

The same erase function is available using the **diag** command. There is more about the **diag** command in “The diag command” on page 384.

3.4 Network Installation Management

Network Installation Management (NIM) allows you to manage an automated installation of the Base Operating System (BOS) and optional software on one or more machines.

You can install a group of machines with a common configuration or customize an installation for your specific needs. The number of machines you can install simultaneously depends on the throughput of your network, the disk access throughput of the installation servers, and the platform type of your servers.

The NIM environment includes client and server machines. A server provides resources (for example, files and programs required for installation) to another machine. A machine that is dependent on a server to provide resources is known as a client. Any machine that receives NIM resources is a client, although the same machine can also be a server in the overall network environment.

Most installation tasks in the NIM environment are performed from one server named the *master*. A set of installation tasks can also be performed from NIM clients. Once the network installation setup is complete, users of stand-alone clients can, from the client, install software that is available on NIM servers.

3.4.1 NIM machines

The types of machines that can be managed in the NIM environment are stand-alone, diskless, and dataless clients.

Stand-alone NIM clients can be booted and operated from local resources. They mount all file systems from local disks and have a local boot image. Stand-alone clients are not dependent upon network servers for operation.

Diskless and dataless clients are machines that are not capable of booting and running without the assistance of servers on a network. As their names imply, diskless clients have no hard disk, and dataless clients have disks that are unable to hold all the data that may be required for operation. Diskless machines must mount paging space and all file systems from remote servers. Dataless machines can only use a local disk for paging space and the /tmp and /home file systems. Neither diskless nor dataless clients have a local boot image, and they must boot from servers in the network.

3.4.2 NIM roles

The NIM environment is composed of two basic machine roles: master and client. The NIM master manages the installation of the rest of the machines in the NIM environment. The master is the only machine that can remotely run NIM commands on the clients. All other machines participating in the NIM environment are clients to the master, including machines that may also serve resources.

3.4.3 NIM resources

All operations on clients in the NIM environment require one or more resources. NIM resource objects represent files and directories that are used to support some type of NIM operation. Because NIM resources are ordinary file system objects in the operating system, most of them are provided to clients with standard Network File System (NFS) software. This means that many resources must reside locally on the servers providing these resources, because NFS can only export file system objects that are stored on local media in the machines from which they are exported. Table 3-2 lists the NIM resources.

Table 3-2 Definition of the NIM resources

NIM resource	Description
adapter_def	Represents a directory that contains secondary adapter configuration files that are used during bos_inst and cust operations.

NIM resource	Description
boot	An internally managed NIM resource used to indicate that a boot image has been allocated to a client.
bosinst_data	Represents a file that contains information for the BOS installation program. Normally, the BOS installation program looks for this information in the /bosinst.data file in the BOS installation image.
dump	Represents a directory in which client dump directories are maintained.
exclude_files	Represents a file that contains a list of files and directories that should be excluded when creating a system backup image.
fb_script	Represents a file that is used to configure devices when a NIM client is booting for the first time after the BOS installation process is completed.
fix_bundle	Represents a file containing fix keywords to be used by the **instfix** command, which is called by the NIM cust and fix_query operations. NIM mounts the fix_bundle resource on the client so it can be used by the local **instfix** command. NIM automatically unmounts the resource when the operation has completed.
home	Represents a directory in which client /home directories are maintained.
image_data	Represents a file that contains information for the BOS installation program. This information describes how physical disks and file systems should be configured in the root volume group during installation.
installp_bundle	Represents a file that contains the names of filesets that should be managed by NIM.
lpp_source	Represents a directory in which software installation images are stored.
mksysb	Represents a file that is a system backup image created using the **mksysb** command. This type of resource can be used as the source for the installation of a client.
nim_script	An internally-managed NIM resource used to indicate that a script should be run by NIM as part of a NIM operation.
paging	Represents a directory where client paging files are maintained.

NIM resource	Description
resolv_conf	Represents a file containing valid /etc/resolv.conf entries that define Domain Name Protocol nameserver information for local resolver routines.
root	Represents a directory in which client root directories are maintained.
shared_home	Represents a directory that can be used as a common /home directory by one or more clients.
SPOT	The Shared Product Object Tree (SPOT) is a fundamental resource in the NIM environment. It is required to install or initialize all types of machine configurations.
tmp	Represents a directory where client /tmp files are maintained.

3.4.4 Using EZ NIM

The SMIT EZ NIM feature organizes the commonly used NIM operations and simplifies frequently used advanced NIM operations.

Features of SMIT EZ NIM include:

- Task-oriented menus.
- Automatic resource naming that includes the level of the software used to create NIM resources.
- A review of what steps will take place before executing a task, whenever possible.

Use the **smitty eznim** fast path to open the EZ NIM main menu. If the NIM environment has not been set up on your system, the EZ NIM main menu will appear as shown in Figure 3-24.

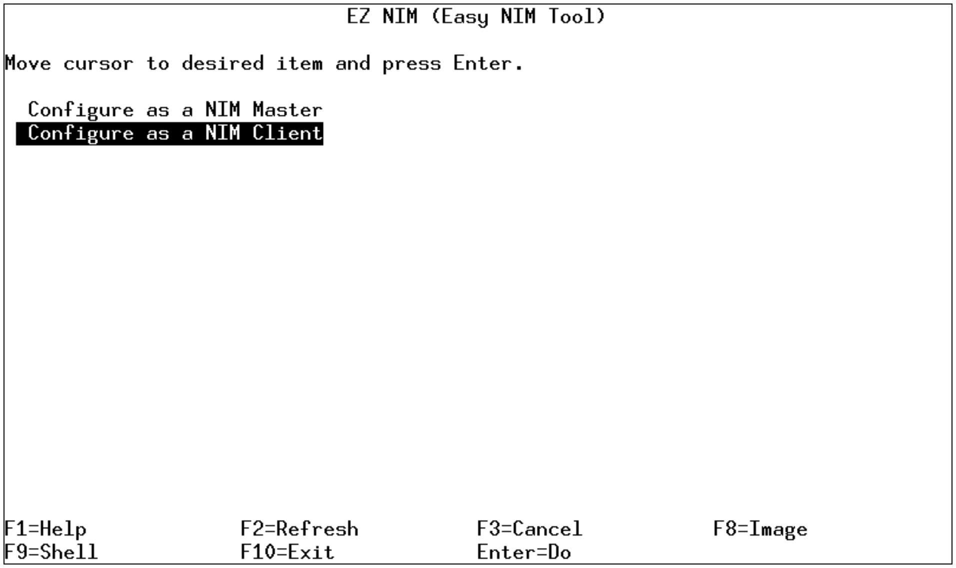


Figure 3-24 EZNIM main menu

Using EZ NIM to configure a NIM master

Follow these steps to configure your current system as a NIM master. Select Configure as a NIM Master and the options shown in Figure 3-25 on page 53 will appear.

```
                            Configure as a NIM Master

Move cursor to desired item and press Enter.

  Setup the NIM Master environment
  Enable Cryptographic Authentication
  Add fixes to the NIM Master environment
  Add client to the NIM environment

  Update clients
  Backup a client
  Reinstall clients
  Reset clients

  Show the NIM environment
  Verify the NIM environment
  Remove NIM environment




F1=Help             F2=Refresh          F3=Cancel           F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 3-25 Configure as a NIM Master menu options from EZNIM

- To configure your current system as a NIM master, select Setup the NIM Master environment. You can select the software source to configure from, the volume group to use for the NIM resources, and the file system to use for the NIM resources. When the NIM master environment is configured, the basic NIM resources are created.
- To view the NIM resources created by EZ NIM, select Show the NIM environment, or run the **lsnim** command on the NIM master.
- To configure your NIM master for SSL authentication, select Enable Cryptographic Authentication. This option allows you to install and configure the cryptographic software in the OpenSSL RPM package. After you configure OpenSSL, NIM clients with OpenSSL installed can request cryptographic authentication during service requests from the NIM master.
- To install updates and maintenance level packages to the NIM master, select Add fixes to the NIM Master environment. This option performs an update installation of a specified set of fixes on the default SPOT resource. A second SPOT resource containing the newly installed fixes is created by this operation. You can optionally select to update all your NIM clients during this operation.
- To update a client using EZNIM, select Update clients. This option allows you to perform an update_all operation on a selected client (or clients) using an lpp_source resource.

- To back up a client using EZ NIM, select Backup a client. This option allows you to create a system backup image of a selected client and store the backup image on the NIM master.
- To reinstall a client using EZ NIM, select Reinstall clients. This option allows you to perform a **mksysb** restore or a native RTE install on a selected client (or clients). You must then select a system backup image to restore or an lpp_source to install and decide whether to reboot and install the client now.
- To reset a NIM client to the ready state, select Reset clients. This option resets the state of a client or clients in the NIM environment. Use this option after a NIM operation has failed, and you want to return the client to the ready state.

Using EZ NIM to configure a NIM client

Follow these steps to configure a NIM client with EZ NIM.

- On a client system, use the **smitty eznim** fast path. Select Configure as a NIM Client, and the options displayed in Figure 3-26 on page 55 appear.
- To define your client in the NIM environment, select Add this system to a NIM environment.
- To configure your NIM client for SSL authentication, select Configure Client Communication Services. This option allows you to install and configure the cryptographic software in the OpenSSL RPM package. After you configure OpenSSL, you can select nimsh as the communication protocol used by the client. Any incoming NIM master service requests are then authenticated through SSL socket connections.

```
                           Configure as a NIM Client

Move cursor to desired item and press Enter.

  Add this system to a NIM environment
  Configure Client Communication Services
  Update this system
  Reinstall this system
  Reset this system
  Show the NIM environment















F1=Help             F2=Refresh          F3=Cancel           F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 3-26 Configure as a NIM Client menu options from EZNIM

- To update your client, select Update this system. This option allows you to perform an update_all operation on your client using an lpp_source resource.
- To reinstall your client, select Reinstall this system. This option allows you to perform a **mksysb** command format restore or native, RTE install on a selected client (or clients). You must then select a system backup image to restore or an lpp_source to install and decide whether to reboot and install the client now.
- To reset your client in the NIM environment, select Reset this system. This option resets the state of the client in the NIM environment. Use this option after a NIM operation has failed, and you want to return the client to the ready state.
- To view the default resources in the EZ NIM environment, select Show the NIM environment. The resources are defined using EZ NIM Master Operations.

3.5 Alternate disk installation

Alternate disk installation, available starting with AIX Version 4.3, allows for system installation on a system while it is still up and running. Install or upgrade downtime decreases considerably. It also allows large facilities to manage an upgrade, because systems can be installed over a longer period of time, while the systems are still running at the existing version. The switch over to the new version can then happen with a simple reboot, with the possibility to roll back to the original situation in case of problems. This facility has received many updates throughout the history of AIX and AIX 5L since its introduction.

Alternate disk installation can be used in one of two ways:

- Cloning the current running rootvg to an alternate disk.

  > **Note:** In this way, you have an online backup available, at the cost of one or more physical disks.

- Installing a mksysb image on another disk.

For more information about alt_disk_install for AIX 5L Version 5.2 and earlier, refer to the respective installation guide.

3.5.1 The alt_disk_install command reference

In AIX 5L Version 5.2 and prior, you used the **`alt_disk_install`** command to cover the alternate disk installation needs. In AIX 5L Version 5.3, this command is obsolete.

3.5.2 New alt_disk_install features in AIX 5L Version 5.3

AIX 5L Version 5.3 has implemented a number of changes to make the **`alt_disk_install`** operations easier to use, document, and maintain.

The following functional changes have been implemented:

- **`alt_disk_install`** has been partitioned into separate modules with separate syntax based on operation and function.
- A library of common functions that can be accessed by the modules has been implemented.
- Error checking and robustness of existing **`alt_disk_install`** operations has been improved.
- Documentation has been improved by creating a separate man page for each module (in the past, there was only one large man page).

The following three new commands have been added:

- **alt_disk_copy** creates copies of rootvg on an alternate set of disks.
- **alt_disk_mksysb** installs an existing mksysb on an alternate set of disks.
- **alt_rootvg_op** performs Wake, Sleep, and Customize operations.

The **alt_disk_install** module will continue to ship as a wrapper to the new modules. However, it will not support any new functions, flags, or features.

3.5.3 Running alternate disk installation using SMIT

To run alternate mksysb installation, perform the following steps:

- At the system prompt, type the **smitty alt_mksysb** fast path.
- Type or select values in the entry fields and press Enter to submit them.

To run alternate rootvg cloning, perform the following steps:

- At the system prompt, type the **smitty alt_clone** fast path.
- Type or select values in the entry fields and press Enter to submit them.

Once the process is finished, the system reboots from the alternate disk, either as **mksysb** or cloned rootvg.

3.5.4 How to rollback alt_disk_install

If the new environment has problems, you can easily rollback by changing the bootlist to point of the original boot disk.

3.6 Cloning an AIX 5L system

Cloning the rootvg to an alternate disk has many advantages. One advantage is having an online backup available, as in the case of a disk crash. Keeping an online backup requires an extra disk or disks to be available on the system.

Another benefit of rootvg cloning occurs when applying new maintenance levels or updates to a system. A copy of the rootvg is made to an alternate disk, then the updates are applied to that copy. The system runs uninterrupted during this process. When it is rebooted, the system boots from the newly updated rootvg for testing. If updates cause problems, the old_rootvg can be retrieved by resetting the bootlist and then rebooting.

With a **mksysb** image, you can clone one system image onto multiple target systems. However, the target systems might not contain the same hardware devices or adapters, or require the same kernel as the source system. All devices and kernels are automatically installed during a BOS installation. As a result, when you create a system backup, the mksysb image contains all the device and kernel support. For example, you can create a system backup from System_A and install System_A's mksysb image onto System_B without having to use product media to boot System_B.

By default, using the **alt_disk_install** command does the following:

1. Creates an /image.data file based on the current rootvg's configuration. A customized image.data file can be used.
2. Creates an alternate rootvg (altinst_rootvg).
3. Creates logical volumes and file systems with the alt_inst prefix.
4. Generates a backup file list from the rootvg, and if an exclude.list file is given, those files are excluded from the list.
5. Copies the final list to the altinst_rootvg's file systems.
6. If specified, the **installp** command installs updates, fixes, or new filesets into the alternate file system.
7. The **bosboot** command creates a boot logical volume on the alternate boot disk.
8. If a customization script is specified, it runs at this point.
9. The file systems are then unmounted, and the logical volumes and file systems are renamed.
10. The logical volume definitions are exported from the system to avoid confusion with identical ODM names, but the altinst_rootvg definition is left as an ODM placeholder.
11. By default, the bootlist is set to the new cloned rootvg for the next reboot.

If you are performing a clone installation, device information will not be restored to the target system by default. During a clone installation, the BOS installation process verifies that the **mksysb** image is from the system you are trying to install. If the target system and the **mksysb** image are different, the device information is not recovered. This behavior is determined by the RECOVER_DEVICES variable in the bosinst.data file. This variable can be set to Default, yes, or no. The following list shows the resulting behaviors for each value:

```
Default
No recovery of devices
yes
Attempted rebuild of ODM
no
No recovery of devices
```

Note: You can override the default value of RECOVER_DEVICES by selecting yes or no in the Backup Restore menu or by editing the value of the attribute in the bosinst.data file.

If the source system does not have the correct passwords and network information, you can make modifications on the target system now. Also, some products (such as graPHIGS) ship device-specific files. If your graphics adapter is different on the target system, verify that the device-specific filesets for graphics-related LPPs are installed.

After the **mksysb** backup installation completes, the installation program automatically installs additional devices and the kernel (uniprocessor or microprocessor) on your system using the original product media you booted from. Information is saved in BOS installation log files. To view BOS installation log files, enter **cd /var/adm/ras** and view the devinst.log file in this directory.

3.7 Troubleshooting your installation

To assist you in problem determination or validation of an install, there are log files that can be viewed after the installation process.

3.7.1 Viewing BOS installation logs

Information based on BOS installation log files might help you determine the cause of installation problems.

To view BOS installation log files, run **cd /var/adm/ras** and view the files in this directory. One example is the devinst.log file, which is a text file that can be viewed with any text editor or paged through.

3.7.2 Viewing BOS installation logs using SMIT

You can use SMIT to view some logs in the /var/adm/ras directory.

To view logs in the /var/adm/ras directory, use the following SMIT fast path:

```
# smit alog_show
```

Figure 3-27 contains all logs that are viewable with the **alog** command. Select bosinst from the list by pressing the F4 key, and then the **alog** file name will be shown as /var/adm/ras/bosinstlog.

```
                                COMMAND STATUS

Command: OK            stdout: yes           stderr: no

Before command completion, additional instructions may appear below.

[TOP]
Preparing target disks.
hdisk0 changed
hdisk1 changed
rootvg
Making boot logical volume.
hd5
Making paging logical volumes.
hd6
Making logical volumes.
hd8
hd4
hd2
[MORE...69]

F1=Help             F2=Refresh          F3=Cancel           F6=Command
F8=Image            F9=Shell            F10=Exit            /=Find
n=Find Next
```

Figure 3-27 Viewing the alog file from within SMIT

3.7.3 Viewing BOS installation logs with the alog command

You can use the **alog** command to view logs in the /var/adm/ras directory.

To view the bosintlog file in the /var/adm/ras directory, run:

```
# alog -o -f bosinstlog
Preparing target disks.
hdisk0 changed
hdisk2 changed
rootvg
Making boot logical volume.
hd5
Making paging logical volumes.
hd6
Making logical volumes.
hd8
hd4
hd2
hd9var
hd3
hd1
```

```
hd10opt
Forming the jfs log.
Making file systems.
File system created successfully.
130864 kilobytes total disk space.
File system created successfully.
130864 kilobytes total disk space.
File system created successfully.
130864 kilobytes total disk space.
File system created successfully.
130864 kilobytes total disk space.
File system created successfully.
130864 kilobytes total disk space.
File system created successfully.
130864 kilobytes total disk space.
Mounting file systems.
Restoring base operating system
Initializing disk environment.
Over mounting /.
Copying Cu* to disk.
Installing additional software.
mkitab: ident entry found in /etc/inittab
Please wait...
Some locale or message software did not install.
See /var/adm/ras/devinst.log for details.
lft0 changed
Initializing dump device.
fwdump
File system created successfully.
393000 kilobytes total disk space.
New File System size is 786432
primary             /dev/hd6
secondary           /dev/sysdumpnull
copy directory      /var/adm/ras
forced copy flag    TRUE
always allow dump   FALSE
dump compression    ON
Creating boot image.
bosboot: Boot image is 22967 512 byte blocks.
Running Customization
Please wait...
```

3.7.4 Installation LED codes

Installation LED codes provide vital information that indicates what step is taking place early on in an installation or system boot.

A list of useful LED codes and their specific explanation when installing on an AIX 5L operating system are as follows:

c40	Configuration files are being restored.
c41	Could not determine the boot type or device.
c42	Extracting data files from diskette.
c43	Cannot access the boot/install tape.
c44	Initializing installation database with target disk information.
c45	Cannot configure the console.
c46	Normal installation processing.
c47	Could not create a physical volume identifier (PVID) on disk.
c48	Prompting you for input.
c49	Could not create or form the JFS log.
c50	Creating root volume group on target disks.
c51	No paging devices were found.
c52	Changing from RAM environment to disk environment.
c53	Not enough space in the /tmp directory to do a preservation installation.
c54	Installing either BOS or additional packages.
c55	Could not remove the specified logical volume in a preservation installation.
c56	Running user-defined customization.
c57	Failure to restore BOS.
c58	Displaying message to turn the key.
c59	Could not copy either device special files, device ODM, or volume group information from RAM to disk.
c61	Failed to create the boot image.
c62	Loading platform dependent debug files.
c63	Loading platform dependent data files.
c64	Failed to load platform dependent data files.

3.8 AIX 5L installation packages

Software products include those shipped with AIX 5L and those purchased separately. Each software product can contain separately installable parts. The following sections explain how software products are organized.

3.8.1 Filesets

A fileset is the smallest installable base unit for the AIX 5L operating system. A fileset includes all files that constitute a complete product, such as bos.net.uucp, or a separately installable part of a product, such as bos.net.nfs.client.

3.8.2 Packages

A package is a group of separately installable filesets that provide a set of related functions. For example, bos.net is a package.

3.8.3 Licensed Program Products

A Licensed Program Product (LPP) is a complete software product including all packages associated with that licensed program. For example, the BOS is a licensed program.

3.8.4 Bundles

A bundle is a list of software that can contain filesets, packages, and LPPs that are suited for a particular use, such as providing personal productivity software or software for a client machine in a network environment. Bundles that are used by default for the system are stored in /usr/sys/inst.data/sys_bundles. Bundles that are user-created are stored in /usr/sys/inst.data/user_bundles. The system defined bundles in AIX 5L are:

Server Bundle A collection of software packages for machines running AIX 5L in a multiuser stand-alone or networked environment. This bundle emphasizes functionality over disk utilization.

Graphics Bundle A collection of software packages that provides support of graphical environments. Graphical support may be automatically installed on some systems during BOS installation.

Migration Bundle This bundle is created when there was not enough disk space available to complete a migration installation during the BOS installation process. The bundle consists of a collection of software packages that must be installed to complete your migration. You must install this bundle to complete the migration installation. Install the bundle using the **smit update_all** fast path.

3.8.5 PTFs and APARs

PTF is an acronym for Program Temporary Fix. A PTF is an updated fileset or a new fileset that fixes a previous system problem. PTFs are installed in the same way as regular filesets by the use of the **installp** command, as described in 3.3, “Base Operating System installation” on page 31.

APAR is an acronym for Authorized Program Analysis Report. An APAR is an emergency fix, or interim fix, to a unique problem on the system. APARs will eventually become PTFs after testing and verification. APARs are applied to the system through the use of the **instfix** command, as described in 3.11.3, “Displaying and updating installed software to the latest level” on page 79.

3.9 Software maintenance

The following sections discuss filesets, packages, and the software installation process for additional software required on a system.

3.9.1 Managing filesets

A software product installation package is a backup-format file containing the files of the software product, required installation control files, and optional installation customization files. The **installp** command is used to install and update software products.

An installation package contains one or more separately installable, logically grouped units called *filesets*. Each fileset in a package must belong to the same product.

A fileset update or update package is a package containing modifications to an existing fileset.

Fileset revision level identification

The fileset level is referred to as the level or, alternatively as the v.r.m.f or VRMF and has the form:

```
Version.Release.Modification.FixLevel
```

where:

- Version is a numeric field of one to two digits that identifies the version number.
- Release is a numeric field of one to two digits that identifies the release number.
- Modification is a numeric field of one to four digits that identifies the modification level.
- FixLevel is a numeric field of one to four digits that identifies the fix level.

A base fileset installation level is the full initial installation level of a fileset. This level contains all files in the fileset, as opposed to a fileset update, which may contain a subset of files from the full fileset.

Note: Fileset level precedence reads from left to right (for example, 5.3.0.0 is a newer level than 5.2.0.0).

3.9.2 Understanding maintenance levels

Once you have installed the base operating system, you can determine the maintenance level with the **oslevel** command.

The general syntax of the **oslevel** command is as follows:

`oslevel [ -l` *Level* `| -g` *Level* `| -q ] [ -r ] [ -f ]`

A brief description of the **oslevel** command flags is given in Table 3-3.

Table 3-3 Commonly used flags for the oslevel command

Flag	Description
-l *Level*	Lists filesets at levels earlier than the maintenance levels specified by the Level parameter.
-g *Level*	Lists filesets at levels later than the current maintenance level.
-q	Lists names of known maintenance levels that can be specified using the -l flag.
-r	Applies all flags to Recommended Maintenance Levels
-f	Forces cache rebuilt

To show the current maintenance level of your system, use the **oslevel** command as follows:

```
# oslevel -r
5300-03
```

The 03 on the end indicates that this system is running with the Recommended Maintenance Level three. Do not confuse this output with the M in V.R.M.F. (Version.Release.Maintenance.Fixlevel).

3.9.3 Installing software

Software can be installed in one of two states: applied or committed. The applied state places software on the system and retains the previous version of the software. When an update is in the applied state, the previous version is stored in the /usr/lpp/*PackageName* directory, where *PackageName* could be bos.net which is a package. This process is useful for deploying or testing new software, where it may be necessary to go back to the previous version of the software in case of errors.

The committed state places software on the system and removes all previous levels of the software from the /usr/lpp/*PackageName* directory. If committed software needs to be removed, you cannot go back to the previous version without a complete reinstall of the previous version software. We recommend you install new software in the applied state, rather than the committed state, until the new software has been thoroughly tested.

Software can be installed either through the command line or through SMIT.

> **Important:** The Web-based System Manager is an intuitive, easy to use tool that can perform every major system administration task available on AIX 5L. It is not mentioned in this redbook because it is not the focus of the certification exam. However, the authors of this redbook completely advocate its use and highly recommend careful attention and exploration of this useful tool. To start the tool, use the **wsm** command. It requires either a PC client for remote access, or an X-Window environment if run on a native system console.

In the command line mode, software is installed by using the **installp** command. The command syntax for the **installp** command to install software in an applied state is:

```
installp -a [ -eLogFile ] [ -V Number ] [ -dDevice ] [ -b ] [ -S ] [ -B ]
[ -D ] [ -I ] [ -p ] [ -Q ] [ -q ] [ -v ] [ -X ] [ -F | -g ] [ -O { [ r ] [ s ]
[ u ] } ] [ -tSaveDirectory ] [ -w ] [ -zBlockSize ] { FilesetName [ Level ]...
| -f ListFile | all }
```

To install software in a committed state, the command syntax is:

```
installp -ac [ -N ] [ -eLogFile ] [ -V Number ] [ -dDevice ] [ -b ] [ -S ]
[ -B ] [ -D ] [ -I ] [ -p ] [ -Q ] [ -q ] [ -v ] [ -X ] [ -F | -g ]
[ -O { [ r ] [ s ] [ u ] } ] [ -tSaveDirectory ] [ -w ]
[ -zBlockSize ] { FilesetName [ Level ]... | -f ListFile | all }
```

For example, to install all filesets within the bos.net software package in /usr/sys/inst.images directory in the applied state, with a checksum check, enter:

```
# installp -avX -d/usr/sys/inst.images bos.net
```

To preview an install of all filesets within the bos.net software package in /usr/sys/inst.images directory in the committed state and to check for disk space requirements, enter:

```
# installp -acpX -d/usr/sys/inst.images bos.net
```

Only a couple of flags are provided for illustration purposes.

Under the RESOURCES section in the output, you will see something similar to:

```
RESOURCES
---------
  Estimated system resource requirements for filesets being installed:
                (All sizes are in 512-byte blocks)
      Filesystem                      Needed Space             Free Space
      /                                      1150                 17624
      /usr                                  54183                 48016
      /var                                      8                 22424
      /tmp                                    300                 63280
      -----                              --------                ------
      TOTAL:                                55641                151344

  NOTE:  "Needed Space" values are calculated from data available prior
  to installation.  These are the estimated resources required for the
  entire operation.  Further resource checks will be made during
  installation to verify that these initial estimates are sufficient.
```

As shown, the /usr file system does not have enough free space for the installation, and the installation would fail.

Note: If you try to run two **installp** commands at a time from the same installation medium, it will fail with an error similar to:

```
0503-430 installp: Either there is an installp process currently running or
there is a previously failed installation. Wait for the process to complete
or run installp -C to cleanup a failed installation.
```

A record of the **installp** output can be found in /var/adm/sw/installp.summary. The following is a sample of the file:

```
# cat /var/adm/sw/installp.summary
0:bos.net.ppp:5:U:5.1.0.0:
0:bos.net.ipsec.rte:5:U:5.1.0.0:
0:bos.net.ppp:5:R:5.1.0.0:
0:bos.net.ipsec.rte:5:R:5.1.0.0:
```

3.9.4 Committing applied updates

The command syntax for the **installp** command to commit applied updates is:

`installp -c [ -e`*LogFile* `] [ -V`*Number* `] [ -b ] [ -g ] [ -p ] [ -v ] [ -X ]`
`[ -O { [ r ] [ s ] [ u ] } ] [ -w ] {` *FilesetName* `[` *Level* `]... | -f` *ListFile* `|`
`all }`

For example, to commit all updates, enter:

```
# installp -cgX all
```

Running this command will commit all the updates and will remove the filesets for the previous version.

3.9.5 Rejecting applied updates

The command syntax for the **installp** command to reject the updates that are in the applied state is:

`installp -r [ -e`*LogFile* `] [ -V`*Number* `] [ -b ] [ -g ] [ -p ] [ -v ] [ -X ]`
`[ -O { [ r ] [ s ] [ u ] } ] [ -w ] {` *FilesetName* `[` *Level* `]... | -f` *ListFile* `}`

For example, to reject all applied updates listed in the file ./reject.list, enter:

```
# installp -rBfX ./reject.list
```

Running this command will remove all the uncommitted updates listed in ./reject.list and bring the system back to the previous maintenance level.

3.9.6 Removing installed software

If you want to remove an installed product, that is, remove all files that belong to that software from the system, use the **installp** command; the command syntax is:

`installp -u [ -e`*LogFile* `] [ -V`*Number* `] [ -b ] [ -g ] [ -p ] [ -v ] [ -X ]`
`[ -O { [ r ] [ s ] [ u ] } ] [ -w ] {` *FilesetName* `[` *Level* `]... | -f` *ListFile* `}`

For example, to preview a remove of bos.net.ipsec.rte and its dependents, with a verbose display of all successes, warnings, and failures, enter:

```
# installp -ugp -V2 bos.net.ipsec.rte
```

Running this command will give you a list of files that will be removed, but will not actually remove them.

3.9.7 Cleaning up after failed installations

If an installation fails, the **installp** command will not be able to install the same software until you have removed those files that succeeded in installing prior to the failure. You can use the **installp** command as follows:

installp -C [-b] [-e*LogFile*]

For example, if all the prerequisites in an installation are not met, the **installp** command might fail. You will not be able to reinstall the product until you have done a cleanup. To do this, run:

```
# installp -C
```

This will remove all the files installed in the failed installation.

3.9.8 Listing all installable software on media

To see what software is available on a particular media, the command syntax for the **installp** command is:

installp { -l | -L } [-e*LogFile*] [-d *Device*] [-B] [-I] [-q]
[-z*BlockSize*] [-O { [s] [u] }]

For example, to list the software that is on your CD-ROM, enter:

```
# installp -L -d /dev/cd0
```

3.10 Installing optional software and service updates

Once you have installed the base operating system, only a limited number of filesets are installed on your system. To install additional software, you can use SMIT, the Web-based System Manager, or the command line. If you decide to use the command line to install your software, you should be familiar with the **installp** command.

3.10.1 Using SMIT for software maintenance

Software installation, un-installation, and maintenance tasks can also be performed through the SMIT menus. SMIT uses the **installp** command to perform these tasks.

> **Note:** SMIT stores a record of software installation, removal, and maintenance tasks in /var/adm/sw/installp.log like the **installp** command, but SMIT also stores a more detailed record in $HOME/smit.log.

Software installation

To install software products:

1. Use the SMIT fast path **smitty install_latest**.

   A screen similar to Figure 3-28 is shown.

```
                Install and Update from LATEST Available Software

Type or select a value for the entry field.
Press Enter AFTER making all desired changes.

                                                    [Entry Fields]
* INPUT device / directory for software            []                     +











F1=Help           F2=Refresh        F3=Cancel         F4=List
F5=Reset          F6=Command        F7=Edit           F8=Image
F9=Shell          F10=Exit          Enter=Do
```

Figure 3-28 Install and Update from LATEST Available Software menu

2. Enter the device name for installation in the INPUT device/directory for software field. A screen similar to Figure 3-29 on page 71 is shown.

```
                Install and Update from LATEST Available Software

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                      [Entry Fields]
* INPUT device / directory for software               /dev/cd0
* SOFTWARE to install                                [_all_latest]          +
  PREVIEW only? (install operation will NOT occur)    no                     +
  COMMIT software updates?                            yes                    +
  SAVE replaced files?                                no                     +
  AUTOMATICALLY install requisite software?           yes                    +
  EXTEND file systems if space needed?                yes                    +
  OVERWRITE same or newer versions?                   no                     +
  VERIFY install and check file sizes?                no                     +
  Include corresponding LANGUAGE filesets?            yes                    +
  DETAILED output?                                    no                     +
  Process multiple volumes?                           yes                    +



F1=Help              F2=Refresh           F3=Cancel            F4=List
F5=Reset             F6=Command           F7=Edit              F8=Image
F9=Shell             F10=Exit             Enter=Do
```

Figure 3-29 Install and Update from LATEST Available Software menu - more

3. In the SOFTWARE to install field, either enter the name, if you know what you have to install, or press F4 to get a list of all the available software. Press Enter once you have selected the products you want to install.
4. We recommend that you first verify that the software you are trying to install meets all the prerequisite and co-requisite requirements. It is a good practice to set the PREVIEW only? (install operation will NOT occur) field to YES. This will give you a detailed listing of whether your installation will be successful or not.
5. It is recommended that you accept the default values for the AUTOMATICALLY install requisite software (default YES) and EXTEND file systems (default YES) fields if space is needed. Your installation might fail if you instruct the **installp** command not to extend the file system. If it runs out of space, an error similar to the one shown below can be encountered:

   ```
   0503-008 installp: There is not enough free disk space in file system
   /usr (506935 more 512-byte blocks are required). An attempt to extend
   this file system was unsuccessful. Make more space available, then retry
   this operation.
   ```

6. Press Enter.
7. Read the error messages, if any, at the end of the command execution when the command status changes to failed. We recommend that you look at your smit.log even if the command status reports OK, since there may be filesets that you wanted to install that the system did not attempt to install.

Committing applied updates

To commit an applied software update:

1. Use the SMIT fast path **`smitty install_commit`**.

   A screen similar to Figure 3-30 is shown.

```
                Commit Applied Software Updates (Remove Saved Files)

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                        [Entry Fields]
* SOFTWARE name                                      [all]                    +
  PREVIEW only? (commit operation will NOT occur)     no                      +
  COMMIT requisites?                                  yes                     +
  EXTEND file systems if space needed?                yes                     +
  DETAILED output?                                    no                      +










F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 3-30 Commit Applied Software Updates (Remove Saved Files) menu

2. In the SOFTWARE to install field, either enter the name, if you know what you want to commit, or press F4 to get a list of all the available software. Press Enter once you have selected the products you want to commit. Leaving the SOFTWARE name field to all will commit all applied filesets installed on the system.
3. Press Enter. The system reports that the software is about to be committed, commits the software, and then removes the copies from the /usr/lpp/*PackageName* directory.

Rejecting applied updates

In order to reject a service update that you have installed:

1. Use the SMIT fast path **`smitty install_reject`**.

   A screen similar to Figure 3-31 is shown.

```
                Reject Applied Software Updates (Use Previous Version)

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                          [Entry Fields]
* SOFTWARE name                                        []                   +
  PREVIEW only? (reject operation will NOT occur)       no                  +
  REJECT dependent software?                            no                  +
  EXTEND file systems if space needed?                  yes                 +
  DETAILED output?                                      no                  +











F1=Help              F2=Refresh           F3=Cancel            F4=List
F5=Reset             F6=Command           F7=Edit              F8=Image
F9=Shell             F10=Exit             Enter=Do
```

Figure 3-31 Reject Applied Software Updates (Use Previous Version) menu

2. Press F4 on the SOFTWARE name field to select the software update you want to reject. All the software updates that are in the applied state will be listed. Select the update that you want to reject, and press Enter.

Removing installed software

1. You can remove installed and committed software by using the SMIT fast path **smitty install_remove**.

   A screen similar to Figure 3-32 is shown.

```
                        Remove Installed Software

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                        [Entry Fields]
* SOFTWARE name                                        []                    +
  PREVIEW only? (remove operation will NOT occur)       yes                  +
  REMOVE dependent software?                            no                   +
  EXTEND file systems if space needed?                  no                   +
  DETAILED output?                                      no                   +










F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 3-32 Remove Installed Software menu

2. Press F4 in the SOFTWARE name field to get a list of all the software that is installed on your system. Select the software you want to remove by pressing F7, followed by Enter, once you are done.
3. The PREVIEW only? (remove operation will NOT occur) field is yes by default. This allows you to preview any remove operations and confirm your choices before you actually do the remove action.
4. Once you are sure that you want to remove this software, change PREVIEW only? (remove operation will NOT occur) field to no, and press Enter. This will remove all the software that you have selected to be removed.

3.11 Maintaining optional software (applying updates)

Software that is distributed to fix a problem in a product is called an update. All software products have a version number and a release number that identify the release level of the product. In addition to this, product updates are assigned a modification level number and a fix level number to identify the level of the update. Suppose that you have your system currently running 5.3.0.0, and all the filesets are at the 5.3.0.0 maintenance level. Then IBM has just released a latest maintenance level for systems on 5.3.0.0. You have to upgrade your system to bring it to the latest maintenance level.

Bringing a system to the latest maintenance level involves a number of steps, as listed below:

- Listing the maintenance level of the software
- Downloading fixes
- Displaying and updating installed software to the latest level

3.11.1 Listing the maintenance level of software

The **lslpp** command displays information about installed filesets or fileset updates. The most common flags used with the **lslpp** command are listed in Table 3-4.

Table 3-4 Commonly used flags for the lslpp command

Flag	Description
-l	Displays the name, most recent level, state, and description of the specified fileset.
-f	Displays the names of the files added to the system during installation of the specified fileset.
-h	Displays the installation and update history information for the specified fileset.
-a	Displays all information about filesets specified when combined with other flags. Cannot be used with the -f flag.

In order to see what maintenance level your filesets are currently on, use the following command:

```
# lslpp -l
```

This will list all the software that is installed on your system showing the current maintenance level. The output will look similar to the following:

```
Fileset                      Level  State      Description
----------------------------------------------------------------------------
Path: /usr/lib/objrepos
  Java14.sdk                1.4.2.10  COMMITTED  Java SDK 32-bit
  Tivoli_Management_Agent.client.rte
                             3.7.1.0  COMMITTED  Management Framework Endpoint
                                                 Runtime"
  X11.Dt.ToolTalk            5.3.0.0  COMMITTED  AIX CDE ToolTalk Support
  X11.Dt.bitmaps             5.3.0.0  COMMITTED  AIX CDE Bitmaps
  X11.Dt.helpmin             5.3.0.0  COMMITTED  AIX CDE Minimum Help Files
  X11.Dt.helprun             5.3.0.0  COMMITTED  AIX CDE Runtime Help
  X11.Dt.lib                5.3.0.30  COMMITTED  AIX CDE Runtime Libraries
  X11.Dt.rte                5.3.0.30  COMMITTED  AIX Common Desktop Environment
                                                 (CDE) 1.0
  X11.adt.bitmaps            5.3.0.0  COMMITTED  AIXwindows Application
                                                 Development Toolkit Bitmap
                                                 Files
  X11.adt.imake             5.3.0.30  COMMITTED  AIXwindows Application
                                                 Development Toolkit imake
  X11.adt.include           5.3.0.30  COMMITTED  AIXwindows Application
                                                 Development Toolkit Include
                                                 Files
  X11.adt.lib                5.3.0.0  COMMITTED  AIXwindows Application
bos.64bit                 5.3.0.30  COMMITTED  Base Operating System 64 bit
                                                 Runtime
  bos.acct                  5.3.0.30  COMMITTED  Accounting Services
  bos.adt.base              5.3.0.30  COMMITTED  Base Application Development
                                                 Toolkit
  bos.adt.include           5.3.0.30  COMMITTED  Base Application Development
                                                 Include Files
  bos.adt.lib               5.3.0.30  COMMITTED  Base Application Development
                                                 Libraries
  bos.alt_disk_install.boot_images
                            5.3.0.30  COMMITTED  Alternate Disk Installation
                                                 Disk Boot Images
  bos.alt_disk_install.rte  5.3.0.30  COMMITTED  Alternate Disk Installation
                                                 Runtime
  bos.cdmount               5.3.0.30  COMMITTED  CD/DVD Automount Facility
  bos.diag.com              5.3.0.30  COMMITTED  Common Hardware Diagnostics
  bos.diag.rte              5.3.0.30  COMMITTED  Hardware Diagnostics
  bos.diag.util             5.3.0.30  COMMITTED  Hardware Diagnostics Utilities
  bos.help.msg.en_US.com    5.3.0.10  COMMITTED  WebSM/SMIT Context Helps -
                                                 U.S. English
```

To list the individual files that are installed with a particular fileset, use the following command:

```
# lslpp -f
```

For example, if you wanted to display all files installed with the bos.64bit fileset, you would enter:

```
# lslpp -f bos.64bit
  Fileset               File
-----------------------------------------------------------------------------
Path: /usr/lib/objrepos
  bos.64bit 5.3.0.30    /usr/lib/methods/cfg64
                        /usr/ccs/bin/shlap
                        /usr/ccs/bin/usla64
                        /usr/lib/drivers/syscalls64.ext
                        /usr/ccs/bin/usla
                        /usr/ccs/bin/shlap64

Path: /etc/objrepos
  bos.64bit 5.3.0.30    NONE
```

To list the installation and update history of filesets, use the following command:

```
# lslpp -h
```

For example, if you wanted to see when the bos.sysmgt.trace fileset was last updated, you would enter:

```
# lslpp -h bos.sysmgt.trace
  Fileset         Level     Action       Status       Date         Time
-----------------------------------------------------------------------------
Path: /usr/lib/objrepos
  bos.sysmgt.trace
                 5.3.0.30   COMMIT       COMPLETE     11/10/05     16:32:42

Path: /etc/objrepos
  bos.sysmgt.trace
                 5.3.0.30   COMMIT       COMPLETE     11/10/05     16:33:09
```

3.11.2 Downloading fixes

AIX 5L development is highly focused on implementing tools and functions that help to fulfill the IBM Autonomic Computing strategy. In a first approach, AIX 5L Version 5.2 provided proactive capabilities through the **compare_report** command. This command and its SMIT interface allow the comparison of installed software or fix repositories to a list of available fixes from the IBM support Web site, enabling system administrators to develop a proactive fix strategy.

AIX 5L Version 5.3 advances this process one step further and introduces automatic download, scheduling, and notification capabilities through the new Service Update Management Assistant (SUMA) tool. SUMA is fully integrated into the AIX 5L Base Operating System and supports scheduled and unattended task-based download of Authorized Program Analysis Reports (APARs), Program Temporary Fixes (PTFs), and recommended maintenance levels (MLs). SUMA can also be configured to periodically check the availability of specific new fixes and entire maintenance levels, so that the time spent on such system administration tasks is reduced. The SUMA implementation allows for multiple concurrent downloads to optimize performance and has no dependency on any Web browser.

All Service Update Management Assistant related tasks and functions are supported by SMIT menus and panels. The new main SUMA menu shown in Figure 3-33 can be directly accessed through the SMIT fast path **suma**. Alternatively, you can select the new Service Update Management Assistant (SUMA) option in either the Software Maintenance and Utilities or the Software Service Management menu. Both menus are listed under the Software Installation and Maintenance option of the SMIT top-level menu.

```
                   Service Update Management Assistant (SUMA)

Move cursor to desired item and press Enter.

  Download Updates Now (Easy)
  Custom/Automated Downloads (Advanced)
  Configure SUMA










F1=Help              F2=Refresh           F3=Cancel            F8=Image
F9=Shell             F10=Exit             Enter=Do
```

Figure 3-33 Service Update Management Assistant (SUMA) on SMIT

3.11.3 Displaying and updating installed software to the latest level

Once you have downloaded all the fixes into the /ptf directory, the next step is to install them and bring your system to the latest maintenance level. In this section, the following procedures are discussed:

- Displaying an individual fix (**instfix** command)
- Installing an individual fix by APAR
- Updating all filesets to the latest level

Displaying an individual fix (instfix command)

You can download an individual fix using FixDist following the same procedure given in 3.11.2, "Downloading fixes" on page 77.

In order to determine if a fix is installed on your system or to install a fix, use the **instfix** command. The general syntax of the **instfix** command is as follows:

```
instfix [ -T ] [ -s String ] [ -S ] [ -k Keyword | -f File ] [ -p ]
[ -d Device ] [ -i [ -c ] [ -q ] [ -t Type ] [ -v ] [ -F ] ] [ -a ]
```

A practical list of flags used with **instfix** command are given in Table 3-5.

Table 3-5 Commonly used flags for the instfix command

Flag	Description
-a	Displays the symptom text associated with a fix. Can be used with the -f, -i, and -k flags.
-d *Device*	Specifies the input device. Required for all flags except -i and -a.
-f *FileName*	Specifies the input file *FileName* containing keywords or fixes. The -T flag produces a suitable input file format for the -f flag.
-i	Displays whether fixes or keywords are installed.
-k *Keyword*	Specifies an APAR number or keyword to be installed. Multiple keywords can be entered. A list of keywords entered with the -k flag must be contained in quotation marks and separated with spaces.
-s *String*	Searches for and displays fixes on the media containing a specified string.
-T	Displays the list of fixes on the media.
-v	Used with the -i flag to specify verbose mode. Displays information about each fileset associated with a fix or keyword.

The **instfix** command allows you to install a fix or set of fixes without knowing any information other than the Authorized Program Analysis Report (APAR) number or other unique keywords that identify the fix.

A fix can have a single fileset or multiple filesets. Fix information is organized in the Table of Contents (TOC) on the installation media. After a fix is installed, fix information is kept on the system in a fix database.

To list fixes that are on a CD-ROM in /dev/cd0, enter the command:

```
# instfix -T -d /dev/cd0
IY73748
```

To determine if APAR IX75893 is installed on the system, enter the command:

```
# instfix -ik IY73748
Not all filesets for IY73748 were found.
```

To examine information about APAR IX75893 and what it does, enter the command:

```
# instfix -aik IY73748
IX75893 Abstract: Process memory is made read-only unnecessarily

IX75893 Symptom Text:
 Resource handler routines not being able to store to process
  memory when a process is dumping core.

----------------------------
Not all filesets for IY73748 were found.
```

To list what maintenance levels have been installed on your system with the **instfix** command, enter the command:

```
# instfix -i | grep ML
    All filesets for 5300-02_AIX_ML were found.
    All filesets for 5.3.0.0_AIX_ML were found.
    All filesets for 5300-01_AIX_ML were found.
    All filesets for 5300-03_AIX_ML were found.
```

To install APAR IY73748 from /dev/cd0, enter the command:

```
# instfix -k IY73748 -d /dev/cd0
```

Note: By default, when the **instfix** command is run from the command line, the command uses stdout and stderr for reporting. If you want to generate an installation report, you will need to redirect the output. For example:

```
# instfix -aik IY73748 >/tmp/instfix.out 2>/tmp/instfix.err
```

You can also use SMIT to determine what fixes are installed on your system.

1. Use the SMIT fast path:

   `smitty show_apar_stat`

   A screen similar to Figure 3-34 is shown.

```
                          Show Fix (APAR) Installation Status

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                         [Entry Fields]
* FIX ID                                             []                         +









F1=Help               F2=Refresh            F3=Cancel             F4=List
F5=Reset              F6=Command            F7=Edit               F8=Image
F9=Shell              F10=Exit              Enter=Do
```

Figure 3-34   Show Fix (APAR) Installation Status menu

2. Press F4 in the FIX ID field to get a list of all the fixes that are installed on the system. The output from this command is similar to the one from the **`instfix -iv`** command.

Installing an individual fix by APAR

The following steps are useful for installing a fix.

1. To install the fixes using SMIT, use the SMIT fast path **`smitty instfix`** or **`smitty update_by_fix`**.

2. In the INPUT device/directory for the software field, enter the name of the device (or directory if you downloaded the fixes to your system) from which to install the fixes, and press Enter. A screen similar to Figure 3-35 is shown.

```
                         Update Software by Fix (APAR)

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                         [Entry Fields]
* INPUT device / directory for software                /ptf/aix431
* FIXES to install                                    []                        +
  PREVIEW only? (update operation will NOT occur)      no                       +
  COMMIT software updates?                             yes                      +
  SAVE replaced files?                                 no                       +
  EXTEND file systems if space needed?                 yes                      +
  VERIFY install and check file sizes?                 no                       +
  DETAILED output?                                     no                       +
  Process multiple volumes?                            yes                      +






F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 3-35 Update Software by Fix (APAR) menu

3. In the FIXES to Install field, press F4 to get a list of fixes that are available on the media and select the fixes you want to install.
4. Press Enter.

The system will update the maintenance level of the fileset you selected and upon completion you will have successfully updated the maintenance level of your software.

Updating all filesets to the latest level

The following steps update filesets to the latest level.

1. To install all new fixes that are available from IBM, use the fast path **smitty update_all**./

   A screen similar to Figure 3-36 on page 83 is shown.

```
             Update Installed Software to Latest Level (Update All)

Type or select a value for the entry field.
Press Enter AFTER making all desired changes.

                                                        [Entry Fields]
* INPUT device / directory for software              []                       +















F1=Help              F2=Refresh           F3=Cancel            F4=List
F5=Reset             F6=Command           F7=Edit              F8=Image
F9=Shell             F10=Exit             Enter=Do
```

Figure 3-36 Update Installed Software to Latest Level (Update All) menu

2. In the INPUT device/directory for software field, enter the name of the device (or directory if you have fixes on your hard disk) from which installation will be carried out.
3. Press Enter.

A screen similar to Figure 3-37 is shown.

```
             Update Installed Software to Latest Level (Update All)

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                           [Entry Fields]
* INPUT device / directory for software                  /ptf/aix53
* SOFTWARE to update                                     _update_all
  PREVIEW only? (update operation will NOT occur)        no                      +
  COMMIT software updates?                               yes                     +
  SAVE replaced files?                                   no                      +
  AUTOMATICALLY install requisite software?              yes                     +
  EXTEND file systems if space needed?                   yes                     +
  VERIFY install and check file sizes?                   no                      +
  DETAILED output?                                       no                      +
  Process multiple volumes?                              yes                     +
  ACCEPT new license agreements?                         no                      +
  Preview new LICENSE agreements?                        no                      +



F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 3-37 Update Installed Software to Latest Level (Update All) menu - more

4. It is best to set the PREVIEW only? (update operation will NOT occur) field to yes by pressing the Tab key. The Preview option makes a dry run of the task you are trying to perform and reports any failures that might be encountered when you do the actual installation. This will ensure that your installation does not fail.

Once you are sure that there are no prerequisites that you are missing, you can do the actual installation. This procedure will update your software to the latest maintenance level.

To view the new maintenance level of your software, run:

```
# lslpp -l
```

This will show you the latest maintenance level of the filesets, including those you just updated.

3.12 Creating installation images on a disk

Installable image files (or installation packages) can be copied to a hard disk for use in future installations. These image files will be copied from your installation media (tape or diskette) to a directory on the disk, so that they may be installed later using the disk directory as the input device. These files will be copied to the directory named /usr/sys/inst.images.

1. To create installation images on your hard disk, use the SMIT fast path **`smitty bffcreate`**.

   A screen similar to Figure 3-38 is shown.

```
               Copy Software to Hard Disk for Future Installation

Type or select a value for the entry field.
Press Enter AFTER making all desired changes.

                                                     [Entry Fields]
* INPUT device / directory for software             []                     +











F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 3-38 Copy Software to Hard Disk for Future Installation menu

2. In the INPUT device/directory for software field, enter the name of your source that will be used to copy the images and press Enter.
3. On the next screen, press F4 on the Software package to copy field to get a list of the software available on the media. Select the installation images you want to copy to your hard disk and press Enter.
4. All the images will be copied to your hard disk in the /usr/sys/inst.images directory, and the /usr/sys/inst.images/.toc file is updated.

For future installations, enter the /usr/sys/inst.images directory in the INPUT device / directory for software field. If, for some reason, your .toc file becomes corrupted, you will receive an error either in SMIT or the command line, depending on what you are using, similar to:

```
0503-005 The format of .toc file is invalid
```

In this case, simply use the **inutoc /usr/sys/inst.images/.toc** command to recreate your .toc file.

This method of creating installation images is helpful in situations where the software you are trying to install has co-requisites that are on different media and your installation process does not let you change the media it is currently processing. In such situations, your installation will fail; therefore, we recommend having all the prerequisites and co-requisites reside in one directory and then do the installation.

To get co-requisites that are on different media, use the **smitty bffcreate** fast path to copy required filesets from the first CD-ROM to the hard disk in /usr/sys/inst.images. Then, use the **smitty bffcreate** fast path to copy the required filesets from the additional CD-ROMs to the hard disk in /usr/sys/inst.images. After all the required filesets have been copied to the hard disk, use the **installp** command or the **smitty install_latest** fast path to install the software. Since the system reads the /usr/sys/inst.images/.toc file for installation, and all the filesets are local to the hard disk, it will not prompt you for a CD-ROM or fail the installation.

3.13 Verifying the integrity of the operating system

The **lppchk** command is used to verify whether the software installed on your system is in a consistent state. We recommend that you should use this command after you install the operating system, after you apply PTFs or corrective APARs, or after you install any additional software on your system.

Every installable software product has three parts located under /, /usr/, and /usr/share. For each of these parts, there are object classes in ODM that are used by Software Vital Product Database (SWVPD) as follows:

- Four classes for the / part of the software are located in the /etc/objrepos file.
- Four classes for the /usr part of the software are located in /usr/lib/objrepos.
- Four classes for the /usr/share part of the software are located in /usr/share/lib/objrepos.

The name of the four classes are:

lpp Contains information about installed software, such as status and description.

inventory Contains information about files installed by software products.

product Contains product information about the installation, requirements, and prerequisites of software products.

history Contains information about history of installation and updates of software products.

The `lppchk` command verifies if the actual state of the software matches with the data stored in ODM in terms of file sizes, checksum values, and symbolic links.

Table 3-6 provides a list of common command flags and their descriptions for the `lppchk` command.

Table 3-6 Commonly used flags for the lppchk command

Flag	Description
-c	Performs a checksum operation on the specified items and verifies that the checksum and the file size are consistent with the SWVPD database.
-f	Checks that the specified items are present and the file size matches the SWVPD database.
-l (lowercase L)	Verifies symbolic links for files as specified in the SWVPD database.
-m [1|2|3]	Displays three levels of information. The levels are as follows: 1 - Error messages only (default). 2 - Error messages and warnings. 3 - Error messages, warnings, and informational messages.
-O {[r][s][u]}	Verifies the specified parts of the program. This flag is not needed with stand-alone systems because without this option all parts are verified by default. The flags specify the following parts: ► r: Indicates the / (root) part is to be verified. ► s: Indicates the /usr/share part is to be verified. ► u: Indicates the /usr part is to be verified.
-u	Updates the SWVPD with new checksum or size information from the system when the system information does not match the SWVPD database. This flag sets symbolic links that are found to be missing. This flag is valid with only the -c or -l flag.
-v	Verifies that the / (root), /usr, and /usr/share parts of the system are valid with each other.

The following example shows how flags -c and -l have reported that some files and links have been deleted:

```
# rm /usr/lpp/X11/include/X11/ObjectP.h
# lppchk -c X11.adt.include
lppchk: 0504-206  File /usr/lpp/X11/include/X11/ObjectP.h could not be located.
# rm /usr/lpp/X11/include/X11/ext/rgb.h
# lppchk -l X11.adt.include
lppchk: 0504-220  No link found from /usr/lpp/X11/include/X11/ext/rgb.h to
/usr/lpp/X11/Xamples/programs/Xserver/include/rgb.h.
#
```

The following example shows how to verify if all filesets are correctly installed and have all their prerequisites:

```
# lppchk -v
lppchk:  The following filesets need to be installed or corrected to bring
         the system to a consistent state:

  bos.64bit 5.3.0.30                     (not installed; requisite fileset)
  bos.adt.include 5.3.0.30               (not installed; requisite fileset)
  bos.alt_disk_install.boot_images 5.3.0.30 (not installed; requisite fileset)
  bos.alt_disk_install.rte 5.3.0.30      (not installed; requisite fileset)
  bos.mp 5.3.0.30                        (not installed; requisite fileset)
  bos.mp64 5.3.0.30                      (not installed; requisite fileset)
  bos.net.ewlm.rte 5.3.0.30              (usr: COMMITTED, root: BROKEN)
  bos.net.ipsec.keymgt 5.3.0.30          (usr: COMMITTED, root: not installed)
  bos.net.ipsec.rte 5.3.0.30             (usr: COMMITTED, root: not installed)
  bos.net.ipsec.websm 5.3.0.30           (usr: COMMITTED, root: not installed)
  bos.net.mobip6.rte 5.3.0.10            (usr: COMMITTED, root: BROKEN)
  bos.net.nfs.cachefs 5.3.0.30           (usr: COMMITTED, root: not installed)
  bos.net.nfs.client 5.3.0.30            (usr: COMMITTED, root: not installed)
  bos.net.nis.client 5.3.0.30            (usr: COMMITTED, root: not installed)
  bos.net.nis.server 5.3.0.30            (usr: COMMITTED, root: not installed)
  bos.net.nisplus 5.3.0.10               (usr: COMMITTED, root: not installed)
  bos.net.ppp 5.3.0.30                   (usr: COMMITTED, root: not installed)
  bos.net.tcp.client 5.3.0.30            (usr: COMMITTED, root: not installed)
  bos.net.tcp.server 5.3.0.30            (usr: COMMITTED, root: BROKEN)
  bos.net.uucp 5.3.0.30                  (usr: COMMITTED, root: not installed)
  csm.diagnostics 1.4.1.10               (not installed; requisite fileset)
  devices.chrp.IBM.HPS.rte 1.2.0.0       (not installed; requisite fileset)
  devices.common.IBM.sni.rte 1.2.0.0     (not installed; requisite fileset)
  devices.scsi.ses.diag 5.3.0.30         (not installed; requisite fileset)
  perfagent.tools 5.3.0.30               (not installed; requisite fileset)
```

3.14 Differences between installp and rpm

This section explains the **installp** and **rpm** commands.

3.14.1 The installp command

The **installp** command is used to install and update software. The **installp** command has a large number of flags. In the following sections, only the most important flags are shown with each command. The **installp** command is also used by all the SMIT scripts to install software.

The flags commonly used with the **installp** command are listed in Table 3-7.

Table 3-7 Commonly used flags for the installp command

Flag	Description
-a	Applies one or more software products or updates. This is the default action. This flag can be used with the -c flag to apply and commit a software product update during installation.
-B	Indicates that the requested action should be limited to software updates.
-C	Cleans up after an interrupted installation and attempts to remove all incomplete pieces of the previous installation.
-c	Commits applied updates to the system.
-d *Device*	Specifies on what device the installation media can be found.
-F	Forces the installation of a software product even if there exists a previously installed version of the software product that is the same version as or newer than the one being installed.
-f *ListFile*	Reads the names of the software products from *ListFile*. If *ListFile* is a - (dash), it reads the list of names from the standard input. Output from the **installp -l** command is suitable for input to this flag.
-g	When used to install or commit, this flag automatically installs or commits, respectively, any software products or updates that are requisites of the specified software product. When used to remove or reject software, this flag automatically removes or rejects dependents of the specified software.
-L	Displays the contents of the media by looking at the table of contents (.toc) and displaying the information in colon-separated output. This flag is used by SMIT to list the content of the media.

Flag	Description
-l (lowercase L)	Lists all the software products and their separately installable options contained on the installation media to the standard output. No installation occurs.
-N	Overrides saving of existing files that are replaced when installing or updating. This flag is valid only with the -ac flags.
-p	Performs a preview of an action by running all pre-installation checks for the specified action. This flag is only valid with the apply, commit, reject, and remove (-a, -c, -r, and -u) flags.
-r	Rejects all software updates that are currently applied but not committed.
-u	Removes the specified software product and any of its installed updates from the system. Removal of any bos.rte fileset is never permitted.
-V *Number*	Specifies the verbose option that can provide up to four levels of detail for pre-installation output, including SUCCESSES, WARNINGS, and FAILURES.
-v	Verifies that all installed files in the fileset have the correct checksum value after installation. Can be used with the -a and -ac flags to confirm a successful installation. If any errors are reported by this flag, it may be necessary to reinstall the software.
-X	Attempts to expand any file systems where there is insufficient space to do the installation.

3.14.2 The RPM Package Manager

The RPM Package Manager (RPM), in addition to **`installp`** formatted packages, has been in AIX 5L since Version 5.1, and allows you to install and perform powerful queries and verification about Open Source Products in your system. RPM maintains a database of installed packages and their files. An RPM is an archive of files specific to a program and contains the program name, version, description, dependencies, and program files.

Use the Web-based System Manager, SMIT, or the **`geninstall`** command to install and un-install these types of packages. The **`geninstall`** command can detect the format type of a specified package and run the appropriate installation command.

The AIX 5L product media contains **installp** packages and RPM packages that are installed during a base operating system (BOS) installation. The **installp** packages are located in the following path:

/mount_point/installp/ppc

The RPM packages are located in the following path:

/mount_point/RPMS/ppc

If you are using the **geninstall** command to install RPM packages, use the prefix type to indicate to the **geninstall** command the type of package that you are installing. The package prefix types are the following:

I	**installp** format
R	RPM format

Table 3-8 shows a description for the commonly used flags for the **geninstall** command.

Table 3-8 Commonly used flags for the geninstall command

Flag	Description
-d device media or directory	Specifies the device or directory containing the images to install.
-f file	Specifies the file containing a list of entries to install. Each entry in the file must be preceded by a format type prefix. Currently, **geninstall** accepts the following prefixes: I:bos.net (Installp) J:WebSphere (ISMP) R:mtools (RPM) U:devices.pci.8602912 (UDI) This information is given in the **geninstall -L** output.
-I installpflags	Specifies the **installp** flags to use when calling the **installp** command. The flags that are used during an install operation for **installp** are the a, b, c, D, e, E, F, g, I, J, M, N, O, p, Q, q, S, t, v, V, w, and X flags. The **installp** flags that should not be used during install are the C, i, r, S, z, A, and I flags. The **installp** command should be called directly to perform these functions. The -u, -d, -L, and -f flags should be given outside the -I flag.

Flag	Description
-L	Lists the contents of the media. The output format is the same as the **installp -Lc** format, with additional fields at the end for ISMP, RPM, and UDI formatted products.
-R ResponseFile	Takes the full path name of the ResponseFile to send to the ISMP installer program.
-u	Performs an un-install of the specified software. For ISMP products, the un-installer listed in the vendor database is called, prefixed by a "J:".
-Y	Agrees to required software license agreements for software to be installed. This flag is also accepted as an **installp** flag with the -I option.
-Z	Tells **geninstall** to invoke the installation in silent mode.

The syntax of the **geninstall** command is:

- Install software from device:

  ```
  geninstall -d Media [ -I installpFlags ] [ -R ResponseFile ] [ -E
  ResponseFile ] [ -N ] [ -Y ] [ -Z ] -f file | install_list... | all
  ```

- Uninstall software:

  ```
  geninstall -u -f file | uninstall_list...
  ```

- List installable software on device:

  ```
  geninstall -L -d media
  ```

The **rpm** command is also used to install, upgrade, query, and delete Linux RPM packages on AIX 5L. The tool is also used to maintain the RPM package database.

4

Boot process

This chapter describes the boot process and the different stages the system uses to prepare the AIX 5L environment.

Topics discussed in this chapter are:

- The boot process
- System initialization
- The /etc/inittab file
- How to recover from a non-responsive boot process
- Run levels
- An introduction to the rc.* files

© Copyright IBM Corp. 2006. All rights reserved.

4.1 The boot process

As a system administrator, you should have a general understanding of the boot process. This knowledge is useful to solving problems that can prevent a system from booting properly. These problems can be both software or hardware. We also recommend that you be familiar with the hardware configuration of your system.

Booting involves the following steps:

- The initial step in booting a system is the Power On Self Test (POST). Its purpose is to verify that the basic hardware is in a functional state. The memory, keyboard, communication, and audio devices are also initialized. You can see an image for each of these devices displayed on the screen. It is during this step that you can press a function key to choose a different boot list. The LED values displayed during this phase are model specific. Both hardware and software problems can prevent the system from booting.

> **Note:** Old systems based on the MCI architecture execute an additional step before this, the so-called Built In Self Test (BIST). This step is no longer required for systems based on the PCI architecture.

- System Read Only Storage (ROS) is specific to each system type. It is necessary for AIX 5L Version 5.3 to boot, but it does not build the data structures required for booting. It will locate and load bootstrap code. System ROS contains generic boot information and is operating system independent.
- Software ROS (also named bootstrap) forms an IPL control block, which is compatible with AIX 5L Version 5.3, that takes control and builds AIX 5L specific boot information. A special file system located in memory and named the RAMFS file system is created. Software ROS then locates, loads, and turns control over to the AIX 5L boot logical volume (BLV). Software ROS is AIX 5L information based on machine type and is responsible for completing machine preparation to enable it to start the AIX 5L kernel.
- A complete list of files that are part of the BLV can be obtained from the /usr/lib/boot directory. The most important components are the following:
  - The AIX 5L kernel
  - Boot commands called during the boot process, such as **`bootinfo`** and **`cfgmgr`**
  - A reduced version of the ODM. Many devices need to be configured before hd4 is made available, so their corresponding methods have to be stored in the BLV. These devices are marked as base in PdDv.
  - The rc.boot script

- The AIX 5L kernel is loaded and takes control. The system will display 0299 on the LED panel. All previous codes are hardware-related. The kernel will complete the boot process by configuring devices and starting the init process. LED codes displayed during this stage will be generic AIX 5L codes.
- So far, the system has tested the hardware, found a BLV, created the RAMFS, and started the init process from the BLV. The rootvg has not yet been activated. From now on, the rc.boot script will be called three times, and is passed a different parameter each time.

4.1.1 Boot phase 1

During this phase, the following steps are taken:

- The init process started from RAMFS executes the boot script rc.boot 1. If the init process fails for some reason, code c06 is shown on the LED display.
- At this stage, the **`restbase`** command is called to copy a partial image of ODM from the BLV into the RAMFS. If this operation is successful, the LED display shows 510; otherwise, LED code 548 is shown.
- After this, the **`cfgmgr -f`** command reads the Config_Rules class from the reduced ODM. In this class, devices with the attribute phase=1 are considered base devices. Base devices are all devices that are necessary to access rootvg. For example, if the rootvg is located on a hard disk, all devices starting from the motherboard up to the disk will have to be initialized. The corresponding methods are called so that rootvg can be activated in boot phase 2.
- At the end of boot phase 1, the **`bootinfo -b`** command is called to determine the last boot device. At this stage, the LED shows 511.

4.1.2 Boot phase 2

In boot phase 2, the rc.boot script is passed to the parameter 2.

During this phase, the following steps are taken.

- The rootvg volume group is varied on with the special version of the **`varyonvg`** command named the **`ipl_varyon`** command. If this command is successful, the system displays 517; otherwise, one of the following LED codes will appear: 552, 554, or 556, and the boot process is halted.
- Root file system hd4 is checked using the **`fsck -f`** command. This will verify whether the file system was unmounted cleanly before the last shutdown. If this command fails, the system will display code 555.
- The root file system (/dev/hd4) is mounted on a temporary mount point (/mnt) in RAMFS. If this fails, 557 will appear in the LED display.

- The /usr file system is verified using the **fsck -f** command and then mounted. If this operation fails, the LED 518 appears.
- The /var file system is verified using the **fsck -f** command and then mounted. The **copycore** command checks if a dump occurred. If it did, it is copied from default dump devices, /dev/hd6, to the default copy directory, /var/adm/ras. Afterwards, /var is unmounted.
- The primary paging space from rootvg, /dev/hd6, will be activated.
- The mergedev process is called and all /dev files from the RAM file system are copied onto disk.
- All customized ODM files from the RAM file system are copied to disk. Both ODM versions from hd4 and hd5 are now synchronized.
- Finally, the root file system from rootvg (disk) is mounted over the root file system from the RAMFS. The mount points for the rootvg file systems become available. Now, the /var and /usr file systems from the rootvg are mounted again on their ordinary mount points.

There is no console available at this stage, so all boot messages will be copied to **alog**. The **alog** command maintains and manages logs.

4.1.3 Boot phase 3

After phase 2 is completed, rootvg is activated and the following steps are taken:

- /etc/init process is started. It reads the /etc/inittab file and calls rc.boot with argument 3.
- The /tmp file system is mounted.
- The rootvg is synchronized by calling the **syncvg** command and launching it as a background process. As a result, all stale partitions from rootvg are updated. At this stage, the LED code 553 is shown.
- At this stage, the **cfgmgr** command is called; if the system is booted in normal mode, the **cfgmgr** command is called with option -p2; if the system is booted in service mode, the **cfgmgr** command is called with option -p3. The **cfgmgr** command reads the Config_rules file from ODM and calls all methods corresponding to either phase=2 or phase=3. All other devices that are not base devices are configured at this time.
- Next, the console is configured by calling the **cfgcon** command. After the configuration of the console, boot messages are sent to the console if no STDOUT redirection is made. However, all missed messages can be found in /var/adm/ras/conslog. LED codes that can be displayed at this time are:
  - c31: Console not yet configured. Provides instructions to select console.
  - c32: Console is an LFT terminal.

- c33: Console is a TTY.
- c34: Console is a file on the disk.

- Finally, the synchronization of the ODM in the BLV with the ODM from the / (root) file system is done by the **savebase** command.
- The **syncd** daemon and **errdemon** are started.
- The LED display is turned off.
- If the file /etc/nologin exists, it will be removed.
- If there are devices marked as missing in CuDv, a message is displayed on the console.
- The message System initialization completed is sent to the console. The execution of rc.boot is has completed. Process init will continue processing the next command from /etc/inittab.

4.2 System initialization

During system startup, after the root file system has been mounted in the pre-initialization process, the following sequence of events occurs:

1. The **init** command is run as the last step of the startup process.
2. The **init** command attempts to read the /etc/inittab file.
3. If the /etc/inittab file exists, the **init** command attempts to locate an initdefault entry in the /etc/inittab file.
   a. If the initdefault entry exists, the **init** command uses the specified run level as the initial system run level.
   b. If the initdefault entry does not exist, the **init** command requests that the user enter a run level from the system console (/dev/console).
   c. If the user enters an S, s, M, or m run level, the **init** command enters the maintenance run level. These are the only run levels that do not require a properly formatted /etc/inittab file.
4. If the /etc/inittab file does not exist, the **init** command places the system in the maintenance run level by default.
5. The **init** command rereads the /etc/inittab file every 60 seconds. If the /etc/inittab file has changed since the last time the **init** command read it, the new commands in the /etc/inittab file are executed.

4.3 The /etc/inittab file

The /etc/inittab file controls the initialization process.

The /etc/inittab file supplies the script to the **init** command's role as a general process dispatcher. The process that constitutes the majority of the **init** command's process dispatching activities is the /etc/getty line process, which initiates individual terminal lines. Other processes typically dispatched by the **init** command are daemons and the shell.

The /etc/inittab file is composed of entries that are position-dependent and have the following format:

```
Identifier:RunLevel:Action:Command
```

Each entry is delimited by a newline character. A backslash (\) preceding a newline character indicates the continuation of an entry. There are no limits (other than maximum entry size) on the number of entries in the /etc/inittab file. The maximum entry size is 1024 characters.

The entry fields are:

Identifier A one to fourteen character field that uniquely identifies an object.

RunLevel The run level at which this entry can be processed.

The run level has the following attributes:

- Run levels effectively correspond to a configuration of processes in the system.
- Each process started by the **init** command is assigned one or more run levels in which it can exist.
- Run levels are represented by the numbers 0 through 9. For example, if the system is in run level 1, only those entries with a 1 in the run-level field are started.
- When you request the **init** command to change run levels, all processes without a matching entry in the run-level field for the target run level receive a warning signal (SIGTERM). There is a 20-second grace period before processes are forcibly terminated by the kill signal (SIGKILL).
- The run-level field can define multiple run levels for a process by selecting more than one run level in any combination from 0 through 9. If no run level is specified, the process is assumed to be valid at all run levels.

– There are four other values that appear in the run-level field, even though they are not true run levels: a, b, c and h. Entries that have these characters in the run level field are processed only when the **telinit** command requests them to be run (regardless of the current run level of the system). They differ from run levels in that the **init** command can never enter run level a, b, c, or h. Also, a request for the execution of any of these processes does not change the current run level. Furthermore, a process started by an a, b, or c command is not killed when the **init** command changes levels. They are only killed if their line in the /etc/inittab file is marked off in the action field, their line is deleted entirely from /etc/inittab, or the **init** command goes into single-user mode.

Action Tells the **init** command how to treat the process specified in the process field. The following actions are recognized by the **init** command:

respawn If the process does not exist, start the process. Do not wait for its termination (continue scanning the /etc/inittab file). Restart the process when it dies. If the process exists, do nothing and continue scanning the /etc/inittab file.

wait When the **init** command enters the run level that matches the entry's run level, start the process and wait for its termination. All subsequent reads of the /etc/inittab file, while the **init** command is in the same run level, will cause the **init** command to ignore this entry.

once When the **init** command enters a run level that matches the entry's run level, start the process, and do not wait for termination. When it dies, do not restart the process. When the system enters a new run level, and the process is still running from a previous run level change, the program will not be restarted.

boot Process the entry only during system boot, which is when the **init** command reads the /etc/inittab file during system startup. Start the process, do not wait for its termination, and when it dies, do not restart the process. In order for the instruction to be meaningful, the run level should be the default or it must match the **init** command's run level at boot time. This action is useful for an initialization function following a hardware reboot of the system.

bootwait Process the entry the first time that the **init** command goes from single-user to multi-user state after the system is booted. Start the process, wait for its termination, and when it dies, do not restart the process. If the initdefault is 2, run the process right after boot.

powerfail Execute the process associated with this entry only when the **init** command receives a power fail signal (SIGPWR).

powerwait Execute the process associated with this entry only when the **init** command receives a power fail signal (SIGPWR), and wait until it terminates before continuing to process the /etc/inittab file.

off If the process associated with this entry is currently running, send the warning signal (SIGTERM), and wait 20 seconds before terminating the process with the kill signal (SIGKILL). If the process is not running, ignore this entry.

ondemand Functionally identical to respawn, except this action applies to the a, b, or c values, not to run levels.

initdefault An entry with this action is only scanned when the **init** command is initially invoked. The **init** command uses this entry, if it exists, to determine which run level to enter initially. It does this by taking the highest run level specified in the run-level field and using that as its initial state. If the run level field is empty, this is interpreted as 0123456789: therefore, the **init** command enters run level 9. Additionally, if the **init** command does not find an initdefault entry in the /etc/inittab file, it requests an initial run level from the user at boot time.

sysinit Entries of this type are executed before the **init** command tries to access the console before login. It is expected that this entry will only be used to initialize devices on which the **init** command might try to ask the run level question. These entries are executed and waited for before continuing.

Command A shell command to execute. The entire command field is prefixed with exec and passed to a forked sh as the **sh -c exec**

command. Any legal **sh** command syntax can appear in this field. Comments can be inserted with the # comment syntax.

The **getty** command overwrites the output of any commands that appear before it in the /etc/inittab file. To record the output of these commands to the boot log, pipe their output to the **alog -tboot** command.

The stdin, stdout, and stderr file descriptors may not be available while the **init** command is processing inittab entries. Any entries writing to stdout or stderr may not work predictably unless they redirect their output to a file or to /dev/console.

The following commands are the only supported methods for modifying the records in the /etc/inittab file:

mkitab Adds records to the /etc/inittab file.

lsitab Lists records in the /etc/inittab file.

chitab Changes records in the /etc/inittab file.

rmitab Removes records from the /etc/inittab file.

For example, you want to add a record on the /etc/inittab file to run the **find** command on the run level 2 and start it again once it has finished:

1. Run the **ps** command and display only those processes that contain the word find:

```
# ps -ef | grep find
   root 19750 13964   0 10:47:23  pts/0  0:00 grep find
#
```

2. Add a record named xcmd on the /etc/inittab using the **mkitab** command:

```
# mkitab "xcmd:2:respawn:find / -type f > /dev/null 2>&1"
```

3. Show the new record with the **lsitab** command:

```
# lsitab xcmd
xcmd:2:respawn:find / -type f > /dev/null 2>&1
#
```

4. Display the processes:

```
# ps -ef | grep find
   root 25462     1   6 10:56:58      -  0:00 find / -type f
   root 28002 13964   0 10:57:00  pts/0  0:00 grep find
#
```

5. Cancel the **find** command process:

```
# kill 25462
```

6. Display the processes:

```
# ps -ef | grep find
    root 23538 13964   0 10:58:24  pts/0  0:00 grep find
    root 28966     1   4 10:58:21      -  0:00 find / -type f
#
```

   Since the action field is configured as respawn, a new process (28966, in this example) is started each time its predecessor finishes.

The process will continue re-spawning, unless you change the action field, for example:

1. Change the action field on the record xcmd from respawn to once:

```
# chitab "xcmd:2:once:find / -type f > /dev/null 2>&1"
```

2. Display the processes:

```
# ps -ef | grep find
    root 20378 13964   0 11:07:20  pts/0  0:00 grep find
    root 28970     1   4 11:05:46      -  0:03 find / -type f
```

3. Cancel the **find** command process:

```
# kill 28970
```

4. Display the processes:

```
# ps -ef | grep find
    root 28972 13964   0 11:07:33  pts/0  0:00 grep find
#
```

To delete this record from the /etc/inittab file, you use the **rmitab** command. For example:

```
# rmitab xcmd
# lsitab xcmd
#
```

Order of the /etc/inittab entries

The base process entries in the /etc/inittab file is ordered as follows:

1. initdefault
2. sysinit
3. Powerfailure Detection (powerfail)
4. Multiuser check (rc)
5. /etc/firstboot (fbcheck)
6. System Resource Controller (srcmstr)
7. Start TCP/IP daemons (rctcpip)

8. Start NFS daemons (rcnfs)
9. cron
10. pb cleanup (piobe)
11. getty for the console (cons)

The System Resource Controller (SRC) has to be started near the beginning of the etc/inittab file since the SRC daemon is needed to start other processes. Since NFS requires TCP/IP daemons to run correctly, TCP/IP daemons are started ahead of the NFS daemons. The entries in the /etc/inittab file are ordered according to dependencies, meaning that if a process (process2) requires that another process (process1) be present for it to operate normally, then an entry for process1 comes before an entry for process2 in the /etc/inittab file.

4.4 How to recover from a non-responsive boot process

If your system does not boot, AIX 5L offers several tools and methods to recover from a hung boot process.

This section discusses situations that may cause system hang during the boot process.

4.4.1 The bootlist command

You can change the order in which your system looks up devices to find its boot code. The **bootlist** command displays and alters the list of boot devices available to the system. The general syntax of the command is as follows:

```
bootlist [ { -m Mode } [ -r ] [  -o  ] [ [ -i ] | [ [ -f File ]
[ Device [ Attr=Value ... ] ... ] ] ]
```

The most common flags used with the **bootlist** command are given in Table 4-1.

Table 4-1 Commonly used flags for the bootlist command

Flag	Description
-m *mode*	Specifies which boot list to display or alter. Possible values for the *mode* variable are normal, service, both, or prevboot.
-f *File*	Indicates that the device information is to be read from the specified file name.
-i	Indicates that the device list specified by the -m flag should be invalidated.
-o	Indicates that the specified boot list is to be displayed after any specified alteration is performed. The output is a list of device names.

Flag	Description
-r	Indicates to display the specified bootlist after any specified alteration is performed. The output is hardware-platform dependent.

The **bootlist** command allows the user to display and alter the list of possible boot devices from which the system may be booted. This command supports the updating of the following:

Normal boot list The normal list designates possible boot devices for when the system is booted in normal mode.

Service boot list The service list designates possible boot devices for when the system is booted in service mode.

Previous boot device This entry designates the last device from which the system booted. Some hardware platforms may attempt to boot from the previous boot device before looking for a boot device in one of the other lists.

Support of these boot lists varies from platform to platform, and some platforms do not have boot lists. When a system is booted, it will scan the boot list searching for a boot device, The system selects the first device in the list and determines if it is bootable. If no boot file system is detected on the first device, the system moves on to the next device in the list. As a result, the ordering of devices in the device list is extremely important.

In order to display a boot list (AIX Version 4.2 or later), use the command:

```
# bootlist -m normal -o
cd0
hdisk0 blv=hd5
rmt0
```

If you want to make changes to your normal boot list and remove, for example, rmt0, use the command:

```
# bootlist -m normal cd0 hdisk0
```

or create a file containing the list of cd0 hdisk0 (separated by white space or one device per line) and use the command:

```
# bootlist -m normal -f filename
```

This will change the normal bootlist to indicate that when the system is booted, it will first attempt to boot from cd0. If it cannot find a boot image on cd0, it will look to hdisk0. If the search fails, it will give an LED code and wait for user intervention. It will not search for rmt0 anymore.

After changing the bootlist, verify the bootlist as follows:

```
# bootlist -m normal -o
hdisk0 blv=hd5
cd0
```

Support of these boot lists varies from platform to platform, and some platforms do not have boot lists. When searching for a boot device, the system selects the first device in the list and determines if it is bootable. If no BLV is detected on the first device, the system moves on to the next device in the list. As a result, the ordering of devices in the device list is extremely important.

Boot device choices

The naming conventions that can be used in your boot list are provided in Table 4-2. Each device that you add to your bootlist must be in the AVAILABLE state. Otherwise, the **bootlist** command will fail, and you will encounter an error similar to:

```
0514-210 bootlist: Device xxxxx is not in the AVAILABLE state
```

Table 4-2 Valid device names for the bootlist command

Device	Description
hdisk*xx*	Physical volume device logical names
cd*xx*	SCSI and IDE CD-ROM device logical names
rmt*xx*	Magnetic tape device logical names
ent*xx*	Ethernet adapter logical names
tok*xx*	Token ring adapters logical names

4.4.2 Accessing a system that will not boot

If you are unable to boot your system, the first step is to access the system and see what is the probable cause of the failure. This procedure enables you to get a system prompt so that you may attempt to recover data from the system or perform corrective action that will enable the system to boot from the hard disk.

The following steps need to be executed to access the system:

1. Turn the system key (if present) to the Service position or alternatively press F5 on a PCI based system to boot from the tape/CD-ROM/DVD-RAM drive (during step 4).

2. Turn on all attached external devices, such as terminals, CD-ROM or DVD-RAM drives, tape drives, monitors, and external disk drives before turning on the system unit. Turn on the system unit to allow the installation media to be loaded.
3. Insert Volume 1 of the installation media into the tape, CD-ROM, or DVD-RAM drive, and power the system unit off.
4. Turn the system unit power switch to the On position. When booting, a screen will appear (before the one in Figure 4-1) asking you to press a function key (such as F1) to select the proper display as the system console. Each display on the system will receive a function key number in order to identify it as the system console. The system begins booting from the installation media. After several minutes, c31 is displayed in the LED (if your system has an LED; otherwise, a screen similar to the one in Figure 4-1 is displayed).

```
                      Welcome to Base Operating System
                      Installation and Maintenance

Type the number of your choice and press Enter.  Choice is indicated by >>>.

>>> 1 Start Install Now with Default Settings

    2 Change/Show Installation Settings and Install

    3 Start Maintenance Mode for System Recovery









    88  Help ?
    99  Previous Menu

>>> Choice [1]:
```

Figure 4-1 BOS installation and maintenance screen

5. Select option 3, Start Maintenance Mode for System Recovery, and press Enter. A screen similar to the one in Figure 4-2 on page 107 is shown.

```
                    Maintenance

Type the number of your choice and press Enter.

>>> 1 Access a Root Volume Group
    2 Copy a System Dump to Removable Media
    3 Access Advanced Maintenance Functions
    4 Erase Disks
    5 Install from a System Backup










    88  Help ?
    99  Previous Menu

>>> Choice [1]:
```

Figure 4-2 Maintenance menu

6. Enter 1, Access a Root Volume Group. A screen similar to the one in Figure 4-3 is shown.

```
                  Warning:

If you choose to access a root volume group, you will not be able to return
to the Base Operating System Installation menus without rebooting.


Type the number of your choice and press Enter.

    0 Continue











    88  Help ?
>>> 99  Previous Menu

>>> Choice [99]:
```

Figure 4-3 Warning screen

7. Take note of the warning. If you want to return to the previous menu, enter 99; otherwise, enter **0** to confirm. This will display the screen shown in Figure 4-4 on page 109.

```
                    Access a Root Volume Group

Type the number for a volume group to display the logical volume information
and press Enter.

   1)   Volume Group 00c478de00004c0000000107b393746c contains these disks:
          hdisk0  70006 05-08-00-3,0




















   Choice:
```

Figure 4-4   List of found volume group(s)

8. Select the volume group whose logical volume information you want to display. This is important, because rootvg will have hd5, that is, the boot logical volume. Enter the number of the volume group and press Enter. A screen similar to Figure 4-5 is shown.

```
                          Volume Group Information

     ------------------------------------------------------------------------------
        Volume Group ID 00c478de00004c0000000107a1137f36 includes the following
        logical volumes:

             hd5         hd6         hd8         hd4         hd2       hd9var
             hd3         hd1      hd10opt      fslv00      fslv01
     ------------------------------------------------------------------------------








    Type the number of your choice and press Enter.

       1) Access this Volume Group and start a shell
       2) Access this Volume Group and start a shell before mounting filesystems

      99) Previous Menu

        Choice [99]:
```

Figure 4-5 List of logical volumes found on the selected volume group

9. Select one of the options from the Volume Group Information screen and press Enter. Each option does the following:

   Choice 1 Selecting this choice imports and activates the volume group and mounts the file systems for this root volume group before providing you with a shell and a system prompt.

   Choice 2 Selecting this choice imports and activates the volume group and provides you with a shell and system prompt before mounting the file systems for this root volume group.

   Choice 99 Entering 99 returns you to the Access a Root Volume Group screen.

   After either choice 1 or 2 is selected and processed, a shell and system prompt are displayed.

10. Take appropriate measures to recover data or take action (such as using the **`bosboot`** command) to enable the system to boot normally.

4.4.3 Common boot LED codes

During system initialization, servers with operator panel displays will show LED codes that provide information about the status of the boot process. Some codes are checkpoints that indicate which point in the boot process the server has reached. These codes depend on the type of server.

For example, on a server with a four character display, E1F1 indicates that the system-defined console has been activated, while FF1 indicates this on a server with a three character display. Codes in the form of Fxx, where *xx* is a hexadecimal number, are generally related to firmware.

Other codes indicate that a fault has been detected. The most common LED codes that indicate boot problems and how to get around them to getting your system up and running again are given in Table 4-3.

Table 4-3 Common startup LEDs and solutions

LED 201 - Damaged boot image
1. Access your rootvg by following the procedure described in 4.4.2, "Accessing a system that will not boot" on page 105. 2. Check the / and /tmp file systems. If they are almost full, create more space. 3. Determine the boot disk by using the **`lslv -m hd5`** command. 4. Recreate the boot image using **`bosboot -a -d /dev/hdisk`*****n***, where *n* is the disk number of the disk containing the boot logical volume. 5. Check for CHECKSTOP errors in the error log. If such errors are found, it is probably failing hardware. 6. Shut down and restart the system.
LED 223-229 - Invalid boot list
1. Set the key mode switch to service (F5 for systems without keylock) and power up the machine. 2. If display continues normally, change the key mode switch to Normal and continue with step 3. If you do not get the prompt, go to step 4. 3. When you get the login prompt, log in and follow the procedure described in 4.4.1, "The bootlist command" on page 103 to change your bootlist. Continue with step 7. 4. Follow the procedure in 4.4.2, "Accessing a system that will not boot" on page 105 to access your rootvg and continue with step 5. 5. Determine the boot disk by using the **`lslv -m hd5`** command. 6. Change the bootlist following the procedure given in 4.4.1, "The bootlist command" on page 103. 7. Shut down and restart your system.

LED 551, 555, and 557 - Corrupted file system, corrupted JFS log, and so on

1. Follow the procedure described in 4.4.2, “Accessing a system that will not boot” on page 105 to access the rootvg before mounting any file systems (choice 2 on the Volume Group Information screen).
2. Verify and correct the file systems as follows:

```
fsck -y /dev/hd1
fsck -y /dev/hd2
fsck -y /dev/hd3
fsck -y /dev/hd4
fsck -y /dev/hd9var
```

3. Format the JFS log again by using the command:

```
/usr/sbin/logform /dev/hd8
```

4. Use **`lslv -m hd5`** to obtain the boot disk.
5. Recreate the boot image using the command:

```
bosboot -a -d /dev/hdiskn
```

   Where *n* is the disk number of the disk containing the boot logical volume.

LED 552, 554, and 556 - Super block corrupted and corrupted customized ODM database

1. Repeat steps 1 through 2 for LEDs 551, 555, and 557.
2. If **fsck** indicates that block 8 is corrupted, the super block for the file system is corrupted and needs to be repaired. Enter the command:
`dd count=1 bs=4k skip=31 seek=1 if=/dev/hdn of=/dev/hdn`
where *n* is the number of the file system.
3. Rebuild your JFS log by using the command:
`/usr/sbin/logform /dev/hd8`
4. If this solves the problem, stop here; otherwise, continue with step 5.
5. Your ODM database is corrupted. Restart your system and follow the procedure given in 4.4.2, “Accessing a system that will not boot” on page 105 to access rootvg with choice 2 on the Volume Group Information screen.
6. Mount the root and usr file systems as follows:
`mount /dev/hd4 /mnt`
`mount /usr`
7. Copy the system configuration to a back up directory:
`mkdir /mnt/etc/objrepos/backup`
`cp /mnt/etc/objrepos/Cu* /mnt/etc/objrepos/backup`
8. Copy the configuration from the RAM file system as follows:
`cp /etc/objrepos/Cu* /mnt/etc/objrepos`
9. Unmount all file systems by using the **umount all** command.
10. Determine the boot disk by using the **lslv -m hd5** command.
11. Save the clean ODM to the boot logical volume by using the command:
`savebase -d/dev/hdiskn`
where *n* is the disk number of the disk containing boot logical volume.
12. Reboot, if the system does not come up, and reinstall BOS.

LED 553 - Corrupted /etc/inittab file

1. Access the rootvg with all file systems mounted by following the procedure described in 4.4.2, “Accessing a system that will not boot” on page 105.
2. Check for free space in /, /var, and /tmp by using the **`df`** command.
3. Check the /etc/inittab file and correct the inittab problems if there is one empty inittab file, missing inittab file, or wrong entry in inittab file.
4. Check problems with:
`/etc/environment file`
`/bin/sh`
`/bin/bsh`
`/etc/fsck`
`/etc/profile`
`/.profile`
5. Shut down the system and reboot.

4.5 Run levels

Before performing maintenance on the operating system or changing the system run level, you might need to examine the various run levels. A run level is a software configuration that allows only a selected group of processes to exist.

4.5.1 Identifying system run levels

This section describes how to identify the run level at which the system is operating and how to display a history of previous run levels. The system can be at one of the run levels listed in Table 4-4.

Table 4-4 Run levels available on AIX 5L

Run level	Description
0-1	Reserved for the future use of the operating system.
2	Contains all of the terminal process and daemons that are run in the multiuser environment. This is the default run level.
3-9	Can be defined according to the user’s preferences
a,b,c,h	These are not true run levels; they differ from run levels in that the **`init`** command cannot request the entire system to enter these run levels. See 4.3, “The /etc/inittab file” on page 98 for more information.

Run level	Description
S,s,M,m	Maintenance mode. When the system enters maintenance mode from another run level, only the system console is used as the terminal.

Identifying the current run level

At the command line, type the following command:

```
# cat /etc/.init.state
2
#
```

The system displays one digit; that is the current run level.

Displaying a history of previous run levels

You can display a history of previous run levels using the **fwtmp** command as follows:

1. Log in as the root user.
2. Type the following command and the system shows information similar to the following:

```
# /usr/lib/acct/fwtmp </var/adm/wtmp |grep run-level
                        run-level 2   1     0 0062 0123 1132072406
                    Tue Nov 15 10:33:26 CST 2005
                        run-level 2   1     0 0062 0123 1132075614
                    Tue Nov 15 11:26:54 CST 2005
                        run-level 2   1     0 0062 0123 1132092491
                    Tue Nov 15 16:08:11 CST 2005
                        run-level 2   1     0 0062 0123 1132094756
                    Tue Nov 15 16:45:56 CST 2005
```

Changing system run levels

When the system starts the first time, it enters the default run level defined by the initdefault entry in the /etc/inittab file. The system operates at that run level until it receives a signal to change it. You can execute the **lsitab** command to find what the default run level is on your system:

```
# lsitab init
init:2:initdefault:
#
```

To change the run level, follow this procedure:

1. Check the /etc/inittab file to confirm that the run level to which you are changing supports the processes that you are running. The getty process is particularly important, since it controls the terminal line access for the system console and other logins. Ensure that the getty process is enabled at all run levels.
2. Use the **`wall`** command to inform all users that you intend to change the run level and request that users log off.
3. Use the **`telinit`** command using the run level you want to switch as a parameter, for example:

   ```
   # telinit M
   ```

The telinit command

The **`telinit`** command directs the actions of the init process (process ID 1) by taking a one-character argument and signaling the init process to perform the appropriate action. In general, the **`telinit`** command sets the system at a specific run level. The following arguments serve as directives that the **`telinit`** command passes to the init process:

0-9 Tells the init process to put the system in one of the run levels 0-9.

S,s,M,m Tells the init process to enter the maintenance mode.

a,b,c Tells the init process to examine only those records in the /etc/inittab file with a, b, or c in the run level field.

Q,q Tells the init process to re-examine the entire /etc/inittab file.

N Sends a signal that stops processes from being respawned.

For example, to enter maintenance mode, type:

```
# telinit M
```

> **Note:** You can also go to maintenance mode by using the **`shutdown -m`** command.

Executing run level scripts

Run level scripts allow users to start and stop selected applications while changing the run level. Scripts beginning with K are stop scripts, while scripts beginning with S are start scripts.

These scripts reside on the subdirectory that is specific to the run level they belong to, and each subdirectory has the form `rc`*`n`*.d, where *`n`* is the run level:

- /etc/rc.d/rc2.d
- /etc/rc.d/rc3.d

- /etc/rc.d/rc4.d
- /etc/rc.d/rc5.d
- /etc/rc.d/rc6.d
- /etc/rc.d/rc7.d
- /etc/rc.d/rc8.d
- /etc/rc.d/rc9.d

The /etc/rc.d/rc script will run the start script it finds in the specified directory, and execute it when the run level changes. The script will first run stop application scripts, then runs start application scripts.

4.6 An introduction to the rc.* files

The rc files are executed as part of the initialization process; they prepare the system to be configured and ready for operation.

4.6.1 rc.boot file

The /sbin/rc.boot file is a shell script that is called by the simple shell init and the standard `init` command to bring up a system. It controls the machine boot process. When the system is booting, the /sbin/rc.boot file is called on each boot phases, each time being passed a different parameter.

Depending upon the type of boot device, the rc.boot file configures devices and also calls the appropriate applications. Appropriate applications include:

- Booting from disk (boot phase 1)
- Varying on a root volume group (boot phase 2)
- Enabling file systems (boot phase 2)
- Calling the BOS installation programs or diagnostics

Note: The rc.boot program is only called by an `init` process; executing the rc.boot script on a system that is already running may cause unpredictable results.

4.6.2 /etc/rc file

The /etc/rc file performs normal startup initialization; its entry in the /etc/inittab file is located after the rc.boot entry. The `init` command reads the /etc/inittab file and creates a process for the /etc/rc file. The contents of the /etc/rc file are installation specific. If all of the necessary operations complete successfully, the file exits with a zero return code that allows the **init** command to start loggers to complete normal initialization and startup.

Many bringup functions are done by the /etc/rc file, such as:

- Vary on all volume groups marked as auto-varyon.
- Activate all paging spaces listed on /etc/swapspaces (using the **swapon -a** command).
- Configure all dump devices (using the **sysdumpdev -q** command).
- Perform file system checks (using the **fsck -fp** command).
- Perform mounting of file systems marked as mount=true on the /etc/filesystems file (using the **mount all** command).

> **Note:** The /, /usr, /var, and /tmp file systems are mounted automatically on the boot process (phase 2). The /etc/rc file does not try to mount these file systems again when it runs the **mount all** command.

4.6.3 rc.net file

The /etc/rc.net file is a shell script that contains network configuration information The stanzas allow you to enable the network interfaces and set the host name, the default gateway, and any static routes for the current host. This file can be used as a one-step configuration alternative to using individually the set of commands and files necessary to configure a host.

The rc.net shell script is run by the configuration manager program during the second phase of configuration. If TCP/IP is installed, a second script, rc.tcpip, is run from the **init** command after the second phase of configuration has completed and after the **init** command has started the SRC master.

Stanzas in the file should appear in the order in which they are presented here.

The /etc/rc.net shell script may also be run by the configuration manager program (cfgmgr) if the **cfgmgr** command is run after system configuration is completed. It is often run at other times to configure new devices that have been added to the system since boot time. If the **cfgmgr** command runs rc.net, both the configuration methods and rc.net itself check to see if networking devices are already in the Available state. If so, the values of device attributes are not changed to avoid overwriting any configuration changes that have been made since boot time.

If /etc/rc.net is run without calling the **cfgmgr** command, device attributes will be reset to the values in the ODM database regardless of the states of the devices.

This allows a system's configuration to be restored to the values specified in the ODM database.

4.6.4 rc.tcpip file

The /etc/rc.tcpip file is a shell script that, when executed, uses SRC commands to initialize selected daemons. The rc.tcpip shell script is automatically executed with each system restart. It can also be executed at any time from the command line.

Most of the daemons that can be initialized by the rc.tcpip file are specific to TCP/IP. These daemons are:

- **inetd** (started by default)
- **gated**
- **routed**
- **named**
- **timed**
- **rwhod**

> **Note:** Running the **gated** and **routed** daemons at the same time on a host may cause unpredictable results.

There are also daemons specific to the base operating system or to other applications that can be started through the rc.tcpip file. These daemons are:

- **lpd**
- **portmap**
- **sendmail**
- **syslogd** (started by default)

The following examples are provided as a reference:

- The following stanza starts the **syslogd** daemon:

  ```
  #Start up syslog daemon (for error and event logging)
  start /usr/sbin/syslogd "$src_running"
  ```

- The following stanza starts the **lpd** daemon:

  ```
  #Start up print daemon
  start /usr/sbin/lpd "$src_running"
  ```

- The following stanza starts the **routed** daemon, but not the **gated** daemon:

```
#Start up routing daemon (only start ONE)
start /usr/sbin/routed "$src_running" -g
#start /usr/sbin/gated "$src_running"
```

Note: To disable a daemon from starting when booting, add a pound sign (#) at the beginning of the line.

5

Configuration

In this chapter, you find more information about the functions of the Object Data Manager (ODM). Configuration management involves adding new devices to the system, their configuration, troubleshooting, and solutions to problems that result.

In addition, topics such as SMIT, network configuration, device configuration, and NFS are covered.

© Copyright IBM Corp. 2006. All rights reserved.

5.1 Object Data Manager (ODM)

The ODM is a repository in which the operating system keeps information regarding your system, such as devices, software, or TCP/IP configuration. The ODM is an object-oriented database that contains vital data, keeps this data consistent with the actual state of the system, and prevents the administrator from altering it by mistake. The ODM is one of the key features that makes AIX 5L different from other versions of UNIX®.

System data managed by the ODM includes:

- Device configuration information
- Display information for SMIT (menus, selectors, and dialogs)
- Vital product data for installation and update procedures
- Communication configuration information
- System resource controller data information
- Error log and dump information
- Network Installation Manager (NIM) information

ODM data is stored in binary format. You cannot modify ODM files with a text editor. You must use special commands that are designed to interact with the ODM.

The basic components of the ODM are

Object classes Each file of the database is an object class. Each object class consists of objects having similar definitions.

Objects Each object is one record in an object class. It is a stand-alone entity and has one or more descriptors.

Descriptors The descriptors describe the layout of the objects. They determine the name and data type of the fields that are part of the object class. The descriptors of an object and their associated values can be located and changed using ODM commands.

Examples of information contained in the ODM and the corresponding classes that store this data include:

- Predefined device information: PdDv, PdAt, and PdCn
- Customized device information: CuDv, CuAt, and CuDep
- Software vital product data: history, inventory, lpp, and product
- SMIT menus: sm_menu_opt, sm_name_hdr, sm_cmd_hdr, and sm_cmd_opt

- Error log, alog and dump information: SWservAt
- System Resource Controller: SRCsubsys and SRCsubsvr
- Network Installation Manager: nim_attr, nim_object, and nim_pdattr

ODM information is divided in three parts in order to support diskless or dataless systems. The names of these three directories as follows:

/usr/lib/objrepos Contains the predefined objects classes, SMIT menu object classes and the four object classes used by SWVPD for the /usr part of the installable software product. The object classes in this repository can be shared across the network by /usr clients, dataless and diskless workstations. Software installed in the /usr part can be shared among several machines with compatible hardware architecture.

/usr/share/lib/objrepos

Contains the four object classes used by the SWVPD for the /usr/share part of the installable software product. The /usr/share part of a software product contains files that are not hardware dependent. They can be shared among several systems, even if these have a different hardware architecture. An example are terminfo files that describe terminal capabilities. Because terminfo is used on many UNIX systems, terminfo files are part of the /usr/share part of the software product.

/etc/objrepos Contains the customized devices object classes and the four object classes used by SWVPD for the / part of the installable software product. To access information in the other directories, this directory contains symbolic links to the predefined devices object classes. These links are needed because the ODMDIR variable points to only /etc/objrepos. It contains the part of the product that cannot be shared with other systems. Most of this software requiring a separate copy for each machine is associated with the configuration of the machine or product.

5.1.1 ODM commands

Having a good knowledge of the ODM is very important for understanding the way your system functions and for analyzing and troubleshooting problems. However, you should be aware that modifying data contained in ODM using ODM commands should be done only when managing your system using the usual command-line commands has become impossible. Both SMIT and command-line commands are designed in such manner to keep ODM and system status synchronized at all times, so ideally you might never have to use ODM commands. Using these commands in a wrong way may render your system unusable, so use them only when you really know what you are doing. However, you should know that these commands exist and have a minimum knowledge about them.

The ODM commands are:

odmadd Adds objects to an object class. The **odmadd** command takes an ASCII stanza file as input and populates object classes with objects found in the stanza file.

odmchange Changes specific objects in a specified object class.

odmcreate Creates empty object classes. The **odmcreate** command takes an ASCII file describing object classes as input and produces C language.h and.c files to be used by the application accessing objects in those object classes.

odmdelete Removes objects from an object class.

odmdrop Removes an entire object class.

odmget Retrieves objects from object classes and puts the object information into **odmadd** command format.

odmshow Displays the description of an object class. The **odmshow** command takes an object class name as input and puts the object class information into the **odmcreate** command format.

Because ODM is a database, ODM queries can accept parameters linked with operators that are common when interrogating databases.

When they execute, ODM commands use the value of the ODMDIR variable. Its default value is /etc/objrepos, but it can be changed.

5.1.2 Examples of using the ODM

The following sections describe different ODM scenarios.

Device configuration

ODM keeps all data necessary for device configuration.

For example, the class named Predefined Devices (PdDv) contains entries for all devices that can be supported by AIX 5L Version 5.3. Important attributes of objects in this class include type, class, subclass, prefix, base, detectable, led, setno, catalog, DvDr, Define, Configure, Change, Unconfigure, Undefine, Start, Stop, and uniquetype.

In Example 5-1, we used the **odmget** command to interrogate ODM PdDv class about all objects whose type start with letters lv.

Example 5-1 Using odmget to interrogate the PdDv class

```
# odmget -q "type LIKE lv*" PdDv
PdDv:
        type = "lvtype"
        class = "logical_volume"
        subclass = "lvsubclass"
        prefix = "lv"
        devid = ""
        base = 1
        has_vpd = 0
        detectable = 0
        chgstatus = 0
        bus_ext = 0
        fru = 0
        led = 0
        setno = 1
        msgno = 699
        catalog = "cmdlvm.cat"
        DvDr = ""
        Define = ""
        Configure = ""
        Change = ""
        Unconfigure = ""
        Undefine = ""
        Start = ""
        Stop = ""
        inventory_only = 0
        uniquetype = "logical_volume/lvsubclass/lvtype"

PdDv:
        type = "lvdd"
```

```
        class = "lvm"
        subclass = "lvm"
        prefix = ""
        devid = ""
        base = 1
        has_vpd = 0
        detectable = 0
        chgstatus = 1
        bus_ext = 0
        fru = 0
        led = 1425
        setno = 1
        msgno = 52
        catalog = "devices.cat"
        DvDr = "hd_pin"
        Define = "/usr/lib/methods/deflvm"
        Configure = "/usr/lib/methods/cfglvdd"
        Change = ""
        Unconfigure = ""
        Undefine = ""
        Start = ""
        Stop = ""
        inventory_only = 0
        uniquetype = "lvm/lvm/lvdd"
```

Software vital product data

The ODM maintains a full inventory of all software products installed on your system.

For example, a class named lpp contains information about currently installed software products, such as ID, name, version, and release.

In Example 5-2, we used the **odmget** command to interrogate the ODM class lpp about all the software installed on the system and we selected the first 30 lines of the output.

Example 5-2 Using odmget to interrogate the lpp class

```
# odmget lpp|head -30
lpp:
        name = "__SWVPD_CTL__"
        size = 0
        state = 0
        cp_flag = 0
        group = ""
        magic_letter = ""
        ver = 0
        rel = 0
```

```
        mod = 0
        fix = 0
        description = ""
        lpp_id = 217

lpp:
        name = "bos.rte"
        size = 0
        state = 5
        cp_flag = 262419
        group = ""
        magic_letter = "I"
        ver = 5
        rel = 3
        mod = 0
        fix = 10
        description = "Base Operating System Runtime"
        lpp_id = 1
```

For example, a class named history contains information about the installation and updates of all software products.

In Example 5-3, we used the **odmget** command to interrogate ODM class history about the updates of a software component having lpp_id 100. Notice how we changed the value of the ODMDIR variable, and the result of the output was different.

Example 5-3 Using odmget to interrogate the history class

```
# echo $ODMDIR
/etc/objrepos
# odmget history|grep -p "lpp_id = 100"
history:
        lpp_id = 100
        event = 1
        ver = 5
        rel = 3
        mod = 0
        fix = 0
        ptf = ""
        corr_svn = ""
        cp_mod = ""
        cp_fix = ""
        login_name = "root"
        state = 1
        time = 6551686
        comment = ""
```

```
# export ODMDIR=/usr/lib/objrepos
# odmget history|grep -p "lpp_id = 100"|more
history:
        lpp_id = 100
        event = 1
        ver = 5
        rel = 3
        mod = 0
        fix = 0
        ptf = ""
        corr_svn = ""
        cp_mod = ""
        cp_fix = ""
        login_name = "root"
        state = 1
        time = 6551634
        comment = ""
```

LVM

The ODM also maintains a copy of all data used by LVM. Commands that affect the LVM are designed so that data from VGDAs located on hard disks are always synchronized with information stored in ODM.

For example, ODM class named CuAt contains customized device-specific attribute information.

In Example 5-4, we used the **odmget** command to interrogate ODM class CuAt about all attributes of the object hdisk0.

Example 5-4 Using odmget to interrogate the CuAt class

```
# odmget -q name=hdisk0 CuAt

CuAt:
        name = "hdisk0"
        attribute = "unique_id"
        value = "2708E6X7643E10IC35L073UCDY10-003IBMscsi"
        type = "R"
        generic = ""
        rep = "nl"
        nls_index = 79

CuAt:
        name = "hdisk0"
        attribute = "pvid"
        value = "00c5e9de00091d6f0000000000000000"
        type = "R"
        generic = "D"
```

```
        rep = "s"
        nls_index = 2

CuAt:
        name = "hdisk0"
        attribute = "size_in_mb"
        value = "73400"
        type = "R"
        generic = "D"
        rep = "nr"
        nls_index = 60

CuAt:
        name = "hdisk0"
        attribute = "led"
        value = "0x57D"
        type = "Z"
        generic = ""
        rep = "nr"
        nls_index = 0

CuAt:
        name = "hdisk0"
        attribute = "message_no"
        value = "87"
        type = "T"
        generic = ""
        rep = "nl"
        nls_index = 0

CuAt:
        name = "hdisk0"
        attribute = "diag_scsd"
        value = "a500000107000507000c0b"
        type = "R"
        generic = ""
        rep = "s"
        nls_index = 0
```

5.2 System Management Interface Tool

The AIX 5L System Management Interface Tool (SMIT) provides an alternative to the typical method of using complex command syntax, valid parameter values, and custom shell path names for managing and maintaining your operating system configuration. Though the Web-based System Manager is a much more modern and intuitive tool for system management, SMIT remains a very popular tool and therefore is covered on the certification exam.

SMIT offers the following features:

- Two modes of operation.
- An interactive, menu-driven user interface.
- User assistance.
- System management activity logging.
- Fast paths to system management tasks.
- User-added SMIT screens.
- AIX 5L library information is listed under Technical Publications.

5.2.1 Modes of operation

SMIT runs in two modes: ASCII (non-graphical) and X-Window (graphical). ASCII SMIT can run on both terminals and graphical displays. The graphical mode, which supports a mouse and point-and-click operations, can be run only on a graphical display and with X-Window support. The ASCII mode is often the preferred way to run SMIT because it can be run from any machine. To start the ASCII mode, type at the command line:

smitty or **`smit -a`**

To start the graphical mode, type:

`smit` or **`smit -m`**

> **Note:** If you execute the above commands from a terminal or your TERM attribute is set to a non-graphical setting, SMIT will always run in the ASCII mode.

5.2.2 End-user interface

SMIT is an interactive, menu-driven user interface that allows you to more easily perform routine system management tasks and to manage and maintain your operating system configuration. System management tasks are grouped by application and presented in a series of menu, selector, and dialog screens. For example, all common software installation tasks are grouped in the Software Installation and Management application. This task-oriented structure makes SMIT easy to use, allowing even novice users to perform routine system administration tasks.

SMIT screens display the actual system configuration. The displayed information varies from system to system, based on what is installed on a particular system. Adding customized system management tasks for your own applications or changing the existing SMIT screen information is one example of what causes this variation. Another example can be seen in the Devices screens. The available system management tasks are based on what type of devices, such as network and storage adapters, disk drives, and other I/O devices, are installed on the system.

5.2.3 SMIT screens

SMIT uses three types of screens: menu, selector, and dialog screens. SMIT uses the data provided in these screens as options and arguments to create and run high-level command strings to perform a selected task. This data is described in stanza files that are stored in the Object Data Manager (ODM). When you press the Enter key or otherwise start a task from SMIT, the dialog executes a shell script that processes the underlying commands to perform the task. In the SMIT graphical mode, the command string associated with the task displays at the top of the screen as it runs. In the ASCII mode, you can see the command string that will be used before you actually run the task by pressing the F6 command key.

Menu screens display a list of items that you can select. Menu items are typically system management tasks or classes of tasks that you can perform. Starting from the System Management menu (the main SMIT menu), you select an item defining a broad range of system tasks. You continue to make selections from menus until you reach the final dialog, which typically collects the information and performs the task.

Selector screens, often presented as a pop-up menu, display a list of items from which you specify or select a particular item. Items in a selector screen are typically system objects, such as printers, or the attributes of objects, such as serial or parallel printer mode. The menu screen provides necessary information that is used by the dialog screen.

Dialog screens are the interface to a command or task that you perform. Each dialog executes one or more commands or shell functions. A command can be run from any number of dialogs.

5.2.4 System management tasks

You can perform most system management tasks from the SMIT interface. Table 5-1 lists the main tasks that display in the System Management menu. Selecting a task from this menu presents additional menus containing tasks, many of which are listed here, that you can perform from that menu.

Table 5-1 System management tasks

Application	System management tasks
Software Installation and Maintenance	Installing new software, updating software, installing fixes, listing installed software, and backing up and restoring the system image.
Software License Management	Adding and deleting node-locked licenses, adding and removing server licenses, managing licenses, and listing licenses.
Devices	Adding, changing, showing, and deleting physical and logical devices, configuring and unconfiguring devices, listing installed devices; and managing PCI hot plugs.
System Storage™ Management (Physical & Logical Storage)	Managing logical volumes, volume groups, physical disk drives, and paging space, managing file systems, managing files and directories, and tasks for backing up and restoring the system.
Security and Users	Managing user accounts and groups, passwords, login controls, and roles.
Communications Applications and Services	Configuring all installed communications options and applications, including TCP/IP, NFS server or client, Network Information System (NIS), and Domain Name Service (DNS).
Print Spooling	Configuring and managing printers, print queues, print jobs, and virtual printers.
Problem Determination	Running hardware diagnostics, performing system traces, initiating system dumps, printing error logs, and verifying software installation and requisites.
Performance and Resource Scheduling	Scheduling jobs, managing resource processes, configuring and enabling Power Management™, configuring and using the Workload Manager, running system traces, and reporting system activity.

Application	System management tasks
System Environments	Starting and stopping the system, configuring and managing system environment parameters such as language, date, user interface, and time, managing system logs, managing the remote reboot facility, and managing system hang detection.
Processes and Subsystems	Managing subsystems, processes, and subservers.

5.3 Linux applications under AIX 5L

This section discusses how Linux applications can be deployed on the AIX 5L platform using open source tools.

5.3.1 Linux affinity

AIX 5L affinity with Linux gives you the capability to easily compile and run Linux applications on AIX 5L. AIX Toolbox for Linux Applications is a group of GNU and open source tools and utilities for building and deploying Linux applications on AIX 5L. It includes a collection of programs that have already been recompiled and tested for the AIX 5L environment.

AIX 5L has been developed using UNIX industry standards and, as such, there is a high degree of compatibility at the API level between AIX 5L and Linux. This degree of similarity is such that many Linux applications can be recompiled and run on AIX 5L using the AIX Toolbox for Linux Applications.

AIX 5L affinity with Linux uses an Application Programming Interface (API) approach to providing Linux application interoperability with AIX 5L. This approach is not an environment or an additional layer or wrapper to run Linux applications in or on. It is the integration of Linux compatible APIs and header files into AIX 5L. Thus, recompiled Linux applications are treated as native AIX 5L applications and have access to all the reliability, scalability, and availability of AIX 5L. The result is a tighter integration of the application to the operating system than can be achieved with an Application Binary Interface (ABI) approach.

The Linux applications deployed on AIX 5L have full access to all AIX 5L functionality, just like an application natively developed for AIX 5L.

Open source software in the Toolbox

The following Web site contains a detailed information about the current content of the Toolbox:

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

The following is a sample of the software that the Toolbox 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

Toolbox RPM packaging format

The tools and applications that come with the Toolbox are all in RPM format. The RPM Package Manager is a 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, un-installing, upgrading, deleting, and building packages.

How to recompile and run a Linux application on AIX 5L

To recompile and run your applications on AIX 5L, take the application source that you are currently using, get the appropriate GNU tools from the AIX Toolbox for Linux Applications (for example, **make**, **automake**, **autoconf**, and **gcc**), use these tools to create the binaries for AIX 5L, and then run the application on AIX 5L. The applications will connect to AIX 5L using the integrated APIs and header files. If you do not have access to the application source code, then contact the developer and ask them to provide a recompiled binary for use on AIX 5L.

The Linux application being recompiled to run on AIX 5L must be written using standard Linux APIs, and use the GNU gcc and g++ compilers.

5.3.2 How to install Linux applications on AIX 5L

This section discusses the topic of how to add open source applications to your AIX 5L system from AIX Toolbox for Linux Applications CD.

The AIX Toolbox for Linux Applications CD that is shipped with your base operating system software contains the most commonly used open source applications that you can use with the AIX 5L operating system. Your options for installing from this CD include:

- Using the SMIT **install_software** fast path to install RPM packages from the AIX Toolbox for Linux Applications CD.
- Using the **geninstall** command to install RPM packages from the AIX Toolbox for Linux Applications CD.
- Installing a bundle. Bundles group the applications you need for a basic Linux operating environment, basic desktop use, GNOME or KDE desktop use, or application development.
- Installing from a directory of packages classified by function. These directory groupings cover a broad range of applications, shell environments, network applications, development tools, application libraries, and so on.
- Installing a single package for a particular application.

Installing RPM packages

To install the cdrecord and mtools RPM packages using SMIT, do the following:

1. Run the SMIT **install_software** fast path.
2. Enter the device name for the AIX Toolbox for Linux Applications CD (for example, /dev/cd0), and press Enter.
3. Press the F4 key to list the contents of the device.
4. Select the cdrecord and mtools packages, and press Enter.
5. Accept the default values for the rest of the Install Software menu fields, and press Enter.
6. Confirm that you do want to install the software, and press Enter.

The software installation process begins at this point.

Install using the geninstall command

To install the cdrecord and mtools RPM packages from the command line, type the following:

```
geninstall -d/dev/cd0 R:cdrecord R:mtools
```

The software installation process begins at this point.

5.3.3 Install using the rpm command

Use the **rpm** command, which is automatically installed with the base operating system for AIX 5L Version 5.1 and later, to install the bundles required for the GNOME desktop and the bc application package. Complete instructions are available in the readme file for the AIX Toolbox for Linux Applications.

1. With your system powered on and AIX 5L Version 5.1 or later running, insert the AIX Toolbox for Linux Applications CD into the CD-ROM drive of your system.
2. With root authority, mount the CD-ROM drive using the following command:

   ```
   # mount -vcdrfs -oro /dev/cd0 /mnt
   ```

   The -v flag specifies the virtual file system type of cdrfs. The -o flag specifies the ro option, which means the mounted file is read-only. The device name is /dev/cd0. The directory in which you want to mount the CD-ROM is /mnt.
3. Change to the /mnt directory by using the following command:

   ```
   # cd /mnt
   ```

4. Use the **ls** command to list the contents of the CD. The listing contains the following, which you can view or print:
   - The readme file contains complete instructions for installing from this CD.
   - The CONTENTS file lists all packages available on this CD and provides a short description of the purpose for each package.
5. In your Web browser, open the /mnt/LICENSES/index.html file to view software licensing information.
6. In your terminal window, change to the ezinstall/ppc directory by using the following command:

   ```
   # cd /mnt/ezinstall/ppc
   ```

In the next step, use the **rpm** command to install GNOME by installing four bundles (Base, Desktop Base, GNOME Base, and GNOME Apps). Alternatively, you can install all necessary packages using the smit **install_bundle** fast path and selecting the GNOME bundle.

7. Install GNOME by using the following sequence of commands:

   ```
   #rpm -Uhv ezinstall/ppc/base/*
   #rpm -Uhv ezinstall/ppc/desktop.base/*
   #rpm -Uhv ezinstall/ppc/gnome.base/*
   #rpm -Uhv ezinstall/ppc/gnome.apps/*
   ```

   The -U flag updates any earlier versions of each package that you might have on your system. The -h flag prints hash marks (#) at timed intervals to indicate that the installation is progressing. The -v flag displays relevant informational or error messages that occur during the installation.

If your **rpm** command returns an error, it is probably caused by one of the following:

– Not enough space in your current file system. Resize the file system or change your mount point.

– The package is already installed. The **rpm** program discovered an existing package of the same name and version level, so it did not replace the package. A script on the CD installs only those packages from a directory that are not already installed on your system, as shown in the following example:

```
# /mnt/contrib/installmissing.sh ezinstall/ppc/desktop.base/*
```

– Failed dependencies. The packages listed in the error message must be installed before you can install this package or bundle.

8. Install the bc application package by using the following command:

```
# rpm -Uhv RPMS/ppc/bc-*.rpm
```

How to enable the use of the Toolbox commands

To execute the Linux version of a 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 some AIX 5L 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 /usr/linux/bin prior to AIX 5L 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, 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.

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 are searched. We show how to set the PATH variable in the current Korn shell environment below:

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

In the following example, first we use the AIX 5L **nl** command, then export the new PATH environment variable, pointing to /usr/linux/bin first, and 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:

```
#root@fenris:/: print $PATH
/usr/bin:/etc:/usr/sbin:/usr/ucb:/usr/bin/X11:/sbin
#root@fenris:/: nl -?
nl: 0653-467 illegal option: -?
Usage: nl [-b Type] [-d Delimiter1 Delimiter2] [-f Type] [-h Type]
[-i Number] [-l Number] [-n Format] [-p] [-s Separator]
[-v Number] [-w Number] [File]
#root@fenris:/: export PATH=/usr/linux/bin:$PATH
#root@fenris:/: print $PATH
/usr/linux/bin:/usr/bin:/etc:/usr/sbin:/usr/ucb:/usr/bin/X11:/sbin
#root@fenris:/: nl -?
/usr/linux/bin/nl: invalid option -- ?
Try `/usr/linux/bin/nl --help' for more information.
```

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 5L, 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:

```
alias command=absolute 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:

```
alias rm=/usr/linux/bin/rm
```

5.4 Network File System

The Network File System (NFS) is a distributed file system that allows users to access files and directories of remote servers as though they were local. For example, you can use operating systems commands to create, remove, read, write, and set file attributes for remote files and directories. NFS is independent of machine types, operating systems, and network architectures because of its use of remote procedure calls (RPC) for these services.

For the successful implementation of an NFS environment, you need the following things:

1. The NFS daemons should be running on the server and the clients.
2. The file systems that need to be remotely available will have to be exported.
3. The exported file systems need to be mounted on the remote (client) systems.

5.4.1 NFS services

NFS provides its services through a client-server relationship.

The following are a list of terms that are used throughout this discussion:

Server A computer that makes its file systems, directories, and other resources available for remote access.

Clients The computers, or their processes, that use a server's resources.

Export The act of making file systems available to remote clients.

Mount The act a client needs to do to access the file systems that a server exports.

Access to exported directories can be restricted to specific clients.

The major services provided by NFS are:

Mount From the **/usr/sbin/rpc.mountd** daemon on the server and the **/usr/sbin/mount** command on the client. The **mountd** daemon is a Remote Procedure Call (RPC) that answers a client request to mount a file system. The **mountd** daemon provides a list of currently mounted file systems and the clients on which they are mounted.

Remote file access From the **/usr/sbin/nfsd** daemon on the server and the **/usr/sbin/biod** daemon on the client. Handles client requests for files. The **biod** daemon runs on all NFS client systems. When a user on a client wants to read or write to a file on a server, the **biod** daemon sends this request to the server.

Start the NFS daemons for each system (whether client or server). The NFS daemons, by default, are not started on a newly installed system. When a system is first installed, all of the files are placed on the system, but the steps to activate NFS are not taken. The daemons can be started by using either of the following two methods:

1. Using the SMIT fast path **smitty mknfs**
2. Using the **mknfs** command to start the NFS daemons immediately, and this should produce the following:

```
# mknfs -N
0513-059 The portmap Subsystem has been started. Subsystem PID is 23734.
Starting NFS services:
0513-059 The biod Subsystem has been started. Subsystem PID is 27264.
0513-059 The nfsd Subsystem has been started. Subsystem PID is 30570.
0513-059 The rpc.mountd Subsystem has been started. Subsystem PID is 28350.
0513-059 The rpc.statd Subsystem has been started. Subsystem PID is 15298.
0513-059 The rpc.lockd Subsystem has been started. Subsystem PID is 30976.
#
```

You can use System Resource Controller to start the NFS daemons if they are not already started.

The NFS daemons can be started individually or all at once. Use the following command to start NFS daemons individually:

`#startsrc -s` *daemon*

where *daemon* is any one of the SRC controlled daemons. For example, to start the **nfsd** daemon, run:

```
#startsrc -s nfsd
```

Use the following command to start all of the NFS daemons:

```
#startsrc -g nfs
```

5.4.2 Exporting NFS directories

This section discusses the use of the **exportfs** command.

Exporting an NFS directory using SMIT

To export file systems using SMIT, follow this procedure:

1. Verify that NFS is already running using the command **lssrc -g nfs**. As in the following example, the output should indicate that the **nfsd** and the **rpc.mountd** daemons are active. If they are not, start NFS using the instructions.

```
#lssrc -g nfs
Subsystem         Group            PID     Status
 biod             nfs              15740   active
 nfsd             nfs              11376   active
 rpc.mountd       nfs              5614    active
 rpc.statd        nfs              16772   active
 rpc.lockd        nfs              15496   active
#
```

2. Use **smitty mknfsexp** on the server to export the directory; the SMIT screen is as shown in Figure 5-1.

```
                       Add a Directory to Exports List

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[TOP]                                                   [Entry Fields]
* Pathname of directory to export                     [/home1]               /
  Anonymous UID                                       [-2]
  Public filesystem?                                   no                    +
* Export directory now, system restart or both         both                  +
  Pathname of alternate exports file                  []
  Allow access by NFS versions                        []                     +
  External name of directory (NFS V4 access only)     []
  Referral locations (NFS V4 access only)             []
  Replica locations                                   []
  Ensure primary hostname in replica list              yes                   +
  Allow delegations?                                   no                    +
* Security method                                     [sys,krb5p,krb5i,krb5,> +
*     Mode to export directory                         read-write            +
[MORE...23]

F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 5-1 Add a Directory to Exports List

3. Specify /home1 in the PATHNAME of directory to export field, set the MODE to export directory field to read-write, and set the EXPORT directory now, system restart, or both field to both.
4. Specify any other optional characteristics you want, or accept the default values by leaving the remaining fields as they are.
5. When you have finished making your changes, SMIT updates the /etc/exports file. If the /etc/exports file does not exist, it will be created.
6. Repeat steps 3 through 5 for directories that need to be exported.
7. If NFS is currently running on the servers, enter:

```
#/usr/sbin/exportfs -a
```

   The -a option tells the **exportfs** command to send all information in the /etc/exports file to the kernel. If NFS is not running, start NFS using the instructions given before.

8. Verify that all file systems have been exported properly as follows:

   On the server:

```
#showmount -e MyServer
export list for MyServer:
/home1     (everyone)
/home2     (everyone)
/home3     (everyone)
/home4     (everyone)
#
```

We have used MyServer as the name of a server with the **showmount** command. We will be using the same name in most of the following discussions.

Exporting an NFS directory using a text editor

To export file systems using a text editor, follow this procedure:

1. Open the /etc/exports file with your favorite text editor, for example:

```
# vi /etc/exports
```

2. Create an entry for each directory to be exported by using the full path name of the directory, as shown in Figure 5-2.

.

```
/home1
/home2
/home3
/home4
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"/etc/exports" 4 lines, 28 characters
```

Figure 5-2 Content of /etc/exports

3. List each directory to be exported starting in the left margin. No directory should include any other directory that is already exported. Save and close the /etc/exports file.
4. If NFS is currently running on the servers, Enter:

```
# /usr/sbin/exportfs -a
```

Exporting an NFS directory temporarily

A file system can be exported when needed, and as such, does not change the /etc/exports file. This is done by entering:

`# exportfs -i` */dirname*

where */dirname* is the name of the file system you want to export. The **`exportfs -i`** command specifies that the /etc/exports file is not to be checked for the specified directory, and all options are taken directly from the command line.

5.4.3 Un-exporting an NFS directory

You can un-export an NFS directory by using one of the following procedures:

- To un-export an NFS directory using SMIT:
  a. On the Server, enter the following command to remove /home4 export:

     ```
     # smitty rmnfsexp
     ```

  b. Enter /home4 in the PATHNAME of exported directory to be removed field.

  The directory is now removed from the /etc/exports file and is un-exported.
- To un-export an NFS directory using a text editor:
  a. Open the /etc/exports file with a text editor.
  b. Find the entry for the directory you wish to un-export, that is, /home4, and then delete that line.
  c. Save and close the /etc/exports file.
  d. If NFS is currently running, enter:

     `# exportfs -u` *dirname*

     where *dirname* is the full path name of the directory (/home4) you just deleted from the /etc/exports file.

5.4.4 Mounting an NFS directory

There are three types of NFS mounts: predefined, explicit, and automatic.

Predefined mounts are specified in the /etc/filesystems file. Each stanza (or entry) in this file defines the characteristics of a mount, as shown in Figure 5-3.

```
/home1:
        dev             = "/home1"
        vfs             = nfs
        nodename        = MyServer
        mount           = true
        options         = bg,hard,intr
        account         = false
```

Figure 5-3 Example NFS stanza in the /etc/filesystems file

Data, such as the host name, remote path, local path, and any mount options, is listed in this stanza. Predefined mounts should be used when certain mounts are always required for proper operation of a client.

Explicit mounts serve the needs of the root user. Explicit mounts are usually made for short periods of time when there is a requirement for occasional unplanned mounts. Explicit mounts can also be used if a mount is required for special tasks, and that mount should not be generally available on the NFS client. These mounts are usually fully qualified on the command line by using the **mount** command with all needed information.

Explicit mounts do not require updating the /etc/filesystems file. File systems mounted explicitly remain mounted unless explicitly unmounted with the **umount** command or until the system is restarted.

Automatic mounts are controlled by the **automount** command, which causes the AutoFS kernel extension to monitor specified directories for activity. If a program or user attempts to access a directory that is not currently mounted, then AutoFS intercepts the request, arranges for the mount of the file system, and then services the request.

NFS mounting process

Clients access files on the server by first mounting a server's exported directories. When a client mounts a directory, it does not make a copy of that directory. Rather, the mounting process uses a series of remote procedure calls to enable a client to access the directories on the server transparently. The following describes the mounting process:

1. When the server starts, the /etc/rc.nfs script runs the **exportfs** command, which reads the server /etc/exports file and then tells the kernel which directories are to be exported and which access restrictions they require.
2. The **rpc.mountd** daemon and several **nfsd** daemons (eight, by default) are then started by the /etc/rc.nfs script.
3. When the client starts, the /etc/rc.nfs script starts several **biod** daemons (eight, by default), which forward client mount requests to the appropriate server.
4. Then the /etc/rc.nfs script executes the **mount** command, which reads the file systems listed in the /etc/filesystems file.
5. The **mount** command locates one or more servers that export the information the client wants and sets up communication between itself and that server. This process is called binding.
6. The **mount** command then requests that one or more servers allow the client to access the directories in the client /etc/filesystems file.
7. The server **rpc.mountd** daemon receives the client mount requests and either grants or denies them. If the requested directory is available to that client, the **rpc.mountd** daemon sends the client's kernel an identifier called a file handle.
8. The client kernel then ties the file handle to the mount point (a directory) by recording certain information in a mount record.

Once the file system is mounted, the client can perform file operations. When the client does a file operation, the **biod** daemon sends the file handle to the server, where the file is read by one of the **nfsd** daemons to process the file request. Assuming the client has access to perform the requested file operation, the **nfsd** daemon returns the necessary information to the client's **biod** daemon.

> **Note:** The mount points for all NFS mounts must exist on your system before you can mount a file system with one exception: If the **automount** daemon is used, it may not be necessary to create mount points. See “Mounting an NFS directory automatically” on page 150.

Establishing predefined NFS mounts

You can establish predefined NFS mounts using one of the following procedures.

> **Note:** Define the bg (background) and intr (interruptible) options in the /etc/filesystems file when establishing a predefined mount that is to be mounted during system startup. Mounts that are non-interruptible and running in the foreground can hang the client if the network or server is down when the client system starts up. If a client cannot access the network or server, the user must start the machine again in maintenance mode and edit the appropriate mount requests.

To establish predefined mounts through SMIT (Figure 5-4), use the following command:

```
# smitty mknfsmnt
```

```
                         Add a File System for Mounting

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[TOP]                                                     [Entry Fields]
* PATHNAME of mount point                               []                  /
* PATHNAME of remote directory                          []
* HOST where remote directory resides                   []
  Mount type NAME                                       []
* Use SECURE mount option?                               no                +
* MOUNT now, add entry to /etc/filesystems or both?      now               +
* /etc/filesystems entry will mount the directory        no                +
   on system RESTART.
* MODE for this NFS file system                          read-write        +
* ATTEMPT mount in foreground or background              background        +
  NUMBER of times to attempt mount                      []                  #
  Buffer SIZE for read                                  []                  #
  Buffer SIZE for writes                                []                  #
[MORE...26]

F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 5-4 Add a File System for Mounting screen

Specify values in this screen for each mount you want predefined. You must specify a value for each required field (those marked with an asterisk (*) in the left margin). You may specify values for the other fields or accept their default values. This method creates an entry in the /etc/filesystems file for the desired mount and attempts the mount.

To establish the NFS default mounts by editing the /etc/filesystems file (only use this method under special circumstances), do the following:

1. Open the /etc/filesystems file on the client with a text editor. Add entries for each of the remote file systems that you want mounted when the system is started. For example:

   ```
   /home1:
         dev = /home1
         mount = false
         vfs = nfs
         nodename = MyServer
         options = ro,soft
         type = nfs_mount
   ```

   This stanza directs the system to mount the /home1 remote directory over the local mount point of the same name. The file system is mounted as read-only (ro). Because it is also mounted as soft, an error is returned in the event the server does not respond. By specifying the type parameter as nfs_mount, the system attempts to mount the /home1 file system (along with any other file systems that are specified in the type = nfs_mount group) when the **mount -t nfs_mount** command is issued.

   The following example stanza directs the system to mount the /home2 file system at system startup time. If the mount fails, the bg option tells the system to continue attempting the mount in the background:

   ```
   /home2:
      dev = /home2
      mount = true
      vfs = nfs
      nodename = MyServer
      options = ro,soft,bg
      type = nfs_mount
   ```

   Note: See “Parameters” on page 151 for additional parameters.

2. Remove any directory entries that you do not want to mount automatically at system startup.
3. Save and close the file.
4. Run the **mount -a** command to mount all the directories specified in the /etc/filesystems file.
5. On other clients, repeat these steps.

The NFS directory is now ready to use.

Mounting an NFS directory explicitly

To mount an NFS directory explicitly, use the following procedure:

1. Verify that the NFS server has exported the directory, using:

   ```
   # showmount -e MyServer
   export list for MyServer:
   /home1     (everyone)
   /home2     (everyone)
   /home3     (everyone)
   /home4     (everyone)
   #
   ```

   where MyServer is the name of the NFS server. This command displays the names of the directories currently exported from the NFS server. If the directory you want to mount is not listed, export the directory from the server.

2. Establish the local mount point using the **mkdir** command. For NFS to complete a mount successfully, a directory that acts as the mount point of an NFS mount must be present. This directory should be empty. This mount point can be created like any other directory, and no special attributes are needed for this directory.

3. On the client machine, enter the following SMIT fast path:

   ```
   # smitty mknfsmnt
   ```

4. Make changes to the following fields that are appropriate for your network configuration. Your configuration may not require completing all of the entries on this screen.

   – PATHNAME of mount point.

   – PATHNAME of remote directory.

   – HOST where remote directory resides.

   – MOUNT now, add entry to /etc/filesystems.

   – /etc/filesystems entry will mount the directory on system RESTART.

   – MODE for this NFS.

   Note: If you are using the ASCII SMIT interface, press the Tab key to change to the correct value for each field, but do not press Enter until you get to step 7.

5. Use the default values for the remaining entries or change them depending on your NFS configuration.

6. When you finish making all the changes on this screen, SMIT mounts the NFS.

7. When the Command: field shows the OK status, exit SMIT.

The NFS is now ready to use.

Mounting an NFS directory automatically

AutoFS relies on the use of the **automount** command to propagate the automatic mount configuration information to the AutoFS kernel extension and start the **automountd** daemon. Through this configuration propagation, the extension automatically and transparently mounts file systems whenever a file or a directory within that file system is opened. The extension informs the **automountd** daemon of mount and unmount requests, and the **automountd** daemon actually performs the requested service.

Because the name-to-location binding is dynamic within the **automountd** daemon, updates to a Network Information Service (NIS) map used by the **automountd** daemon are transparent to the user. Also, there is no need to pre-mount shared file systems for applications that have hard-coded references to files and directories, nor is there a need to maintain records of which hosts must be mounted for particular applications.

AutoFS allows file systems to be mounted as needed. With this method of mounting directories, all file systems do not need to be mounted all of the time, only those being used are mounted.

For example, to mount the /backup NFS directory automatically:

1. Verify that the NFS server has exported the directory by entering:

   ```
   # showmount -e MyServer
   export list for MyServer:
   /backup
   #
   ```

   This command displays the names of the directories currently exported from the NFS server.

2. Create an AutoFS map file. AutoFS will mount and unmount the directories specified in this map file. For example, suppose you want to use AutoFS to mount the /backup directory as needed from the Accounts server onto the remote /backup directory. In this example, the map file name is /tmp/mount.map. An example of a map file can be found in /usr/samples/nfs.

3. Ensure that the AutoFS kernel extension is loaded and the **automountd** daemon is running. This can be accomplished in two ways:

   a. Using SRC, enter:

      ```
      # lssrc -s automountd
      ```

If the automountd subsystem is not running, issue **startsrc -s automountd**.

b. Using the **automount** command, issue **/usr/sbin/automount -v**. Define the map file using the command-line interface by entering:

```
# /usr/sbin/automount -v /backup /tmp/mount.map
```

where /backup is the AutoFS mount point on the client. Now, if a user runs the **cd /backup** command, the AutoFS kernel extension will intercept access to the directory and will issue a remote procedure call to the **automountd** daemon, which will mount the /backup directory and then allow the **cd** command to complete.

4. To stop the automountd, issue the **stopsrc -s automountd** command.

   If, for some reason, the **automountd** daemon was started without the use of SRC, issue:

```
# kill automountd_PID
```

   where automountd_PID is the process ID of the **automountd** daemon. (Running the **ps -e** command will display the process ID of the **automountd** daemon.) The **kill** command sends a SIGTERM signal to the **automountd** daemon.

Parameters

The parameters required for stanzas pertaining to NFS mounts are:

dev=*file_system_name* Specifies the path name of the remote file system being mounted.

mount=[true|false] If true, specifies that the NFS will be mounted when the system boots. If false, the NFS will not be mounted when the system boots.

nodename=*hostname* Specifies the host machine on which the remote file system resides.

vfs=nfs Specifies that the virtual file system being mounted is an NFS.

If you do not set the following options, the kernel automatically sets them to the following default values:

- biods=6
- fg
- retry=10000
- rsize=8192
- wsize=8192

- timeo=7
- retrans=3
- port=NFS_PORT
- hard
- secure=off
- acregmin=3
- acregmax=60
- acdirmin=30
- acdirmax=60

5.4.5 Changing an exported file system

This section explains how you can change an exported NFS.

Changing an exported NFS directory using SMIT

The following procedure will guide you through changing an exported file system using SMIT.

1. Un-export the file system on the server by entering:

   `# exportfs -u` */dirname*

   where */dirname* is the name of the file system you want to change. In this case, /home3.

2. On the server, enter:

   ```
   # smitty chnfsexp
   ```

   The resulting screen is shown in Figure 5-5 on page 153.

3. Enter the appropriate path name in the PATHNAME of exported directory field. In this case, /home3.

```
                    Change Attributes of an Exported Directory

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                          [Entry Fields]
* PATHNAME of Directory to Export                       /home3
* MODE to export directory                              read-write            +
  HOSTS & NETGROUPS allowed client access              []
  Anonymous UID                                        [-2]
  HOSTS allowed root access                            []
  HOSTNAME list. If exported read-mostly               []
  Use SECURE OPTION?                                    no                    +
  Public filesystem?                                    no                    +
* CHANGE export now, system restart or both             both                  +
  PATHNAME of alternate Exports file                   []





F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 5-5 Change the Attributes of an Exported Directory

Make whatever changes you need and then press Enter.

4. Exit SMIT.
5. Re-export the file system by entering:

   `# exportfs` */dirname*

   where */dirname* is the name of the file system you just changed, in this case, /home3.

Changing an exported NFS directory using a text editor

The following procedure will guide you through changing an exported file system using a text editor.

1. Unexport the file system by entering:

   `# exportfs -u` */dirname*

   where */dirname* is the name of the file system you want to change, in this case, /home3.
2. Open the /etc/exports file with your favorite text editor.
3. Make whatever changes you want.
4. Save and close the /etc/exports file.

5. Re-export the file system by entering:

   `# exportfs /dirname`

   where */dirname* is the name of the file system you just changed, in this case, /home3.

5.4.6 Un-mounting a mounted file system

To unmount an explicitly or automatically mounted NFS directory, enter **`umount /directory`** or **`unmount /directory`**, for example:

```
# umount /backup
```

The **`rmfs`** command can be used to remove any file systems you created.

5.5 Network configuration

This section discusses various aspects of a TCP/IP network and related network security. Paging space, which is an important part of any system configuration, as well as configurations and attributes of system devices, are also addressed in the following section.

5.5.1 Initializing TCP/IP daemons

At IPL time, the /init process runs /etc/rc.tcpip after starting the SRC. The /etc/rc.tcpip file is a shell script that, when executed, uses the SRC commands to initialize selected daemons. It can also be executed at any time from the command line.

Most of the daemons that can be initialized by the rc.tcpip file are specific to TCP/IP. These daemons are:

- **`inetd`** (started by default)
- **`gated`**
- **`routed`**
- **`named`**
- **`timed`**
- **`rwhod`**

> **Note:** Running the **`gated`** and **`routed`** daemons at the same time on a host may cause unpredictable results.

There are also daemons specific to the base operating system or to other applications that can be started through the rc.tcpip file. These daemons are:

- **lpd**
- **portmap**
- **sendmail**
- **syslogd** (started by default)

5.5.2 Stopping and restarting TCP/IP daemons

The subsystems started from rc.tcpip can be stopped using the **stopsrc** command and restarted using the **startsrc** command.

5.5.3 Stopping TCP/IP daemons using the /etc/tcp.clean script

The script **/etc/tcp.clean** can be used to stop TCP/IP daemons. It will stop the following daemons and remove the /etc/locks/lpd TCP/IP lock files:

- **ndpd-host**
- **lpd**
- **routed**
- **gated**
- **sendmail**
- **inetd**
- **named**
- **timed**
- **rwhod**
- **iptrace**
- **snmpd**
- **rshd**
- **rlogind**
- **telnetd**
- **syslogd**

Note: The script /etc/tcp.clean does not stop the **portmap** and **nfsd** daemons. If you want to stop the **portmap** and the **nfsd** daemons, use the **stopsrc -s portmap** and the **stopsrc -s nfsd** commands.

5.5.4 Restarting TCP/IP daemons

The /etc/rc.tcpip script can be used to restart TCP/IP daemons. Alternatively, you can use the **startsrc -s** command to start individual TCP/IP daemons.

> **Note:** Do not restart TCP/IP daemons using the command:
>
> ```
> #startsrc -g tcpip
> ```
>
> It will start all subsystems defined in the ODM for the tcpip group, which includes both routed and gated.

5.5.5 System boot without starting rc.tcpip

Connections using TCP/IP are often peer-to-peer. There are no master/slave relations. The applications, however, use a client/server model for communications.

Removing the rc.tcpip entry in /etc/inittab means that you are not starting any server applications during IPL.

Without the server applications started, you will not be able to **telnet** or **ftp** to this machine from another host.

However, as long as you have not brought down the network interface, you can still utilize the client network services. You can still **ping** other hosts, and you can still **telnet** or **ftp** to other hosts.

The **ping** command sends an Internet Control Message Protocol (ICMP) ECHO_REQUEST to obtain an ICMP ECHO_RESPONSE from a host and does not need a server application. Therefore, even without starting any server application, the machine will still respond to a **ping** command request from other hosts.

5.5.6 The inetd daemon

The **/usr/sbin/inetd** daemon provides Internet service management for a network. This daemon reduces system load by invoking other daemons only when they are needed and by providing several simple Internet services internally without invoking other daemons.

Starting and refreshing inetd

When the **inetd** daemon starts, it reads its configuration information from the file specified in the ConfigurationFile parameter for the **/usr/sbin/inetd** daemon. If the parameter is not specified, the **inetd** daemon reads its configuration information from the /etc/inetd.conf file. Once started, the **inetd** daemon listens for connections on certain Internet sockets in the /etc/inetd.conf and either handles the service request itself or invokes the appropriate server once a request on one of these sockets is received.

The /etc/inetd.conf file can be updated by using the System Management Interface Tool (SMIT), the System Resource Controller (SRC), or by editing the /etc/inetd.conf.

If you change the /etc/inetd.conf using SMIT, then the **inetd** daemon will be refreshed automatically and will read the new /etc/inetd.conf file. If you change the file using an editor, run the **refresh -s inetd** or **kill -1 InetdPID** commands to inform the **inetd** daemon of the changes to its configuration file. You will not receive a message if you use the **kill -1** command, as shown in Figure 5-6.

```
# refresh -s inetd
0513-095 The request for subsystem refresh was completed successfully.
# ps -ef |grep inetd
    root 17840  2900   0 09:17:31      -  0:00 /usr/sbin/inetd
    root 20606 20016   1 09:19:14  pts/2  0:00 grep inetd
# kill -1 17840
# ps -ef |grep inetd
    root 17482 20016   2 09:19:37  pts/2  0:00 grep inetd
    root 17840  2900   0 09:17:31      -  0:00 /usr/sbin/inetd
# █
```

Figure 5-6 Refreshing the inetd daemon using refresh or kill

Subservers controlled by inetd

The **inetd** daemon is a subsystem that controls the following daemons (subservers):

- **comsat**
- **ftpd**
- **fingerd**
- **rlogind**
- **rexecd**
- **rshd**
- **talkd**

- **telnetd**
- **tftpd**
- **uucpd**

The **ftpd**, **rlogind**, **rexecd**, **rshd**, **talkd**, **telnetd**, and **uucpd** daemons are started by default. The **tftpd**, **fingerd**, and **comsat** daemons are not started by default.

To start any one of them, remove the pound (#) sign in column one of the respective entry in the /etc/inetd.conf file. You can check the details of subservers started in inetd by using the **lssrc -ls** command, as shown in Figure 5-7.

```
# lssrc -ls inetd
Subsystem         Group            PID     Status
 inetd            tcpip            17840   active

Debug          Not active

Signal         Purpose
 SIGALRM       Establishes socket connections for failed services.
 SIGHUP        Rereads the configuration database and reconfigures services.

 SIGCHLD       Restarts the service in case the service ends abnormally.

Service        Command                  Description              Status
 xmquery       /usr/bin/xmservd         xmservd -p3              active
 ttdbserver    /usr/dt/bin/rpc.ttdbserver rpc.ttdbserver 100083 1  active
 cmsd          /usr/dt/bin/rpc.cmsd     cmsd 100068 2-5          active
 dtspc         /usr/dt/bin/dtspcd       /usr/dt/bin/dtspcd       active
 time          internal                                          active
 daytime       internal                                          active
 discard       internal                                          active
 echo          internal                                          active
 time          internal                                          active
 daytime       internal                                          active
 chargen       internal                                          active
 discard       internal                                          active
 pcnfsd        /usr/sbin/rpc.pcnfsd     pcnfsd 150001 1-2        active
 sprayd        /usr/lib/netsvc/spray/rpc.sprayd sprayd 100012 1          active
 rwalld        /usr/lib/netsvc/rwall/rpc.rwalld rwalld 100008 1          active
 rusersd       /usr/lib/netsvc/rusers/rpc.rusersd rusersd 100002 1-2       active
 rstatd        /usr/sbin/rpc.rstatd     rstatd 100001 1-3        active
 ntalk         /usr/sbin/talkd          talkd                    active
 klogin        /usr/sbin/krlogind       krlogind                 active
 login         /usr/sbin/rlogind        rlogind                  active
 kshell        /usr/sbin/krshd          krshd                    active
 shell         /usr/sbin/rshd           rshd                     active
 telnet        /usr/sbin/telnetd        telnetd                  active
 ftp           /usr/sbin/ftpd           ftpd                     active
#
```

Figure 5-7 Subservers started in inetd

The /etc/services file

The /etc/services file contains information about the known services used in the DARPA Internet network by inetd. Each service listed in /etc/services runs on a specific port number for communications, in a specific format, such as TCP or UDP.

Each service is listed on a single line corresponding to the form:

```
ServiceName PortNumber/ProtocolName Aliases
A sample section from /etc/services may look like the following:
echo          7/tcp
echo          7/udp
discard       9/tcp      sink null
discard       9/udp      sink null
daytime      13/tcp
daytime      13/udp
chargen      19/tcp      ttytst source
chargen      19/udp      ttytst source
ftp          21/tcp
time         37/tcp      timeserver
time         37/udp      timeserver
```

If you edit the /etc/services file, run the **refresh -s inetd** command, in order for your changes to be used.

Stopping inetd

Use the **stopsrc -s inetd** command to stop the **inetd** daemon, as shown in Figure 5-8.

```
# stopsrc -s inetd
0513-044 The stop of the /usr/sbin/inetd Subsystem was completed successfully.
# █
```

Figure 5-8 Stopping inetd

When the **inetd** daemon is stopped, the previously started subserver processes are not affected. However, new service requests for the subservers can no longer be satisfied. If you try to **telnet** or **ftp** to the server with inetd down, you will see the messages shown in Figure 5-9.

```
$ telnet sv1166f
Trying...
telnet: connect: A remote host refused an attempted connect operation.
$ ftp sv1166f
ftp: connect: A remote host refused an attempted connect operation.
ftp> bye
$ █
```

Figure 5-9 Telnet and FTP when inetd on sv1166f is down

In other words, existing sessions are not affected when the **inetd** daemon is stopped, but no new **telnet** and **ftp** sessions can be established without first restarting the **inetd** daemon.

5.5.7 The portmap daemon

The **portmap** daemon converts remote procedure call (RPC) program numbers into Internet port numbers.

When an RPC server starts up, it registers with the **portmap** daemon. The server tells the daemon which port number it is listening to and which RPC program numbers it serves. Thus, the **portmap** daemon knows the location of every registered port on the host and which programs are available on each of these ports.

A client consults the **portmap** daemon only once for each program the client tries to call. The **portmap** daemon tells the client which port to send the call to. The client stores this information for future reference.

Since standard RPC servers are normally started by the **inetd** daemon, the **portmap** daemon must be started before the **inetd** daemon is invoked.

> **Note:** If the **portmap** daemon is stopped or comes to an abnormal end, all RPC servers on the host must be restarted.
>
> The nfsd is a common RPC server.

5.5.8 Internet addressing

If you want your machines to communicate with each other across a TCP/IP network, you must give them unique IP addresses. Each host is assigned a unique 32-bit logical address (in the case of IPv4) that is divided into two main parts: the network number and the host number. The network number identifies a logical network to which the host belongs and must be the same across the subnet. The host number identifies a host on the specific logical network.

IP address format

The IP address is the 32-bit address, grouped eight bits at a time, separated by dots and represented in decimal format called dotted decimal notation. Each bit in the octet has a binary weight (128, 64, 32,16, 8, 4, 2, or 1). The minimum value for an octet is 0, and the maximum value for an octet is 255.

Internet address classes

IP addressing supports five different address classes: A, B, C, D, and E. Classes A, B, and C are available for commercial use. You can determine the network class of an IP address by checking the bits in the first octet of a network address.

To determine the class of an IP address, refer to Table 5-2.

Table 5-2 IP address classes

IP address class	Format	First octet	Address range	Number bits network / host	Number of hosts
A	N.H.H.H	0	1.0.0.0 127.0.0.0	7 / 24	2^{24} - 2
B	N.N.H.H	10	128.1.0.0 191.254.0.0	14 / 16	2^{16} - 2
C	N.N.N.H	110	192.0.1.0 223.255.254.0	22 / 8	2^{8} - 2
D	-	1110	224.0.0.0 239.255.255.255	-	-
N - Network number H - Host number					

For example, in the IP address 195.116.119.2, the first octet is 195. Because 195 falls between 192 and 223, 195.116.119.2 is a class C address.

Class A, B, and C addresses also provide address ranges that are useful to define a private network without InterNIC (Internet Network Information Center) authorization. A private network can have the following address ranges:

Class A	10.0.0.0 to 10.255.255.255
Class B	172.16.0.0 to 172.31.255.255
Class C	192.168.0.0 to 192.168.255.255

Special Internet addresses

There are a few IP addresses that cannot be used as host addresses. Those addresses are used for special occasions.

The loopback interface allows a client and server on the same host to communicate with each other using TCP/IP. The network class A with network address 127 is reserved for the loopback interface lo0. AIX 5L assigns the IP address 127.0.0.1 to this interface and assigns it the name localhost.

To check attributes of any interface, use the **ifconfig** or **lsattr** commands as follows:

```
# ifconfig lo0
lo0:
flags=e08084b<UP,BROADCAST,LOOPBACK,RUNNING,SIMPLEX,MULTICAST,GROUPRT,64BIT>
        inet 127.0.0.1 netmask 0xff000000 broadcast 127.255.255.255
        inet6 ::1/0
# lsattr -El lo0
netaddr   127.0.0.1 Internet Address                       True
state     up        Current Interface Status               True
netmask             Subnet Mask                            True
mtu       16896     Maximum IP Packet Size for This Device True
netaddr6  ::1       N/A                                    True
prefixlen           Subnet Mask                            True
```

The network address is an IP address with all host address bits set to 0. If you have the IP address 195.116.119.2, the network address for this will be 195.116.119.0. This type of address is used in the routing table as the network destination address. An example routing table is shown in the following (0 is omitted in the routing tables):

```
# netstat -nr
Routing tables
Destination      Gateway           Flags   Refs     Use  If   PMTU  Exp  Groups

Route Tree for Protocol Family 2 (Internet):
default          9.3.240.1         UGc       0        0  tr0     -    -
9.3.240/24       9.3.240.58        U        30   130787  tr0     -    -
127/8            127.0.0.1         U        54     1300  lo0     -    -
```

```
195.116.119/24   195.116.119.2     U        0        2  en0     -   -
```

The limited broadcast address is 255.255.255.255 (an address with all host address and network address bits set to 1). This can be used as the destination address for all hosts regardless of their network number. Routers never forward a limited broadcast; it only appears on the local cable.

The directed broadcast address is an IP address, with all the host address bits set to 1. It is used to simultaneously address all hosts within the same network. For example, consider an IP address 195.116.119.2; because it is a class C address, the network address for this address is 195.116.119. Therefore, the directed broadcast for this network will be 195.116.119.255. To check the broadcast setting for interface en0, enter:

```
# ifconfig en0
en0:
flags=e080863<UP,BROADCAST,NOTRAILERS,RUNNING,SIMPLEX,MULTICAST,GROUPRT,64BIT>
inet 195.116.119.2 netmask 0xffffff00 broadcast 195.116.119.255
```

The last column of Table 5-2 on page 161 shows the number of hosts in the appropriate network class. Notice that the two hosts were subtracted. This was done so that one address is reserved for the broadcast address, and one address is reserved for the network address.

Subnetting

Subnet addressing allows an autonomous network made up of multiple systems to share the same Internet address class. The subnetwork capability of TCP/IP also makes it possible to divide a single network into multiple logical networks (subnets). This makes sense for class A and class B addresses, since attaching thousands of hosts to a single network is impossible.

A standard IP address has two fields (see “IP address format” on page 161): a network address and a host address. A subnet address is created by borrowing bits from the host field and designating them as the subnet field. The number of borrowed subnet bits varies and it depends of the chosen subnet mask. Figure 5-10 shows how bits are borrowed from the host address field to create the subnet address field and how the subnet mask works.

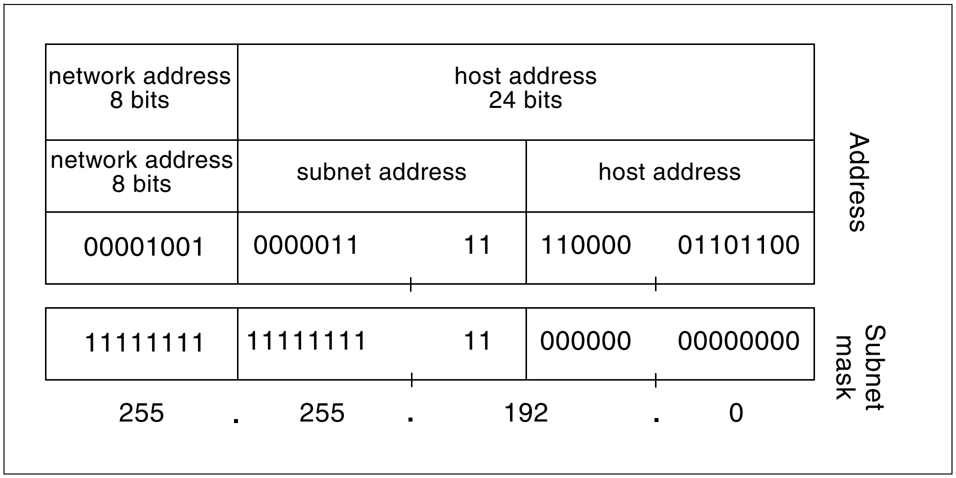


Figure 5-10 Subnetting example

Subnet masks

The subnet mask tells the system what the subnet partitioning scheme is. This bit mask consists of the network address portion and subnet address portion of the IP address.

When a host sends a message to a destination, the system must determine whether the destination is on the same network as the source or if the destination must be reached through a gateway. The system compares the destination address to the host address using the subnet mask. If the destination is not on the local network, the system sends the packet to a gateway. Gateways are systems that handle the routing of information to other systems and networks. They perform the same comparison to see if the destination address is on a network they can reach locally. Gateways (also called a routes) are systems or network devices that will route information onto other systems or networks.

5.5.9 Host name resolution and /etc/netsvc.conf

TCP/IP provides a naming system that supports both flat and hierarchical network organizations so that users can use meaningful, easily remembered names instead of Internet addresses.

In flat TCP/IP networks, each machine in the network has a file (/etc/hosts) containing the name-to-Internet-address mapping information for every host in the network.

When TCP/IP networks become very large, as on the Internet, naming is divided hierarchically. Typically, the divisions follow the network's organization. In TCP/IP, hierarchical naming is known as the domain name system (DNS) and uses the DOMAIN protocol. The DOMAIN protocol is implemented by the **named** daemon in TCP/IP.

The default order in resolving host names is:

1. BIND/DNS (named, using the /etc/resolv.conf file))
2. Network Information Service (NIS)
3. Local /etc/hosts file

When a process receives a symbolic host name and needs to resolve it into an address, it calls a resolver routine. By default, resolver routines attempt to resolve names using the default order.

If all of the services were unavailable, then the resolver routine will return with the message SERVICE_UNAVAILABLE.

The default order can be overwritten by creating the configuration file, /etc/netsvc.conf, and specifying the desired order. The following is a sample /etc/netsvc.conf file:

```
#cat /etc/netsvc.conf
hosts = nis, local, bind
```

In this example, the system will use NIS to attempt to resolve host names first, followed by the /etc/hosts file, and then BIND/DNS last.

Both the default and /etc/netsvc.conf can be overwritten with the environment variable NSORDER. If it is not set, the default will be as though you issued the command:

```
# export NSORDER=bind,nis,local
```

Note: The NSORDER environment variable will override the host name resolution list in /etc/netsvc.conf. If your /etc/netsvc.conf file does not seem to be working properly, you may want to examine the NSORDER environment variable by running the command:

```
# echo $NSORDER

NSORDER=local,bind
```

/etc/hosts

The /etc/hosts file provides a list of server names or aliases and their IP addresses.

The following is a sample /etc/hosts file:

```
# cat /etc/hosts
# The format of this file is:
# Internet Address      Hostname        # Comments
# Items are separated by any number of blanks and/or tabs.  A '#'
# indicates the beginning of a comment; characters up to the end of the
# line are not interpreted by routines which search this file.  Blank
# lines are allowed.

# Internet Address      Hostname        # Comments
127.0.0.1               loopback localhost      # loopback (lo0) name/address
9.3.240.57      sample2.itsc.austin.ibm.com
```

The /etc/resolv.conf file

Domain name servers (DNS) allow IP addresses to be represented symbolically for ease. For example, the system with the IP address of 207.25.253.26 is named service.software.ibm.com.

The /etc/resolv.conf file has two functions. First, it indicates to a system that it should go to a specific domain name server to resolve a name. Second, it defines to the system to what domain it belongs.

For example, the system named *sample* has the following /etc/resolv.conf file:

```
# cat /etc/resolv.conf
nameserver      9.3.1.74
domain  itsc.austin.ibm.com
```

The system *sample* would use the name server 9.3.1.74 for name resolution. Since it belongs to the itsc.austin.ibm.com domain, the full host name of the system is sample.itsc.austin.ibm.com.

If the specified domain name server is unreachable, you will not be able to communicate with systems by host name and will receive errors. The **ping** command can be used to verify if a name server is unreachable.

For example, if the domain name server 9.3.1.74 is not available, it takes some time before an error message from the **ping** command is returned:

```
# ping 9.3.1.121
PING 9.3.1.121: (9.3.1.121): 56 data bytes
^C
----9.3.1.121 PING Statistics----
```

```
3 packets transmitted, 0 packets received, 100% packet loss
# date +%H:%M:%S
16:20:10
# ping sv1166f
0821-062 ping: host name sv1166f NOT FOUND
# date +%H:%M:%S
16:20:19
```

If the name server is down, however, you will still be able to communicate with other servers by IP address.

> **Note:** If your name server has a host name assigned to it, do not enter this name into the nameserver field. Enter in the IP address of the name server; otherwise, your system will not be able to automatically communicate with the name server.

The /etc/resolv.conf file defines Domain Name Protocol (DOMAIN) name-server information for local resolver routines. If the /etc/resolv.conf file does not exist, then BIND/DNS is considered to be not set up or running and, therefore, not available. The system will attempt name resolution using the default paths, the /etc/netsvc.conf file, or the NSORDER environment variable.

Related problems with /etc/resolv.conf

When you have problems resolving a host name, and you are using a name server, you should:

1. Verify that you have a /etc/resolv.conf file specifying the correct domain name and Internet address of a name server. If you try to access a host by name with an incorrect entry in /etc/resolv.conf, and if the host is also not defined in /etc/hosts, you will get an error message, as shown in the following example:

   ```
   # ping olympus
   0821-062 ping: host name olympus NOT FOUND
   ```

2. If /etc/resolv.conf contains the correct data, verify that the host acting as the local name server is up by issuing the **ping** command with the IP address of the name server found in the /etc/resolv.conf file.
3. If the local name server is up, verify that the **named** daemon on that local system is active by issuing the **lssrc -s named** command on that host.
4. If you are running the **syslogd** daemon, there could be error messages logged. The output for these messages is defined in the /etc/syslog.conf file.

5.5.10 Adding network routes

For those systems that need to access a private network, use the SMIT fast path **smit mkroute** to add a route to the private network through the gateway between two networks. A sample of **smit mkroute** is shown in Figure 5-11.

```
                              Add Static Route

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                      [Entry Fields]
  Destination TYPE                                   net                    +
* DESTINATION Address                               [192.168.1]
  (dotted decimal or symbolic name)
* Default GATEWAY Address                           [9.3.1.124]
  (dotted decimal or symbolic name)
* METRIC (number of hops to destination gateway)    [1]                     #
  Network MASK (hexadecimal or dotted decimal)      [255.255.255.0]









F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 5-11 Adding a Static Route menu

Instead of using SMIT, you can also use the command:

```
route add -net 192.168.1 -netmask 255.255.255.0 9.3.1.124
```

The procedure, shown in Figure 5-12 on page 169, shows that:

- A host cannot access the IP addresses 192.168.1.1 and 192.168.1.2.
- A route is added using the **route add** command specifying that 9.3.1.124 should be used as the gateway to the network 192.168.1.
- The **traceroute** command shows the route taken to reach both 192.168.1.1 and 192.168.1.2.

```
# ping 192.168.1.1
PING 192.168.1.1: (192.168.1.1): 56 data bytes
^C
----192.168.1.1 PING Statistics----
2 packets transmitted, 0 packets received, 100% packet loss
# ping 192.168.1.2
PING 192.168.1.2: (192.168.1.2): 56 data bytes
^C
----192.168.1.2 PING Statistics----
2 packets transmitted, 0 packets received, 100% packet loss
# ping 9.3.1.124
PING 9.3.1.124: (9.3.1.124): 56 data bytes
64 bytes from 9.3.1.124: icmp_seq=0 ttl=255 time=1 ms
64 bytes from 9.3.1.124: icmp_seq=1 ttl=255 time=1 ms
^C
----9.3.1.124 PING Statistics----
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 1/1/1 ms
# route add -net 192.168.1 -netmask 255.255.255.0 9.3.1.124
9.3.1.124 net 192.168.1: gateway 9.3.1.124
# traceroute 192.168.1.2
trying to get source for 192.168.1.2
source should be 9.3.1.33
traceroute to 192.168.1.2 (192.168.1.2) from 9.3.1.33 (9.3.1.33), 30 hops max
outgoing MTU = 1492
 1  192.168.1.2 (192.168.1.2)  13 ms  2 ms  2 ms
# traceroute 192.168.1.1
trying to get source for 192.168.1.1
source should be 9.3.1.33
traceroute to 192.168.1.1 (192.168.1.1) from 9.3.1.33 (9.3.1.33), 30 hops max
outgoing MTU = 1492
 1  sv1166f.itsc.austin.ibm.com (9.3.1.124)  13 ms  2 ms  2 ms
 2  192.168.1.1 (192.168.1.1)  5 ms  4 ms  3 ms
#
```

Figure 5-12 Adding a route using the route add command

5.5.11 Changing IP addresses using SMIT

If you are moving your machine from one network segment to another, and need to change IP addresses, use **smit mktcpip** the same way as the first time you configured TCP/IP. You may need to change the host name, IP address, and the default gateway address. A sample screen is shown in Figure 5-13.

```
                        Minimum Configuration & Startup

 To Delete existing configuration data, please use Further Configuration menus

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                     [Entry Fields]
* HOSTNAME                                          [sv1050e]
* Internet ADDRESS (dotted decimal)                 [9.3.1.96]
  Network MASK (dotted decimal)                     [255.255.255.0]
* Network INTERFACE                                  tr0
  NAMESERVER
           Internet ADDRESS (dotted decimal)        [9.3.1.74]
           DOMAIN Name                              [itsc.austin.ibm.com]
  Default GATEWAY Address                           [9.3.1.74]
  (dotted decimal or symbolic name)
  RING Speed                                        [autosense]          +
  START Now                                          no                  +


F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 5-13 Minimum Configuration & Startup menu

> **Note:** Do not perform this task in a **Telnet** command session, as you will lose your connection when the change is made.

If you are not moving across network segments, and simply want to change the IP address, you can change the field START Now shown in Figure 5-13 to yes. This will start the TCP/IP daemons automatically or refresh them if they are already started.

5.5.12 The ifconfig command

The **ifconfig** command can allow you to configure and modify properties of network interfaces directly, without the use of SMIT. Often, administrators find this easier than using the SMIT panels for network administration.

The syntax of the **ifconfig** command for configuring and modifying network interfaces is as follows:

`ifconfig` *Interface* `[` *AddressFamily* `[` *Address* `[` *DestinationAddress* `] ]` `[`*Parameters*`... ] ]`

There are three address families that can be used with the **ifconfig** command:

inet The default dotted decimal notation for a system that is part of the DARPA-Internet. This is the address family that **ifconfig** uses by default.

inet6 The default dotted decimal notation for a system that is part of the DARPA-Internet running IPv6.

ns The default dotted hexadecimal notation for a system that is part of a Xerox Network Systems family.

Table 5-3 is a list of common command parameters and their functions for the **ifconfig** command.

Table 5-3 Commonly used parameters for the ifconfig command

Parameter	Description
alias	Establishes an additional network address for the interface.
delete	Removes the specified network address from the interface.
detach	Removes an interface from the network interface list.
down	Marks an interface as inactive (down), which keeps the system from trying to transmit messages through that interface.
mtu *Value*	Sets the maximum IP packet size to *Value* bytes, (maximum transmission unit), ranging from 60 to 65535.
netmask *Mask*	Specifies how much of the address to reserve for subdividing networks into subnetworks.
up	Marks an interface as active (up).

Identifying network interfaces

Before you use the **ifconfig** command to perform administration on network interfaces, it is helpful to identify all interfaces on your server. There are two ways to identify network interfaces on your server. The first command that you can run is:

```
# lsdev -Cc if
```

This will produce a simple list of all interfaces on the system, whether they are being actively used by the system or not. For example:

```
# lsdev -Cc if
en0 Defined   10-80 Standard Ethernet Network Interface
en1 Defined   20-60 Standard Ethernet Network Interface
et0 Defined   10-80 IEEE 802.3 Ethernet Network Interface
et1 Defined   20-60 IEEE 802.3 Ethernet Network Interface
lo0 Available       Loopback Network Interface
tr0 Available 10-68 Token Ring Network Interface
```

The second command that you can run is:

```
# ifconfig -a
```

This will produce a list of all network interfaces on the system that have IP addresses assigned and are actively being used by the system. For example:

```
# ifconfig -a
tr0: flags=e0a0043<UP,BROADCAST,RUNNING,ALLCAST,MULTICAST,GROUPRT,64BIT>
        inet 10.1.2.2 netmask 0xffffff00 broadcast 10.1.2.255
lo0:
flags=e08084b<UP,BROADCAST,LOOPBACK,RUNNING,SIMPLEX,MULTICAST,GROUPRT,64BIT
>
        inet 127.0.0.1 netmask 0xff000000 broadcast 127.255.255.255
        inet6 ::1/0
```

To get information about one specific network interface, including state, IP address, and netmask, run the command:

```
# ifconfig Interface
```

To get information about tr0, for example, run the command:

```
# ifconfig tr0
tr0: flags=e0a0043<UP,BROADCAST,RUNNING,ALLCAST,MULTICAST,GROUPRT,64BIT>
        inet 10.1.2.2 netmask 0xffffff00 broadcast 10.1.2.255
```

Activating a network interface

Before messages can be transmitted through a network interface, the interface must be placed in the up or active state. To activate an interface using **ifconfig**, run the command:

```
# ifconfig Interface [Address] [netmask Netmask] up
```

To activate a network interface using **ifconfig**, such as tr0, run the command:

```
# ifconfig tr0 up
```

To activate a network interface, such as the loopback interface (lo0) and assign it an IP address, run the command:

```
# ifconfig lo0 127.0.0.1 up
```

To activate a network interface, such as a token ring interface (tr0), and assign it an IP address and netmask, run the command:

```
# ifconfig tr0 10.1.2.3 netmask 255.255.255.0 up
```

Deactivating a network interface

To stop messages from being transmitted through an interface, the interface must be placed in the down or inactive state. To deactivate an interface using **ifconfig**, run the command:

```
# ifconfig Interface down
```

For example, to deactivate the network interface tr0, run the command:

```
# ifconfig tr0 down
```

> **Note:** This command does not remove any IP addresses assigned to the interface from the system or remove the interface from the network interface list.

Deleting an address from a network interface

To remove a network address from an interface, the address must be deleted from the interface definition. To delete a network address from an interface using **ifconfig**, run the command:

```
# ifconfig Interface [Address] [netmask Netmask] delete
```

For example, to delete the network address from tr0, run the command:

```
# ifconfig tr0 delete
```

> **Note:** This command does not place the interface in the down state, nor does it remove the interface from the network interface list.

Detaching a network interface

To remove an interface from the network interface list, the interface must be detached from the system. This command can be used when a network interface card has physically been removed from a system or when an interface no longer needs to be defined within the system. To detach a network interface from the system using **ifconfig,** run the command:

```
# ifconfig Interface detach
```

For example, to remove the interface tr0 from the network interface list, run the command:

```
# ifconfig tr0 detach
```

Note: This command removes all network addresses assigned to the interface and removes the interface from the output of the **ifconfig -a** command. To add an interface back to the system, or to add a new interface to the network interface list, run the command:

```
# ifconfig Interface
```

where *Interface* is the network interface you want to add.

Creating an IP alias for a network interface

Through the **ifconfig** command, you can bind multiple network addresses to a single network interface by defining an alias. This is a useful tool for such activities as providing two different initial home pages through a Web server application. To bind an alias to a network interface, run the command:

```
# ifconfig Interface Address [netmask Netmask] alias
```

For example, to bind the IP address of 10.1.2.3 to tr0 with a netmask of 255.255.255.0, run the command:

```
# ifconfig tr0 10.1.2.3 netmask 255.255.255.0 alias
```

Note: There will be no ODM record created of the alias by this command. You will need to invoke the same command every time you reboot your system to preserve the alias. If your system configuration has a local startup script defined in the /etc/inittab file, this command should be included in that local startup script.

When this alias is no longer required, you can remove it using the command:

```
# ifconfig tr0 10.1.2.3 netmask 255.255.255.0 delete
```

Note: If you do not specify which alias is to be removed from a network interface, the system will default and remove the primary network address from the interface. After this occurs, the first alias in the list of network addresses for the interface will become the primary network address for the interface. To remove all aliases from an interface, you must delete each alias individually.

Changing the MTU size of a network interface

When messages are transmitted through a network interface, they travel in bundles of information called packets. These packets can vary in length from 60 bytes to 65535 bytes per packet. By default, a 16 Mb token-ring interface will transmit packets that are 1492 bytes long, and Ethernet interfaces will transmit packets that are 1500 bytes long. For AIX 5L systems, these packets are governed by the maximum transmission unit (MTU) size variable.

> **Note:** The minimum and maximum MTU sizes for specific interfaces may vary.

The MTU size is critical for proper network communications. Packets that are too small in length may be lost during transmission. Packets that are too long in length may collide with other packets that are being transmitted. These factors can lead to slower transmission rates and other network problems as packets must then be retransmitted.

To determine the MTU size for a network interface, run the command:

`# lsattr -El` *Interface*

The output will look similar to the following:

```
# lsattr -El en0
alias4                         IPv4 Alias including Subnet Mask           True
alias6                         IPv6 Alias including Prefix Length         True
arp           on               Address Resolution Protocol (ARP)          True
authority                      Authorized Users                           True
broadcast                      Broadcast Address                          True
mtu           1500             Maximum IP Packet Size for This Device     True
netaddr       9.3.5.195        Internet Address                           True
netaddr6                       IPv6 Internet Address                      True
netmask       255.255.255.0    Subnet Mask                                True
prefixlen                      Prefix Length for IPv6 Internet Address    True
remmtu        576              Maximum IP Packet Size for REMOTE Networks True
rfc1323                        Enable/Disable TCP RFC 1323 Window Scaling True
security      none             Security Level                             True
state         up               Current Interface Status                   True
tcp_mssdflt                    Set TCP Maximum Segment Size               True
tcp_nodelay                    Enable/Disable TCP_NODELAY Option          True
tcp_recvspace                  Set Socket Buffer Space for Receiving      True
tcp_sendspace                  Set Socket Buffer Space for Sending        True
#
```

The **ifconfig** command can adjust the MTU size for a network interface. To change the MTU size, run the command:

`# ifconfig` *Interface* `mtu` *Value*

For example, to change the MTU size of tr1 to 2000 bytes in length, run the command:

```
# ifconfig en0 mtu 2000
```

> **Note:** The MTU size cannot be changed while the interface is in use. All systems that are on the same local area network (LAN) must have the same MTU size, so all systems must change MTU size simultaneously to prevent problems.

5.6 The ntp.conf file

The ntp.conf file controls how the Network Time Protocol (NTP) daemon **xntpd** operates and behaves.

In the ntp.conf file, comments begin with a # character and extend to the end of the line. Blank lines are ignored. Options consist of an initial keyword followed by a list of arguments, which may be optional, separated by white space. These options may not be continued over multiple lines. Arguments may be host names, host addresses written in numeric (dotted decimal) form, integers, floating point numbers (when specifying times in seconds), and text strings.

A detailed description of the available options and their functionality can be found in the *AIX 5L Version 5.3 Files Reference, System File* that is part of the AIX 5L Version 5.3 product documentation.

5.7 Network security

Network security is a prevalent issue for system administrators. There is a great need for secure connections, trusted networks, and other ways of communications that do not allow for unauthorized system access. This section briefly describes some of the more common ways you can prevent unauthorized access to your systems over the networks.

5.7.1 Trusted and non-trusted processes

A trusted program, or trusted process, is a shell script, a daemon, or a program that meets a particular standard of security. These security standards are set and maintained by the U.S. Department of Defense, which also certifies some trusted programs.

TCP/IP contains several trusted daemons and many non-trusted daemons. The trusted daemons have been tested to ensure that they operate within particular security standards, such as granting users a particular level of access and only permitting users to perform certain tasks.

Examples of trusted daemons are:

- **ftpd**
- **rexecd**
- **telnetd**

The trusted types of daemons require verification and authentication of the user wishing to communicate with the server. Typically, this is done through the use of a login and password.

Examples of non-trusted daemons are:

- **rshd**
- **rlogind**
- **tftpd**

The non-trusted types of daemons do not always require verification or authentication of the user wishing to communicate with the server. A login and password is not necessarily required for the use of these types of daemons. Caution should be used in enabling these processes to run on your system.

5.7.2 Network configuration files

There are many different configuration files used for network support. This section briefly describes some of the more important files you will encounter and modify while performing network support.

Note: All files in this section should have permissions of 600 and should be owned by root whenever possible. These files often contain sensitive information, and unauthorized modification of these files can lead to security exposures and risks. Most of these files will not function properly unless their permissions are set to 600.

$HOME/.netrc

The $HOME/.netrc file contains information used by the automatic login feature of the **rexec** and **ftp** commands. It is a hidden file in a user's home directory and must be owned either by the user executing the command or by the root user. If the .netrc file contains a login password, the file's permissions must be set to 600 (read and write by owner only). The login password is in plain text. Even with permissions set to 600, passwords for remote systems are vulnerable to being revealed to any user with root authority.

Entries in the $HOME/.netrc file are stored in the following format (separated by spaces, tabs, or new lines):

machine *HostName* The *HostName* variable is the name of a remote host. This entry begins the definition of the automatic login process for the specified host. All the following entries up to the next machine entry or the end of the file apply to that host.

login *UserName* The *UserName* variable is the full domain user name for use at the remote host. If this entry is found, the automatic login process initiates a login using the specified name. If this entry is missing, the automatic login process is unsuccessful.

password *Password* The *Password* variable is the login password to be used. The automatic login process supplies this password to the remote server. A login password must be established at the remote host, and that password must be entered in the .netrc file. Otherwise, the automatic login process is unsuccessful, and the user is prompted for the login password.

account *Password* The *Password* variable is the account password to be used. If this entry is found, and an account password is required at the remote host, the automatic login process supplies the password to the remote server. If the remote host requires an account password, but this entry is missing, the automatic login process prompts for the account password.

macdef *MacroName* The *MacroName* variable is the name of an FTP subcommand macro. The macro is defined to contain all of the following FTP subcommands up to the next blank line or the end of the file. If the macro is named init, the **ftp** command executes the macro upon successful completion of the automatic login process. The **rexec** command does not recognize a macdef entry.

A sample $HOME/.netrc file is shown Figure 5-14.

```
$ cat .netrc
machine service.software.ibm.com login anonymous password pw0rd@ macdef init
bin
lcd /ptf/
site exec lfixdist "devices.buc.00004001.rte.4.3.1.1:0:0:202615808:125:IBM:fixdi
stm:0:usrname@hostname"
get /aix/fixes/v4/os/bos.64bit.4.3.1.4.bff bos.64bit.4.3.1.4.bff
get /aix/fixes/v4/os/bos.64bit.4.3.1.4.info bos.64bit.4.3.1.4.info
quit

$ █
```

Figure 5-14 A sample .netrc file

$HOME/.forward

When mail is sent to a local user, the **sendmail** command checks for the $HOME/.forward file for that user. The $HOME/.forward file can contain one or more addresses or aliases. Any messages are then sent to the addresses or aliases in the $HOME/.forward file.

/etc/hosts.equiv and $HOME/.rhosts

The /etc/hosts.equiv file, along with any local $HOME/.rhosts files, defines the hosts (computers on a network) and user accounts that can invoke remote commands on a local host without supplying a password. A user or host that is not required to supply a password is considered trusted, though the daemons that initiate the connections may be non-trusted in nature (for example, rlogind).

When a local host receives a remote command request, the appropriate local daemon first checks the /etc/hosts.equiv file to determine if the request originates with a trusted user or host. For example, if the local host receives a remote login request, the **rlogind** daemon checks for the existence of a hosts.equiv file on the local host. If the file exists, but does not define the host or user, the system checks the appropriate $HOME/.rhosts file. This file is similar to the /etc/hosts.equiv file, except that it is maintained for individual users.

Note: If a remote command request is made by the root user, the /etc/hosts.equiv file is ignored and only the /.rhosts file is read.

Both files, /etc/hosts.equiv and $HOME/.rhosts, must have permissions denying write access to group and other (600). If either group or other have write access to a file, that file will be ignored. Do not give write permission to the /etc/hosts.equiv file to group and others, as this can lead to security vulnerabilities and undesired user access to the local host.

The format of the /etc/hosts.equiv and $HOME/.rhosts files is as follows:

`[ + | - ] HostName [ + | - ]` *UserName* `or` *@NetGroup*

> **Note:** Both /etc/hosts.equiv and $HOME/.rhosts are read top to bottom, so the order of placing entries into these files can change the desired results. The deny, or - (minus sign), statements must precede the accept, or + (plus sign), statements in the lists.

For example, to allow all the users on the hosts toaster and machine to log in to the local host, you would enter:

```
toaster
machine
```

To only allow the user bob to log in from the host machine, you would enter:

```
toaster
machine bob
```

To allow the user lester to log in from any host, you would enter:

```
toaster
machine bob
+ lester
```

To allow all users from the host tron to log in, while requesting users joel and mike for a password to log in, you would enter:

```
toaster
machine bob
+ lester
tron -joel
tron -mike
tron
```

To deny all members of the forum netgroup from logging in automatically, you would enter:

```
toaster
machine bob
+ lester
tron -joel
tron -mike
tron
- @forum
```

Note: Netgroups is a feature of NIS that allows an administrator to easily classify hosts and users into common groups. The use of netgroups is outside the scope of this book. More information about netgroups can be found in the *AIX 5L Version 5.3 System Management Guide: Communications and Networks* that is part of the AIX 5L Version 5.3 product documentation.

5.8 Operations on a network adapter

The following sections discuss configuration methods on a network adapter.

5.8.1 Adding a network adapter

To add a network adapter to the system, perform the following steps:

Note: You should perform this procedure during a system maintenance window, as this procedure will require the shutdown of the system and may interfere with the work of users on the system.

1. Examine what network adapters and interfaces are already on the system by running the following commands:

```
# lscfg |grep -i adapter
+ mg20              20-58             GXT130P Graphics Adapter
+ fda0              01-D1             Standard I/O Diskette Adapter
* siokma0           01-K1             Keyboard/Mouse Adapter
+ sioka0            01-K1-00          Keyboard Adapter
+ sioma0            01-K1-01          Mouse Adapter
+ ppa0              01-R1             Standard I/O Parallel Port Adapter
+ ent0              10-80             IBM PCI Ethernet Adapter (22100020)

# lsdev -Cc if
en0 Defined   10-80 Standard Ethernet Network Interface
et0 Defined   10-80 IEEE 802.3 Ethernet Network Interface
lo0 Available       Loopback Network Interface
```

2. Shut down and power off the system (for systems without hot plug cards).
3. Physically install the new network adapter.
4. Power on the system in normal mode.
5. When the system is fully up, run the **cfgmgr** command. This will automatically detect the network adapter and add network interfaces for the adapter.

Note: After running **cfgmgr**, you may receive an error message as in the following example:

```
# cfgmgr
cfgmgr: 0514-621 WARNING: The following device packages are required for
      device support but are not currently installed.
devices.pci.ethernet:devices.pci.14100401:devices.pci.ae120100:devices.pc
i.pciclass.020000
```

If a message such as this appears, you will need to install the additional filesets listed to enable the network adapter on the system.

6. Run the following commands to confirm the network adapter has been properly added to the system:

```
# lscfg |grep -i adapter
+ mg20              20-58             GXT130P Graphics Adapter
+ ent1              20-60             Gigabit Ethernet-SX PCI Adapter
+ fda0              01-D1             Standard I/O Diskette Adapter
* siokma0           01-K1             Keyboard/Mouse Adapter
+ sioka0            01-K1-00          Keyboard Adapter
+ sioma0            01-K1-01          Mouse Adapter
+ ppa0              01-R1             Standard I/O Parallel Port Adapter
+ ent0              10-80             IBM PCI Ethernet Adapter (22100020)
# lsdev -Cc if
en0 Defined   10-80 Standard Ethernet Network Interface
en1 Defined   20-60 Standard Ethernet Network Interface
et0 Defined   10-80 IEEE 802.3 Ethernet Network Interface
et1 Defined   20-60 IEEE 802.3 Ethernet Network Interface
lo0 Available       Loopback Network Interface
tr0 Available 10-68 Token Ring Network Interface
```

5.8.2 Removing a network adapter

To remove a network adapter from the system, use the **ifconfig** command to remove the network adapter and interface definitions from the system prior to physically removing the network adapter. The **ifconfig** command allows you to perform configurations and modifications directly to network interfaces.

To remove a network adapter, perform the following steps:

> **Note:** Before performing this procedure, make sure that no applications or processes still use the network adapter or network interfaces associated with the network adapter you intend to remove. This can potentially cause system problems if the applications and processes on the system attempt to use network adapters or interfaces that are no longer present on the system.
>
> You should perform this procedure during a system maintenance window, as this procedure may require the shutdown of the system and may interfere with the work of users on the system.

1. Deactivate (down) all network interface definitions for the network adapter by running the command:

   ```
   # ifconfig Interface down
   ```

   This step is technically not required, but is a good fail-safe to ensure that no applications or processes are using the specified network interfaces.

2. Remove (detach) all network interface definitions from the network interface list by running the command:

   ```
   # ifconfig Interface detach
   ```

   This step will remove all attributes associated with the network interface from the system, including attributes like IP address and MTU size.

3. Delete the network interface definitions from the system by running the command:

   ```
   # rmdev -l Interface -d
   ```

4. Delete the network adapter definition from the system by running the command:

   ```
   # rmdev -l Adapter -d
   ```

5. Shut down and power off the system (on non hot plug machines).
6. Physically remove the network adapter.
7. Power on the system.

> **Note:** If you do not physically remove the network adapter before powering on the system, the **cfgmgr** program will detect the network adapter during the system boot, unless it is an ISA adapter, and redefine the device in the Customized Devices object database.

5.8.3 Adapter and interface configuration problems

Network adapters that are properly added to your system may have difficulty communicating to other systems or networks, although all the hardware appears to be functioning correctly. These problems can be caused by incorrect adapter and interface configuration.

Media speed configuration problems

Ethernet adapters communicate at a certain rate of data throughput called media speed. For most Ethernet adapters, communication is done at either a 10 or 100 Mbps ring speed in one of two modes: full duplex or half duplex.

An incorrect media speed will prevent the system from communicating with other systems or networks. Some symptoms of a problem due to incorrect media speed may include:

- Connection timeouts (**telnet**)
- No packet transmission or response (**ping**)
- Unusual pauses and hangs when initializing communication daemons (inetd)

To correct media speed problems, perform the following steps:

1. Obtain the proper media speed from your network administrator.
2. Remove (detach) all network interfaces for the network adapter by running the command:

   ```
   # ifconfig Interface detach
   ```

   > **Note:** Removing all network interfaces by using the detach parameter will remove all configuration for the network interfaces, including IP addresses and MTU size definitions. You should keep a copy of all necessary configuration for the network interfaces for reconfiguration after the ring or media speed is changed.

3. To change the media speed for an Ethernet adapter, run the command **smitty chgenet**. You will see a submenu similar to the one shown in Figure 5-15 on page 185.

```
                          Ethernet Adapter

Move cursor to desired item and press Enter. Use arrow keys to scroll.

  ent1 Available 08-09 2-Port 10/100/1000 Base-TX PCI-X Adapter (1410890
  ent0 Available 08-08 2-Port 10/100/1000 Base-TX PCI-X Adapter (1410890
  ent2 Available 0A-08 10/100/1000 Base-TX PCI-X Adapter (14106902)

F1=Help                 F2=Refresh              F3=Cancel
F8=Image                F10=Exit                Enter=Do
/=Find                  n=Find Next
```

Figure 5-15 Ethernet Adapter submenu

Use the arrow keys to select the adapter you want, and press Enter.

4. You will see a menu similar to the one shown in Figure 5-16.

```
              Change / Show Characteristics of an Ethernet Adapter

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[TOP]                                                    [Entry Fields]
  Ethernet Adapter                                     ent1
  Description                                          2-Port 10/100/1000 Ba>
  Status                                               Available
  Location                                             08-09
  Rcv descriptor queue size                           [1024]                +#
  TX descriptor queue size                            [512]                 +#
  Software transmit queue size                        [8192]                +#
  Transmit jumbo frames                                no                   +
  Enable hardware TX TCP resegmentation                yes                  +
  Enable hardware transmit and receive checksum        yes                  +
  Media speed                                          Auto_Negotiation     +
  Enable ALTERNATE ETHERNET address                    no                   +
  ALTERNATE ETHERNET address                          [0x000000000000]      +
[MORE...2]

F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 5-16 Change/Show Characteristics of an Ethernet Adapter menu

Use the arrow keys to select the Media Speed field, and press F4.

5. A submenu will pop-up, providing the ring speed options, similar to the one shown in Figure 5-17.

```
               Change / Show Characteristics of an Ethernet Adapter

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                     [Entry Fields]
  Ethernet Adapter                                  ent1
  Description                                       IBM 10/100 Mbps Ethern>
  Status                                            Available
   +--------------------------------------------------------------------------+
   |                               Media Speed                                |+#
   |                                                                          |+#
   | Move cursor to desired item and press Enter.                             |+#
   |                                                                          |+
   |   10_Half_Duplex                                                         |+#
   |   10_Full_Duplex                                                         |+
   |   100_Half_Duplex                                                        |+
   |   100_Full_Duplex                                                        |+
   |   Auto_Negotiation                                                       |
   |                                                                          |
   | F1=Help                 F2=Refresh              F3=Cancel                |
F1 | F8=Image                F10=Exit                Enter=Do                 |
F5 | /=Find                  n=Find Next                                      |
F9 +--------------------------------------------------------------------------+
```

Figure 5-17 Media Speed submenu

Select the media speed you want, and press Enter. Press Enter again to change the media speed for the Ethernet adapter.

Cable type configuration problems

Ethernet adapters can use several different types of cable connections, such as bnc, dix, or tp. If the cable type is set incorrectly, the system may not be able to communicate properly.

To set the cable type, run **`smitty tcpip`** and, in the Minimum Configuration & Startup screen, select the Ethernet interface you want to change, as shown in Figure 5-18 on page 187.

```
                            Network Interface Selection

Move cursor to desired item and press Enter.

  List All Network Interfaces
  Add a Network Interface
  Change / Show Characteristics of a Network Interface
  Remove a Network Interface

                         Available Network Interfaces

     Move cursor to desired item and press Enter.

       en0    08-08    Standard Ethernet Network Interface
       en1    08-09    Standard Ethernet Network Interface
       en2    0A-08    Standard Ethernet Network Interface
       et0    08-08    IEEE 802.3 Ethernet Network Interface
       et1    08-09    IEEE 802.3 Ethernet Network Interface
       et2    0A-08    IEEE 802.3 Ethernet Network Interface

     F1=Help                F2=Refresh             F3=Cancel
     F8=Image               F10=Exit               Enter=Do
F1   /=Find                 n=Find Next
F9
```

Figure 5-18 Further Configuration menu for CABLE type

Scroll down using the arrow keys to the Your CABLE Type field and press F4. A submenu similar to the one shown in Figure 5-19 will pop up.

```
                          Minimum Configuration & Startup

 To Delete existing configuration data, please use Further Configuration menus

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[MORE...2]                                                  [Entry Fields]
  Network MASK (dotted decimal)                          []
* Network INTERFACE                                       en1

                               Your CABLE Type

     Move cursor to desired item and press Enter.

       bnc                                                                  #
       dix                                                                 +
       tp                                                                  +
       N/A                                                                 +
[B
     F1=Help                F2=Refresh             F3=Cancel
F1   F8=Image               F10=Exit               Enter=Do
F5   /=Find                 n=Find Next
F9
```

Figure 5-19 Your CABLE Type submenu

Use the arrow keys to select your cable type, and press Enter. Press Enter again to change the cable type for the Ethernet definition.

5.9 Virtual Ethernet

Virtual Ethernet enables inter-partition communication without the need for physical network adapters assigned to each partition. Virtual Ethernet allows the administrator to define in-memory connections between partitions handled at system level (POWER Hypervisor and operating systems interaction). These connections exhibit characteristics similar to physical high-bandwidth Ethernet connections and support the industry standard protocols (such as IPv4, IPv6, ICMP, or ARP).

The POWER Hypervisor is a layer of system firmware that supports virtualization technologies, logical partitioning, and dynamic resource movement across multiple operating system environments, including AIX 5L, Linux, and i5/OS®. With support for dynamic resource movement across multiple environments, you can move processors, memory, and I/O between partitions on the system as you move workloads among the environments.

The POWER Hypervisor supports many advanced functions, including sharing of processors, virtual I/O, high-speed communications between partitions using Virtual LAN, and concurrent maintenance. It also enforces partition security and can provide virtual LAN channels between physical partitions, reducing the need for physical Ethernet adapters and releasing I/O adapter slots.

Virtual Ethernet requires an IBM System p5 or IBM eServer pSeries with either AIX 5L Version 5.3 or the appropriate level of Linux and a Hardware Management Console (HMC) or Integrated Virtualization Manager (IVM) to define the virtual Ethernet devices. Virtual Ethernet does not require the purchase of any additional features or software, such as the Advanced POWER Virtualization feature, which is needed for shared Ethernet adapters and Virtual I/O Servers.

IBM eServer p5 and IBM eServer i5 support inter-LPAR communication using virtual networking. Virtual Ethernet adapters are connected to an IEEE 802.1q (VLAN)-style virtual Ethernet switch. Using this switch function, logical partitions can communicate with each other by using virtual Ethernet adapters and assigning VIDs (VLAN ID) that enable them to share a common logical network. The virtual Ethernet adapters are created and the VID assignments are done using the Hardware Management Console (HMC). The system transmits packets by copying the packet directly from the memory of the sender partition to the receive buffers of the receiver partition without any intermediate buffering of the packet.

For AIX 5L, a virtual Ethernet adapter is not much different from a real Ethernet adapter. It can be used:

- To configure an Ethernet interface with an IP address onto it
- To configure VLAN adapters (one per VID) onto it
- As a member of a Network Interface Backup adapter

But it cannot be used for EtherChannel or Link Aggregation.

The following example is a brief look at the virtual Ethernet adapters in a system device configuration:

```
# lsdev -Cc adapter
ent0    Available 08-08 2-Port 10/100/1000 Base-TX PCI-X Adapter (14108902)
ent1    Available 08-09 2-Port 10/100/1000 Base-TX PCI-X Adapter (14108902)
ent2    Available 0A-08 10/100/1000 Base-TX PCI-X Adapter (14106902)
ide0    Available 03-08 ATA/IDE Controller Device
lai0    Available 07-00 GXT135P Graphics Adapter
sisioa0 Available 05-08 PCI-X Dual Channel U320 SCSI RAID Adapter
sisioa1 Available 09-08 PCI-X Dual Channel U320 SCSI RAID Adapter
usbhc0  Available 02-08 USB Host Controller (33103500)
usbhc1  Available 02-09 USB Host Controller (33103500)
vsa0    Available       LPAR Virtual Serial Adapter
vsa1    Available       LPAR Virtual Serial Adapter
```

5.10 Paging space

To accommodate a large virtual memory space with a limited real memory space, the system uses real memory as a work space and keeps inactive data and programs on disk. The area of the disk that contains this data is called the system paging space. This chapter discusses the management of system paging space related functions.

5.10.1 Paging space overview

A page is a unit of virtual memory that holds 4 KB of data and can be transferred between real and auxiliary storage.

A paging space, also called a swap space, is a logical volume with the attribute type equal to paging. This type of logical volume is referred to as a paging space logical volume or simply paging space. When the amount of free real memory in the system is low, programs or data that have not been used recently are moved from real memory to paging space to release real memory for other activities.

The installation creates a default paging logical volume (hd6) on drive hdisk0, also referred as primary paging space.

The default paging space size is determined during the system customizing phase of AIX 5L installation according to the following characteristics:

- Paging space can use no less than 64 MB.
- If real memory is less than 256 MB, paging space is two times real memory.
- If real memory is greater than or equal to 256 MB, paging space is 512 MB.

The following sections describe how to determine low paging space and some tips on the sizing of new paging spaces.

5.10.2 Low paging space

If any of the following messages appear on the console or in response to a command on any terminal, it indicates a low paging space:

```
"INIT: Paging space is low"
"ksh: cannot fork no swap space"
"Not enough memory"
"Fork function failed"
"fork () system call failed"
"Unable to fork, too many processes"
"Fork failure - not enough memory available"
"Fork function not allowed. Not enough memory available."
"Cannot fork: Not enough space"
```

5.10.3 Paging space tips

The following tips, in general, will help in creating or increasing the paging space:

- Do not allocate more than one paging space logical volume on a physical volume. All processes started during the boot process are allocated paging space on the default paging space logical volume (hd6). After the additional paging space logical volumes are activated, paging space is allocated in a round robin manner in 4 KB blocks. If you have paging space on multiple physical volumes and have more than one paging space on one physical volume, you are no longer spreading paging activity over multiple physical volumes.
- Avoid putting a paging space logical volume on a heavily active logical volume, for example, a volume that is being used by a database. It is not necessary to put a paging space logical volume on each physical volume.

- Make each paging space logical volume equal in size. If you have paging spaces of different sizes, and the smaller ones become full, you will no longer be spreading your paging activity across all of the physical volumes.
- Do not extend a paging space logical volume onto multiple physical volumes. If a paging space logical volume is spread over multiple physical volumes, you will not be spreading paging activity across all the physical volumes. If you want to allocate space for paging on a physical volume that does not already have a paging space logical volume, create a new paging space logical volume on that physical volume.
- For the best performance on a system having multiple disk controllers, allocate paging space logical volumes on physical volumes that are each attached to a different disk controller.
- The default paging space may be altered while in maintenance mode.

5.10.4 Managing paging space

The AIX 5L installation defaults to a paging logical volume (hd6) on the first hard disk in rootvg, which contains part or all of the busy / (root) and /usr file systems.

Displaying paging space usage

To display the usage of the paging space, issue the **lsps** command with the -a parameter. The following example shows a 1% usage of the paging space.

```
# lsps -a
Page Space      Physical Volume   Volume Group    Size %Used Active  Auto  Type
hd6             hdisk0            rootvg         512MB     1   yes   yes    lv
#
```

Increasing paging space

You can use the **chps -s** command to dynamically increase the size of a paging space, including hd6.

For example, if you want to increase the size of hd6 with 3 LP, you issue the following command:

```
# chps -s 3 hd6
#
```

Reducing paging space

You can use the **chps -d** command to dynamically reduce the size of a paging space, including hd6. If you decrease the primary paging space, a temporary boot image and a temporary /sbin/rc.boot pointing to a temporary primary paging space will be created to make sure the system is always in a state where it can be safely rebooted.

Note: The **chps -d** command will prevent you from decreasing the size of hd6 below 32 MB or actually deleting it.

For example, if you want to decrease the size of hd6 with 1 LP, you issue the following command:

```
# chps -d 1 hd6
shrinkps: Temporary paging space paging00 created.
shrinkps: Dump device moved to temporary paging space.
shrinkps: Paging space hd6 removed.
shrinkps: Paging space hd6 recreated with new size.
#
```

Moving the hd6 paging space to another volume group

Moving a paging space with the name hd6 from rootvg to another volume group is not recommended, because the name is hard-coded in several places.

Only the paging spaces in rootvg will be active during the second phase of the boot process, and having no paging space in rootvg could severely affect system boot performance. If you want the majority of paging space on other volume groups, it is better to make hd6 as small as possible (the same size as physical memory) and then create larger paging spaces on other volume groups.

Moving a paging space within the same VG

Moving a paging space (including hd6) from the default location to a different disk within the same volume group does not require system reboot.

The following example shows the command used to move the default (hd6) paging space from hdisk0 to hdisk1:

```
# migratepv -l hd6 hdisk0 hdisk1
#
```

This may take a few minutes, depending upon the size of paging space.

Removing paging space (not hd6)

To remove a paging space, you have to deactivate the paging space.

You can use the **swapoff** command, which deactivates paging spaces without requiring a reboot.

The **swapoff** command syntax is as follows:

```
# swapoff DeviceName { DeviceName ... }
```

Use the command **swapoff /dev/paging03** to deactivate paging space paging03.

Note: It is necessary to move all pages in use on the paging space being deactivated to other paging spaces, so there must be enough space available in the other active paging spaces.

After the **swapoff** of paging03, you issue the **rmps** command as follows:

```
# rmps paging03
rmlv: Logical volume paging03 is removed.
#
```

5.11 Device configurations

The following sections discuss how to manage device configurations.

5.11.1 Determining the existing device configuration

To determine a specific subset of the configuration of your system, you can use the following commands:

lscfg Displays configuration, diagnostic, and vital product data (VPD) information about the system.

lsdev Displays devices in the system and their characteristics.

lsattr Displays attribute characteristics and possible values of attributes for devices in the system.

There are also device-specific and class-specific commands that you can use to obtain detailed information about your system, for example, **bindprocessor**, **ifconfig**, **lspv**, and **lslpp**.

Using the lscfg command

You can use the **lscfg** command to display summary or detailed data about devices. If you run the **lscfg** command without any flags, it displays the name, location, and description of each device found in the current Customized VPD object class that is a child device of the sys0 object. Information on a specific device can be displayed with the -l flag.

You can also use the **lscfg** command to display vital product data (VPD), such as part numbers, serial numbers, and engineering change levels from either the Customized VPD object class or platform specific areas. Not all devices contain VPD data.

The general command syntax of the **lscfg** command is as follows:

```
lscfg [ -v ] [ -p ] [ -s ] [ -l Name ]
```

Some of the most commonly used flags with the **lscfg** command are given in Table 5-4.

Table 5-4 Commonly used flags for the lscfg command

Flag	Description
-l *Name*	Displays device information for the named device.
-p	Displays the platform-specific device information. This flag only applies to AIX Version 4.2.1 or later.
-v	Displays the VPD found in the Customized VPD object class. Also, on AIX Version 4.2.1 or later, displays platform specific VPD when used with the -p flag.

The following examples show the usage of **lscfg**:

- To display the VPD for about rmt0, execute:

```
# lscfg -v -p -l rmt0
  rmt0    U787B.001.DNW108F-P1-T14-L0-L0  LVD SCSI Tape Drive (80000 MB)

        Manufacturer................IBM
        Machine Type and Model......VXA-2
        Device Specific.(Z1)........2107
        Serial Number...............20173584
        Device Specific.(LI)........A170029C
        Part Number.................19P4897
        FRU Number..................19P4898
        EC Level....................H28141
        Device Specific.(Z0)........0180020283000130
        Device Specific.(Z3)........L1

  PLATFORM SPECIFIC

  Name:  st
    Node:  st
    Device Type:  byte
```

- To obtain the physical location and firmware version of eth0, execute:

```
# lscfg -v -p -l ent2
  ent2            U787B.001.DNW108F-P1-C1-T1  10/100/1000 Base-TX PCI-X
                  Adapter (14106902)

      10/100/1000 Base-TX PCI-X Adapter:
        Part Number.................00P6130
        FRU Number..................00P6130
```

```
        EC Level....................H12818
        Manufacture ID..............YL1021
        Network Address.............000255D3DD00
        ROM Level.(alterable).......GOL021
        Device Specific.(YL)........U787B.001.DNW108F-P1-C1-T1

  PLATFORM SPECIFIC

  Name:  ethernet
    Node:  ethernet@1
    Device Type:  network
    Physical Location: U787B.001.DNW108F-P1-C1-T1
```

The location is in slot C1 and the ROM level is at version GOL021.

Using the lsdev command

You can use the **lsdev** command to display information about devices in the device configuration database. You can use this command to display information from either the Customized Devices object class in ODM using the -C flag or the Predefined Devices object class in ODM using the -P flag.

The general command syntax of the **lsdev** command is as follows:

lsdev [-C][-c *Class*] [-s *Subclass*] [-t *Type*] [-f *File*]
[-F *Format* | -r *ColumnName*] [-h] [-H] [-l *Name*] [-S *State*]
[-p *Parent*]
lsdev -P [-c *Class*] [-s *Subclass*] [-t *Type*] [-f *File*]
[-F *Format* | -r *ColumnName*] [-h] [-H]

Some of the most commonly used flags with the **lsdev** command are given in Table 5-5.

Table 5-5 Commonly used flags for the lsdev command

Flag	Description
-C	Lists information about a device that is in the Customized Devices object class. The default information displayed is name, status, location, and description. This flag cannot be used with the -P flag.
-c *Class*	Specifies a device class name. This flag can be used to restrict output to devices in a specified class.
-H	Displays headers above the column output.
-h	Displays the command usage message.
-P	Lists information about a device that is in the Predefined Devices object class. The default information displayed is class, type, subclass, and description. This flag cannot be used with the -C, -l, or -S flags.

Flag	Description
-S *State*	Lists all devices in a specified state as named by the State parameter.

The following examples show the usage of the **lsdev** command:

- To show the disk drives on your system:

```
# lsdev -C -c disk
hdisk0 Available 05-08-00-3,0 16 Bit LVD SCSI Disk Drive
hdisk1 Available 05-08-00-4,0 16 Bit LVD SCSI Disk Drive
hdisk2 Available 05-08-00-5,0 16 Bit LVD SCSI Disk Drive
hdisk3 Available 05-08-00-8,0 16 Bit LVD SCSI Disk Drive
hdisk4 Available 09-08-00-3,0 16 Bit LVD SCSI Disk Drive
```

- To show the tape devices on your system:

```
# lsdev -C -c tape
rmt0 Available 09-08-00-0,0 LVD SCSI Tape Drive
```

- To show all the adapters on your system:

```
# lsdev -C -c adapter
ent0    Available 08-08 2-Port 10/100/1000 Base-TX PCI-X Adapter (14108902)
ent1    Available 08-09 2-Port 10/100/1000 Base-TX PCI-X Adapter (14108902)
ent2    Available 0A-08 10/100/1000 Base-TX PCI-X Adapter (14106902)
ide0    Available 03-08 ATA/IDE Controller Device
lai0    Available 07-00 GXT135P Graphics Adapter
sisioa0 Available 05-08 PCI-X Dual Channel U320 SCSI RAID Adapter
sisioa1 Available 09-08 PCI-X Dual Channel U320 SCSI RAID Adapter
usbhc0  Available 02-08 USB Host Controller (33103500)
usbhc1  Available 02-09 USB Host Controller (33103500)
vsa0    Available       LPAR Virtual Serial Adapter
vsa1    Available       LPAR Virtual Serial Adapter
```

Using the lsattr command

You can use the **lsattr** command to display information about the attributes of a given device or kind of device. If you do not specify the device's logical name (-l *Name*), you must use a combination of one or all of the -c *Class*, -s *Subclass*, and -t *Type* flags to uniquely identify the predefined device. The general syntax of the **lsattr** command is as follows:

`lsattr { -D [ -O ] | -E [ -O ] | -F` *Format* `[ -Z` *Character* `] } -l` *Name* `[ -a` *Attribute* `] ... [ -f` *File* `] [ -h ] [ -H ]`
`lsattr { -D [ -O ] | -F` *Format* `[ -Z` *Character* `] } { [ -c` *Class* `] [ -s` *Subclass* `] [ -t` *Type* `] } [ -a` *Attribute* `] ... [ -f` *File* `] [ -h ] [ -H ]`
`lsattr -R { -l` *Name* `| [ -c` *Class* `] [ -s` *Subclass* `] [ -t` *Type* `] } -a` *Attribute* `[ -f` *File* `] [ -h ] [ -H ]`

The flags commonly used with the **lsattr** command are given in Table 5-6.

Table 5-6 Commonly used flags for the lsattr command

Flag	Description
-D	Displays the attribute names, default values, descriptions, and user-set flag values for a specific device when not used with the -O flag. The -D flag displays only the attribute name and default value in colon format when used with the -O flag.
-E	Displays the attribute names, current values, descriptions, and user-settable flag values for a specific device when not used with the -O flag. The -E flag only displays the attribute name and current value in colon format when used with the -O flag. This flag cannot be used with the -c, -D, -F, -R, -s, or -t flags.
-F *Format*	Displays the output in a user-specified format.
-a *Attribute*	Displays information for the specified attributes of a specific device or kind of device.
-c *Class*	Specifies a device class name. This flag cannot be used with the -E or -l flags.
-f *File*	Reads the needed flags from the File parameter.
-H	Displays headers above the column output. To use the -H flag with either the -O or the -R flags is meaningless; the -O or -R flag prevails.
-l *Name*	Specifies the device logical name in the Customized Devices object class whose attribute names or values are to be displayed.
-O	Displays all attribute names separated by colons and, on the second line, displays all the corresponding attribute values separated by colons.
-R	Displays the legal values for an attribute name. The -R flag cannot be used with the -D, -E, -F, and -O flags, but can be used with any combination of the -c, -s, and -t flags that uniquely identifies a device from the Predefined Devices object class or with the -l flag. The -R flag displays the list attribute values in a vertical column as follows: Value1 Value2 . . ValueN The -R flag displays the range attribute values as x...n(+i) where *x* is the start of the range, *n* is the end of the range, and *i* is the increment.

Flag	Description
-s *Subclass*	Specifies a device subclass name. This flag can be used to restrict the output to devices for a specified subclass. This flag cannot be used with the -E or -l flags.
-t *Type*	Specifies a device type name. This flag can be used to restrict the output to that of devices of a specified class. This flag cannot be used with the -E or -l flag.

When displaying the effective values of the attributes for a customized device, the information is obtained from the Configuration Database, not the device. Generally, the database values reflect how the device is configured unless it is reconfigured with the **chdev** command using the -P or -T flag. If this has occurred, the information displayed by the **lsattr** command might not correctly indicate the current device configuration until after the next system boot.

If you use the -D or -E flags, the output defaults to the values for the attribute's name, value, description, and user-settable strings unless also used with the -O flag. The -O flag displays the names of all attributes specified separated by colons. On the next line, the -O flag displays all the corresponding attribute values separated by colons. The -H flag can be used with either the -D, -E, or -F flags to display headers above the column names. You can define the format of the output with a user-specified format where the format parameter is a quoted list of column names separated by non-alphanumeric characters or white space using the -F *Format* flag.

The following examples show the usage of **lsattr**:

- To learn more about a particular processor, use the following **lsattr** command:

```
# lsattr -El proc0
frequency   1654344000     Processor Speed       False
smt_enabled true           Processor SMT enabled False
smt_threads 2              Processor SMT threads False
state       enable         Processor state       False
type        PowerPC_POWER5 Processor type        False
```

- To discover how much memory is installed, use the following command:

```
# lsattr -El sys0 | grep realmem
realmem     3883008     Amount of usable physical memory in Kbytes false
```

- To discover if ent2 supports jumbo frames transmission, use the following command:

```
# lsattr -EHl ent0 -a jumbo_frames
attribute    value description          user_settable

jumbo_frames no    Transmit jumbo frames True
```

  This example shows that ent2 supports jumbo frames but the option is currently not enabled.

- To discover if device driver software for the 14100401 class of adapters (gigabit Ethernet) is installed, use the **lslpp** command as follows:

```
# lslpp -l | grep 14100401
devices.pci.14100401.diag  5.3.0.0  COMMITTED  Gigabit Ethernet-SX PCI
devices.pci.14100401.rte   5.3.0.0  COMMITTED  Gigabit Ethernet-SX PCI
devices.pci.14100401.rte   5.3.0.0  COMMITTED  Gigabit Ethernet-SX PCI
```

5.11.2 Remove a device configuration

To unload an existing device from the system, you have two possibilities: either change the state from AVAILABLE to DEFINED or permanently delete all entries from the ODM. The command **rmdev** executes this task. The general syntax of **rmdev** is as follows:

`rmdev { -l | -p }` *Name* `[ -d | -S ] [ -f` *File* `] [ -h ] [ -q ] [ -R ]`

The flags commonly used with the **rmdev** command are given in Table 5-7.

Table 5-7 Commonly used flags for the rmdev command

Flag	Description
-d	Removes the device definition from the Customized Devices object class. This flag cannot be used with the -S flag.
-q	Suppresses the command output messages from standard output and standard error
-f *File*	Reads the needed flags from the File parameter.
-l *Name*	Specifies the logical device, indicated by the Name parameter, in the Customized Devices object class. This flag cannot be used with the -p flag.
-p *Name*	Specifies the parent logical device (indicated by the Name parameter) in the Customized Devices object class, with children that must be removed. This flag may not be used with the -l flag.
-R	Unconfigures the device and its children. When used with the -d or -S flags, the children are undefined or stopped, respectively.
-S	Makes the device unavailable by only calling the Stop method if the device has a Stop method. This flag cannot be used with the -d flag.

The following example shows the process to change the state:

```
# lsdev -Cctape
rmt0 Available 09-08-00-0,0 LVD SCSI Tape Drive
# rmdev -l rmt0
rmt0 Defined
# lsdev -Cctape
rmt0 Defined 09-08-00-0,0 LVD SCSI Tape Drive
```

To unload the device configuration from the ODM, use the -d option:

```
# rmdev -dl rmt0
rmt0 deleted
# lsdev -Cctape
#
```

Result: rmt0 is completely removed. To redetect the device, use the **cfgmgr** command:

```
# cfgmgr
# lsdev -Cctape
rmt0 Available 09-08-00-0,0 LVD SCSI Tape Drive
```

5.11.3 Modify an existing device configuration

To modify an existing device parameter, use the **chdev** command.

The general syntax of the **chdev** command is as follows:

chdev -l *Name* [-a *Attribute=Value* ...] [-f *File*] [-h] [-p *ParentName*] [-P | -T] [-q] [-w *ConnectionLocation*]

The flags commonly used with the **chdev** command are given in Table 5-8.

Table 5-8 Commonly used flags for the chdev command

Flag	Description
-a *Attribute=Value*	Specifies the device attribute-value pairs used for changing specific attribute values. The *Attribute=Value* parameter can use one attribute value pair or multiple attribute value pairs for one -a flag. If you use an -a flag with multiple attribute value pairs, the list of pairs must be enclosed in quotes with spaces between the pairs. For example, entering -a *Attribute=Value* lists one attribute value pair per flag, while entering -a '*Attribute1=Value1 Attribute2=Value2*' lists more than one attribute value pair.
-f *File*	Reads the needed flags from the File parameter.
-l *Name*	Specifies the device logical name in the Customized Devices object class whose characteristics are to be changed.
-p *ParentName*	Specifies the new device logical name of the parent device in the Customized Devices object class. Use this flag only when changing the parent of the device. Not all devices support the -p flag.
-P	Changes the device's characteristics permanently in the Customized Devices object class without actually changing the device. This is useful for devices that cannot be made unavailable and cannot be changed while in the available state. The change is made to the database, and the changes are applied to the device when the system is rebooted. This flag cannot be used with the -T flag. Not all devices support the -P flag.

Flag	Description
-T	Changes the characteristics of the device temporarily without changing the Customized Devices object class for the current start of the system. This flag cannot be used with the -P flag. Not all devices support the -T flag.
-w *ConnectionLocation*	Specifies the new connection location of the device on the parent. Use this flag only when changing the connection location of the device. Not all devices support the -w flag.

The device logical name is specified with the -l Name flag. The device can be in the Defined, Stopped, or Available state. Some changes may not be allowed when the device is in the Available state. When changing the device characteristics, you can supply the flags either on the command line or in a file specified with the -f File flag.

When neither the -P or the -T flags are specified, the **chdev** command applies the changes to the device and updates the database to reflect the changes. If the -P flag is specified, only the database is updated to reflect the changes, and the device itself is left unchanged. This is useful in cases where a device cannot be changed because it is in use; in which case, the changes will be applied to the device when the system is restarted. The -T flag is used to make a temporary change in the device without the change being reflected in the database. It is temporary in that the device will revert to the characteristics described in the database when the system is restarted. Not all devices support the -P and -T flags. If a device that is in the Defined state, changes are only applied to the database.

Attention: To protect the configuration database, the **chdev** command is not interruptible. Stopping this command before it is complete could result in a corrupted database.

To change the retention instructions of the rmt0 4 mm SCSI tape drive so that the drive does not move the tape to the beginning, then to the end, and then back to the beginning each time a tape is inserted or the drive is powered on, type the following:

```
chdev -l rmt0 -a ret=no
```

The system displays a message similar to the following:

```
rmt0 changed
```

To change one or more attributes of the tok0 token-ring adapter to preset values as described in the changattr file, type the following:

```
chdev -l tok0 -f changattr
```

The system displays a message similar to the following:

```
tok0 changed
```

To change the SCSI ID of the available scsi0 SCSI adapter that cannot be changed (made unavailable due to available disk drives connected to it), type the following:

```
chdev  -l scsi0 -a id=6 -P
```

The system displays a message similar to the following:

```
scsi0 changed
```

To apply the change to the adapter, shut down and restart the system.

To move the defined tty11 TTY device to port 0 on the sa5 serial adapter, type the following:

```
chdev  -l tty11  -p sa5  -w 0
```

The system displays a message similar to the following:

```
tty11 changed
```

To change the maximum number of processes allowed per user to 100, type the following:

```
chdev -l sys0 -a maxuproc=100
```

The system displays a message similar to the following:

```
sys0 changed
```

To change a parameter of a device, use the **lsattr** command to look up the name of the attribute. The following example shows the output for a tape drive:

```
# lsattr -El rmt0
block_size    1024  BLOCK size (0=variable length)       True
compress      yes   Use data COMPRESSION                 True
density_set_1 129   DENSITY setting #1                   True
density_set_2 128   DENSITY setting #2                   True
extfm         no    Use EXTENDED file marks              True
mode          yes   Use DEVICE BUFFERS during writes     True
ret           no    RETENSION on tape change or reset    True
ret_error     no    RETURN error on tape change or reset True
size_in_mb    80000 Size in Megabytes                    False
```

If you want to change the block size parameter, use the following **chdev** command:

```
# chdev -l rmt0 -a block_size=512
rmt0 changed
```

To verify your applied setting, execute the **lsattr** command again:

```
# lsattr -El rmt0
block_size    512   BLOCK size (0=variable length)       True
compress      yes   Use data COMPRESSION                 True
density_set_1 129   DENSITY setting #1                   True
density_set_2 128   DENSITY setting #2                   True
extfm         no    Use EXTENDED file marks              True
mode          yes   Use DEVICE BUFFERS during writes     True
ret           no    RETENSION on tape change or reset    True
ret_error     no    RETURN error on tape change or reset True
size_in_mb    80000 Size in Megabytes                    False
```

5.11.4 SMIT fast paths for devices configuration

SMIT fast paths can be used to access the device configuration menus: **smitty devices** is the same as **smitty chdev** and **smitty rmdev**.

For example, the example used in 5.11.3, “Modify an existing device configuration” on page 201 executed with SMIT, looks as follows:

1. Start SMIT with the fast path **chdev**, as seen in Chapter 5-20, “SMIT chdev example” on page 205.

```
                                    Devices

Move cursor to desired item and press Enter.

[TOP]
  Install/Configure Devices Added After IPL
  Printer/Plotter
  TTY
  Asynchronous Adapters
  PTY
  Console
  MPIO Management
  Fixed Disk
  Disk Array
  CD ROM Drive
  Read/Write Optical Drive
  Diskette Drive
  Tape Drive
  Communication
  Graphic Displays
[MORE...16]

F1=Help              F2=Refresh           F3=Cancel            F8=Image
F9=Shell             F10=Exit             Enter=Do
```

Figure 5-20 SMIT chdev example

2. Choose the Tape Drive menu; the next menu appears, as shown in Figure 5-21.

```
                                   Tape Drive

Move cursor to desired item and press Enter.

  List All Defined Tape Drives
  List All Supported Tape Drives
  Add a Tape Drive
  Change / Show Characteristics of a Tape Drive
  Remove a Tape Drive
  Configure a Defined Tape Drive
  Generate Error Report
  Trace a Tape Drive

F1=Help              F2=Refresh           F3=Cancel            F8=Image
F9=Shell             F10=Exit             Enter=Do
```

Figure 5-21 SMIT Tape Drive menu

3. Choose Change / Show Characteristics of a Tape Drive; a list with all available tape drives on your system is shown. You can find an example in Figure 5-22.

```
                                  Tape Drive

Move cursor to desired item and press Enter.

  List All Defined Tape Drives
  List All Supported Tape Drives
  Add a Tape Drive
  Change / Show Characteristics of a Tape Drive
  Remove a Tape Drive
  Configure a Defined Tape Drive
  Generate Error Report
  Trace a Tape Drive

  +--------------------------------------------------------------------------+
  |                               Tape Drive                                 |
  |                                                                          |
  | Move cursor to desired item and press Enter.                             |
  |                                                                          |
  |   rmt0 Available 09-08-00-0,0 LVD SCSI Tape Drive                        |
  |                                                                          |
  | F1=Help                  F2=Refresh               F3=Cancel              |
  | F8=Image                 F10=Exit                 Enter=Do               |
F1| /=Find                   n=Find Next                                     |
F9+--------------------------------------------------------------------------+
```

Figure 5-22 Sample Tape Drive selection window

4. Choose the tape drive for which you want to change the settings, in this example, rmt0. In the following screen, change the BLOCK size parameter from 1024 to 512, as described in Figure 5-23 on page 207. Press Enter to confirm the changes.

```
                Change / Show Characteristics of a Tape Drive

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[TOP]                                                  [Entry Fields]
  Tape Drive                                          rmt0
  Tape Drive type                                     scsd
  Tape Drive interface                                scsi
  Description                                         LVD SCSI Tape Drive
  Status                                              Available
  Location                                            09-08-00-0,0
  Parent adapter                                      scsi2
  Connection address                                  0,0
  BLOCK size (0=variable length)                     [512]                   +#
  Use DEVICE BUFFERS during writes                    yes                     +
  RETURN error on tape change or reset                no                      +
  RETENSION on tape change or reset                   no                      +
  DENSITY setting #1                                  129                     +
[MORE...3]

F1=Help              F2=Refresh           F3=Cancel            F4=List
F5=Reset             F6=Command           F7=Edit              F8=Image
F9=Shell             F10=Exit             Enter=Do
```

Figure 5-23 SMIT characteristics change of a tape drive

5. The result of the command is shown in Figure 5-24.

```
                                 COMMAND STATUS

Command: OK            stdout: yes          stderr: no

Before command completion, additional instructions may appear below.

rmt0 changed













F1=Help              F2=Refresh           F3=Cancel            F6=Command
F8=Image             F9=Shell             F10=Exit             /=Find
n=Find Next
```

Figure 5-24 SMIT command status output for the chdev command

5.11.5 Special device configurations

The following example shows how device parameters can have an impact on the output of **lsdev** and **lscfg**:

The machine in the given example is a p550, with two physical processors installed. This can be confirmed by executing the following command:

```
# lscfg | grep proc
+ proc0                                               Processor
+ proc2                                               Processor
```

You receive the same output with the **lsdev** command:

```
# lsdev -Cc processor
proc0 Available 00-00 Processor
proc2 Available 00-02 Processor
```

However, if you execute the **bindprocessor** command with the -q parameter, it will query the available processors. The following result is obtained:

```
# bindprocessor -q
The available processors are:  0 1 2 3
```

In the output of several other commands, the four processors appear; for example, with the **sar** command, an extract from the output is shown as follows:

```
# sar

AIX server2 3 5 00C478DE4C00    11/11/05

System configuration: lcpu=4
```

The output lcpu=4 is an indication what is going on. There is a difference between the physical and the logical processors.

A sample output of the **lsattr** command on proc0 shows the reason:

```
# lsattr -E -l proc0
frequency   1654344000     Processor Speed       False
smt_enabled true           Processor SMT enabled False
smt_threads 2              Processor SMT threads False
state       enable         Processor state       False
type        PowerPC_POWER5 Processor type        False
```

The parameter smt_threads indicates that for this proc0, two simultaneous threads can be handled.

To change the simultaneous multithreading settings of your server, use the **smtctl** command.

Examples of the **smtctl** command on the same system:

- To turn off simultaneous multithreading immediately, without rebooting, use the following command:

```
# smtctl -m off -w now
smtctl: SMT is now disabled.
# bindprocessor -q
The available processors are:  0 1
```

- To turn on simultaneous multithreading after the next reboot, use the following command:

```
# smtctl -m on -w boot
smtctl: SMT will be enabled on the next reboot.
        Note that the boot image must be remade with the bosboot
        command before the next reboot.
# bindprocessor -q
The available processors are:  0 1
```

5.12 Reconfiguring a system from a 32-bit to a 64-bit kernel

AIX 5L Version 5.3 allows you to install a 64-bit kernel as well as the previously available 32-bit kernel during installation. If your system has 64-bit processors, the 64-bit kernel is automatically installed with the Base Operating System (BOS). However, it is not mandatory to run the 64-bit kernel on a system with 64-bit processors. Some products and devices that can be used in systems with 64-bit processors do not support the 64-bit kernel yet. Unless your system environment requires a 64-bit kernel, chances are that you will not need to enable it during installation.

To identify the type of system hardware you have, either 32-bit or 64-bit, execute the **bootinfo -y** command. If the command returns a 32, you cannot use the 64-bit kernel.

Important: The AIX 5L operating system previously contained both a uniprocessor 32-bit and a multiprocessor 32-bit kernel. Effective with AIX 5L Version 5.3, the operating system supports only the multiprocessor kernel, regardless of the number of physical processors.

If you want to enable 64-bit kernel after system installation, you will need to instruct the system to use the 64-bit kernel information stored in the /usr/lib/boot directory. There are two kernels available in the /usr/lib/boot directory:

unix_mp 32-bit kernel for multiprocessor systems

unix_64 64-bit kernel for 64-bit processor systems

The following example provides the commands to run to enable the 64-bit kernel after system installation:

```
# ln -sf /usr/lib/boot/unix_64 /unix
# ln -sf /usr/lib/boot/unix_64 /usr/lib/boot/unix
# bosboot -ad /dev/ipldevice

bosboot: Boot image is 23137 512 byte blocks.
# shutdown -r
```

After the system has rebooted, it will be running the 64-bit kernel. To reactivate the 32-bit kernel, follow the same procedure, substituting unix_mp for unix_64, depending on your system type.

To verify your settings, execute the following command:

```
# ls -al /unix
lrwxrwxrwx   1 root  system  21 Nov 11 11:30 /unix -> /usr/lib/boot/unix_64
```

6

Disk storage management

In this chapter, we describe the fundamentals of Logical Volume Management (LVM).

© Copyright IBM Corp. 2006. All rights reserved.

6.1 Storage management concepts

The fundamental concepts used by LVM are physical volumes, volume groups, physical partitions, logical volumes, logical partitions, file systems, and raw devices. Some of their characteristics are presented as follows:

- Each individual disk drive is a named physical volume (PV) and has a name such as hdisk0 or hdisk1.
- One or more PVs can make up a volume group (VG). A physical volume can belong to a maximum of one VG.
- You cannot assign a fraction of a PV to one VG. A physical volume is assigned entirely to a volume group.
- Physical volumes can be assigned to the same volume group even though they are of different types, such as SCSI or SSA.
- Storage space from physical volumes is divided into physical partitions (PPs). The size of the physical partitions is identical on all disks belonging to the same VG.
- Within each volume group, one or more logical volumes (LVs) can be defined. Data stored on logical volumes appears to be contiguous from the user point of view, but can be spread on different physical volumes from the same volume group.
- Logical volumes consist of one or more logical partitions (LPs). Each logical partition has at least one corresponding physical partition. A logical partition and a physical partition always have the same size. You can have up to three copies of the data located on different physical partitions. Usually, physical partitions storing identical data are located on different physical disks for redundancy purposes.
- Data from a logical volume can be stored in an organized manner, having the form of files located in directories. This structured and hierarchical form of organization is named a file system.
- Data from a logical volume can also be seen as a sequential string of bytes. This type of logical volumes are named raw logical volumes. It is the responsibility of the application that uses this data to access and interpret it correctly.
- The volume group descriptor area (VGDA) is an area on the disk that contains information pertinent to the volume group that the physical volume belongs to. It also includes information about the properties and status of all physical and logical volumes that are part of the volume group. The information from VGDA is used and updated by LVM commands. There is at least one VGDA per physical volume. Information from VGDAs of all disks that are part of the same volume group must be identical. The VGDA internal architecture and

location on the disk depends on the type of the volume group (original, big, or scalable).

- The volume group status area (VGSA) is used to describe the state of all physical partitions from all physical volumes within a volume group. The VGSA indicates if a physical partition contains accurate or stale information. VGSA is used for monitoring and maintained data copies synchronization. The VGSA is essentially a bitmap and its architecture and location on the disk depends on the type of the volume group.
- A logical volume control block (LVCB) contains important information about the logical volume, such as the number of the logical partitions or disk allocation policy. Its architecture and location on the disk depends on the type of the volume group it belongs to. For standard volume groups, the LVCB resides on the first block of user data within the LV. For big volume groups, there is additional LVCB information in VGDA on the disk. For scalable volume groups, all relevant logical volume control information is kept in the VGDA as part of the LVCB information area and the LV entry area.

Note: The layout of the hard disk drives is defined by the /usr/include/sys/hd_psn.h header file. The LVM records starts at sector 7 on the disk. All LVM record structures are defined in the /usr/include/lvmrec.h header file.

6.1.1 Limitations of logical volume storage

The LVM layer for AIX 5L Version 5.3 provides an increased level of flexibility in disk management. There are limitations that you have to be aware of, which are listed in Table 6-1.

Table 6-1 LVM limitations for AIX 5L Version 5.3

VG Type	Maximum PVs	Maximum LVs	Maximum PPs per VG	Maximum PP size
Normal VG	32	256	32512 (1016*32)	1 GB
Big VG	128	512	130048 (1016*128)	1 GB
Scalable VG	1024	4096	2097152	128 GB

6.2 Physical volumes

When a disk drive is initially added to the system, it is seen a simple device. The disk is not yet accessible for LVM operations. To be made accessible, it has to be assigned to a volume group, which means changing from a disk to a physical volume. For each disk, two device drivers will be created under the /dev directory: one block device driver and one character device driver. The disk drive is assigned an 32-bit unique identifier that is called a physical volume identifier (PVID).

The **lspv** command, used without any parameters, displays all physical volumes, their PVIDs, the volume groups whom the PVs belong to, and the status of volume groups, as shown in Example 6-1.

Example 6-1 Using **lspv** *command to display physical volumes*

```
# lspv
hdisk0          00c478de09a40b16                    rootvg          active
hdisk1          00c478de09caf163                    rootvg          active
hdisk2          00c478de09caf37f                    None
hdisk3          00c478de49630c6a                    None
hdisk4          00c478de00655246                    None
hdisk5          00c478de008a399b                    None
hdisk6          00c478de008a3ba1                    None
hdisk7          00c478de6c9883b7                    None
```

6.2.1 PVID

When the PVID is generated, the system uses its own serial number and a time stamp to ensure that two disks would never have the same PVID.

The PVIDs are stored also in ODM. They are used by LVM commands and possibly referenced by external applications such as HACMP.

The following command changes an available disk device to a physical volume by assigning a PVID, if it does not already have one:

```
chdev -l hdisk7 -a pv=yes
```

This command has no effect if the disk is already a physical volume.

The following commands clears the PVID from the physical volume:

```
chdev -l hdisk7 -a pv=clear
```

Note: You can list the PVID of a disk using intermediate level commands, such as **lquerypv**, as follows:

```
# lquerypv -h /dev/hdisk2 80 10
00000080   00C478DE 09CAF37F 00000000 00000000  |..x.............|
```

6.2.2 Listing information about physical volumes

You can use the **lspv** command and pass the name of the physical volume as a parameter in order to find more details about a physical volume, as shown in Example 6-2.

Example 6-2 Using lspv to display detailed information about a physical volume

```
# lspv hdisk2
PHYSICAL VOLUME:    hdisk2                   VOLUME GROUP:     testvg
PV IDENTIFIER:      00c478de09caf37f VG IDENTIFIER
00c478de00004c00000001078fc3497d
PV STATE:           active
STALE PARTITIONS:   0                        ALLOCATABLE:      yes
PP SIZE:            128 megabyte(s)          LOGICAL VOLUMES:  1
TOTAL PPs:          546 (69888 megabytes)    VG DESCRIPTORS:   2
FREE PPs:           542 (69376 megabytes)    HOT SPARE:        no
USED PPs:           4 (512 megabytes)        MAX REQUEST:      256 kilobytes
FREE DISTRIBUTION:  110..105..109..109..109
USED DISTRIBUTION:  00..04..00..00..00
```

The meaning of the fields in Example 6-2 are:

PHYSICAL VOLUME The name of the physical volume.

PV IDENTIFIER The physical volume identifier.

PV STATE The state of the physical volume: active, missing, or removed. The status can be changed using the **chpv** command.

STALE PARTITIONS The number of stale partitions.

PP SIZE The size of a physical partition.

TOTAL PPs The total number of physical partitions residing on the disk (free and used).

FREE PPs The number of free partitions available on the physical volume.

USED PPs The number of used partitions on the physical volume.

FREE DISTRIBUTION
The number of free physical partitions available on each intra-physical volume area.

USED DISTRIBUTION
The number of used physical partitions located on each intra-physical volume area.

VOLUME GROUP The name of the volume group to which the physical volume belongs.

VG IDENTIFIER The identifier of the volume group to which the physical volume belongs.

ALLOCATABLE Allocation permission for the physical volume determines if free PPs can be allocated to logical volumes.

LOGICAL VOLUMES The number of the logical volumes that have at least one LP located on this physical volume.

VG DESCRIPTORS The number of VGDAs located on this physical volume.

HOT SPARE The physical volume is defined as hot spare or not.

MAX REQUEST The LTG size for this physical volume.

The **lspv** command can be used with the -l flag to display the names of all the logical volumes that have at least one partition located on the physical volume, the total number of partitions located on that physical volume, the total number of PPs that correspond to the LPs, the distribution of PPs corresponding to each intra-physical disk area, the mounting point of the logical volume, if it exists. An example of the output obtained when running this command is presented in Example 6-3.

Example 6-3 Using the lspv -l command

```
# lspv -l hdisk0
hdisk0:
LV NAME               LPs     PPs     DISTRIBUTION          MOUNT POINT
hd3                   1       1       00..00..01..00..00    /tmp
hd10opt               1       1       00..00..01..00..00    /opt
hd5                   1       1       01..00..00..00..00    N/A
hd8                   1       1       00..00..01..00..00    N/A
hd6                   11      11      00..00..11..00..00    N/A
hd2                   8       8       00..00..08..00..00    /usr
hd9var                1       1       00..00..01..00..00    /var
hd1                   1       1       00..00..01..00..00    /home
hd4                   1       1       00..00..01..00..00    /
```

If you want to display the number, location of partitions, and the logical volumes they correspond to, use the **lspv -p** command, as shown in Example 6-4.

Example 6-4 Using lspv to display the allocation of PPs to logical volumes

```
# lspv -p hdisk0
hdisk0:
PP RANGE  STATE   REGION        LV NAME             TYPE       MOUNT POINT
  1-1     used    outer edge    hd5                 boot       N/A
  2-110   free    outer edge
111-219   free    outer middle
220-220   used    center        hd8                 jfs2log    N/A
221-221   used    center        hd2                 jfs2       /usr
222-222   used    center        hd3                 jfs2       /tmp
223-223   used    center        hd10opt             jfs2       /opt
224-230   used    center        hd2                 jfs2       /usr
231-241   used    center        hd6                 paging     N/A
242-328   free    center
329-437   free    inner middle
438-546   free    inner edge
```

You can make a detailed map of the disk layout and display the relationship between each physical and logical partition by using the **lspv -M** command, as shown in Example 6-5.

Example 6-5 Using the lspv -M command to display the layout of a physical volume

```
# lspv -M hdisk0|more
hdisk0:1       hd5:1
hdisk0:2-110
hdisk0:111     lv1:1:1
hdisk0:112     lv1:2:1
hdisk0:113     lv1:3:1
hdisk0:114     lv1:4:1
hdisk0:115     lv1:5:1
hdisk0:116     lv1:6:1
hdisk0:117     lv1:7:1
hdisk0:118     lv1:8:1
hdisk0:119     lv1:9:1
hdisk0:120     lv1:10:1
hdisk0:121     lv1:11:1
hdisk0:122     lv1:12:1
hdisk0:123-219
hdisk0:220     hd8:1
hdisk0:221     hd2:1
hdisk0:222     hd3:1
hdisk0:223     hd10opt:1
hdisk0:224     hd2:2
hdisk0:225     hd2:3
hdisk0:226     hd2:4
hdisk0:227     hd2:5
hdisk0:228     hd2:6
```

```
hdisk0:229      hd2:7
hdisk0:230      hd2:8
hdisk0:231      hd6:1
hdisk0:232      hd6:2
hdisk0:233      hd6:3
hdisk0:234      hd6:4
hdisk0:235      hd6:5
hdisk0:236      hd6:6
hdisk0:237      hd6:7
hdisk0:238      hd6:8
hdisk0:239      hd6:9
hdisk0:240      hd6:10
hdisk0:241      hd6:11
hdisk0:242-546
```

6.2.3 Changing the allocation permission for a physical volume

The allocation permission for a physical volume determines if physical partitions located on that physical volume, which have not been allocated to a logical volume yet, can be allocated to logical volumes. The operation of logical volumes that reside on that physical volume is not affected.

In Example 6-6, we disabled the ability to allocate new free physical partitions from hdisk2 to any logical volume. We tried to create a logical volume that would use PPs from hdisk2 and received an error message stating that partitions from that physical volume were not allocatable.

Example 6-6 Disabling partition allocation for a physical volume

```
# chpv -an hdisk2
# lspv hdisk2
PHYSICAL VOLUME:    hdisk2                   VOLUME GROUP:     testvg
PV IDENTIFIER:      00c478de09caf37f VG IDENTIFIER
00c478de00004c00000001078fc3497d
PV STATE:           active
STALE PARTITIONS:   0                        ALLOCATABLE:     no
PP SIZE:            128 megabyte(s)          LOGICAL VOLUMES:  1
TOTAL PPs:          546 (69888 megabytes)    VG DESCRIPTORS:   2
FREE PPs:           542 (69376 megabytes)    HOT SPARE:        no
USED PPs:           4 (512 megabytes)        MAX REQUEST:      256 kilobytes
FREE DISTRIBUTION:  110..105..109..109..109
USED DISTRIBUTION:  00..04..00..00..00
# mklv -y test -t jfs2 testvg 10 hdisk2
0516-823 lquerypv: Physical Volume hdisk2 is not allocatable.
0516-848 lquerypv: Failure on physical volume hdisk2, it may be missing
        or removed.
0516-822 mklv: Unable to create logical volume.
```

To turn on the allocation permission, use the following command:

```
chpv -ay hdisk2
```

6.2.4 Changing the availability of a physical volume

The availability of a physical volume defines whether any logical input/output operations can be performed to the specified physical volume. VGDA and VGSA copies on the physical volume will not be taken into account for any subsequent LVM operations. Also, information regarding the physical volume will be removed from the VGDAs of the other physical volumes within the volume group. The physical volume will be marked as removed.

In Example 6-7 on page 220, we show the concept of being available and its relation with VGDAs as follows:

- The **lsvg testvg** command shows that the VG is active, contains two PVs, both PVs are active, and the VG has three VGDAs.
- The **lsvg -p testvg** command shows that testvg contains disks hdisk2 and hdisk3, and both are active.
- **lspv hdisk3** shows that hdisk3 is active and has two VGDAs.
- **lspv hdisk2** shows that hdisk2 is active and has one VGDA.
- **chpv -vr hdisk3** makes hdisk3 unavailable.
- **lspv hdisk3** confirms that hdisk3 is removed and does not have any VGDAs on it.
- **lspv hdisk2** confirms that hdisk2 is active and now contains two VGDAs, because any volume group must contain at least one VGDA.
- **lsvg -p testvg** shows that hdisk3 has been removed.
- **lsvg testvg** shows that the volume group is still active, one PV of two is active, and the total number of VGDAs has been changed to two.
- **chpv -va hdisk3** makes hdisk3 available again.
- **lspv hdisk3** shows that hdisk3 is active and contains only one VGDA.
- **lsvg -p testvg** confirms that both disks are now active.

Example 6-7 Using the chpv command to change the availability of physical volumes

```
# lsvg testvg
VOLUME GROUP:       testvg                   VG IDENTIFIER:
00c478de00004c00000001078fc3497d
VG STATE:           active                   PP SIZE:        128 megabyte(s)
VG PERMISSION:      read/write               TOTAL PPs:      1092 (139776
megabytes)
MAX LVs:            256                      FREE PPs:       1092 (139776
megabytes)
LVs:                0                        USED PPs:       0 (0 megabytes)
OPEN LVs:           0                        QUORUM:         2
TOTAL PVs:          2                        VG DESCRIPTORS: 3
STALE PVs:          0                        STALE PPs:      0
ACTIVE PVs:         2                        AUTO ON:        yes
MAX PPs per VG:     32512
MAX PPs per PV:     1016                     MAX PVs:        32
LTG size (Dynamic): 256 kilobyte(s)          AUTO SYNC:      no
HOT SPARE:          no                       BB POLICY:      relocatable
# lsvg -p testvg
testvg:
PV_NAME           PV STATE          TOTAL PPs   FREE PPs    FREE DISTRIBUTION
hdisk2            active            546         546
110..109..109..109..109
hdisk3            active            546         546
110..109..109..109..109
# lspv hdisk3
PHYSICAL VOLUME:    hdisk3                   VOLUME GROUP:     testvg
PV IDENTIFIER:      00c478de49630c6a VG IDENTIFIER
00c478de00004c00000001078fc3497d
PV STATE:           active
STALE PARTITIONS:   0                        ALLOCATABLE:      yes
PP SIZE:            128 megabyte(s)          LOGICAL VOLUMES:  0
TOTAL PPs:          546 (69888 megabytes)    VG DESCRIPTORS:   2
FREE PPs:           546 (69888 megabytes)    HOT SPARE:        no
USED PPs:           0 (0 megabytes)          MAX REQUEST:      256 kilobytes
FREE DISTRIBUTION:  110..109..109..109..109
USED DISTRIBUTION:  00..00..00..00..00
# lspv hdisk2
PHYSICAL VOLUME:    hdisk2                   VOLUME GROUP:     testvg
PV IDENTIFIER:      00c478de09caf37f VG IDENTIFIER
00c478de00004c00000001078fc3497d
PV STATE:           active
STALE PARTITIONS:   0                        ALLOCATABLE:      yes
PP SIZE:            128 megabyte(s)          LOGICAL VOLUMES:  0
TOTAL PPs:          546 (69888 megabytes)    VG DESCRIPTORS:   1
FREE PPs:           546 (69888 megabytes)    HOT SPARE:        no
USED PPs:           0 (0 megabytes)          MAX REQUEST:      256 kilobytes
FREE DISTRIBUTION:  110..109..109..109..109
```

```
USED DISTRIBUTION:  00..00..00..00..00
# chpv -vr hdisk3
# lspv hdisk3
PHYSICAL VOLUME:    hdisk3                   VOLUME GROUP:     testvg
PV IDENTIFIER:      00c478de49630c6a VG IDENTIFIER
00c478de00004c00000001078fc3497d
PV STATE:           removed
STALE PARTITIONS:   0                        ALLOCATABLE:      yes
PP SIZE:            128 megabyte(s)          LOGICAL VOLUMES:  0
TOTAL PPs:          546 (69888 megabytes)    VG DESCRIPTORS:   0
FREE PPs:           546 (69888 megabytes)    HOT SPARE:        no
USED PPs:           0 (0 megabytes)          MAX REQUEST:      256 kilobytes
FREE DISTRIBUTION:  110..109..109..109..109
USED DISTRIBUTION:  00..00..00..00..00
# lspv hdisk2
PHYSICAL VOLUME:    hdisk2                   VOLUME GROUP:     testvg
PV IDENTIFIER:      00c478de09caf37f VG IDENTIFIER
00c478de00004c00000001078fc3497d
PV STATE:           active
STALE PARTITIONS:   0                        ALLOCATABLE:      yes
PP SIZE:            128 megabyte(s)          LOGICAL VOLUMES:  0
TOTAL PPs:          546 (69888 megabytes)    VG DESCRIPTORS:   2
FREE PPs:           546 (69888 megabytes)    HOT SPARE:        no
USED PPs:           0 (0 megabytes)          MAX REQUEST:      256 kilobytes
FREE DISTRIBUTION:  110..109..109..109..109
USED DISTRIBUTION:  00..00..00..00..00
# lsvg -p testvg
testvg:
PV_NAME           PV STATE          TOTAL PPs   FREE PPs    FREE DISTRIBUTION
hdisk2            active            546         546
110..109..109..109..109
hdisk3            removed           546         546
110..109..109..109..109
# lsvg testvg
VOLUME GROUP:       testvg                   VG IDENTIFIER:
00c478de00004c00000001078fc3497d
VG STATE:           active                   PP SIZE:        128 megabyte(s)
VG PERMISSION:      read/write               TOTAL PPs:      1092 (139776
megabytes)
MAX LVs:            256                      FREE PPs:       1092 (139776
megabytes)
LVs:                0                        USED PPs:       0 (0 megabytes)
OPEN LVs:           0                        QUORUM:         2
TOTAL PVs:          2                        VG DESCRIPTORS: 2
STALE PVs:          0                        STALE PPs:      0
ACTIVE PVs:         1                        AUTO ON:        yes
MAX PPs per VG:     32512
MAX PPs per PV:     1016                     MAX PVs:        32
LTG size (Dynamic): 256 kilobyte(s)          AUTO SYNC:      no
```

```
HOT SPARE:          no                        BB POLICY:      relocatable
# chpv -va hdisk3
# lspv hdisk3
PHYSICAL VOLUME:    hdisk3                    VOLUME GROUP:     testvg
PV IDENTIFIER:      00c478de49630c6a VG IDENTIFIER
00c478de00004c00000001078fc3497d
PV STATE:           active
STALE PARTITIONS:   0                         ALLOCATABLE:      yes
PP SIZE:            128 megabyte(s)           LOGICAL VOLUMES:  0
TOTAL PPs:          546 (69888 megabytes)     VG DESCRIPTORS:   1
FREE PPs:           546 (69888 megabytes)     HOT SPARE:        no
USED PPs:           0 (0 megabytes)           MAX REQUEST:      256 kilobytes
FREE DISTRIBUTION:  110..109..109..109..109
USED DISTRIBUTION:  00..00..00..00..00
# lsvg -p testvg
testvg:
PV_NAME           PV STATE          TOTAL PPs   FREE PPs    FREE DISTRIBUTION
hdisk2            active            546         546
110..109..109..109..109
hdisk3            active            546         546
110..109..109..109..109
```

Before changing the availability of any physical volume, you have to close any logical volume residing on that disk and ensure that the volume group meets quorum requirements after the disk is removed.

6.2.5 Cleaning the boot record from a physical volume

To clear the boot record located on physical volume hdisk1, use the command:

```
chpv -c hdisk1
```

6.2.6 Declaring a physical volume hot spare

You can use the **chpv** command to define a physical volume as a hot spare. This command also disables the allocation permission for the physical volume. The disk size has to be at least equal with the size of the smallest disk already existing in the volume group.

To define hdisk3 as a hot spare, use the command:

```
chpv -hy hdisk3
```

To remove hdisk3 from the hot spare pool of its volume group, use the command:

```
chpv -hn hdisk3
```

6.2.7 Migrating data from physical volumes

Physical partitions located on a physical volume can be moved to one or more physical volumes contained in the same volume group.

In Example 6-8, we offer an example of migrating data from a physical volume as follows:

- **lsvg -p rootvg** displays all PVs that are contained in rootvg.
- **lsvg -M hdisk1** displays the map of all physical partitions located on hdisk1.
- **lspv -M hdisk5** shows that all partitions of hdisk5 are not allocated.
- **migratepv hdisk1 hdisk5** migrates the data from hdisk1 to hdisk5.
- **lspv -M hdisk1** confirms that hdisk1 has all partitions free.
- **chpv -c hdisk1** clears the boot record from hdisk1.
- **lspv -M hdisk5** confirms that all physical partitions have been migrated to hdisk5.

Example 6-8 Migrating physical partition from one disk to another

```
# lsvg -p rootvg
rootvg:
PV_NAME           PV STATE          TOTAL PPs   FREE PPs    FREE DISTRIBUTION
hdisk0            active            546         523
109..109..87..109..109
hdisk1            active            546         538
109..105..106..109..109
hdisk5            active            546         546
110..109..109..109..109
# lspv -M hdisk1
hdisk1:1        hd5:1:2
hdisk1:2-122
hdisk1:123      fslv00:1
hdisk1:124      fslv00:2
hdisk1:125      fslv00:3
hdisk1:126      fslv00:4
hdisk1:127-219
hdisk1:220      hd4:1
hdisk1:221      hd9var:1
hdisk1:222      hd1:1
hdisk1:223-546
# lspv -M hdisk5
hdisk5:1-546
# migratepv hdisk1 hdisk5
0516-1011 migratepv: Logical volume hd5 is labeled as a boot logical volume.
0516-1246 migratepv: If hd5 is the boot logical volume, please run 'chpv -c
hdisk1'
```

```
        as root user to clear the boot record and avoid a potential boot
        off an old boot image that may reside on the disk from which this
        logical volume is moved/removed.
# lspv -M hdisk1
hdisk1:1-546
# chpv -c hdisk1
# lspv -M hdisk5
hdisk5:1        hd5:1:2
hdisk5:2-110
hdisk5:111      fslv00:1
hdisk5:112      fslv00:2
hdisk5:113      fslv00:3
hdisk5:114      fslv00:4
hdisk5:115-219
hdisk5:220      hd4:1
hdisk5:221      hd9var:1
hdisk5:222      hd1:1
hdisk5:223-546
```

If you migrate data from a physical volume that contains a boot image, you should also update the boot list.

It is possible to migrate only data from partitions that belong to a specific logical volume. To migrate only physical partitions that belong to logical volume testlv from hdisk1 to hdisk5, use the command:

```
migratepv -l testlv hdisk1 hdisk5
```

6.2.8 Migrating partitions

You can move data from one partition located on a physical disk to another physical partition on a different disk.

In Example 6-9 on page 225, we offer an example of migrating data from a physical partition to another as follows:

- **`lspv -M hdisk1`** displays the map of all physical partitions located on hdisk1. Note that the second copy of the logical partition number 1 of logical volume testlv resides on physical partition 115.
- **`lspv -M hdisk5`** shows that all partitions of hdisk5 are not allocated.
- **`migratelp testlv/1/2 hdisk5/123`** migrates the data from the second copy of the logical partition number 1 of logical volume to hdisk5 on physical partition 123.
- **`lsvg -M hdisk1`** displays the map of all physical partitions located on hdisk1. Note that physical partition 115 is free.

- **lspv -M hdisk5** confirms that the second copy of the logical partition number 1 of logical volume testlv now resides on physical partition 123 of hdisk5.

Example 6-9 Migrating a partition to another partition on a different physical volume

```
# lspv -M hdisk1
hdisk1:1        hd5:1:2
hdisk1:2-110
hdisk1:111      fslv00:1
hdisk1:112      fslv00:2
hdisk1:113      fslv00:3
hdisk1:114      fslv00:4
hdisk1:115      testlv:1:2
hdisk1:116-219
hdisk1:220      hd4:1
hdisk1:221      hd9var:1
hdisk1:222      hd1:1
hdisk1:223-546
# lspv -M hdisk5
hdisk5:1-546
# migratelp testlv/1/2 hdisk5/123
migratelp: Mirror copy 2 of logical partition 1 of logical volume
        testlv migrated to physical partition 123 of hdisk5.
# lspv -M hdisk1
hdisk1:1        hd5:1:2
hdisk1:2-110
hdisk1:111      fslv00:1
hdisk1:112      fslv00:2
hdisk1:113      fslv00:3
hdisk1:114      fslv00:4
hdisk1:115-219
hdisk1:220      hd4:1
hdisk1:221      hd9var:1
hdisk1:222      hd1:1
hdisk1:223-546
# lspv -M hdisk5
hdisk5:1-122
hdisk5:123      testlv:1:2
hdisk5:124-546
```

6.2.9 Finding the LTG size

Logical track group (LTG) size is the maximum allowed transfer size for an I/O disk operation.

You can use the **lquerypv** command to find the LTG size for a physical disk, as shown in Example 6-10. A disk can support multiple LTG sizes, which are discovered using an ioctl call.

Example 6-10 Using lquerypv command to find LTG size

```
# lquerypv -M hdisk0
256
```

6.3 Volume groups

When the operating system is installed, one volume group named rootvg is created by default. Additional volume groups can be created on the system using one or more physical volumes that have not been allocated to other volume groups yet and are in an available state. All physical volumes will be divided in physical partitions having the same size. The size of the physical partitions cannot be changed after the volume group is created.

This section discusses the operations that can be performed on volume groups.

6.3.1 Creating a volume group

You can use the **mkvg** command to create volume groups. Each volume group is assigned an volume group identifier (VGID) that will be used internally by LVM commands. For each volume group, two device driver files are created under directory /dev, as shown in Example 6-11. Both files will have the major device number equal to the major number of the volume group.

Example 6-11 Two files under /dev created for each volume group

```
# lsvg
rootvg
testvg
vg1
vg2
# cd /dev
# ls -l|grep vg
crw-rw----   1 root     system       10,  0 Nov 10 11:35 IPL_rootvg
crw-------   1 root     system       10,  0 Nov 15 19:08 __vg10
crw-------   1 root     system      100,  0 Nov 16 10:24 __vg100
crw-------   1 root     system       46,  0 Nov 15 18:48 __vg46
```

```
crw-------   1 root     system       47,  0 Nov 16 10:24 __vg47
crw-rw----   1 root     system       10,  0 Nov 10 11:00 rootvg
crw-rw----   1 root     system       46,  0 Nov 14 11:12 testvg
crw-rw----   1 root     system       47,  0 Nov 16 10:21 vg1
crw-rw----   1 root     system      100,  0 Nov 16 10:21 vg2
```

For each volume group varied on, there is a file under /etc/vg having the name identical to the VGID, as shown in Example 6-12.

Example 6-12 Handle files for volume groups

```
# lsvg -o
vg1
testvg
rootvg
# cd /etc/vg
# ls -l
total 0
-rw-r--r--   1 root     system            0 Nov 15 16:15
vg00C478DE00004C00000001077B1E974A
-rw-rw----   1 root     system            0 Nov 15 15:49
vg00C478DE00004C00000001078FC3497D
-rw-rw----   1 root     system            0 Nov 16 10:37
vg00C478DE00004C000000010799E10D8E
# lsvg rootvg |grep -i identifier
VOLUME GROUP:       rootvg                   VG IDENTIFIER:
00c478de00004c00000001077b1e974a
# lsvg testvg |grep -i identifier
VOLUME GROUP:       testvg                   VG IDENTIFIER:
00c478de00004c00000001078fc3497d
# lsvg vg1 |grep -i identifier
VOLUME GROUP:       vg1                      VG IDENTIFIER:
00c478de00004c000000010799e10d8e
```

In Example 6-13, we use the **mkvg** command to create an original volume group named vg1, with a physical partition size of 64 MB, having major number 99 and using the physical volumes hdisk4.

Example 6-13 Creating an original volume group

```
# mkvg -y vg1 -s64 -V99 hdisk4
vg1
```

In Example 6-14, we tried to create an original volume group that would exceed the limitations of this type of volume group in terms of maximum number of PPs.

Example 6-14 Failing to create an original volume group

```
# lsattr -El hdisk4
PCM             PCM/friend/scsiscsd Path Control Module          False
algorithm       fail_over           Algorithm                    True
dist_err_pcnt   0                   Distributed Error Percentage True
dist_tw_width   50                  Distributed Error Sample Time True
hcheck_interval 0                   Health Check Interval        True
hcheck_mode     nonactive           Health Check Mode            True
max_transfer    0x40000             Maximum TRANSFER Size        True
pvid            none                Physical volume identifier   False
queue_depth     3                   Queue DEPTH                  False
reserve_policy  single_path         Reserve Policy               True
size_in_mb      73400               Size in Megabytes            False
# mkvg -y testvg -s 4 -f hdisk4
0516-1254 mkvg: Changing the PVID in the ODM.
0516-1208 mkvg: Warning, The Physical Partition Size of 4 requires the
        creation of 17501 partitions for hdisk4.  The system limitation is
16256
        physical partitions per disk at a factor value of 16. Specify a larger
        Physical Partition Size or a larger factor value in order create a
        volume group on this disk.
0516-862 mkvg: Unable to create volume group.
```

In Example 6-15, we use the **mkvg** command to force the creation of a big volume group named vg2, with a physical partition size of 128 MB, having major number 101 and using physical volume hdisk6. We created this volume group with auto varyon flag set to no so that it will not be automatically varied on at system reboot.

Example 6-15 Creating a big volume group

```
# mkvg -B -y vg2 -s 128 -f -n -V 101 hdisk6
vg2
```

In Example 6-16 on page 229, we tried to create a big volume group that would exceed the limitations of this type of volume groups in terms of maximum number of PPs.

Example 6-16 Failing to create a big volume group

```
# lsattr -El hdisk4
PCM             PCM/friend/scsiscsd               Path Control Module
False
algorithm       fail_over                         Algorithm
True
dist_err_pcnt   0                                 Distributed Error Percentage
True
dist_tw_width   50                                Distributed Error Sample Time
True
hcheck_interval 0                                 Health Check Interval
True
hcheck_mode     nonactive                         Health Check Mode
True
max_transfer    0x40000                           Maximum TRANSFER Size
True
pvid            none                Physical volume identifier
Falsequeue_depth    3                               Queue DEPTH
False
reserve_policy  single_path                       Reserve Policy
True
size_in_mb      73400                             Size in Megabytes
False
# mkvg -B -y test1vg -s 1 -f hdisk4
0516-1254 mkvg: Changing the PVID in the ODM.
0516-1208 mkvg: Warning, The Physical Partition Size of 1 requires the
        creation of 70006 partitions for hdisk4.  The system limitation is
65024
        physical partitions per disk at a factor value of 64. Specify a larger
        Physical Partition Size or a larger factor value in order create a
        volume group on this disk.
0516-862 mkvg: Unable to create volume group.
```

In Example 6-17, we use the **mkvg** command to force the creation of a scalable volume group named testvg, with a physical partition size of 1 MB, and using physical volumes from hdisk3 to hdisk7.

Example 6-17 Creating a scalable volume group

```
# lsattr -El hdisk3|grep -i size_in_mb
size_in_mb      73400                            Size in Megabytes
False
# lsattr -El hdisk4|grep -i size_in_mb
size_in_mb      73400                            Size in Megabytes
False
# lsattr -El hdisk5|grep -i size_in_mb
size_in_mb      73400                            Size in Megabytes
False
# lsattr -El hdisk6|grep -i size_in_mb
size_in_mb      73400                            Size in Megabytes
False
# lsattr -El hdisk7|grep -i size_in_mb
size_in_mb      73400                            Size in Megabytes
False
# mkvg -S -y testvg -s 1 -f hdisk3 hdisk4 hdisk5 hdisk6 hdisk7
0516-1254 mkvg: Changing the PVID in the ODM.
0516-1254 mkvg: Changing the PVID in the ODM.
0516-1254 mkvg: Changing the PVID in the ODM.
0516-1254 mkvg: Changing the PVID in the ODM.
0516-1254 mkvg: Changing the PVID in the ODM.
testvg
# lsvg testvg
VOLUME GROUP:       testvg                  VG IDENTIFIER:
00c5e9de00004c0000000107a5572082
VG STATE:           active                   PP SIZE:        1 megabyte(s)
VG PERMISSION:      read/write               TOTAL PPs:      349690 (349690
megabytes)
MAX LVs:            256                      FREE PPs:       349690 (349690
megabytes)
LVs:                0                        USED PPs:       0 (0 megabytes)
OPEN LVs:           0                        QUORUM:         3
TOTAL PVs:          5                        VG DESCRIPTORS: 5
STALE PVs:          0                        STALE PPs:      0
ACTIVE PVs:         5                        AUTO ON:        yes
MAX PPs per VG:     524288                   MAX PVs:        1024
LTG size (Dynamic): 256 kilobyte(s)          AUTO SYNC:      no
HOT SPARE:          no                       BB POLICY:      relocatable
```

The **mkvg** command will automatically vary on the newly created volume group by calling the **varyonvg** command.

For volume groups created with the **-I** flag, the LTG size is set to 128, 256, 512, or 1024 KB. The value will be equal to the minimum of the transfer size of disks that are part of the volume group.

By default, in AIX 5L Version 5.3, volume groups created without using the **-I** flag will use a variable LTG size.

6.3.2 Listing information about volume groups

The **lsvg** command displays information about the volume groups currently known to the system.

In Example 6-18, we use the **lsvg** command to display all volume groups that are known to the system, either varied on or not.

Example 6-18 Using lsvg to display all volume groups known to a system

```
# lsvg
rootvg
dumpvg
test2vg
test1vg
```

In Example 6-19, we use the **lsvg -o** command to display all volume groups that are varied on.

Example 6-19 Using lsvg to display all active volume groups

```
# lsvg -o
test1vg
dumpvg
rootvg
```

You can use the **lsvg** command and pass the name of the volume group as an argument to obtain more details about the volume group, as shown in Example 6-20.

Example 6-20 Using lsvg to display details about a specific volume group

```
# lsvg test1vg
VOLUME GROUP:       test1vg                  VG IDENTIFIER:
00c5e9de00004c0000000107a5b596ab
VG STATE:           active                   PP SIZE:        512 megabyte(s)
VG PERMISSION:      read/write               TOTAL PPs:      408 (208896
megabytes)
MAX LVs:            256                      FREE PPs:       398 (203776
megabytes)
LVs:                3                        USED PPs:       10 (5120
megabytes)
OPEN LVs:           0                        QUORUM:         2
TOTAL PVs:          3                        VG DESCRIPTORS: 3
STALE PVs:          0                        STALE PPs:      0
ACTIVE PVs:         3                        AUTO ON:        yes
MAX PPs per VG:     32512
MAX PPs per PV:     1016                     MAX PVs:        32
LTG size (Dynamic): 256 kilobyte(s)          AUTO SYNC:      no
HOT SPARE:          no                       BB POLICY:      relocatable
```

The meanings of the fields in Example 6-20 are:

VOLUME GROUP The name of the volume group.

VG STATE The state of the volume group.

VG PERMISSION Access permission: read-only or read-write.

MAX LVs Maximum number of LVs allowed in the volume group.

OPEN LVs The number of logical volumes currently open.

TOTAL PVs The total number of PVs in the volume group.

STALE PVs The number of PVs that contain stale partitions.

ACTIVE PVs The number of PVs currently active.

MAX PPs per VG Maximum number of PPs allowed in the volume group.

MAX PPs per PV Maximum number of PPs per physical volume.

LTG size LTG size of the volume group.

HOT SPARE Hot spare policy of the volume group.

VG IDENTIFIER The numerical identifier of the volume group.

PP SIZE The size of physical partitions from the volume group.

TOTAL PPs The total number of PPs within the volume group.

FREE PPs The total number of PPs within the volume group that have not yet been allocated to any logical volume.

USED PPs The total number of PPs within the volume group that have been already allocated to logical volumes.

QUORUM The number of physical volumes needed for quorum.

VG DESCRIPTORS The number of VGDAs within the volume group.

STALE PPs The number of stale PPs within the volume group.

AUTO ON Determines if the volume group will automatically vary on after system reboot.

MAX PVs The maximum number of PVs allowed in the volume group.

AUTO SYNC Determines if stale partitions are synchronized automatically.

BB POLICY Bad block relocation policy.

In Example 6-21, we use the **lsvg -l** command to display all logical volumes that are part of rootvg.

Example 6-21 Using lsvg -l to display logical volumes contained in a volume group

```
# lsvg -l rootvg
rootvg:
LV NAME             TYPE       LPs    PPs    PVs  LV STATE      MOUNT POINT
hd5                 boot       1      1      1    closed/syncd  N/A
hd6                 paging     3      3      1    open/syncd    N/A
hd8                 jfs2log    1      1      1    open/syncd    N/A
hd4                 jfs2       1      1      1    open/syncd    /
hd2                 jfs2       9      9      1    open/syncd    /usr
hd9var              jfs2       1      1      1    open/syncd    /var
hd3                 jfs2       1      1      1    open/syncd    /tmp
hd1                 jfs2       1      1      1    open/syncd    /home
hd10opt             jfs2       1      1      1    open/syncd    /opt
fwdump              jfs2       3      3      1    open/syncd
/var/adm/ras/platform
paging00            paging     1      1      1    open/syncd    N/A
fslv00              jfs2       2      2      1    open/syncd    /app01
dumpdev             jfs        3      3      1    closed/syncd  N/A
fslv01              jfs2       8      8      1    open/syncd    /kdb
```

In Example 6-22, we use the **lsvg -p** command to display all physical volumes that are part of the test1vg volume group.

Example 6-22 Using lsvg to display all physical volumes contained in a volume group

```
# lsvg -p test1vg
test1vg:
PV_NAME           PV STATE          TOTAL PPs   FREE PPs    FREE DISTRIBUTION
hdisk5            active            136         129         28..20..27..27..27
hdisk6            active            136         133         28..24..27..27..27
hdisk7            active            136         136         28..27..27..27..27
```

In situations when you investigate LVM metadata corruption, you can use **lsvg -n** command to obtain information about a volume group read from a VGDA located on a specific disk.

6.3.3 Changing volume group characteristics

The following section discusses the tasks required to modify a volume group's characteristics.

Auto varyon flag

The following command changes the volume group testvg to be activated automatically the next time the system is restarted:

```
# chvg -ay newvg
```

The following command changes the volume group testvg to not be activated automatically next time the system is restarted:

```
# chvg -an newvg
```

Quorum

The following commands change the quorum for the volume group testvg. This attribute determines if the volume group will be varied off or not after losing the simple majority of its physical volumes.

To turn off the quorum, use the command:

```
#chvg -Qn testvg
```

To turn on the quorum, use the command:

```
# chvg -Qy testvg
```

Maximum number of physical partitions per physical volume

You can change the maximum number of physical partitions per physical volume, as shown in Example 6-23.

Example 6-23 Changing the maximum number of physical partitions per physical volume

```
# lsvg testvg
VOLUME GROUP:       testvg                   VG IDENTIFIER:
00c5e9de00004c0000000107a58c754e
VG STATE:           active                   PP SIZE:        16 megabyte(s)
VG PERMISSION:      read/write               TOTAL PPs:      4375 (70000
megabytes)
MAX LVs:            256                      FREE PPs:       4375 (70000
megabytes)
LVs:                0                        USED PPs:       0 (0 megabytes)
OPEN LVs:           0                        QUORUM:         2
TOTAL PVs:          1                        VG DESCRIPTORS: 2
STALE PVs:          0                        STALE PPs:      0
ACTIVE PVs:         1                        AUTO ON:        yes
MAX PPs per VG:     32512
MAX PPs per PV:     5080                     MAX PVs:        6
LTG size (Dynamic): 256 kilobyte(s)          AUTO SYNC:      no
HOT SPARE:          no                       BB POLICY:      relocatable
# chvg -t 16 testvg
0516-1164 chvg: Volume group testvg changed.  With given characteristics testvg
        can include upto 1 physical volumes with 16256 physical partitions
each.
# lsvg testvg
VOLUME GROUP:       testvg                   VG IDENTIFIER:
00c5e9de00004c0000000107a58c754e
VG STATE:           active                   PP SIZE:        16 megabyte(s)
VG PERMISSION:      read/write               TOTAL PPs:      4375 (70000
megabytes)
MAX LVs:            256                      FREE PPs:       4375 (70000
megabytes)
LVs:                0                        USED PPs:       0 (0 megabytes)
OPEN LVs:           0                        QUORUM:         2
TOTAL PVs:          1                        VG DESCRIPTORS: 2
STALE PVs:          0                        STALE PPs:      0
ACTIVE PVs:         1                        AUTO ON:        yes
MAX PPs per VG:     32512
MAX PPs per PV:     16256                    MAX PVs:        2
LTG size (Dynamic): 256 kilobyte(s)          AUTO SYNC:      no
HOT SPARE:          no                       BB POLICY:      relocatable
```

Changing a volume group format

You can change the format of an original volume group to either big or scalable. Once the volume group has been converted to a scalable format, it cannot be changed into a different format. Before changing the format of a volume group you must vary off the volume group.

In Example 6-24, we use the **chvg -G** command to change the format of the volume group tttt from original to scalable.

Example 6-24 Changing a volume group to scalable vg format

```
# lsvg tttt
VOLUME GROUP:       tttt                     VG IDENTIFIER:
00c478de00004c0000000107d3af5798
VG STATE:           active                   PP SIZE:        128 megabyte(s)
VG PERMISSION:      read/write               TOTAL PPs:      546 (69888
megabytes)
MAX LVs:            256                      FREE PPs:       546 (69888
megabytes)
LVs:                0                        USED PPs:       0 (0 megabytes)
OPEN LVs:           0                        QUORUM:         2
TOTAL PVs:          1                        VG DESCRIPTORS: 2
STALE PVs:          0                        STALE PPs:      0
ACTIVE PVs:         1                        AUTO ON:        no
MAX PPs per VG:     32512
MAX PPs per PV:     1016                     MAX PVs:        32
LTG size (Dynamic): 256 kilobyte(s)          AUTO SYNC:      no
HOT SPARE:          no                       BB POLICY:      relocatable
# varyoffvg tttt
# chvg -G tttt
0516-1224 chvg: WARNING, once this operation is completed, volume group tttt
        cannot be imported into AIX 5.2 or lower versions. Continue (y/n) ?
y
0516-1712 chvg: Volume group tttt changed.  tttt can include up to 1024
physical volumes with 2097152 total physical partitions in the volume group.
# varyonvg tttt
# lsvg tttt
VOLUME GROUP:       tttt                     VG IDENTIFIER:
00c478de00004c0000000107d3af5798
VG STATE:           active                   PP SIZE:        128 megabyte(s)
VG PERMISSION:      read/write               TOTAL PPs:      545 (69760
megabytes)
MAX LVs:            256                      FREE PPs:       545 (69760
megabytes)
LVs:                0                        USED PPs:       0 (0 megabytes)
OPEN LVs:           0                        QUORUM:         2
TOTAL PVs:          1                        VG DESCRIPTORS: 2
STALE PVs:          0                        STALE PPs:      0
ACTIVE PVs:         1                        AUTO ON:        no
```

```
MAX PPs per VG:     32768                    MAX PVs:        1024
LTG size (Dynamic): 256 kilobyte(s)          AUTO SYNC:      no
HOT SPARE:          no                       BB POLICY:      relocatable
```

As shown, the maximum number of physical partitions is no longer defined on a per disk basis, but rather applies to the entire volume group. As a consequence, the **lsvg** command will no longer display the maximum number of physical volumes per volume group for scalable volume groups.

Changing LTG size

By default, volume groups in AIX 5L Version 5.3 are created with a variable logical track group size. For volume groups created to be compatible with a previous version of AIX 5L, you can change the LTG size to 0, 128, 256, 512, or 1024. The new LTG size should be less than or equal to the smallest of the maximum transfer size of all disks in the volume group. You can change the LTG size for the testvg volume group using the following command:

```
chvg -L 128 testvg
```

Changing the hot spare policy

To improve data availability, one or more disks from a volume group can be designated as hot spares. Physical volumes that are to be used as a hot spare must have all physical partitions free. All logical volumes from the volume group that contain hot spare disks must be mirrored. Information corresponding to a physical partition located on a failing disk will be copied from its mirror copy to one or more disks from the hot spare pool according to the hot spare policy of the volume group.

In Example 6-25 on page 238, we offer an example of implementing a hot spare policy mechanism as follows:

- **lsvg -p test1vg** displays physical volumes that are part of test1vg.
- **chpv -hy hdisk5** tries to designate hdisk5 as a hot spare, but fails because hdisk5 contains physical partitions that are allocated.
- **chpv -hy hdisk4** designates hdisk4 as a hot spare.
- **lspv hdisk4** shows that hdisk4 does not contain any allocated physical partitions. The allocatable permission for this physical volume is set to no.
- **chvg -hy test1vg** changes the hot spare policy of the volume group to migrate data from a failing disk to one spare disk, as confirmed by **lsvg test1vg**.
- **chvg -hY test1vg** changes the hot spare policy of the volume group to migrate data from a failing disk to the entire pool of spare disks, as confirmed by **lsvg test1vg**.

- **`chvg -hn test1vg`** disables the hot spare policy of the volume group, as confirmed by **`lsvg test1vg`**.

Example 6-25 Changing the hot spare policy of a volume group.

```
# lsvg -p test1vg
test1vg:
PV_NAME          PV STATE          TOTAL PPs   FREE PPs    FREE DISTRIBUTION
hdisk5           active            136         129         28..20..27..27..27
hdisk6           active            136         133         28..24..27..27..27
hdisk4           active            136         136         28..27..27..27..27
# chpv -hy hdisk5
0516-1302 chpv: Cannot make the hdisk5 as a hot spare disk because some of the
physical partitions
        of hdisk5 are allocated.
# chpv -hy hdisk4
# lspv hdisk4
PHYSICAL VOLUME:    hdisk4                   VOLUME GROUP:     test1vg
PV IDENTIFIER:      00c5e9deb9bd80c3 VG IDENTIFIER
00c5e9de00004c0000000107a5b596ab
PV STATE:           active
STALE PARTITIONS:   0                        ALLOCATABLE:      no
PP SIZE:            512 megabyte(s)          LOGICAL VOLUMES:  0
TOTAL PPs:          136 (69632 megabytes)    VG DESCRIPTORS:   1
FREE PPs:           136 (69632 megabytes)    HOT SPARE:        yes
USED PPs:           0 (0 megabytes)          MAX REQUEST:      256 kilobytes
FREE DISTRIBUTION:  28..27..27..27..27
USED DISTRIBUTION:  00..00..00..00..00
# chvg -hy test1vg
# lsvg test1vg
VOLUME GROUP:       test1vg                  VG IDENTIFIER:
00c5e9de00004c0000000107a5b596ab
VG STATE:           active                   PP SIZE:        512 megabyte(s)
VG PERMISSION:      read/write               TOTAL PPs:      408 (208896
megabytes)
MAX LVs:            256                      FREE PPs:       398 (203776
megabytes)
LVs:                3                        USED PPs:       10 (5120
megabytes)
OPEN LVs:           0                        QUORUM:         2
TOTAL PVs:          3                        VG DESCRIPTORS: 3
STALE PVs:          0                        STALE PPs:      0
ACTIVE PVs:         3                        AUTO ON:        yes
MAX PPs per VG:     32512
MAX PPs per PV:     1016                     MAX PVs:        32
LTG size (Dynamic): 256 kilobyte(s)          AUTO SYNC:      yes
HOT SPARE:          yes (one to one)         BB POLICY:      relocatable
# chvg -hY test1vg
# lsvg test1vg
```

```
VOLUME GROUP:       test1vg                  VG IDENTIFIER:
00c5e9de00004c0000000107a5b596ab
VG STATE:           active                   PP SIZE:        512 megabyte(s)
VG PERMISSION:      read/write               TOTAL PPs:      408 (208896
megabytes)
MAX LVs:            256                      FREE PPs:       398 (203776
megabytes)
LVs:                3                        USED PPs:       10 (5120
megabytes)
OPEN LVs:           0                        QUORUM:         2
TOTAL PVs:          3                        VG DESCRIPTORS: 3
STALE PVs:          0                        STALE PPs:      0
ACTIVE PVs:         3                        AUTO ON:        yes
MAX PPs per VG:     32512
MAX PPs per PV:     1016                     MAX PVs:        32
LTG size (Dynamic): 256 kilobyte(s)          AUTO SYNC:      no
HOT SPARE:          yes (one to many)        BB POLICY:      relocatable
# chvg -hn test1vg
# lsvg test1vg
VOLUME GROUP:       test1vg                  VG IDENTIFIER:
00c5e9de00004c0000000107a5b596ab
VG STATE:           active                   PP SIZE:        512 megabyte(s)
VG PERMISSION:      read/write               TOTAL PPs:      408 (208896
megabytes)
MAX LVs:            256                      FREE PPs:       398 (203776
megabytes)
LVs:                3                        USED PPs:       10 (5120
megabytes)
OPEN LVs:           0                        QUORUM:         2
TOTAL PVs:          3                        VG DESCRIPTORS: 3
STALE PVs:          0                        STALE PPs:      0
ACTIVE PVs:         3                        AUTO ON:        yes
MAX PPs per VG:     32512
MAX PPs per PV:     1016                     MAX PVs:        32
LTG size (Dynamic): 256 kilobyte(s)          AUTO SYNC:      no
HOT SPARE:          no                       BB POLICY:      relocatable
```

Changing the synchronization policy

Synchronization policy controls automatic synchronization of stale partitions within the volume group. This flag has significance only for partitions that correspond to mirrored logical volumes.

In Example 6-26, we use the **chvg -s** command to change the synchronization policy. This is confirmed by the **lsvg test1vg** command.

Example 6-26 Changing synchronization policy of a volume group

```
# chvg -sy test1vg
# lsvg test1vg
VOLUME GROUP:       test1vg                  VG IDENTIFIER:
00c5e9de00004c0000000107a5b596ab
VG STATE:           active                   PP SIZE:        512 megabyte(s)
VG PERMISSION:      read/write               TOTAL PPs:      408 (208896
megabytes)
MAX LVs:            256                      FREE PPs:       398 (203776
megabytes)
LVs:                3                        USED PPs:       10 (5120
megabytes)
OPEN LVs:           0                        QUORUM:         2
TOTAL PVs:          3                        VG DESCRIPTORS: 3
STALE PVs:          0                        STALE PPs:      0
ACTIVE PVs:         3                        AUTO ON:        yes
MAX PPs per VG:     32512
MAX PPs per PV:     1016                     MAX PVs:        32
LTG size (Dynamic): 256 kilobyte(s)          AUTO SYNC:      yes
HOT SPARE:          yes (one to many)        BB POLICY:      relocatable
```

Changing the maximum number of physical partitions

In Example 6-27, we use the **chvg -P** command to change the maximum number of physical partitions within a volume group. This is confirmed with the **lsvg testvg** command.

Example 6-27 Changing the maximum number of physical partitions

```
# lsvg testvg
VOLUME GROUP:       testvg                   VG IDENTIFIER:
00c5e9de00004c00000
00107a58c754e
VG STATE:           active                   PP SIZE:        16 megabyte(s)
VG PERMISSION:      read/write               TOTAL PPs:      4370 (69920
megabytes)
MAX LVs:            256                      FREE PPs:       4370 (69920
megabytes)
LVs:                0                        USED PPs:       0 (0 megabytes)
OPEN LVs:           0                        QUORUM:         2
TOTAL PVs:          1                        VG DESCRIPTORS: 2
STALE PVs:          0                        STALE PPs:      0
ACTIVE PVs:         1                        AUTO ON:        yes
MAX PPs per VG:     32768                    MAX PVs:        1024
LTG size (Dynamic): 256 kilobyte(s)          AUTO SYNC:      no
```

```
HOT SPARE:          no                        BB POLICY:      relocatable
# chvg -P 2048 testvg
# lsvg testvg
VOLUME GROUP:       testvg                    VG IDENTIFIER:
00c5e9de00004c0000000107a58c754e
VG STATE:           active                    PP SIZE:        16 megabyte(s)
VG PERMISSION:      read/write                TOTAL PPs:      4370 (69920
megabytes)
MAX LVs:            256                       FREE PPs:       4370 (69920
megabytes)
LVs:                0                         USED PPs:       0 (0 megabytes)
OPEN LVs:           0                         QUORUM:         2
TOTAL PVs:          1                         VG DESCRIPTORS: 2
STALE PVs:          0                         STALE PPs:      0
ACTIVE PVs:         1                         AUTO ON:        yes
MAX PPs per VG:     2097152                   MAX PVs:        1024
LTG size (Dynamic): 256 kilobyte(s)           AUTO SYNC:      no
HOT SPARE:          no                        BB POLICY:      relocatable
```

Changing the maximum number of logical volumes

In Example 6-28, we used the **chvg -v** command to change the maximum number of logical volumes within a volume group. This is confirmed by **lsvg testvg**.

Example 6-28 Changing the maximum number of logical volumes

```
# lsvg testvg
VOLUME GROUP:       testvg                    VG IDENTIFIER:
00c5e9de00004c0000000107a58c754e
VG STATE:           active                    PP SIZE:        16 megabyte(s)
VG PERMISSION:      read/write                TOTAL PPs:      4370 (69920
megabytes)
MAX LVs:            256                       FREE PPs:       4370 (69920
megabytes)
LVs:                0                         USED PPs:       0 (0 megabytes)
OPEN LVs:           0                         QUORUM:         2
TOTAL PVs:          1                         VG DESCRIPTORS: 2
STALE PVs:          0                         STALE PPs:      0
ACTIVE PVs:         1                         AUTO ON:        yes
MAX PPs per VG:     2097152                   MAX PVs:        1024
LTG size (Dynamic): 256 kilobyte(s)           AUTO SYNC:      no
HOT SPARE:          no                        BB POLICY:      relocatable
# chvg -v 4096 testvg
# lsvg testvg
VOLUME GROUP:       testvg                    VG IDENTIFIER:
00c5e9de00004c0000000107a58c754e
VG STATE:           active                    PP SIZE:        16 megabyte(s)
```

```
VG PERMISSION:      read/write               TOTAL PPs:      4370 (69920
megabytes)
MAX LVs:            4096                     FREE PPs:       4370 (69920
megabytes)
LVs:                0                        USED PPs:       0 (0 megabytes)
OPEN LVs:           0                        QUORUM:         2
TOTAL PVs:          1                        VG DESCRIPTORS: 2
STALE PVs:          0                        STALE PPs:      0
ACTIVE PVs:         1                        AUTO ON:        yes
MAX PPs per VG:     2097152                  MAX PVs:        1024
LTG size (Dynamic): 256 kilobyte(s)          AUTO SYNC:      no
HOT SPARE:          no                       BB POLICY:      relocatable
```

6.3.4 Unlocking a volume group

A volume group can become locked after an abnormal termination of an LVM command. You can remove the lock using the **chvg -u** command.

6.3.5 Extending a volume group

You can increase the space available in a volume group by adding new physical volumes using the **extendvg** command. Before adding a new disk, you have to ensure that the disk is in an available state.

The system queries the disk to be added to discover if it already contains an VGDA. If the disk has one VGDA corresponding to another already varied on volume group, the command exits. If the VGDA belongs to a volume group that is varied off, the system will prompt the user for confirmation in continuing with command execution. If the user says yes, the old VGDA is erased and all previous data on that disk will be unavailable.

For volume groups created prior to AIX 5L Version 5.3 or for volume groups created on AIX 5L Version 5.3 but varied on with the **varyonvg -M** command, the **extendvg** will fail if the physical volume has a maximum transfer size smaller than the LTG of the volume group. For volume groups created on AIX 5L Version 5.3 and varied on without the **varyonvg -M** command, **extendvg** will dynamically lower the LTG of the VG if the physical volume has a maximum transfer size smaller than the LTG of the VG.

In Example 6-29 on page 243, we showed how the extendvg command works by adding hdisk4 and hdisk7 to test1vg as follows:

- **lspv** shows that hdisk6 has already an PVID, while hdisk7 is clear.
- **extendvg test1vg hdisk7** assigns an PVID to hdisk7 and adds it to the volume group test1vg, as confirmed by **lsvg -p testvg**.

- **`extendvg test1vg hdisk4`** senses that hdisk4 appears to belong to a volume that is not varied on and asks the user use the force flag.
- **`extendvg -f test1vg hdisk4`** forcibly adds hdisk4 to volume group test1vg, as confirmed by **`lsvg -p test1vg`**.
- **`extendvg -f test1vg hdisk1`** tries to forcibly add group physical volume hdisk1, belonging to volume group rootvg, to test1vg volume, and fails.

Example 6-29 Using the extendvg command to add disks to a volume group

```
# lspv
hdisk0          00c5e9de00091d6f                    rootvg          active
hdisk1          00c5e9de00838438                    rootvg          active
hdisk2          00c5e9de0083864d                    dumpvg          active
hdisk3          00c5e9dea557184b                    test2vg
hdisk4          00c5e9deb9bd80c3                    None
hdisk5          00c5e9dea5571a32                    test1vg         active
hdisk6          00c5e9deb8f111ed                    test1vg         active
hdisk7          none                                None
# extendvg test1vg hdisk7
0516-1254 extendvg: Changing the PVID in the ODM.
# lsvg -p test1vg
test1vg:
PV_NAME           PV STATE          TOTAL PPs   FREE PPs    FREE DISTRIBUTION
hdisk5            active            136         129         28..20..27..27..27
hdisk6            active            136         133         28..24..27..27..27
hdisk7            active            136         136         28..27..27..27..27
# extendvg test1vg hdisk4
0516-1398 extendvg: The physical volume hdisk4, appears to belong to
another volume group. Use the force option to add this physical volume
to a volume group.
0516-792 extendvg: Unable to extend volume group.
# extendvg -f test1vg hdisk4
# lsvg -p test1vg
test1vg:
PV_NAME           PV STATE          TOTAL PPs   FREE PPs    FREE DISTRIBUTION
hdisk5            active            136         129         28..20..27..27..27
hdisk6            active            136         133         28..24..27..27..27
hdisk7            active            136         136         28..27..27..27..27
hdisk4            active            136         136         28..27..27..27..27
# extendvg -f test1vg hdisk1
0516-029 extendvg: The Physical Volume is a member of a currently
        varied on Volume Group and this cannot be overidden.
0516-1397 extendvg: The physical volume hdisk1, will not be added to
the volume group.
0516-792 extendvg: Unable to extend volume group.
```

6.3.6 Reducing a volume group

You can remove a physical disk from a volume group using the **reducevg** command. The volume group must be varied on. When you remove the last physical volume from the volume group, the VG will also be removed. For volume groups created on AIX 5L Version 5.3 and varied on without using **varyonvg -M**, **reducevg** will dynamically raise the LTG size if the remaining disks permit it. All logical volumes residing on the disk to be reduced have to be closed before. If the logical volumes on the physical volume specified to be removed also span other physical volumes in the volume group, the removal operation may destroy the integrity of those logical volumes, regardless of the physical volume on which they reside.

In Example 6-30, we show how the **reducevg** command works by deleting hdisk7 from testvg as follows:

- **lsvg -p testvg** shows that testvg contains physical volumes hdisk6 and hdisk7.
- **lsvg -l testvg** shows that testvg contains two opened mirrored logical volumes.
- **lslv -l loglv01** shows that the log of this volume group is located on hdisk6.
- **reducevg testvg hdisk7** tries to delete physical volume hdisk7, but fails because the physical volume contains opened logical volumes.
- **reducevg -d testvg hdisk7** tries to forcibly delete physical volume hdisk7, but fails because the physical volume contains opened logical volumes.
- We close logical volumes lv1, lv2, and loglv01 by unmounting the corresponding file systems
- **reducevg testvg hdisk7** still does not work.
- **reducevg -f testvg hdisk7** prompts the user for confirmation, deletes the data located on physical volume hdisk7, and removes the disk definition from the testvg volume group.

Example 6-30 Using the reducevg command

```
# lsvg -p testvg
testvg:
PV_NAME           PV STATE          TOTAL PPs    FREE PPs    FREE DISTRIBUTION
hdisk6            active            546          541
110..104..109..109..109
hdisk7            active            546          542
110..105..109..109..109
# lsvg -l testvg
testvg:
LV NAME             TYPE       LPs    PPs    PVs  LV STATE      MOUNT POINT
```

```
lv1                 jfs2       2     4     2    open/syncd    /fs1
lv2                 jfs2       2     4     2    open/syncd    /fs2
loglv01             jfs2log    1     1     1    open/syncd    N/A
# lslv -l loglv01
loglv01:N/A
PV                COPIES        IN BAND       DISTRIBUTION
hdisk6            001:000:000   100%          000:001:000:000:000
# reducevg testvg hdisk7
0516-016 ldeletepv: Cannot delete physical volume with allocated
        partitions. Use either migratepv to move the partitions or
        reducevg with the -d option to delete the partitions.
0516-884 reducevg: Unable to remove physical volume hdisk7.
# reducevg -d testvg hdisk7
0516-914 rmlv: Warning, all data belonging to logical volume
        lv1 on physical volume hdisk7 will be destroyed.
rmlv: Do you wish to continue? y(es) n(o)? y
0516-1008 rmlv: Logical volume lv1 must be closed.  If the logical
        volume contains a filesystem, the umount command will close
        the LV device.
0516-884 reducevg: Unable to remove physical volume hdisk7.
# umount /fs1
# umount /fs2
# reducevg testvg hdisk7
0516-016 ldeletepv: Cannot delete physical volume with allocated
        partitions. Use either migratepv to move the partitions or
        reducevg with the -d option to delete the partitions.
0516-884 reducevg: Unable to remove physical volume hdisk7.
# reducevg -d testvg hdisk7
0516-914 rmlv: Warning, all data belonging to logical volume
        lv1 on physical volume hdisk7 will be destroyed.
rmlv: Do you wish to continue? y(es) n(o)? y
0516-914 rmlv: Warning, all data belonging to logical volume
        lv2 on physical volume hdisk7 will be destroyed.
rmlv: Do you wish to continue? y(es) n(o)? y
# lsvg -p testvg
testvg:
PV_NAME           PV STATE          TOTAL PPs   FREE PPs    FREE DISTRIBUTION
hdisk6            active            546         541
110..104..109..109..109
```

6.3.7 Resynchronizing the device configuration database

During normal operations, the information about volume groups from ODM remains consistent with the LVM information contained in metadata like LVCBs and VGDAs. If the ODM becomes corrupted, you first need to understand the circumstances of the corruption before trying any corrective action.

The **synclvodm** command can be used to synchronize or rebuild information from ODM, device files, and LVM metadata structures, such as VGDA or LVCB. Items that will be synchronized or rebuilt include:

- The list of physical volumes that are part of the volume group
- The name, major number, and characteristics of the volume group
- VGDAs for the volume group
- Timestamps
- Name and characteristics of logical volumes
- Device special files corresponding to logical volumes and volume group
- Stanzas from ODM referring to the volume group, logical and physical volumes

> **Note:** The **synclvodm** command does not resolve all corruption scenarios and is not intended to be a substitute for problem determination and solving techniques. Using the **synclvodm** command improperly can make things worse.

The volume groups for which the system has inaccurate information must be active for the resynchronization to occur.

To synchronize ODM to contain the latest LVM information for volume group testvg, use the following command:

```
# synclvodm testvg
```

6.3.8 Exporting a volume group

There are situations when all data from a volume group needs to be moved from one system to another system. You will need to delete any reference to that data from the originating system

The **exportvg** command only removes volume group definition from the ODM and does not delete any data from the physical disks. It clears the stanzas from /etc/filesystem that correspond to the logical volumes contained in the exported volume group, but it will not delete the mounting point. You cannot export a volume group that contains an active paging space.

The mount point information of a logical volume will be missing from the LVCB if it is longer than 128 characters.

To export the volume group testvg, use the command:

```
# exportvg testvg
```

6.3.9 Importing a volume group

Importing a volume group means recreating the reference to the volume group data and making that data available.

The following example shows the import volume group testvg using hdisk7:

```
# importvg -y testvg hdisk7
```

The **importvg** command reads the VGDA of one the physical volumes that are part of the volume group. It uses **redefinevg** to find all other disks that belong to the volume group. It will add corresponding entries into the ODM database and update /etc/filesystems with the new values (if possible) for the new logical volumes and their corresponding mount points.

If the specified volume group name is already in use, the **importvg** command will fail, because duplicate volume group names are not allowed.

It is possible that some of the imported logical volume names may conflict with those already existing on the system. The **importvg** command will automatically assign system default names to those that have been imported and send an error message.

When you import a classical concurrent volume group, you will have to change the type of the volume group to enhanced concurrent.

Example 6-31 on page 248 provides an example of how the **importvg** command works as follows:

- **lsvg -l test2vg** shows that the volume group named test2vg contains a logical volume named loglv0 of type jfslog, and a logical volume named test1lv of type jfs and has mounting point /testmp.
- There is a stanza in /etc/filesystems for /testmp.
- On a different system, we created another volume group containing one physical volume and the same logical volumes as volume group test2vg. We exported the volume group from that system and attached the disk drive to our system. The disk drive is seen as hdisk5.
- **importvg -y test2vg hdisk5** tries to import the volume group from hdisk5 using the name test2vg, which is already allocated and fails.
- **importvg -y test1vg hdisk5** tries to import the volume group from hdisk5 under the new name test1vg and the operation completes successfully. However, the names of logical volumes that reside on hdisk5 are changed to names that are system generated. The **importvg** command also informs us that mounting point corresponding to logical volume test1lv from hdisk5 already existed at the time of import.

- **lsvg -l test1vg** shows the new names for two logical volumes imported. Logical volume fslv02 needs to have a new mounting point defined.

Example 6-31 Importing a volume group

```
# lsvg -l test2vg
test2vg:
LV NAME             TYPE       LPs    PPs    PVs  LV STATE      MOUNT POINT
test1lv             jfs        2      2      1    closed/syncd  /testmp
loglv00             jfslog     1      1      1    closed/syncd  N/A
# cat /etc/filesystems|grep -ip test
/testmp:
        dev             = /dev/test1lv
        vfs             = jfs
        log             = /dev/loglv00
        mount           = false
        options         = rw
        account         = false
# importvg -y test2vg hdisk5
0516-360 getvgname: The device name is already used; choose a
        different name.
0516-776 importvg: Cannot import hdisk5 as test2vg.
# importvg -y test1vg hdisk5
0516-530 synclvodm: Logical volume name test1lv changed to fslv02.
0516-530 synclvodm: Logical volume name loglv00 changed to loglv01.
imfs: Warning: mount point /testmp already exists in /etc/filesystems.
test1vg
# lsvg -l test1vg
test1vg:
LV NAME             TYPE       LPs    PPs    PVs  LV STATE      MOUNT POINT
fslv02              jfs2       2      2      1    closed/syncd  N/A
loglv01             jfs2log    1      1      1    closed/syncd  N/A
```

An imported volume group is automatically varied on, unless it is concurrent capable.

You should run the **fsck** command before mounting the file systems.

6.3.10 Varying on a volume group

An already defined volume group can be activated using the **varyonvg** command and made available for use. This process involves the following steps:

- The **varyonvg** command will open the corresponding file from /etc/vg to obtain a lock for the volume group.

- Timestamps from each VGDA are read to make sure that information from VGDA is not stale. The latest consistent copy of the VGDA is selected to be further used as a reference point.
- If a majority of physical volumes are not accessible, the varyon fails. You will need to forcibly varyon the volume group in order to activate it.
- The LVM metadata structures from all physical volumes are updated with latest information about all physical volumes' status.
- All physical volumes are updated to contain the latest consistent copy of the VGDA.
- The LVM device driver is updated to contain the latest information about the volume group.
- The **syncvg** command is called to synchronize stale partitions, if any.

You can use the **varyonvg -f** command to forcibly vary on a volume group that cannot be activated normally. This action does not guarantee data integrity and should be done only in emergency cases.

You can use the **varyonvg -n** command to vary on a volume group, without synchronizing stale partitions automatically. This flag can be very useful when you encountered disk problems and want to control the synchronization to be sure that you have the correct data available.

In Example 6-32 on page 250, we use the **varyonvg** command to activate the test1vg volume group as follows:

- **lsvg testvg** shows that all three physical volumes of testvg are active and that quorum is active.
- We physically remove hdisk7 from the system.
- The **varyoffvg testvg** and **varyonvg testvg** commands force the system to verify the availability of the physical volumes. hdisk7 is declared as missing, as confirmed by **lsvg -p testvg**.
- We disable the quorum for testvg using **chvg -Qn testvg** and varyoff the volume group testvg.
- **varyonvg testvg** fails to activate the volume group because one of the physical volumes hdisk7 is missing.
- **varyonvg -f testvg** forcibly activates the volume group and declares hdisk7 missing.
- **chvg -Qy testvg** activates the quorum.
- We physically remove hdisk6 from the system and varyoff volume group testvg.

- **`varyonvg testvg`** fails to activate the volume group, since there are not enough active physical volumes to meet the quorum.
- **`varyonvg -f testvg`** forcibly activates the volume group and puts hdisk6 and hdisk7 in the removed status.

Example 6-32 Using the varyonvg command

```
# lsvg testvg
VOLUME GROUP:       testvg                   VG IDENTIFIER:
00c5e9de00004c0000000107d47002be
VG STATE:           active                   PP SIZE:        128 megabyte(s)
VG PERMISSION:      read/write               TOTAL PPs:      1638 (209664
megabytes)
MAX LVs:            256                      FREE PPs:       1638 (209664
megabytes)
LVs:                0                        USED PPs:       0 (0 megabytes)
OPEN LVs:           0                        QUORUM:         2
TOTAL PVs:          3                        VG DESCRIPTORS: 3
STALE PVs:          0                        STALE PPs:      0
ACTIVE PVs:         3                        AUTO ON:        yes
MAX PPs per VG:     32512
MAX PPs per PV:     1016                     MAX PVs:        32
LTG size (Dynamic): 256 kilobyte(s)          AUTO SYNC:      no
HOT SPARE:          no                       BB POLICY:      relocatable
# varyoffvg testvg
# varyonvg testvg
PV Status:      hdisk5  00c5e9dea5571a32       PVACTIVE
                hdisk6  00c5e9deb8f111ed       PVACTIVE
                hdisk7  00c5e9deb9bda044       PVMISSING
varyonvg: Volume group testvg is varied on.
# lsvg -p testvg
testvg:
PV_NAME           PV STATE          TOTAL PPs   FREE PPs    FREE DISTRIBUTION
hdisk5            active            546         546
110..109..109..109..109
hdisk6            active            546         546
110..109..109..109..109
hdisk7            missing           546         546
110..109..109..109..109
# chvg -Qn testvg
# varyoffvg testvg
# varyonvg testvg
0516-056 varyonvg: The volume group is not varied on because a
        physical volume is marked missing. Run diagnostics.
# varyonvg -f testvg
PV Status:      hdisk5  00c5e9dea5571a32       PVACTIVE
                hdisk6  00c5e9deb8f111ed       PVACTIVE
                hdisk7  00c5e9deb9bda044       PVMISSING
varyonvg: Volume group testvg is varied on.
```

```
# chvg -Qy testvg
# varyoffvg testvg
# varyonvg testvg
0516-052 varyonvg: Volume group cannot be varied on without a
        quorum. More physical volumes in the group must be active.
        Run diagnostics on inactive PVs.
# varyonvg -f testvg
PV Status:      hdisk5  00c5e9dea5571a32        PVACTIVE
                hdisk6  00c5e9deb8f111ed        PVREMOVED
                hdisk7  00c5e9deb9bda044        PVREMOVED
varyonvg: Volume group testvg is varied on.
```

Varying off a volume group

The **varyoffvg** command will deactivate a volume group and logical volumes within it. All logical volumes must be closed, which requires that all file systems associated with logical volumes be unmounted.

In Example 6-33, we use the **varyoffvg** command to deactivate volume group test1vg, which initially failed because logical volumes were opened.

Example 6-33 Using varyoffvg command

```
# varyoffvg test1vg
0516-012 lvaryoffvg: Logical volume must be closed.  If the logical
        volume contains a filesystem, the umount command will close
        the LV device.
0516-942 varyoffvg: Unable to vary off volume group test1vg.
# lsvg -l test1vg
test1vg:
LV NAME             TYPE       LPs    PPs    PVs  LV STATE      MOUNT POINT
lv2                 jfs2       2      2      1    open/syncd    /fs2
logggg              jfs2log    2      2      1    open/syncd    N/A
# umount /fs2
# varyoffvg test1vg
# lsvg -o
dumpvg
test2vg
rootvg
```

6.3.11 Reorganizing a volume group

The **reorgvg** command is used to reorganize physical partitions within a volume group. The physical partitions will be rearranged on the disks according to the intra-physical and inter-physical policy allocation for each logical volume.

For this command to work, the volume group must have at least one free partition and a relocatable flag of each of the logical volumes that you would like to organize must be set.

In Example 6-34, the effects of the **reorgvg** command are as follows:

- **lslv -l lv1** shows that the percentage of logical partitions that belong to logical volume lv1 and are compliant with intra-physical allocation policies of physical volumes hdisk6, hdisk5, and hdisk7.
- After running the **reorgvg** command, the percentage of logical partitions compliant with intra-physical allocation policies of physical volumes hdisk6, hdisk5, and hdisk7 has increased to 100%.

Example 6-34 Using the reorgvg command

```
# lslv -l lv1
lv1:/fs1
PV                  COPIES         IN BAND        DISTRIBUTION
hdisk6              003:001:000    50%            002:002:000:000:000
hdisk5              003:000:000    33%            001:001:001:000:000
hdisk7              002:000:000    0%             001:000:000:000:001
# reorgvg test1vg
# lslv -l lv1
lv1:/fs1
PV                  COPIES         IN BAND        DISTRIBUTION
hdisk5              003:000:000    100%           000:003:000:000:000
hdisk6              003:000:000    100%           000:003:000:000:000
hdisk7              003:000:000    100%           000:003:000:000:000
```

To reorganize only logical volumes lv1 and lv1 from volume group testvg, use:

```
reorgvg testvg lv1 lv2
```

To reorganize only partitions located on physical volumes hdisk6 and hdisk7 that belong to logical volumes lv1 and lv2 from volume group testvg, use:

```
echo "hdisk6 hdisk7" | reorgvg -i testvg lv1 lv2
```

6.3.12 Synchronizing a volume group

The **syncvg** command is used to synchronize stale physical partitions. It accepts names of logical volumes, physical volumes, or volume groups as parameters. The synchronization process can be time consuming, depending on the hardware characteristics and the total amount of data.

When the -f flag is used, synchronization is forced and an uncorrupted physical copy is chosen and propagated to all other copies of the logical partition, whether or not they are stale.

To synchronize the copies located on physical volumes hdisk4 and hdisk5, use:

```
syncvg -p hdisk4 hdisk5
```

To synchronize the all physical partitions from volume group testvg, use:

```
syncvg -v testvg
```

6.3.13 Mirroring a volume group

You can use the **mirrorvg** command to mirror all logical volumes within a volume group.

In Example 6-35, after we extended rootvg to contain a second physical volume, we used the **mirrorvg** command to create a copy for each logical volume within rootvg. Notice that the rootvg volume group has all logical volumes mirrored and that quorum has been disabled. Also, a new BLV was created on the newly added disk, the disk was included in the bootlist, and the system rebooted.

Example 6-35 Using the mirrorvg command to mirror rootvg volume group

```
# lsvg -p rootvg
rootvg:
PV_NAME           PV STATE          TOTAL PPs   FREE PPs    FREE DISTRIBUTION
hdisk0            active            546         526
109..105..94..109..109
# extendvg rootvg hdisk1
0516-1254 extendvg: Changing the PVID in the ODM.
# mirrorvg rootvg
0516-1124 mirrorvg: Quorum requirement turned off, reboot system for this
        to take effect for rootvg.
0516-1126 mirrorvg: rootvg successfully mirrored, user should perform
        bosboot of system to initialize boot records.  Then, user must modify
        bootlist to include:  hdisk0 hdisk1.
# bosboot -ad /dev/hdisk1

bosboot: Boot image is 23795 512 byte blocks.
# bootlist -m normal hdisk0 hdisk1
# lsvg -l rootvg
rootvg:
LV NAME             TYPE       LPs   PPs   PVs  LV STATE      MOUNT POINT
hd5                 boot       1     2     2    closed/syncd  N/A
hd6                 paging     4     8     2    open/syncd    N/A
hd8                 jfs2log    1     2     2    open/syncd    N/A
hd4                 jfs2       1     2     2    open/syncd    /
hd2                 jfs2       9     18    2    open/syncd    /usr
hd9var              jfs2       1     2     2    open/syncd    /var
hd3                 jfs2       1     2     2    open/syncd    /tmp
hd1                 jfs2       1     2     2    open/syncd    /home
```

```
hd10opt          jfs2      1     2     2    open/syncd     /opt
#shutdown -Fr
```

6.3.14 Splitting and rejoining copies of a volume group

You can use the **splitvg** command to split a copy of a mirrored volume group into a snapshot volume group. To split a volume group, all logical volumes in the volume group must have a mirror copy and the mirror must be located on a disk or a set of disks that contain only this set of mirrors. The original volume group will stop using the disks that are part of the snapshot volume group. New logical volumes and new mounting points will be created in the snapshot volume group. Both volume groups will monitor changes of any physical partition so that when the snapshot volume group is rejoined with the original volume group, the data will remain consistent.

Note that the **splitvg** command should not be used on a volume group with paging spaces (such as the rootvg in default installations).

In Example 6-36, we use the **splitvg** command to split a volume group as follows:

- **lsvg -o** displays all active volume groups.
- **lsvg -l test1vg** shows that all logical volumes are mirrored (your results may differ).
- **lsvg -p test1vg** shows that test1vg contains two physical volumes.
- **splitvg -y newvg -c 1 test1vg** splits test1vg and creates a snapshot volume group named newvg.
- **lsvg -o** confirms that newvg is active.
- **lsvg -l newvg** displays the new names for logical volumes within testvg.
- **lsvg -p testvg** shows that hdisk6 is marked as snapshotpv.

Example 6-36 Using the splitvg command to split a volume group

```
# lsvg -o
test1vg
test2vg
dumpvg
rootvg
# lsvg -l test1vg
test1vg:
LV NAME          TYPE      LPs   PPs   PVs  LV STATE      MOUNT POINT
lv1              jfs2      9     18    2    closed/syncd  /fs1
lv4              sysdump   2     4     2    closed/syncd  N/A
# lsvg -p test1vg
```

```
test1vg:
PV_NAME           PV STATE          TOTAL PPs   FREE PPs    FREE DISTRIBUTION
hdisk6            active            136         118         28..17..25..27..21
hdisk7            active            136         125         28..27..25..27..18
# splitvg -y newvg -c 1 test1vg
# lsvg -o
newvg
test1vg
test2vg
dumpvg
rootvg
# lsvg -l test1vg
test1vg:
LV NAME             TYPE       LPs    PPs    PVs  LV STATE      MOUNT POINT
lv1                 jfs2       9      18     2    closed/syncd  /fs1
lv4                 sysdump    2      4      2    closed/syncd  N/A
# lsvg -l newvg
newvg:
LV NAME             TYPE       LPs    PPs    PVs  LV STATE      MOUNT POINT
fslv1               jfs2       9      9      1    closed/syncd  N/A
fslv2               jfs2       2      2      1    closed/syncd  N/A
# lsvg -p test1vg
test1vg:
PV_NAME           PV STATE          TOTAL PPs   FREE PPs    FREE DISTRIBUTION
hdisk6            snapshotpv        136         118         28..17..25..27..21
hdisk7            active            136         118         28..20..25..27..18
```

To rejoin the two copies of the volume group test1vg, use the command:

```
joinvg test1vg
```

6.4 Managing logical volumes

Logical volumes provide applications with the ability to access data as though it was stored contiguously. A logical volume consists of a sequence of one or more numbered logical partitions. Each logical partition has at least one and a maximum of three corresponding physical partitions that can be located on different physical volumes. The location on the disk for physical partitions is determined by intra-physical and inter-physical allocation policies.

For each logical volume, there are two corresponding device files under the /dev directory. The first one is a character device and the second one is a block device.

6.4.1 Creating a logical volume

You can create logical volumes using the **mklv** command. This command allows you to specify the name of the logical volume and its characteristics, such as the number of logical partitions and their location.

In Example 6-37, we use the mklv command to create, within volume group test1vg, a logical volume named lv3, of type jfs2, having 10 logical partitions, located on hdisk5.

Example 6-37 Using the mklv command

```
# mklv -y lv3 -t jfs2 -a im test1vg 10 hdisk5
lv3
# lslv lv3
LOGICAL VOLUME:     lv3                    VOLUME GROUP:   test1vg
LV IDENTIFIER:      00c5e9de00004c0000000107a5b596ab.4 PERMISSION:
read/write
VG STATE:           active/complete        LV STATE:       closed/syncd
TYPE:               jfs2                   WRITE VERIFY:   off
MAX LPs:            512                    PP SIZE:        512 megabyte(s)
COPIES:             1                      SCHED POLICY:   parallel
LPs:                10                     PPs:            10
STALE PPs:          0                      BB POLICY:      relocatable
INTER-POLICY:       minimum                RELOCATABLE:    yes
INTRA-POLICY:       inner middle           UPPER BOUND:    32
MOUNT POINT:        N/A                    LABEL:          None
MIRROR WRITE CONSISTENCY: on/ACTIVE
EACH LP COPY ON A SEPARATE PV ?: yes
Serialize IO ?:     NO
```

In Example 6-38, we use the **mklv** command to create, within volume group test1vg, a logical volume named lv4, of type sysdump, having two logical partitions, each having three copies, located on the center of three different disks (hdisk5, hdisk6 and hdisk7), being labeled demo-label and having a maximum of five logical partitions.

Example 6-38 Using the mklv command

```
# mklv -y lv4 -t sysdump -a c -e x -c3 -L demo-label -x5 test1vg 2 hdisk5
hdisk6 hdisk7
lv4
# lslv lv4
LOGICAL VOLUME:     lv4                    VOLUME GROUP:   test1vg
LV IDENTIFIER:      00c5e9de00004c0000000107a5b596ab.5 PERMISSION:
read/write
VG STATE:           active/complete        LV STATE:       closed/syncd
TYPE:               sysdump                WRITE VERIFY:   off
MAX LPs:            5                      PP SIZE:        512 megabyte(s)
```

```
COPIES:             3                        SCHED POLICY:   parallel
LPs:                2                        PPs:            6
STALE PPs:          0                        BB POLICY:      relocatable
INTER-POLICY:       maximum                  RELOCATABLE:    yes
INTRA-POLICY:       center                   UPPER BOUND:    32
MOUNT POINT:        N/A                      LABEL:          demo-label
MIRROR WRITE CONSISTENCY: on/ACTIVE
EACH LP COPY ON A SEPARATE PV ?: yes
Serialize IO ?:     NO
```

In Example 6-39, we use the **mklv** command to create, within volume group test1vg, a logical volume named lv5, of type jfs2, having three logical partitions, each having two pinned copies on different disks, read/write operations on the two disks being done sequential, write-verify being enabled, and I/O operations being serialized.

Example 6-39 Using the mklv command

```
# mklv -y lv5 -t jfs2 -c2 -rn -bn -ds -vy -oy test1vg 2 hdisk5 hdisk6
lv5
# lslv lv5
LOGICAL VOLUME:     lv5                      VOLUME GROUP:   test1vg
LV IDENTIFIER:      00c5e9de00004c0000000107a5b596ab.6 PERMISSION:
read/write
VG STATE:           active/complete          LV STATE:       closed/syncd
TYPE:               jfs2                     WRITE VERIFY:   on
MAX LPs:            512                      PP SIZE:        512 megabyte(s)
COPIES:             2                        SCHED POLICY:   sequential
LPs:                3                        PPs:            6
STALE PPs:          0                        BB POLICY:      non-relocatable
INTER-POLICY:       minimum                  RELOCATABLE:    no
INTRA-POLICY:       middle                   UPPER BOUND:    32
MOUNT POINT:        N/A                      LABEL:          None
MIRROR WRITE CONSISTENCY: on/ACTIVE
EACH LP COPY ON A SEPARATE PV ?: yes
Serialize IO ?:     YES
```

6.4.2 Removing a logical volume

The **rmlv** command is used to remove a logical volume. This command removes only the logical volume, but does not remove other entities, such as file systems or paging spaces that were using the logical volume.

In Example 6-40 on page 258, we offer an example of using **rmlv** to delete logical volumes as follows:

- **lsvg -l test1vg** displays all LVs that are contained in test1vg.

- **rmlv lv7** prompts for user confirmation and then deletes lv7.
- **lslv -l lv1** shows physical partitions of lv1 that are located on hdisk5, hdisk6, and hdisk7.
- **rmlv -p hdisk7 lv1** tries to delete partitions of lv1 located on hdisk7 and prompts for user confirmation. Because lv1 is opened, the operation cannot be executed.
- **umount /fs1** closes lv1.
- **rmlv -p hdisk7 lv1** tries to delete partitions of lv1 located on hdisk7, prompts for uses confirmation, and completes successfully.
- **lslv -l lv1** confirms that physical partitions of lv1 located on hdisk7 were deleted.

Example 6-40 Removing a logical volume

```
# lsvg -l test1vg
test1vg:
LV NAME             TYPE       LPs   PPs   PVs  LV STATE      MOUNT POINT
lv1                 jfs2       3     9     3    open/syncd    /fs1
lv2                 jfs2       2     4     2    closed/syncd  /fs2
logggg              jfs2log    2     2     1    open/syncd    N/A
lv3                 jfs2       10    10    1    closed/syncd  N/A
lv4                 sysdump    2     6     3    closed/syncd  N/A
lv5                 jfs2       3     6     2    closed/syncd  N/A
lv6                 jfs2       2     4     2    closed/syncd  N/A
lv7                 jfs2       2     4     2    closed/syncd  N/A
# rmlv lv7
Warning, all data contained on logical volume lv7 will be destroyed.
rmlv: Do you wish to continue? y(es) n(o)? y
rmlv: Logical volume lv7 is removed.
# lslv -l lv1
lv1:/fs1
PV                COPIES        IN BAND       DISTRIBUTION
hdisk5            003:000:000   100%          000:003:000:000:000
hdisk6            003:000:000   100%          000:003:000:000:000
hdisk7            003:000:000   100%          000:003:000:000:000
# rmlv -p hdisk7 lv1
0516-914 rmlv: Warning, all data belonging to logical volume
        lv1 on physical volume hdisk7 will be destroyed.
rmlv: Do you wish to continue? y(es) n(o)? y
0516-1008 rmlv: Logical volume lv1 must be closed.  If the logical
        volume contains a filesystem, the umount command will close
        the LV device.
# umount /fs1
# rmlv -p hdisk7 lv1
0516-914 rmlv: Warning, all data belonging to logical volume
        lv1 on physical volume hdisk7 will be destroyed.
```

```
rmlv: Do you wish to continue? y(es) n(o)? y
# lslv -l lv1
lv1:/fs1
PV                COPIES         IN BAND       DISTRIBUTION
hdisk5            003:000:000    100%          000:003:000:000:000
hdisk6            003:000:000    100%          000:003:000:000:000
```

6.4.3 Listing information about logical volumes

The **lslv** command displays information about the logical volumes.

In Example 6-41, we use the **lslv** command to display all details about a specific logical volume.

Example 6-41 Using lslv to display details about a specific logical volume

```
# lslv lv1
LOGICAL VOLUME:     lv1                    VOLUME GROUP:   test1vg
LV IDENTIFIER:      00c5e9de00004c0000000107a5b596ab.1 PERMISSION:
read/write
VG STATE:           active/complete        LV STATE:       closed/syncd
TYPE:               jfs2                   WRITE VERIFY:   off
MAX LPs:            512                    PP SIZE:        512 megabyte(s)
COPIES:             2                      SCHED POLICY:   parallel
LPs:                9                      PPs:            18
STALE PPs:          0                      BB POLICY:      relocatable
INTER-POLICY:       minimum                RELOCATABLE:    yes
INTRA-POLICY:       inner edge             UPPER BOUND:    32
MOUNT POINT:        /fs1                   LABEL:          None
MIRROR WRITE CONSISTENCY: on/ACTIVE
EACH LP COPY ON A SEPARATE PV ?: yes
Serialize IO ?:     NO
```

The meaning of the fields in Example 6-41 are:

LOGICAL VOLUME The name of the logical volume.

LV IDENTIFIER The identifier of the logical volume.

VG STATE The state of the volume group.

TYPE The LV type.

MAX LPs The maximum number of logical partitions of the logical volume.

COPIES The number of copies for each LP of the logical volume.

LPs The number of LPs contained in the logical volume.

STALE PPs The number of LPs that are stale.

INTER-POLICY Inter-physical allocation policy.

INTRA-POLICY Intra-physical allocation policy.

MOUNT POINT Mounting point for the file system residing on logical volume, if existent.

MIRROR WRITE CONSISTENCY
MWC activated or not.

EACH LP COPY ON A SEPARATE PV
The degree of strictness.

Serialize IO Serialization of overlapping IO states activated or not.

VOLUME GROUP The name of the volume group to which LV belongs.

PERMISSION Access permissions.

LV STATE State of the logical volume.

WRITE VERIFY Write verify policy on or off.

PP SIZE The size of each partition.

SCHED POLICY Sequential or parallel scheduling policy for I/O operations.

PPs The total number of physical partitions.

BB POLICY Bad block relocation policy.

RELOCATABLE Partitions belonging to this logical volume can be relocated or not during reorganization.

UPPER BOUND Maximum number of disks in a mirror copy.

LABEL Label for the logical volume.

In Example 6-42, we use the **lslv -l** command to display the distribution of the physical partitions corresponding to logical volume lv1 across regions of physical disks and the percentage of the physical partitions that are compliant with intra-physical allocation policy.

Example 6-42 Using lslv -l to display logical volumes contained in a volume group

```
# lslv -l lv1
lv1:/fs1
PV                COPIES        IN BAND       DISTRIBUTION
hdisk5            009:000:000   66%           000:003:000:000:006
hdisk6            009:000:000   66%           000:003:000:000:006
```

In Example 6-43 on page 261, we use the **lslv -m** command to display the numbers of logical partitions and their corresponding physical partitions.

Example 6-43 Using lslv -l to display LPs and PPs number

```
# lslv -m lv1
lv1:/fs1
LP    PP1  PV1               PP2  PV2               PP3  PV3
0001  0029 hdisk5            0029 hdisk6
0002  0030 hdisk5            0030 hdisk6
0003  0031 hdisk5            0031 hdisk6
0004  0110 hdisk5            0111 hdisk6
0005  0110 hdisk6            0112 hdisk5
0006  0111 hdisk5            0112 hdisk6
0007  0113 hdisk5            0113 hdisk6
0008  0114 hdisk5            0114 hdisk6
0009  0115 hdisk5            0115 hdisk6
```

In Example 6-44, we use the **lslv -n** command to display information about the logical volumes read from the VGDAs of two different disks as follows:

- **lsvg -l testvg** shows that testvg contains logical volume testlv.
- **lsvg -p testvg** shows that testvg contains hdisk5, hdisk6, and hdisk7.
- **lslv -m testlv** shows that testlv has all partitions located on hdisk5 and hdisk7.
- **lslv -n hdisk6 testlv** displays information about logical volume testlv read from VGDA located on hdisk6, although hdisk6 does not contain any single partition belonging to testlv.

Example 6-44 Using lslv -n to display information about logical volumes

```
# lsvg -l testvg
testvg:
LV NAME             TYPE       LPs   PPs   PVs  LV STATE      MOUNT POINT
testlv              jfs2       3     6     2    closed/syncd  /test
loglv00             jfs2log    1     2     2    closed/syncd  N/A
copylv              jfs2       3     6     2    closed/syncd  /copy
# lsvg -p testvg
testvg:
PV_NAME           PV STATE          TOTAL PPs   FREE PPs    FREE DISTRIBUTION
hdisk5            active            273         266         55..48..54..54..55
hdisk6            active            273         273         55..55..54..54..55
hdisk7            active            273         266         55..48..54..54..55
# lslv -m testlv
testlv:/test
LP    PP1  PV1               PP2  PV2               PP3  PV3
0001  0056 hdisk5            0059 hdisk7
0002  0057 hdisk5            0060 hdisk7
0003  0058 hdisk5            0061 hdisk7
# lslv -n hdisk6 testlv
LOGICAL VOLUME:     testlv                 VOLUME GROUP:   testvg
```

```
LV IDENTIFIER:      00c478de00004c0000000107c4419ccf.1 PERMISSION:
read/write
VG STATE:           active/complete        LV STATE:      closed/syncd
TYPE:               jfs2                   WRITE VERIFY:  off
MAX LPs:            512                    PP SIZE:       256 megabyte(s)
COPIES:             2                      SCHED POLICY:  parallel
LPs:                3                      PPs:           6
STALE PPs:          0                      BB POLICY:     relocatable
INTER-POLICY:       minimum                RELOCATABLE:   yes
INTRA-POLICY:       middle                 UPPER BOUND:   32
MOUNT POINT:        /test                  LABEL:         /test
MIRROR WRITE CONSISTENCY: on/ACTIVE
EACH LP COPY ON A SEPARATE PV ?: yes
Serialize IO ?:     NO
```

You can display the LVCB of a logical volume using intermediate level command **getlvcb**, as shown in Example 6-45.

Example 6-45 Using getlvcb to display the LVCB

```
# getlvcb -AT lv1
         AIX LVCB
         intrapolicy = ie
         copies = 3
         interpolicy = m
         lvid = 00c5e9de00004c0000000107a5b596ab.1
         lvname = lv1
         label = None
         machine id = 5E9DE4C00
         number lps = 9
         relocatable = y
         strict = y
         stripe width = 0
         stripe size in exponent = 0
         type = jfs2
         upperbound = 32
         fs =
         time created  = Wed Nov 23 15:30:44 2005
         time modified = Wed Nov 23 18:08:52 2005
```

6.4.4 Increasing the size of a logical volume

Additional logical partitions can be added to an already existing logical volume using the **extendlv** command. By default, the logical volume is expanded while preserving its characteristics. You can change these characteristics for the partitions to be added only using flags. The initial characteristics of the whole volume group will remain unchanged. You can specify one or multiple disks that will accommodate newly defined partitions. You cannot exceed the maximum number of partitions that was defined for the volume group. You can also specify blocks whose size is measured in KB, MB, or GB. The system will automatically determine the minimum number of partitions needed to fulfill the request.

In Example 6-46, we use the **extendlv** command to extend logical volume lv1 with three logical partitions located on the inner edges of both hdisk5 and hdisk6.

Example 6-46 Using the extendlv command

```
# lslv -l lv1
lv1:/fs1
PV              COPIES        IN BAND       DISTRIBUTION
hdisk5          003:000:000   100%          000:003:000:000:000
hdisk6          003:000:000   100%          000:003:000:000:000
# extendlv -a ie -ex lv1 3 hdisk5 hdisk6
# lslv -l lv1
lv1:/fs1
PV              COPIES        IN BAND       DISTRIBUTION
hdisk5          006:000:000   50%           000:003:000:000:003
hdisk6          006:000:000   50%           000:003:000:000:003
```

6.4.5 Copying a logical volume

You can copy the content of a logical volume to either a new or an already existing logical volume. To preserve data integrity, you should ensure that the destination logical volume is at least the size of the source logical volume.

The following example shows the use of the **cplv** command to copy logical volume lv1 to the dumpvg volume group under the name of lv8:

```
cplv -v dumpvg -y lv8 lv1
```

6.4.6 Creating copies of logical volumes

You can use the **mklvcopy** command to increase the number of copies of logical partitions. The logical volume will keep its characteristics. The new copies can be synchronized either manually or automatically.

In Example 6-47, we used the **mklvcopy** command to create and synchronize one extra copy of each of the logical partitions of logical volume lv1. Newly created copies will be located on hdisk7.

Example 6-47 Using mklvcopy to create and synchronize extra copies of logical partitions

```
# lslv -m lv1
lv1:/fs1
LP    PP1  PV1               PP2  PV2               PP3  PV3
0001  0029 hdisk5            0029 hdisk6
0002  0030 hdisk5            0030 hdisk6
0003  0031 hdisk5            0031 hdisk6
0004  0110 hdisk5            0111 hdisk6
0005  0110 hdisk6            0112 hdisk5
0006  0111 hdisk5            0112 hdisk6
0007  0113 hdisk5            0113 hdisk6
0008  0114 hdisk5            0114 hdisk6
0009  0115 hdisk5            0115 hdisk6
# mklvcopy -k lv1 3 hdisk7 &
# lslv -m lv1
lv1:/fs1
LP    PP1  PV1               PP2  PV2               PP3  PV3
0001  0029 hdisk5            0029 hdisk6            0110 hdisk7
0002  0030 hdisk5            0030 hdisk6            0111 hdisk7
0003  0031 hdisk5            0031 hdisk6            0112 hdisk7
0004  0110 hdisk5            0111 hdisk6            0113 hdisk7
0005  0110 hdisk6            0112 hdisk5            0114 hdisk7
0006  0111 hdisk5            0112 hdisk6            0115 hdisk7
0007  0113 hdisk5            0113 hdisk6            0116 hdisk7
0008  0114 hdisk5            0114 hdisk6            0117 hdisk7
0009  0115 hdisk5            0115 hdisk6            0118 hdisk7
```

6.4.7 Changing characteristics of logical volumes

You can use the **chlv** command to change the characteristics of an already existing logical volume. If you change attributes that affect the location of physical partitions, they will not affect already existing partitions, but only partitions that will be subsequently added or deleted. If the changes you make affect the file system residing on that logical volume, you will have to update the file system characteristics as well.

In Example 6-48 on page 265, we use the **chlv** command to change, for logical volume lv1, the maximum number of logical partitions to 1000 and the scheduling policy for I/O operations to parallel/round-robin.

Example 6-48 Changing logical volume characteristics

```
# lslv lv1
LOGICAL VOLUME:     lv1                     VOLUME GROUP:   test1vg
LV IDENTIFIER:      00c5e9de00004c0000000107a5b596ab.1 PERMISSION:
read/write
VG STATE:           active/complete         LV STATE:       opened/syncd
TYPE:               jfs2                    WRITE VERIFY:   off
MAX LPs:            512                     PP SIZE:        512 megabyte(s)
COPIES:             3                       SCHED POLICY:   parallel
LPs:                9                       PPs:            27
STALE PPs:          0                       BB POLICY:      relocatable
INTER-POLICY:       minimum                 RELOCATABLE:    yes
INTRA-POLICY:       inner edge              UPPER BOUND:    32
MOUNT POINT:        /fs1                    LABEL:          None
MIRROR WRITE CONSISTENCY: on/ACTIVE
EACH LP COPY ON A SEPARATE PV ?: yes
Serialize IO ?:     NO
# chlv -x 1000 -d pr lv1
# lslv lv1
LOGICAL VOLUME:     lv1                     VOLUME GROUP:   test1vg
LV IDENTIFIER:      00c5e9de00004c0000000107a5b596ab.1 PERMISSION:
read/write
VG STATE:           active/complete         LV STATE:       closed/syncd
TYPE:               jfs2                    WRITE VERIFY:   off
MAX LPs:            1000                    PP SIZE:        512 megabyte(s)
COPIES:             3                       SCHED POLICY:   parallel/round robin
LPs:                9                       PPs:            27
STALE PPs:          0                       BB POLICY:      relocatable
INTER-POLICY:       minimum                 RELOCATABLE:    yes
INTRA-POLICY:       inner edge              UPPER BOUND:    32
MOUNT POINT:        /fs1                    LABEL:          None
MIRROR WRITE CONSISTENCY: on/ACTIVE
EACH LP COPY ON A SEPARATE PV ?: yes
Serialize IO ?:     NO
```

6.4.8 Splitting a logical volume

You can use the **splitlvcopy** command to split a logical volume that has at least two copies of each logical partition into two different logical volumes. The newly created logical volume will have the same characteristics as the original. We recommend closing the logical volume to be split. If the original logical volume contains a file system, the data from the newly created logical volume will have to be accessed as a different file system.

In Example 6-49, we use the **splitlvcopy** command to split a logical volume in two copies as follows (without maintaining file system data):

- **lsvg -l testvg** shows that testvg contains logical volume testlv of type jfs2 and having /test as a mounting point.
- **lslv -m testlv** shows that testlv has three mirror copies located on hdisk5, hdisk6, and hdisk7.
- **splitlvcopy -y copylv testlv 2** tries to split the logical volume and prompts for user confirmation, because testlv is open and data may be corrupted.
- **umount /test** closes logical volume testlv.
- **splitlvcopy -y copylv testlv 2** splits the logical volume.
- **lsvg -l testvg** shows that the new logical volume copylv has been created.
- **lslv -m testlv** shows that testlv now has only two mirrored copies, located on hdisk5 and hdisk6.
- **lslv -m copylv** shows that copylv contains partitions from hdisk7.
- **lslv copylv** displays characteristics of the newly created logical volume copylv. Notice that the logical volume does not have a mounting point.
- **crfs -v jfs2 -d /dev/copylv -m /copy** creates the file system structure for copylv. Note that this command will destroy any file system data.

If you want to maintain the file system data on the original logical volume, instead of running the **crfs** command in the last step, perform the following:

- **mkdir /copy** creates a copy directory.
- **mount /dev/copylv /copy** mounts the copied file system.
- Edit the /etc/filesystems file manually and add an entry for the /copy mount point.

Example 6-49 Using splitlvcopy

```
# lsvg -l testvg
testvg:
LV NAME             TYPE       LPs    PPs    PVs   LV STATE       MOUNT POINT
testlv              jfs2       3      9      3     open/syncd     /test
loglv00             jfs2log    1      1      1     open/syncd     N/A
# lslv -m testlv
testlv:/test
LP    PP1  PV1               PP2  PV2               PP3  PV3
0001  0056 hdisk5            0056 hdisk6            0056 hdisk7
0002  0057 hdisk5            0057 hdisk6            0057 hdisk7
0003  0058 hdisk5            0058 hdisk6            0058 hdisk7
# splitlvcopy -y copylv testlv 2
splitlvcopy: WARNING! The logical volume being split, testlv, is open.
```

```
        Splitting an open logical volume may cause data loss or corruption
        and is not supported by IBM. IBM will not be held responsible for
        data loss or corruption caused by splitting an open logical volume.
        Do you wish to continue? y(es) n(o)? n
# umount /test
# splitlvcopy -y copylv testlv 2
copylv
# lsvg -l testvg
testvg:
LV NAME             TYPE       LPs   PPs   PVs  LV STATE      MOUNT POINT
testlv              jfs2       3     6     2    closed/syncd  /test
loglv00             jfs2log    1     1     1    closed/syncd  N/A
copylv              jfs2       3     3     1    closed/syncd  N/A
# lslv -m testlv
testlv:/test
LP    PP1  PV1               PP2  PV2               PP3  PV3
0001  0056 hdisk5            0056 hdisk6
0002  0057 hdisk5            0057 hdisk6
0003  0058 hdisk5            0058 hdisk6
# lslv -m copylv
copylv:N/A
LP    PP1  PV1               PP2  PV2               PP3  PV3
0001  0056 hdisk7
0002  0057 hdisk7
0003  0058 hdisk7
# lslv copylv
LOGICAL VOLUME:     copylv                 VOLUME GROUP:   testvg
LV IDENTIFIER:      00c478de00004c0000000107c4419ccf.3 PERMISSION:
read/write
VG STATE:           active/complete        LV STATE:       closed/syncd
TYPE:               jfs2                   WRITE VERIFY:   off
MAX LPs:            512                    PP SIZE:        256 megabyte(s)
COPIES:             1                      SCHED POLICY:   parallel
LPs:                3                      PPs:            3
STALE PPs:          0                      BB POLICY:      relocatable
INTER-POLICY:       minimum                RELOCATABLE:    yes
INTRA-POLICY:       middle                 UPPER BOUND:    32
MOUNT POINT:        N/A                   LABEL:          /test
MIRROR WRITE CONSISTENCY: on/ACTIVE
EACH LP COPY ON A SEPARATE PV ?: yes
Serialize IO ?:     NO
# crfs -v jfs2 -d /dev/copylv -m /copy
File system created successfully.
786204 kilobytes total disk space.
New File System size is 1572864
```

6.4.9 Removing a copy of a logical volume

You can use the **rmlvcopy** command to remove copies of logical partitions of a logical volume.

In Example 6-50, we use the **rmlvcopy** command to remove a set of copies for the logical partitions testlv as follows:

- **lslv -m testlv** shows that testlv has three mirror copies located on hdisk5, hdisk6, and hdisk7.
- **rmlvcopy testlv 2 hdisk6** removes copies located on hdisk6 and leaves two mirror copies.
- **lslv -m testlv** shows that testlv now has two mirror copies located on hdisk5 and hdisk7.

Example 6-50 Using rmlvcopy

```
# lslv -m testlv
testlv:/test
LP    PP1  PV1               PP2  PV2               PP3  PV3
0001  0056 hdisk5            0056 hdisk6            0059 hdisk7
0002  0057 hdisk5            0057 hdisk6            0060 hdisk7
0003  0058 hdisk5            0058 hdisk6            0061 hdisk7
# rmlvcopy testlv 2 hdisk6
# lslv -m testlv
testlv:/test
LP    PP1  PV1               PP2  PV2               PP3  PV3
0001  0056 hdisk5            0059 hdisk7
0002  0057 hdisk5            0060 hdisk7
0003  0058 hdisk5            0061 hdisk7
```

6.5 Summary of the LVM commands

All LVM commands have corresponding menus in SMIT.

Table 6-2 on page 269 provides a summary of LVM commands, their corresponding SMIT fast path, and a short description of each command.

Table 6-2 Summary of LVM commands

Command	SMIT fast path	Description
`chpv`	`smit chpv`	Changes the characteristics of a physical volume.
`lspv`	`smit lspv`	Lists information about physical volumes.
`migratepv`	`smit migratepv`	Migrates physical partitions from one physical volume to other(s).
`mkvg`	`smit mkvg`	Creates a volume group.
`lsvg`	`smit lsvg`	Lists information about volume groups.
`reducevg`	`smit reducevg`	Removes a physical volume from a volume group.
`chvg`	`smit chvg`	Changes the characteristics of a volume group.
`importvg`	`smit importvg`	Imports the definition of a volume group into the system.
`exportvg`	`smit exportvg`	Removes the definition of a volume group from the system.
`varyonvg`	`smit varyonvg`	Activates a volume group.
`varyoffvg`	`smit varyoffvg`	Deactivates a volume group.
`mklv`	`smit mklv`	Creates a logical volume.
`lslv`	`smit lslv`	Lists information about a logical volume.
`chlv`	`smit chlv`	Changes the characteristics of a logical volume.
`rmlv`	`smit rmlv`	Deletes a logical volume.
`extendlv`	`smit extendlv`	Extends a logical volume.
`mklvcopy`	`smit mklvcopy`	Creates a copy of a logical volume.
`rmlvcopy`	`smit rmlvcopy`	Removes a copy of a logical volume.

7

File systems

File systems represent a convenient way to store and access data from a logical volume. A file system is comprised of files, directories, and other data structures. File systems maintain information and identify the location of a file or directory's data.

© Copyright IBM Corp. 2006. All rights reserved.

7.1 File system types

The following types of file systems are supported on an AIX 5L Version 5.3:

Journaled file system
This type of file system is named journaled because the system uses journaling techniques to maintain the integrity of control structures. Each journaled file system must reside on a distinct jfs logical volume. Therefore, the file system size will be a multiple of the size of a logical partition.

Enhanced journaled file system
This is the enhanced version of the initial journalized file system. It uses extent based allocation to allow higher performance, larger file systems, and a larger file size. Each enhanced journaled file system must reside on a distinct jfs2 logical volume. When the operating system is installed using the default options, it creates JFS2 file systems.

Network file system The network file system (NFS) is a distributed file system that allows users to access files and directories located on remote computers and use those files and directories as though they are local.

CD-ROM file system The CD-ROM file system (CDRFS) is a file system type that allows you to access the contents of a CD-ROM through the normal file system interfaces.

7.2 File system structure

The journaled file systems use data structures, such as superblock, allocation groups, inodes, blocks, fragments, and device logs.

7.2.1 Superblock

The superblock contains control information about a file system, such as the overall size of the file system in 512 byte blocks, the file system name, the file system log device, the version number, the number of inodes, the list of free inodes, the list of free data blocks, date and time of creation, and file system state. All this data is stored in the first logical block of the file system. Corruption of this data may render the file system unusable. This is why the system keeps a second copy of the superblock on logical block 31.

7.2.2 Allocation group

An allocation group consists of inodes and its corresponding data blocks. An allocation groups spans multiple adjacent disk blocks and improves the speed of I/O operations. Both JFS and JFS2 file systems use allocation groups. For a JFS file system, the allocation group size can be specified when the file system is created.

7.2.3 Inodes

The inode contains control information about the file, such as type, size, owner, and the date and time when the file was created, modified, or last accessed. It also contains pointers to data blocks that store the actual data of the file. Every file has a corresponding inode.

For JFS file systems, the maximum number of inodes, and hence the maximum number of files, is determined by the number of bytes per inode (nbpi) value, which is specified when the file system is created. For every nbpi bytes of your file system, there will be an inode created. The total number of inodes is fixed. The nbpi values needs to be correlated with allocation group size.

The JFS restricts all file systems to 16 MB (2^{24}) inodes.

JFS2 file systems manages the necessary space for inodes dynamically so there is not any nbpi parameter.

7.2.4 Data blocks

Data blocks store the actual data of the file or pointers to other data blocks. The default value for disk block size is 4 KB.

7.2.5 Fragments

Fragments of logical blocks can be used to support files smaller than the standard size of the logical block (4 KB). This rule applies only to the last block of a file smaller than 32 KB.

For JFS file systems only, you have the option to use compression to allow all logical blocks of a file to be stored as a sequence of contiguous fragments. Compression for a file system will increase the amount of CPU and I/O activity when using that file system.

These features can be useful to support a large number of small files. Fragment size must be specified for a file system at installation time. Different file systems can have different fragment sizes.

7.3 Device logs

The journaled file system log stores transactional information about file system metadata changes.This data can be used to roll back incomplete operations if the machine crashes. JFS file systems are used for logging logical volumes of type jfslog, while JFS2 file systems are used for logging logical volumes of type jfs2log.

Data from data blocks are not journaled. Log devices ensure file system integrity, not data integrity.

After the operating system is installed, all file systems within the rootvg volume group use logical volume hd8 as a common log.

You can create a JFS2 file system that can use inline logs. This means the log data is written into the same logical volume as the file system, and not into the log logical volume.

7.4 File system differences

Table 7-1 displays some of the differences between JFS and JFS2 file systems.

Table 7-1 Journaled file system differences

Function	JFS	JFS2
Architectural maximum file system size	1 TB	4 PB
Architectural maximum file size	64 GB	4 PB
Number of i-nodes	Fixed, set at file system creation	Dynamic
Inode size	128 bytes	512 bytes
Fragment size	512	512
Block size	4096	4096
Directory organization	Linear	B-tree
Compression	Yes	No
Default ownership at creation	sys.sys	root.system
SGID of default file mode	SGID=on	SGID=off
Quotas	Yes	Yes

JFS and JFS2 file system can coexist on the same systems.

If you to migrate data from a JFS file system to a JFS2 file system you have to back up the JFS file system and restore the data on the JFS2 file systems.

7.5 File system management

The follow sections describe basic file system management.

7.5.1 Creating a file system

Every file system in AIX 5L Version 5.3 must reside on a logical volume.

When you create a file system, if you do not specify a previously created logical volume the system will create a logical volume with a system-generated name that has default values for the logical volume's characteristics. The size of the file system is rounded up to closest multiple of the logical partition size. The size of the file system cannot exceed the standard maximum number of logical partitions for a logical volume. A stanza containing the details of the file system will be added to /etc/filesystems and new data will be written to ODM.

In Example 7-1 on page 276, we show how to use the **crfs** command as follows:

- **lsvg -l testvg** shows that testvg does not contain any logical volumes of type jfs, jfs2, jfslog, and jfs2log.
- **crfs -v jfs -g testvg -a size=10M -m /fs1** creates, within volume group testvg, a jfs file system of 10 MB with mounting point /fs1. There was not an existing jfs logical volume, so the system created one and assigned it the name lv00. There was not an existing jfs log device, so the system created one and assigned it the name loglv00.
- **crfs -v jfs2 -g testvg -a size=10M -p ro -m /fs2** creates, within volume group testvg, a jfs2 file system of 10 MB with mounting point /fs2 and having read only permissions. There was not an existing jfs2 logical volume, so the system created one and assigned it the name fslv00. There was not an existing jfs2 log device, so the system created one and assigned it the name loglv01.
- **lslv lv00 and lslv fslv00** shows that logical volumes were created using their default characteristics.
- **cat /etc/filesystems|grep -ip fs1** displays the stanza from /etc/filesystems that corresponds to fs1.

Example 7-1 Creating file systems without specifying logical volumes

```
# lsvg -l testvg
testvg:
LV NAME             TYPE       LPs    PPs    PVs  LV STATE      MOUNT POINT
# crfs -v jfs -g testvg -a size=10M -m /fs1
Based on the parameters chosen, the new /fs1 JFS file system
is limited to a maximum size of 134217728 (512 byte blocks)

New File System size is 262144
# crfs -v jfs2 -g testvg -a size=10M -p ro -m /fs2
File system created successfully.
130864 kilobytes total disk space.
New File System size is 262144
# lsvg -l testvg
testvg:
LV NAME             TYPE       LPs    PPs    PVs  LV STATE      MOUNT POINT
loglv00             jfslog     1      1      1    closed/syncd  N/A
lv00                jfs        1      1      1    closed/syncd  /fs1
loglv01             jfs2log    1      1      1    closed/syncd  N/A
fslv00              jfs2       1      1      1    closed/syncd  /fs2
# lslv lv00
LOGICAL VOLUME:     lv00                   VOLUME GROUP:   testvg
LV IDENTIFIER:      00c478de00004c0000000107d96de510.2 PERMISSION:
read/write
VG STATE:           active/complete        LV STATE:       closed/syncd
TYPE:               jfs                    WRITE VERIFY:   off
MAX LPs:            512                    PP SIZE:        128 megabyte(s)
COPIES:             1                      SCHED POLICY:   parallel
LPs:                1                      PPs:            1
STALE PPs:          0                      BB POLICY:      relocatable
INTER-POLICY:       minimum                RELOCATABLE:    yes
INTRA-POLICY:       middle                 UPPER BOUND:    32
MOUNT POINT:        /fs1                   LABEL:          /fs1
MIRROR WRITE CONSISTENCY: on/ACTIVE
EACH LP COPY ON A SEPARATE PV ?: yes
Serialize IO ?:     NO
# lslv fslv00
LOGICAL VOLUME:     fslv00                 VOLUME GROUP:   testvg
LV IDENTIFIER:      00c478de00004c0000000107d96de510.4 PERMISSION:
read/write
VG STATE:           active/complete        LV STATE:       closed/syncd
TYPE:               jfs2                   WRITE VERIFY:   off
MAX LPs:            512                    PP SIZE:        128 megabyte(s)
COPIES:             1                      SCHED POLICY:   parallel
LPs:                1                      PPs:            1
STALE PPs:          0                      BB POLICY:      relocatable
INTER-POLICY:       minimum                RELOCATABLE:    yes
INTRA-POLICY:       middle                 UPPER BOUND:    32
```

```
MOUNT POINT:       /fs2                   LABEL:          /fs2
MIRROR WRITE CONSISTENCY: on/ACTIVE
EACH LP COPY ON A SEPARATE PV ?: yes
Serialize IO ?:     NO
# cat /etc/filesystems|grep -ip fs1
/fs1:
        dev             = /dev/lv00
        vfs             = jfs
        log             = /dev/loglv00
        mount           = false
        account         = false
```

You can specify the name of a previously created logical volume. The type of the logical volume must match the type of the file system. The size of the file system is rounded up to closest multiple of the logical partition size. A stanza containing the details of the file system will be added to /etc/filesystems and the ODM will be updated.

In Example 7-2, we show how to use the **crfs** command as follows:

- **lsvg -l testvg** shows that jfs2 type testlv logical volume does not have any file system associated and there is already a jfs2 log device defined. testlv has only one partition of 128 MB.
- **crfs -v jfs2 -d /dev/testlv -a logname=loglv01 -m /test -a size=130M** creates a jfs2 file system located on already existing logical volume named testlv, using jfs2 log device loglv01 and having /test as the mounting point. Because we specified, for the file system, a size bigger than the logical volume itself, the size parameter is ignored and the final size of the file system will be rounded to the size of the logical volume.
- **cat /etc/filesystems|grep -ip test** displays the stanza from /etc/filesystems corresponding to test.

Example 7-2 Creating file systems on previously defined logical volumes

```
# lsvg -l testvg
testvg:
LV NAME             TYPE       LPs   PPs   PVs  LV STATE      MOUNT POINT
loglv00             jfslog     1     1     1    closed/syncd  N/A
lv00                jfs        1     1     1    closed/syncd  /fs1
loglv01             jfs2log    1     1     1    closed/syncd  N/A
fslv00              jfs2       1     1     1    closed/syncd  /fs2
testlv              jfs2       1     1     1    closed/syncd  N/A
# crfs -v jfs2 -d /dev/testlv -a logname=loglv01 -m /test -a size=130M
crfs: Warning: device name given, size parameter ignored.
File system created successfully.
130864 kilobytes total disk space.
New File System size is 262144
```

```
# cat /etc/filesystems|grep -ip test
/test:
        dev             = /dev/testlv
        vfs             = jfs2
        log             = /dev/loglv01
        mount           = false
        account         = false
```

7.5.2 Mounting and unmounting file systems

Mounting is a concept that makes file systems, files, directories, devices, and special files available for use at a particular location. It is the only way a file system is made accessible.

The mounting point must be created before the file system is mounted. When a file system is mounted over a directory, the permissions of the root directory of the mounted file system take precedence over the permissions of the mount point.

In Example 7-3, we use the **mount** command as follows:

- **mount /dev/fslv02 /testfs** tries to mount /test under a nonexisting directory and generates an error.
- **mount /dev/fslv02 /test** mounts the file system under an existing directory.
- **umount /testfs** tries to unmount a nonexistent file system and generates an error.
- **umount /test** unmounts the /test file system.

Example 7-3 Mounting and un-mounting file systems

```
# mount /dev/fslv02 /testfs
mount: 0506-324 Cannot mount /dev/fslv02 on /testfs: A file or directory in the
path name does not exist.
# umount /dev/fslv02 /test
umount: 0506-347 Cannot find anything to unmount.
# mount /dev/fslv02 /test
# umount /testf
umount: 0506-347 Cannot find anything to unmount.
# umount /test
```

If you want to mount all the file systems, you can use the following command to mount all the file systems at one time:

```
mount {-a|all}
```

7.5.3 Displaying mounted file systems

You can use the **mount** command without any flags to display information about all the currently mounted file systems, as shown in Example 7-4.

Example 7-4 Displaying mounted file systems using the mount command

```
# mount
  node       mounted        mounted over    vfs       date        options
-------- --------------- --------------- ------ ------------ ---------------
         /dev/hd4        /               jfs2   Nov 27 12:36 rw,log=/dev/hd8
         /dev/hd2        /usr            jfs2   Nov 27 12:36 rw,log=/dev/hd8
         /dev/hd9var     /var            jfs2   Nov 27 12:36 rw,log=/dev/hd8
         /dev/hd3        /tmp            jfs2   Nov 27 12:36 rw,log=/dev/hd8
         /dev/hd1        /home           jfs2   Nov 27 12:36 rw,log=/dev/hd8
         /proc           /proc           procfs Nov 27 12:36 rw
         /dev/hd10opt    /opt            jfs2   Nov 27 12:36 rw,log=/dev/hd8
         /dev/testlv     /test           jfs2   Nov 28 19:54
rw,log=/dev/loglv01
```

You can use the **lsfs** command to display the characteristics of file systems, as shown in Example 7-5. This command will display data about all types of file systems

Example 7-5 Using the lsfs command

```
# lsfs -a
Name            Nodename   Mount Pt               VFS   Size    Options    Auto
Accounting
/dev/hd4        --         /                      jfs2  262144  --         yes
no
/dev/hd1        --         /home                  jfs2  262144  --         yes
no
/dev/hd2        --         /usr                   jfs2  2359296 --         yes
no
/dev/hd9var     --         /var                   jfs2  262144  --         yes
no
/dev/hd3        --         /tmp                   jfs2  262144  --         yes
no
/proc           --         /proc                 procfs --      --         yes
no
/dev/hd10opt    --         /opt                   jfs2  262144  --         yes
no
/dev/lv00       --         /fs1                   jfs   262144  --         no
no
/dev/fslv00     --         /fs2                   jfs2  262144  ro         no
no
# lsfs -q
```

```
Name            Nodename   Mount Pt               VFS   Size    Options    Auto
Accounting
/dev/hd4        --         /                      jfs2  262144  --         yes
no
  (lv size: 262144, fs size: 262144, block size: 4096, sparse files: yes,
inline log: no, inline log size: 0, EAformat: v1, Quota: no, DMAPI: no, VIX:
no)
/dev/hd1        --         /home                  jfs2  262144  --         yes
no
  (lv size: 262144, fs size: 262144, block size: 4096, sparse files: yes,
inline log: no, inline log size: 0, EAformat: v1, Quota: no, DMAPI: no, VIX:
no)
/dev/hd2        --         /usr                   jfs2  2359296 --         yes
no
  (lv size: 2359296, fs size: 2359296, block size: 4096, sparse files: yes,
inline log: no, inline log size: 0, EAformat: v1, Quota: no, DMAPI: no, VIX:
no)
/dev/hd9var     --         /var                   jfs2  262144  --         yes
no
  (lv size: 262144, fs size: 262144, block size: 4096, sparse files: yes,
inline log: no, inline log size: 0, EAformat: v1, Quota: no, DMAPI: no, VIX:
no)
/dev/hd3        --         /tmp                   jfs2  262144  --         yes
no
  (lv size: 262144, fs size: 262144, block size: 4096, sparse files: yes,
inline log: no, inline log size: 0, EAformat: v1, Quota: no, DMAPI: no, VIX:
no)
/proc           --         /proc                  procfs --     --         yes
no
/dev/hd10opt    --         /opt                   jfs2  262144  --         yes
no
  (lv size: 262144, fs size: 262144, block size: 4096, sparse files: yes,
inline log: no, inline log size: 0, EAformat: v1, Quota: no, DMAPI: no, VIX:
no)
/dev/lv00       --         /fs1                   jfs   262144  --         no
no
  (lv size: 262144, fs size: 262144, frag size: 4096, nbpi: 4096, compress: no,
bf: false, ag: 8)
/dev/fslv00     --         /fs2                   jfs2  262144  ro         no
no
  (lv size: 262144, fs size: 262144, block size: 4096, sparse files: yes,
inline log: no, inline log size: 0, EAformat: v1, Quota: no, DMAPI: no, VIX:
no)
```

7.5.4 Removing a file system

You can use the **rmfs** command to remove a file system. The command will delete the corresponding stanza from the /etc/filesystems and the logical volume used on which the file system resides.

Example 7-6 Using the rmfs command

```
# lsvg -l testvg
testvg:
LV NAME             TYPE       LPs   PPs   PVs  LV STATE      MOUNT POINT
loglv00             jfslog     1     1     1    closed/syncd  N/A
lv00                jfs        1     1     1    closed/syncd  /fs1
loglv01             jfs2log    1     1     1    open/syncd    N/A
fslv00              jfs2       1     1     1    closed/syncd  /fs2
testlv              jfs2       1     1     1    open/syncd    /test
# rmfs /test
rmfs: 0506-921 /test is currently mounted.
# umount /test
# rmfs /test
rmlv: Logical volume testlv is removed.
# lsvg -l testvg

testvg:
LV NAME             TYPE       LPs   PPs   PVs  LV STATE      MOUNT POINT
loglv00             jfslog     1     1     1    closed/syncd  N/A
lv00                jfs        1     1     1    closed/syncd  /fs1
loglv01             jfs2log    1     1     1    closed/syncd  N/A
fslv00              jfs2       1     1     1    closed/syncd  /fs2
# cat /etc/filesystems|grep test
#
```

7.5.5 Changing the attributes of a file system

You can use the **chfs** command to change some of the attributes of a file system, such as mounting point permissions, log device, or size, as shown in Example 7-7. If the new size for the file system is larger than the size of the logical volume, the logical volume will be extended to accommodate the file system, provided that it does not exceed the maximum number of logical partitions.

Example 7-7 Changing attributes of a file system

```
# lsfs -a
Name            Nodename   Mount Pt               VFS   Size    Options    Auto
Accounting
/dev/hd4        --         /                      jfs2  262144  --         yes
no
```

```
/dev/hd1          --          /home                  jfs2  262144  --         yes
no
/dev/hd2          --          /usr                   jfs2  2359296 --         yes
no
/dev/hd9var       --          /var                   jfs2  262144  --         yes
no
/dev/hd3          --          /tmp                   jfs2  262144  --         yes
no
/proc             --          /proc                  procfs --     --         yes
no
/dev/hd10opt      --          /opt                   jfs2  262144  --         yes
no
/dev/lv00         --          /fs1                   jfs   262144  --         no
no
/dev/fslv00       --          /fs2                   jfs2  262144  ro         no
no
# chfs -a size=250M -p rw /fs2
Filesystem size changed to 524288
# lsfs -a
Name              Nodename    Mount Pt               VFS   Size    Options    Auto
Accounting
/dev/hd4          --          /                      jfs2  262144  --         yes
no
/dev/hd1          --          /home                  jfs2  262144  --         yes
no
/dev/hd2          --          /usr                   jfs2  2359296 --         yes
no
/dev/hd9var       --          /var                   jfs2  262144  --         yes
no
/dev/hd3          --          /tmp                   jfs2  262144  --         yes
no
/proc             --          /proc                  procfs --     --         yes
no
/dev/hd10opt      --          /opt                   jfs2  262144  --         yes
no
/dev/lv00         --          /fs1                   jfs   262144  --         no
no
/dev/fslv00       --          /fs2                   jfs2  524288  rw         no
no
```

7.5.6 Checking file system consistency

The **fsck** command checks file system consistency and interactively repairs the file system. You should not run the **fsck** command on a mounted file system. You must be able to read the device file on which the file system resides. The **fsck** command tries to repair file system metadata structures, displays information about the inconsistencies found, and prompts you for permission to repair them.

It does not recover the data from data blocks. If you lost data, you have to restore it from a backup.

Orphaned files and directories detected by the **fsck** command are placed under the lost+found directory located in the root directory of the file system.

When the system boots, the **fsck** command is called to verify the /, /usr, /var, and /tmp file systems. An unsuccessful result prevents the system from booting.

7.5.7 Log devices

The following sections describe basic information about file system logs.

Creating log devices

A dedicated log device is created on hd8 for rootvg when the system is installed. When the size of your file system is increasing, you should consider either increasing the size of the default log or creating new log devices.

To create a log devices, you should use the **mklv** command and specify, for the type of the logical volume, jfslog or jfs2log.

Initializing log devices

The log devices are initialized using the **logform** command by clearing all log records. The same command is used for jfslog devices, jfs2log2 devices, or inline logs. The **logform** command does not affect the data itself.

To initialize the jf2log device named loglv01, use the following command:

```
logform /dev/loglv01
```

7.6 Defragmenting a file system

The use of fragments and compression, as well as the creation and deletion of a large number of files, can decrease the amount of contiguous free disk space. The **defragfs** command can be used to improve or report the status of contiguous space within a file system. To defragment the file system /home, use the following command:

```
defragfs /home
```

7.7 Displaying information about inodes

You can use the **istat** command to display information regarding a particular inode, as shown in Example 7-8

Example 7-8 Using istat command

```
# istat /etc/passwd
Inode 66 on device 10/4 File
Protection: rw-r--r--
Owner: 0(root)          Group: 7(security)
Link count:   1         Length 644 bytes

Last updated:   Thu Dec  1 18:14:12 CST 2005
Last modified:  Thu Dec  1 18:14:12 CST 2005
Last accessed:  Mon Dec  5 17:04:17 CST 2005
```

7.8 Troubleshooting file system problems

This section discusses some of the problems related to file systems and how to resolve them.

7.8.1 Recovering from super block errors

If you receive one of the following errors from the **fsck** or **mount** commands, the problem may be a corrupted superblock:

```
fsck: Not an AIX3 file system
fsck: Not an AIXV3 file system
fsck: Not an AIX4 file system
fsck: Not an AIXV4 file system
fsck: Not a recognized file system type
mount: invalid argument
```

The problem can be resolved by restoring the backup of the superblock over the primary superblock using one of the following commands:

```
dd count=1 bs=4k skip=31 seek=1 if=/dev/lv00 of=/dev/lv00
```

Once the restoration process is completed, check the integrity of the file system using the **fsck** command. If this does not resolve the problem, recreate the file system, and restore the data from a backup.

7.8.2 Cannot unmount file systems

A file system cannot be unmounted if any references are still active within that file system. The following situations can leave open references to a mounted file system:

- Files are open within a file system. Close these files before the file system can be unmounted. The **fuser** command is often the best way to determine the process IDs for all processes that have open references within a specified file system. The process having an open reference can be killed by using the **kill** command and the unmount can be accomplished.
- If the file system is still busy and not getting unmounted, this could be due to a kernel extension that is loaded, but exists within the source file system. The **fuser** command will not show these kinds of references, because a user process is not involved. However, the **genkex** command will report on all loaded kernel extensions.
- File systems are still mounted within that file system. If any file system is mounted within a file system, this leaves open references in the source file system at the mount point of the other file system. Unmount all the file systems that are mounted within the file system to be unmounted.
- A user is using a directory within the file system as their current working directory. The **fuser** command appends the letter "c" to the process IDs of all processes that are using a directory as their current working directory, and the -u flag identifies the owner of the process. It can be used with the **find** command, as shown in the following example:

```
# find /home -type d -exec fuser -u {} \;
/home:
/home/lost+found:
/home/guest:
/home/kenzie:     3548c(kenzie)
```

7.8.3 Full file systems

Full file systems can produce a wide variety of problems. For example, some of the system-created file systems, such as /, /usr, /var, and /tmp, are used by some system commands. Lack of space in these files systems can induce unexpected results. Always make sure that you have enough free space in the file systems. You can monitor the space usage for your file systems using such commands as **df** or **du**.

8

Monitoring and performance tuning

This chapter provides information about concepts, tools, and techniques for assessing and tuning the performance of AIX 5L Version 5.3. Part of the job of the system administrator is performance analysis: To understand the system behavior and identify the usage of resources.

The concepts, tools, and techniques discussed in this chapter are not intended to be a total list, and as such, you are encouraged to seek additional information from the appropriate AIX 5L product documentation.

© Copyright IBM Corp. 2006. All rights reserved.

8.1 Monitoring file system growth

On the AIX 5L system, the administrator needs to monitor the file system growth to prevent a full file system.

The disk quota system, based on the Berkeley Disk Quota System, provides an effective way to control the use of disk space. The quota system can be defined for individual users or groups, and is maintained for each journaled file system.

The disk quota system establishes limits based on the following parameters that can be changed with the **edquota** command:

- User's or group's soft limits
- User's or group's hard limits
- Quota grace period

The soft limit defines the number of 1 KB disk blocks or files under which the user must remain. The hard limit defines the maximum amount of disk blocks or files the user can accumulate under the established disk quotas. The quota grace period allows the user to exceed the soft limit for a short period of time (the default value is one week). If the user fails to reduce usage below the soft limit during the specified time, the system will interpret the soft limit as the maximum allocation allowed, and no further storage is allocated to the user. The user can reset this condition by removing enough files to reduce usage below the soft limit.

The disk quota system tracks user and group quotas in the quota.user and quota.group files located in the root directories of file systems enabled with quotas. These files are created with the **quotacheck** and **edquota** commands and are readable with the **quota** commands.

Recovering from over-quota conditions

To reduce file system usage when you have exceeded quota limits, you can use the following methods:

- Kill the current process that caused the file system to reach its limit, remove surplus files to bring the limit below quota, and retry the failed program.
- If you are running an editor such as vi, use the shell escape sequence to check your file space, remove surplus files, and return without losing your edited file. Alternatively, if you are using the C or Korn shells, you can suspend the editor with the Ctrl-Z key sequence, issue the file system commands, and then return with the **fg** (foreground) command.
- Temporarily write the file to a file system where quota limits have not been exceeded, delete surplus files, and then return the file to the correct file system.

The **quotaon** command enables disk quotas for one or more file systems specified by the File System parameter. The specified file system must be defined with quotas in the /etc/filesystems file, and must be mounted. The **quotaon** command looks for the quota.user and quota.group default quota files in the root directory of the associated file system. These file names may be changed in the /etc/filesystems file.

By default, both user and group quotas are enabled. The -u flag enables only user quotas; the -g flag enables only group quotas. Specifying both the -g and -u flags is equivalent to the default option. The -a flag specifies that all file systems with disk quotas, as indicated by the /etc/filesystems file, are enabled.

The **quotaoff** command disables disk quotas for one or more file systems. By default, both user and group quotas are disabled. The -a, -g, and -u flags operate as with the **quotaon** command.

The -v flag prints a message for each file system in which quotas are turned on or off with the **quotaon** and **quotaoff** commands, respectively. Only the root user can execute this command.

The most commonly used command flags are provided in Table 8-1.

Table 8-1 Commonly used flags for the quota command

Flag	Description
-a	Enables or disables all file systems that are read-write and have disk quotas, as indicated by the /etc/filesystems file. When used with the -g flag, only group quotas in the /etc/filesystems file are enabled or disabled; when used with the -u flag, only user quotas in the /etc/filesystems file are enabled or disabled.
-g	Specifies that only group quotas are enabled or disabled.
-u	Specifies that only user quotas are enabled or disabled.
-v	Prints a message for each file system in which quotas are turned on or off.

Typically, only those file systems that contain user home directories and files require disk quotas. Consider implementing the disk quota system under the following conditions:

- Your system has limited disk space.
- You require more file system security.
- Your disk-usage levels are large, such as in many departments.

If these conditions do not apply to your environment, you might not want to create disk-usage limits by implementing the disk quota system.

The disk quota system can be used only with the journaled file system.

> **Note:** Do not establish disk quotas for the /tmp file system.

To set up the disk quota system, use the following procedure:

1. Log in with root authority.
2. Determine which file systems require quotas.

> **Note:** Because many editors and system utilities create temporary files in the /tmp file system, it must be free of quotas.

3. Use the **chfs** command to include the userquota and groupquota quota configuration attributes in the /etc/filesystems file. The following example uses the **chfs** command to enable user quotas on the /home file system:

   ```
   # chfs -a "quota = userquota" /home
   ```

   To enable both user and group quotas on the /home file system, type:

   ```
   # chfs -a "quota = userquota,groupquota" /home
   ```

   The corresponding entry in the /etc/filesystems file is displayed as follows:

   ```
   /home:
   dev         = /dev/hd1
   vfs         = jfs
   log         = /dev/hd8
   mount       = true
   check       = true
   quota       = userquota,groupquota
   options     = rw
   ```

4. Optionally, specify alternate disk quota file names. The quota.user and quota.group file names are the default names located at the root directories of the file systems enabled with quotas. You can specify alternate names or directories for these quota files with the userquota and groupquota attributes in the /etc/filesystems file.

   The following example uses the **chfs** command to establish user and group quotas for the /home file system, and names the myquota.user and myquota.group quota files:

   ```
   # chfs -a "userquota = /home/myquota.user"
                      -a "groupquota = /home/myquota.group" /home
   ```

The corresponding entry in the /etc/filesystems file is displayed as follows:

```
/home:
dev        = /dev/hd1
vfs        = jfs
log        = /dev/hd8
mount      = true
check      = true
quota      = userquota,groupquota
userquota  = /home/myquota.user
groupquota = /home/myquota.group
options    = rw
```

5. If they are not previously mounted, mount the specified file systems.
6. Set the desired quota limits for each user or group. Use the **edquota** command to create each user or group's soft and hard limits for allowable disk space and maximum number of files.

   The following example entry shows quota limits for the gpsilva user:

   ```
   Quotas for user gpsilva:
   /home: blocks in use: 30, limits (soft = 100, hard = 150)
           inodes in use: 73, limits (soft = 200, hard = 250)
   ```

   This user has used 30 KB of the maximum 100 KB of disk space. Of the maximum 200 files, gpsilva has created 73. This user has buffers of 50 KB of disk space and 50 files that can be allocated to temporary storage.

   When establishing disk quotas for multiple users, use the -p flag with the **edquota** command to duplicate a user's quotas for another user.

   To duplicate the quotas established for user gpsilva for user tneiva, type:

   ```
   # edquota -p gpsilva tneiva
   ```

7. Enable the quota system with the **quotaon** command. The **quotaon** command enables quotas for a specified file system, or for all file systems with quotas (as indicated in the /etc/filesystems file) when used with the -a flag.
8. Use the **quotacheck** command to check the consistency of the quota files against actual disk usage.

> **Note:** It is recommended that you do this each time you first enable quotas on a file system and after you reboot the system.

To enable this check and to turn on quotas during system startup, add the following lines at the end of the /etc/rc file:

```
echo " Enabling filesystem quotas "
/usr/sbin/quotacheck -a
/usr/sbin/quotaon -a
```

There are related commands, namely the **edquota** command, **quotacheck** command, and **repquota** command.

The following examples show the commands in typical uses:

1. To enable user quotas for the /usr/Tivoli/tsm/server/db file system, enter:

```
# quotaon -u /usr/Tivoli/tsm/server/db
```

2. To disable user and group quotas for all file systems in the /etc/filesystems file and print a message, enter:

```
# quotaoff -v -a
```

3. To display your quotas as user neivac, type:

```
$ quota
```

The system displays the following information:

```
User quotas for user neivac (uid 502):
Filesystem  blocks  quota  limit  grace  Files  quota limit grace
        /u      20     55     60            20     60    65
```

4. To display quotas as the root user for user gpsilva, type:

```
quota -u gpsilva
```

The system displays the following information:

```
User quotas for user gpsilva (uid 2702):
Filesystem  blocks  quota  limit  grace  files  quota limit grace
        /u      48     50     60             7     60    60
```

8.2 Recovering from a full file system

A full file system occurs when too many files fill up the allotted space. This can be caused by a runaway process that creates many unnecessary files. You can use the following procedures to correct the problem.

8.2.1 Fix a full / (root) file system

Check the following when the root file system (/) has become full:

- Use the **who** command to read the contents of the /etc/security/failedlogin file:

```
# who /etc/security/failedlogin
```

The condition of TTYs respawning too rapidly can create failed login entries. To clear the file after reading or saving the output, execute the following command:

```
# cp /dev/null /etc/security/failedlogin
```

- Check the /dev directory for a device name that is typed incorrectly. If a device name is typed incorrectly, such as rmto instead of rmt0, a file will be created in /dev called rmto. The command will normally proceed until the entire root file system is filled before failing. /dev is part of the root (/) file system. Look for entries that are not devices (that do not have a major or minor number). To check for this situation, use the following command:

```
# cd /dev
# ls -l | pg
total 40
drwxrwx---   2 root     system        4096 Nov 29 14:19 .SRC-unix
crw-rw----   1 root     system       10,  0 Nov 10 17:18 IPL_rootvg
srwxrwxrwx   1 root     system            0 Nov 28 17:30 SRC
crw-------   1 root     system       22,  1 Nov 23 12:14 __pv22.1.323758
crw-------   1 root     system       22,  1 Nov 23 12:14 __pv22.1.381014
crw-------   1 root     system       22,  1 Nov 23 12:15 __pv22.1.397388
.
.
.
(lines ommited)
.
.
.
crw-rw-rwT   1 root     system       17,  0 Nov 10 16:22 usb0
crw-rw-rw-   1 root     system       15,  0 Nov 10 16:22 usbhc0
crw-rw-rw-   1 root     system       15,  1 Nov 10 16:22 usbhc1
crw-------   1 root     system       11,  0 Nov 10 16:22 vio0
crw-rw-rw-   1 root     system       21,  0 Nov 10 16:22 vty0
crw-rw-rw-   1 root     system       21,  1 Nov 10 16:22 vty1
drwxr-xr-x   2 root     system         256 Nov 10 17:18 xti
crw-rw-rw-   1 root     system        2,  3 Nov 10 17:18 zero
#
```

  In the same location that would indicate a file size for an ordinary file, a device file has two numbers separated by a comma. For example:

```
crw-rw-rw-   1 root     system       24,  0 Nov 10 16:22 rmt0
```

  If the file name or size location indicates an invalid device, as shown in the following example, remove the associated file:

```
crw-rw-rw-   1 root     system   9375473 Oct 25 10:19 rmto
```

  Notes:

  1. Do not remove valid device names in the /dev directory. One indicator of an invalid device is an associated file size that is larger than 500 bytes.
  2. If system auditing is running, the default /audit directory can rapidly fill up and require attention.

Check for very large files that might be removed using the **find** command. For example, to find all files in the root (/) directory larger than 1 MB, use the following command:

```
# find / -xdev -size  +1024 -ls |sort -r  +6
```

This command finds all files greater than 1 MB and sorts them in reverse order with the largest files first. Other flags for the **find** command, such as -newer, might be useful in this search. For detailed information, see the command description for the **find** command.

> **Note:** When checking the root directory, major and minor numbers for devices in the /dev directory will be interspersed with real files and file sizes. Major and minor numbers, which are separated by a comma, can be ignored.

Before removing any files, use the **fuser** command to ensure a file is not currently in use by a user process:

```
fuser filename
```

Where *filename* is the name of the suspect large file. If a file is open at the time of removal, it is only removed from the directory listing. The blocks allocated to that file are not freed until the process holding the file open is killed.

8.2.2 Fix a full /var file system

Check the following when the /var file system has become full:

- You can use the **find** command to look for large files in the /var directory. For example:

  ```
  # find /var -xdev -size  +2048 -ls| sort -r  +6
  ```

  For detailed information, see the command description for the **find** command.
- Check for obsolete or leftover files in /var/tmp.
- Check the size of the /var/adm/wtmp file, which logs all logins, rlogins, and telnet sessions. The log will grow indefinitely unless system accounting is running. System accounting clears it out nightly. The /var/adm/wtmp file can be cleared out or edited to remove old and unwanted information. To clear it, use the following command:

  ```
  # cp /dev/null  /var/adm/wtmp
  #
  ```

To edit the /var/adm/wtmp file, first copy the file temporarily with the following command:

```
# /usr/sbin/acct/fwtmp < /var/adm/wtmp >/tmp/out
#
```

Edit the /tmp/out file to remove unwanted entries, then replace the original file with the following command:

```
# /usr/sbin/acct/fwtmp -ic < /tmp/out > /var/adm/wtmp
```

- Clear the error log in the /var/adm/ras directory using the following procedure. The error log is never cleared unless it is manually cleared.

> **Note:** Never use the **`cp /dev/null`** command to clear the error log. A zero length errlog file disables the error logging functions of the operating system and must be replaced from a backup.

a. Stop the **`error`** daemon using the following command:

```
# /usr/lib/errstop
```

b. Remove or move the error log file to a different file system by using one of the following commands:

```
# rm /var/adm/ras/errlog
```

or

```
# mv /var/adm/ras/errlog filename
```

Where *filename* is the name of the moved errlog file.

> **Note:** The historical error data is deleted if you remove the error log file.

c. Restart the **`error`** daemon using the following command:

```
# /usr/lib/errdemon
#
```

> **Note:** Consider limiting the errorlog by running the following entries in cron:
>
> ```
> 0 11 * * * /usr/bin/errclear -d S,O 30
> 0 12 * * * /usr/bin/errclear -d H 90
> ```

- Check whether the trcfile file in this directory is large. If it is large and a trace is not currently being run, you can remove the file using the following command:

```
# rm /var/adm/ras/trcfile
```

- If your dump device is set to hd6 (which is the default), there might be a number of vmcore* files in the /var/adm/ras directory. If their file dates are old or you do not want to retain them, you can remove them with the **rm** command.
- Check the /var/spool directory, which contains the queuing subsystem files. Clear the queuing subsystem using the following commands:

```
# stopsrc -s qdaemon
0513-044 The qdaemon Subsystem was requested to stop.
# rm /var/spool/lpd/qdir/*
# rm /var/spool/lpd/stat/*
# rm /var/spool/qdaemon/*
# startsrc -s qdaemon
0513-059 The qdaemon Subsystem has been started. Subsystem PID is 291042.
#
```

- Check the /var/adm/acct directory, which contains accounting records. If accounting is running, this directory may contain several large files.
- Check the /var/preserve directory for terminated vi sessions. Generally, it is safe to remove these files. If a user wants to recover a session, you can use the **vi -r** command to list all recoverable sessions. To recover a specific session, use **vi -r *filename***.
- Modify the /var/adm/sulog file, which records the number of attempted uses of the **su** command and whether each was successful. This is a flat file and can be viewed and modified with a favorite editor. If it is removed, it will be recreated by the next attempted **su** command.
- Modify the /var/tmp/snmpd.log, which records events from the **snmpd** daemon. If the file is removed, it will be recreated by the **snmpd** daemon.

> **Note:** The size of the /var/tmp/snmpd.log file can be limited so that it does not grow indefinitely. Edit the /etc/snmpd.conf file to change the number (in bytes) in the appropriate section for size.

8.2.3 Fix a full user defined file system

Use this procedure to fix an overflowing user-defined file system.

- Remove old backup files and core files. The following command removes all *.bak, .*.bak, a.out, core, *, or ed.hup files:

```
find / \( -name "*.bak" -o -name core -o -name a.out -o \
-name "...*" -o -name ".*.bak" -o -name ed.hup \) \
-atime +1 -mtime +1 -type f -print | xargs -e rm -f
```

- To prevent files from regularly overflowing the disk, run the **skulker** command as part of the cron process and remove files that are unnecessary or temporary.

  The **skulker** command purges files in the /tmp directory, files older than a specified age, a.out files, core files, and ed.hup files. It is run daily as part of an accounting procedure run by the **cron** command during off peak periods (assuming you have turned on accounting).

  The **cron** daemon runs shell commands at specified dates and times. Regularly scheduled commands such as **skulker** can be specified according to instructions contained in the crontab files. Submit crontab files with the **crontab** command. To edit a system or root crontab file, you must have root user authority. Users can edit their own crontab files.

8.2.4 Fix other file systems and general search techniques

Use the **find** command with the -size flag to locate large files or, if the file system recently overflowed, use the -newer flag to find recently modified files. To produce a file for the -newer flag to find against, use the following touch command:

```
touch mmddhhmm filename
```

Where *mm* is the month, *dd* is the date, *hh* is the hour in 24–hour format, *mm* is the minute, and *filename* is the name of the file you are creating with the **touch** command.

After you have created the touched file, you can use the following command to find newer large files:

```
find /filesystem_name -xdev -newer touch_filename -ls
```

You can also use the **find** command to locate files that have been changed in the last 24 hours, as shown in the following example:

```
find /filesystem_name -xdev -mtime 0 -ls
```

8.2.5 Fix a damaged file system

File systems can get corrupted when the i-node or superblock information for the directory structure of the file system gets corrupted. This can be caused by a hardware related ailment or by a program that gets corrupted that accesses the i-node or superblock information directly. (Programs written in assembler and C can bypass the operating system and write directly to the hardware.) One symptom of a corrupt file system is that the system cannot locate, read, or write data located in the particular file system.

To fix a damaged file system, you must diagnose the problem and then repair it. The **fsck** command performs low level diagnosis and repairs.

1. With root authority, unmount the damaged file system using one of the following SMIT fast paths: **smit unmountfs** (for a file system on a fixed disk drive) or **smit unmntdsk** (for a file system on a removable disk).
2. Assess file system damage by running the **fsck** command. In the following example, the **fsck** command checks the unmounted file system located on the /dev/myfilelv device:

   ```
   # fsck /dev/myfilelv
   ```

   The **fsck** command checks and interactively repairs inconsistent file systems. Normally, the file system is consistent, and the **fsck** command merely reports on the number of files, used blocks, and free blocks in the file system. If the file system is inconsistent, the **fsck** command displays information about the inconsistencies found and prompts you for permission to repair them. The **fsck** command is conservative in its repair efforts and tries to avoid actions that might result in the loss of valid data. In certain cases, however, the **fsck** command recommends the destruction of a damaged file.
3. If the file system cannot be repaired, restore it from backup.

8.3 The system error log

AIX 5L provides a error logging facility for recording hardware and software failures in an error log. This error log can be used for information purposes or for fault detection and corrective actions.

The error logging process begins when an operating system module detects an error. The error detecting segment of code then sends error information to either the errsave and errlast kernel service or the errlog application subroutine where the information is, in turn, written to the /dev/error special file. This process then adds a time stamp to the collected data. You can use the **errpt** command to retrieve an error record from the error log.

8.3.1 Starting and stopping error logging

Error logging is automatically started by the rc.boot script during system initialization and is automatically stopped by the shutdown script during system shutdown.

The errdemon program starts the error logging daemon, reads error records from the /dev/error file, and writes entries to the system error log. The default system error log is maintained in the /var/adm/ras/errlog file. The last error entry is placed in nonvolatile random access memory (NVRAM). During system startup, this last error entry is read from NVRAM and added to the error log when the error logging daemon is started.

The errdemon program

The errdemon program is normally started automatically during system startup, however, if it has been terminated for any reason and you need to restart it, enter the following command:

```
# /usr/lib/errdemon
```

The errstop command

The **errstop** command stops the error logging daemon initiated by the errdemon program:

```
# /usr/lib/errstop
#
```

> **Attention:** Running the **errstop** command disables diagnostic and recovery functions. The error log should never be stopped during normal operations. Stopping the error logging daemon can cause error data temporarily stored in internal buffers to be overwritten before it can be recorded in the error log file. The **errstop** command should only be used during special circumstances when it is absolutely required and the consequences are clearly understood.

Error log file

To determine the path to your system's error log file, run the following command:

```
# /usr/lib/errdemon -l
Error Log Attributes
--------------------------------------------
Log File                /var/adm/ras/errlog
Log Size                1048576 bytes
Memory Buffer Size      32768 bytes
Duplicate Removal       true
Duplicate Interval      10000 milliseconds
Duplicate Error Maximum 1000
#
```

To change the maximum size of the error log file, enter:

```
# /usr/lib/errdemon -s 2000000
#
```

To change the size of the error log device driver's internal buffer, enter:

```
# /usr/lib/errdemon -B 64000
0315-175 The error log memory buffer size you supplied will be rounded upto a
multiple of 4096 bytes.
#
```

8.3.2 The errpt command

To retrieve the entries in the error log, use the **errpt** command. The **errpt** command generates an error report from entries in an error log. It includes flags for selecting errors that match specific criteria. By using the default condition, you can display error log entries in the reverse order in which they occurred and were recorded.

Some of the most commonly used flags used with the **errpt** command are shown in Table 8-2.

Table 8-2 Commonly used flags for the errpt command

Flag	Description
-a	Displays information about errors in the error log file in a detailed format. If used in conjunction with the - t flag, all the information from the template file is displayed.
-j *ErrorID[,ErrorID]*	Includes only the error-log entries specified by the ErrorID (error identifier) variable. The ErrorID variables can be separated by commas (,) or enclosed in double quotation marks ("") and separated by commas (,) or space characters. When combined with the -t flag, entries are processed from the error-template repository.
-s *StartDate*	Specifies all records posted after the StartDate variable, where the StartDate variable has the form *mmddhhmmyy* (month, day, hour, minute, and year).
-e *EndDate*	Specifies all records posted prior to and including the EndDate variable, where the EndDate variable has the form mmddhhmmyy (month, day, hour, minute, and year).

The following examples show some uses of the **errpt** command:

- To display a complete summary report of the errors that have been recorded, run:

```
# errpt
IDENTIFIER TIMESTAMP  T C RESOURCE_NAME  DESCRIPTION
26120107   1128173005 U S LIBLVM         PHYSICAL VOLUME DEFINED AS MISSING
EC0BCCD4   1128173005 T H ent0           ETHERNET DOWN
67145A39   1128173005 U S SYSDUMP        SYSTEM DUMP
F48137AC   1128173005 U O minidump       COMPRESSED MINIMAL DUMP
9DBCFDEE   1128173005 T O errdemon       ERROR LOGGING TURNED ON
EA88F829   1128134005 I O SYSJ2          USER DATA I/O ERROR
A39F8A49   1128134005 T S syserrlg       ERROR LOGGING BUFFER OVERFLOW
8527F6F4   1128134005 P S SYSVMM         NO PAGING SPACE AVAILABLE
#
```

- To display all the errors which have an specific error ID, run:

```
# errpt -j 8527F6F4
IDENTIFIER TIMESTAMP  T C RESOURCE_NAME  DESCRIPTION
8527F6F4   1128134005 P S SYSVMM         NO PAGING SPACE AVAILABLE
#
```

- To display all the errors logged in a specific period of time, run:

```
# errpt -s 1122164405 -e 1123100405
IDENTIFIER TIMESTAMP  T C RESOURCE_NAME  DESCRIPTION
2BFA76F6   1123092705 T S SYSPROC        SYSTEM SHUTDOWN BY USER
A6DF45AA   1122183705 I O RMCdaemon      The daemon is started.
9DBCFDEE   1122164405 T O errdemon       ERROR LOGGING TURNED ON
#
```

8.3.3 The errclear command

The **errclear** command deletes entries from the error log. For example:

- To delete all the entries from the error log, type:

```
# errclear 0
#
```

- To delete all entries in the error log classified as software errors, type:

```
# errclear -d S 0
#
```

8.3.4 The errlogger command

The **errlogger** command allows you to log operator messages to the system error log. These messages can be up to 1024 bytes in length.

The use of the **errlogger** command and its output are shown in the following example:

```
# errlogger "This is a test of the errlogger command"
# errpt
IDENTIFIER TIMESTAMP  T C RESOURCE_NAME  DESCRIPTION
AA8AB241   1129134705 T O OPERATOR       OPERATOR NOTIFICATION
F89FB899   1128150005 P O dumpcheck      The copy directory is too small.
A6DF45AA   1127123605 I O RMCdaemon      The daemon is started.
9DBCFDEE   1127123605 T O errdemon       ERROR LOGGING TURNED ON
A6DF45AA   1127121105 I O RMCdaemon      The daemon is started.
#
```

To display the operator notification generated (id AA8AB241) in the previous example, type:

```
# errpt -a -j AA8AB241
-------------------------------------------
LABEL:          OPMSG
IDENTIFIER:     AA8AB241

Date/Time:       Tue Nov 29 13:47:43 CST 2005
Sequence Number: 24
Machine Id:      00C478DE4C00
Node Id:         server2
Class:           O
Type:            TEMP
Resource Name:   OPERATOR

Description
OPERATOR NOTIFICATION

User Causes
User Causes
ERRLOGGER COMMAND

        Recommended Actions
        REVIEW DETAILED DATA

Detail Data
MESSAGE FROM ERRLOGGER COMMAND
This is a test of the errlogger command
#
```

8.3.5 Extracting error records from a system dump

The **errdead** command extracts error records from a system dump containing the internal buffer maintained by the /dev/error file. The **errdead** command extracts the error records from the dump file and adds those error records directly to the error log.

Note: The error log daemon must not be running when the **errdead** command is run.

For example, to capture error log information from a dump image that resides in the /dev/hd7 file, enter:

```
# /usr/lib/errdead /dev/hd7
```

Error logging information is in the dump image if the **errdemon** daemon was not running when the dump occurred.

8.3.6 Redirecting syslog messages to error log

Some applications use syslog for logging errors and other events. To list error log messages and syslog messages in a single report, you can redirect the syslog messages to the error log. You can do this by specifying errlog as the destination in the /etc/syslog.conf configuration file.

8.3.7 Other commands for manipulating error messages

There are other error commands that are not used to identify errors on the system but to create customized error messages and install them on the system's error log message catalog:

errinstall Installs messages in the error logging message sets.

errupdate Updates the error record template repository.

errmsg Adds a message to the error log message catalog.

errupdate Updates the error record template repository.

ras_logger Provides a way to log any error from the command line. It can be used to test newly created error templates and provides a way to log an error from a shell script.

8.4 The system log configuration

To log system messages, AIX 5L uses syslogd. The **syslogd** daemon reads a datagram socket and sends each message line to a destination described by the /etc/syslog.conf configuration file. The **syslogd** daemon reads the configuration file when it is activated and when it receives a hang-up signal.

The **syslogd** daemon creates the /etc/syslog.pid file. This file contains a single line with the command process ID of the **syslogd** daemon. It is used to end or reconfigure the **syslogd** daemon.

A terminate signal sent to the **syslogd** daemon ends the daemon. The **syslogd** daemon logs the end-signal information and terminates immediately.

Each message is one line. A message can contain a priority code marked by a digit enclosed in angle braces (< >) at the beginning of the line. Messages longer than 900 bytes may be truncated.

The /usr/include/sys/syslog.h include file defines the facility and priority codes used by the configuration file. Locally written applications use the definitions contained in the syslog.h file to log messages using the **syslogd** daemon.

The general syntax of the **syslogd** command is as follows:

`syslogd [ -d ] [ -s ] [ -f` *ConfigurationFile* `] [ -m` *MarkInterval* `] [-r]`

The flags commonly used when starting syslogd are provided in Table 8-3.

Table 8-3 Commonly used flags for the syslogd daemon

Flag	Description
-d	Turns on debugging.
-f *Config File*	Specifies an alternate configuration file.
-m *MarkInterval*	Specifies the number of minutes between the **mark** command messages. If you do not use this flag, the **mark** command sends a message with LOG_INFO priority every 20 minutes. This facility is not enabled by a selector field containing an * (asterisk), which selects all other facilities.
-s	Specifies to forward a shortened message to another system (if it is configured to do so) for all the forwarding syslogd messages generated on the local system.
-r	Suppresses logging of messages received from remote hosts.

The **syslogd** daemon uses a configuration file to determine where to send a system message depending on the message's priority level and the facility that generated it. By default, syslogd reads the default configuration file /etc/syslog.conf, but if you specify the -f flag, you can specify an alternate configuration file.

8.4.1 The syslogd configuration file

The /etc/syslog.conf file controls the behavior of the **syslogd** daemon. For example, syslogd uses /etc/syslog.conf file to determine where to send the error messages or how to react to different system events. The following is a part of the default /etc/syslog.conf file:

```
# @(#)34        1.11  src/bos/etc/syslog/syslog.conf, cmdnet, bos530 4/27/04
14:
47:53
# IBM_PROLOG_BEGIN_TAG
# This is an automatically generated prolog.
#
# bos530 src/bos/etc/syslog/syslog.conf 1.11
#
# Licensed Materials - Property of IBM
#
# (C) COPYRIGHT International Business Machines Corp. 1988,1989
# All Rights Reserved
#
# US Government Users Restricted Rights - Use, duplication or
# disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
#
# IBM_PROLOG_END_TAG
#
.
.
.
(lines omitted)
.
.
.
# /etc/syslog.conf - control output of syslogd
#
#
# Each line must consist of two parts:-
#
# 1) A selector to determine the message priorities to which the
#    line applies
# 2) An action.
#
# Each line can contain an optional part:-
#
# 3) Rotation.
#
# The fields must be separated by one or more tabs or spaces.
#
# format:
#
```

```
# <msg_src_list> <destination> [rotate [size <size> k|m] [files <files>] [time
<
time> h|d|w|m|y] [compress] [archive <archive>]]
#
# where <msg_src_list> is a semicolon separated list of <facility>.<priority>
# where:
#
# <facility> is:
#       * - all (except mark)
#       mark - time marks
.
.
.
(lines omitted)
.
.
.
# example:
# "mail messages, at debug or higher, go to Log file. File must exist."
# "all facilities, at debug and higher, go to console"
# "all facilities, at crit or higher, go to all users"
#  mail.debug           /usr/spool/mqueue/syslog
#  *.debug              /dev/console
#  *.crit                       *
#  *.debug              /tmp/syslog.out     rotate size 100k files 4
#  *.crit               /tmp/syslog.out     rotate time 1d
```

In addition to the /etc/syslog.conf file that contains the settings for the **syslogd** daemon, the /etc/syslog.pid file contains the process ID of the running **syslogd** daemon.

8.4.2 The format of the configuration file

This section describes what the format of the /etc/syslog.conf file is and how you can interpret the different entries in this file. Lines in the configuration file for the **syslogd** daemon contain a selector field and an action field separated by one or more tabs.

The selector field names a facility and a priority level. Separate the facility names with a comma (,), separate the facility and priority-level portions of the selector field with a period (.), and separate multiple entries in the same selector field with a semicolon (;). To select all facilities, use an asterisk (*).

The action field identifies a destination (file, host, or user) to receive the messages. If routed to a remote host, the remote system will handle the message as indicated in its own configuration file. To display messages on a user's terminal, the destination field must contain the name of a valid, logged-in system user.

Facilities

Table 8-4 lists some of the facilities used in the /etc/syslog.conf file. You can use these system facility names in the selector field.

Table 8-4 Facilities used in the /etc/syslog.conf file

Facility	Description
kern	Kernel
user	User level
mail	Mail subsystem
daemon	System daemons
auth	Security or authorization
syslog	`syslogd` daemon
lpr	Line-printer subsystem
news	News subsystem
uucp	uucp subsystem
*	All facilities

Priority levels

Table 8-5 lists the priority levels used in the /etc/syslog.conf file. You can use the message priority levels in the selector field. Messages of the specified priority level and all levels above it are sent as directed.

Table 8-5 Priority levels for the /etc/syslog.conf file

Priority Level	Description
emerg	Specifies emergency messages (LOG_EMERG). These messages are not distributed to all users. LOG_EMERG priority messages can be logged into a separate file for reviewing.
alert	Specifies important messages (LOG_ALERT), such as a serious hardware error. These messages are distributed to all users.
crit	Specifies critical messages not classified as errors (LOG_CRIT), such as improper login attempts. LOG_CRIT and higher-priority messages are sent to the system console.
err	Specifies messages that represent error conditions (LOG_ERR), such as an unsuccessful disk write.
warning	Specifies messages for abnormal, but recoverable, conditions (LOG_WARNING).
notice	Specifies important informational messages (LOG_NOTICE). Messages without a priority designation are mapped into this priority. These are more important than informational messages, but not warnings.
info	Specifies informational messages (LOG_INFO). These messages can be discarded, but are useful in analyzing the system.
debug	Specifies debugging messages (LOG_DEBUG). These messages may be discarded.
none	Excludes the selected facility. This priority level is useful only if preceded by an entry with an * (asterisk) in the same selector field.

Destinations

Table 8-6 lists a few of the destinations that are used in the /etc/syslog.conf file. You can use these message destinations in the action field.

Table 8-6 Destination description for the /etc/syslog.conf file

Destination	Description
File Name	Full path name of a file opened in append mode.
@Host	Host name, preceded by @ (at sign).
User[, User][...]	User names.
*	All users.

8.4.3 Using the system log

To customize the /etc/syslog.conf file so that your required conditions are met, the system log should be updated by editing the /etc/syslog.conf file. After you have edited and added your lines to the /etc/syslog.conf file, you need to restart the **syslogd** daemon. You can do this by running the following commands:

1. Check to see what the **syslogd** daemon process ID is. In this case, it is 217228:

```
# ps -ef | grep syslogd
    root 217228 114906   0   Nov 16      -  0:00 /usr/sbin/syslogd
    root 430306 290870   0 14:18:11  pts/0  0:00 grep syslogd
```

2. Use the **stopsrc** command to stop the **syslogd** daemon as follows:

```
# stopsrc -s syslogd
0513-044 The syslogd Subsystem was requested to stop.
```

3. Check if the **syslogd** daemon has been stopped successfully:

```
# ps -ef | grep syslogd
root 364610 290870   0 14:20:22  pts/0  0:00 grep syslogd
```

4. Use the **startsrc** command to restart the **syslogd** daemon:

```
# startsrc -s syslogd
0513-059 The syslogd Subsystem has been started. Subsystem PID is 471258.
```

The following are a few examples of the /etc/syslog.conf file usage:

- To log all mail facility messages at the debug level or above to the file /tmp/mailsyslog, enter:

  ```
  mail.debug /tmp/mailsyslog
  ```

  Where:

  – *mail* is the Facility, as per Table 8-4 on page 307.

  – *debug* is the Priority Level, as per Table 8-5 on page 308.

  – */tmp/mailsyslog* is the Destination, as per Table 8-6 on page 309.

- To send all system messages except those from the mail facility to a host named rigil, enter:

  ```
  *.debug;mail.none @rigil
  ```

  Where:

  – * and *mail* are the Facilities, as per Table 8-4 on page 307.

  – *debug* and *none* are the Priority Levels, as per Table 8-5 on page 308.

  – *@rigil* is the Destination, as per Table 8-6 on page 309.

- To send messages at the emerg priority level from all facilities and messages at the crit priority level and above from the mail and daemon facilities to users nick and jam, enter:

  ```
  *.emerg;mail,daemon.crit nick, jam
  ```

  Where:

  – *, *mail*, and *daemon* are the Facilities, as per Table 8-4 on page 307.

  – *emerg* and *crit* are the Priority Levels, as per Table 8-5 on page 308.

  – *nick* and *jam* are the Destinations, as per Table 8-6 on page 309.

- To send all mail facility messages to all users' terminal screens, enter:

  ```
  mail.debug *
  ```

  Where:

  – *mail* is the Facility, as per Table 8-4 on page 307.

  – *debug* is the Priority Level, as per Table 8-5 on page 308.

  – * is the Destination, as per Table 8-6 on page 309.

8.5 Performance tools overview

The following sections introduce the most useful performance tools.

8.5.1 The vmstat command

The **vmstat** command reports statistics about kernel threads, virtual memory, disks, traps, and CPU activity.

> **Note:** Beginning with AIX 5L V5.3, the **vmstat** command reports the number of physical processors consumed (pc), and the percentage of entitlement consumed (ec), in Micro-Partitioning™ environments. These metrics will only be displayed on Micro-Partitioning environments.

Reports generated by the **vmstat** command can be used to balance system load activity. These system-wide statistics (among all processors) are calculated either as averages for values expressed as percentages, or as sums. If the **vmstat** command is invoked without flags, the report contains a summary of the virtual memory activity since system startup.

The syntax of the **vmstat** command is:

```
vmstat [ -f ] [ -i ] [ -s  ] [ -I ] [ -t ] [ -v ] [ -w] [ -l ]
[ PhysicalVolume ... ] [ Interval [ Count ] ]
```

Table 8-7 provides the key flags for the **vmstat** command.

Table 8-7 Commonly used flags for the vmstat command

Flag	Description
-f	Reports the number of forks since system startup.
-i	Displays the number of interrupts taken by each device since system startup.
-s	Writes the contents of the sum structure to standard output, which contains an absolute count of paging events since system initialization. The -s option is exclusive of the other **vmstat** command options.

The PhysicalVolume parameter can be used to specify one to four names. Transfer statistics are given for each specified drive in the order specified. This count represents logical and physical requests to the physical device. It does not imply an amount of data that was read or written. Several logical requests can be combined into one physical request.

The Interval parameter specifies the amount of time in seconds between each report. The first report contains statistics for the time since system startup. Subsequent reports contain statistics collected during the interval since the previous report. If the Interval parameter is not specified, the **vmstat** command generates a single report and then exits.

The Count parameter can only be specified with the Interval parameter. If the Count parameter is specified, its value determines the number of reports generated and the number of seconds apart. If the Interval parameter is specified without the Count parameter, reports are continuously generated. A Count parameter of zero (0) is not allowed. The kernel maintains statistics for kernel threads, paging, and interrupt activity, which the **vmstat** command accesses. The disk input/output statistics are maintained by device drivers. For disks, the average transfer rate is determined by using the active time and number of transfers information. The percent active time is computed from the amount of time the drive is busy during the report.

The vmstat command report output

The reports generated by the **vmstat** command contains the column headings shown in Table 8-8.

Table 8-8 vmstat output column headings

Column	Description
Kthr: Kernel thread state.	
r	Average number of runnable kernel threads over the sampling interval. Runnable refers to threads that are ready but waiting to run and to those threads already running.
b	Average number of kernel threads placed in the VMM wait queue (awaiting resource and awaiting input/output) over the sampling interval.
Memory: Information about the usage of virtual and real memory. Virtual pages are considered active if they have been accessed.	
avm	Active virtual pages, that is, the total number of pages allocated in page space. A high value is not an indicator of poor performance.
fre	Size of the free list. A large portion of real memory is utilized as a cache for file system data. It is not unusual for the size of the free list to remain small.
Page: information about page faults and paging activity. These are averaged over the interval and given in units per second.	
re	Pager input/output list.

Column	Description
pi	Pages paged in from paging space.
po	Pages paged out to paging space.
fr	Pages freed (page replacement).
sr	Pages scanned by page-replacement algorithm.
cy	Clock cycles used by page-replacement algorithm.
Faults: Trap and interrupt rate averages per second over the sampling interval.	
in	Device interrupts.
sy	System calls.
cs	Kernel thread context switches.
CPU: breakdown of percentage usage of CPU time.	
us	User time
sy	System time
id	CPU idle time
wa	CPU idle time during which the system had outstanding disk/NFS I/O request(s).
pc	Number of physical processors consumed. Displayed only if the partition is running with shared processor. (only displayed on micro-partitioned environments).
ec	The percentage of entitled capacity consumed. Displayed only if the partition is running with shared processor. (only displayed on micro-partitioned environments).
Disk: Provides the number of transfers per second to the specified physical volumes that occurred in the sample interval.	

The following examples show some uses for the **vmstat** command:

- To display five summaries at 1 second intervals, type the following command:

```
# vmstat 1 5

System configuration: lcpu=4 mem=3792MB

kthr    memory              page              faults        cpu
----- ----------- ------------------------ ------------ -----------
 r  b   avm   fre  re  pi  po  fr   sr  cy  in   sy  cs us sy id wa
 0  0 159012 649264   0   0   0   0    0   0   9 1266  82  0  0 99  0
 0  0 159013 649263   0   0   0   0    0   0   6 1154  84  0  0 99  0
 0  0 159013 649263   0   0   0   0    0   0   4 1152  82  0  0 99  0
 0  0 159013 649263   0   0   0   0    0   0   6 1158  90  0  0 99  0
 0  0 159013 649263   0   0   0   0    0   0   4 1160  85  0  1 99  0
#
```

- To display the count of various events, type the following command:

```
# vmstat -s
               367947 total address trans. faults
                22409 page ins
                 4607 page outs
                    0 paging space page ins
                    0 paging space page outs
                    0 total reclaims
               215640 zero filled pages faults
                 4540 executable filled pages faults
                    0 pages examined by clock
                    0 revolutions of the clock hand
                    0 pages freed by the clock
                50503 backtracks
                    0 free frame waits
                    0 extend XPT waits
                 9018 pending I/O waits
                27017 start I/Os
                15751 iodones
                69848 cpu context switches
                15486 device interrupts
                27013 software interrupts
                90354 decrementer interrupts
                  400 mpc-sent interrupts
                  400 mpc-receive interrupts
                  234 phantom interrupts
                    0 traps
              1163018 syscalls
#
```

- To display five summaries for hdisk0 and hdisk1 at 2 seconds interval, type the following command:

```
# vmstat hdisk0 hdisk1 2 5

System configuration: lcpu=4 mem=3792MB drives=7

kthr    memory              page              faults        cpu      disk xfer
----- ----------- ------------------------ ------------ ----------- -----------
 r  b   avm   fre  re  pi  po  fr   sr  cy  in   sy  cs us sy id wa  1  2  3  4
 0  0 139021 817059   0   0   0   0    0   0   4 1200  93  0  0 99  0 9277 1151
 -  -
 0  0 139022 817058   0   0   0   0    0   0   2 1142  82  0  0 99  0  0  0  -
-
 0  0 139022 817058   0   0   0   0    0   0   3 1148  78  0  0 99  0  0  0  -
-
 0  0 139022 817058   0   0   0   0    0   0   3 1143  81  0  0 99  0  0  0  -
-
 0  0 139022 817058   0   0   0   0    0   0   2 1142  78  0  0 99  0  0  0  -
-
#
```

- To display the number of forks since system startup, type the following command:

```
# vmstat -f
                 2792 forks
```

Start some processes and check them after the count of forks:

```
# for proc in 1 2 3 4 5
> do
>    lsvg > /dev/null
> done
# vmstat -f
                 2798 forks
#
```

The difference from the last execution is 6, one for the **for** command, and five for the **lsvg > /dev/null** command.

8.5.2 The sar command

The **sar** command collects, reports, or saves system activity information.

The default report of the **sar** command (CPU utilization report) might be one of the first facilities the system administrator must run to begin system activity investigation, because it monitors major system resources.

You can select information about specific system activities using flags. Not specifying any flags selects only system unit activity. Specifying the -A flag selects all activities. The **sar** command prints the number of CPUs and the number of disks that are currently active before starting to print the statistics.

Note: Beginning with AIX 5L V5.3, the **sar** command reports utilization metrics physc and %entc, which are related to Micro-Partitioning and simultaneous multithreading environments. These metrics will only be displayed on Micro-Partitioning and simultaneous multithreading environments. physc indicates the number of physical processors.

sar report output

The reports generated by the **sar** command contains the column headings shown in Table 8-9.

Table 8-9 Column headings of the sar command

Column	Description
%idle	Reports the percentage of time the CPU or CPUs were idle with no outstanding disk I/O requests.
%sys	Reports the percentage of time the CPU or CPUs spent in execution at the system (or kernel) level.
%usr	Reports the percentage of time the CPU or CPUs spent in execution at the user (or application) level.
physc	Reports the number of physical processors consumed. This will be reported only if the partition is running with shared processors or simultaneous multithreading enabled.
%entc	Reports the percentage of entitled capacity consumed. This will be reported only if the partition is running with shared processors.

If CPU utilization is near 100 percent (%user + %sys columns), the workload sampled is CPU-bound. If a considerable percentage of time is spent in I/O wait, it implies that CPU execution is blocked waiting for disk I/O. The I/O may be required file accesses or it may be I/O associated with paging due to a lack of sufficient memory.

Note: The **sar** command reports system unit activity if no other specific content options are requested. If the -P flag is used and the partition is running with shared processors, and if the partition capacity usage is what is allocated, then a CPU row with cpuid U will be reported to show the system-wide unused capacity.

If the partition is running with shared processors in uncapped mode, then %entc will report the percentage of granted entitled capacity against each CPU row and percentage of allocated entitled capacity in the system-wide CPU row.

The following examples show some uses for the **sar** command:

- To report current activity for each two seconds for the next five seconds, enter:

```
# sar 2 5

AIX server2 3 5 00C478DE4C00    12/02/05

System configuration: lcpu=4

16:26:29    %usr    %sys    %wio   %idle   physc
16:26:31       0       0       0     100    2.00
16:26:33       0       0       0     100    2.00
16:26:35       0       0       0     100    2.00
16:26:37       0       0       0     100    2.00
16:26:39       0       0       0     100    2.00

Average        0       0       0     100    2.00
#
```

- To report activity for the first two processors for each second for the next five seconds, enter:

```
# sar -u  -P 0,1 1 5

AIX server2 3 5 00C478DE4C00    12/02/05

System configuration: lcpu=4

16:28:55 cpu    %usr    %sys    %wio   %idle   physc
16:28:56  0        0       0       0     100    0.48
          1        0       0       0     100    0.48
16:28:57  0        0       0       0     100    0.53
          1        0       0       0     100    0.48
16:28:58  0        0       0       0     100    0.53
          1        0       0       0     100    0.48
```

```
16:28:59  0         0        0        0      100     0.53
          1         0        0        0      100     0.48
16:29:00  0         0        0        0      100     0.53
          1         0        0        0      100     0.48

Average   0         0        0        0      100     0.52
          1         0        0        0      100     0.48
#
```

8.5.3 The topas command

The **topas** command reports vital statistics about the activity on the local system on a character terminal. The **topas** command extracts and displays statistics from the system with a default interval of two seconds. The **topas** command offers the following alternate screens:

- Overall system statistics
- List of busiest processes
- WLM statistics

> **Note:** The bos.perf.tools and perfagent.tools filesets must be installed on the system to run the **topas** command.

Overall system statistics screen of the topas monitor

The output of the overall system statistics screen consists of one fixed section and one variable section. The top two lines at the left of the output shows the name of the system that the **topas** program is running on, the date and time of the last observation, and the monitoring interval.

The variable part of the **topas** display can have one, two, three, four, or five subsections. If more than one subsection displays, they are always shown in the following order:

- CPU
- Network Interfaces
- Physical Disks
- Workload Management Classes
- Processes

CPU utilization

This subsection displays a bar chart showing cumulative CPU usage. If more than one CPU exists, a list of CPUs can be displayed by pressing the c key twice.

Pressing the c key only once will turn this subsection off. The following fields are displayed by both formats:

User% This shows the percentage of CPU used by programs executing in user mode. (Default sorted by User%)

Kern% This shows the percentage of CPU used by programs executing in kernel mode.

Wait% This shows the percentage of time spent waiting for IO.

Idle% This shows the percentage of time the CPU(s) is idle.

Physg Number of physical processors granted to the partition (if Micro-Partitioning).

%Entg Percentage of Entitled Capacity granted to a partition (if Micro-Partitioning).

Network interfaces

This subsection displays a list of hot network interfaces. The maximum number of interfaces displayed is the number of hot interfaces being monitored, as specified with the -n flag. A smaller number of interfaces will be displayed if other subsections are also being displayed. Pressing the n key turns off this subsection. Pressing the n key again shows a one-line report summary of the activity for all network interfaces. Both reports display the following fields:

Interf The name of the network interface.

KBPS The total throughput in megabytes per second over the monitoring interval. This field is the sum of kilobytes received and kilobytes sent per second.

I-Pack The number of data packets received per second over the monitoring interval.

O-Pack The number of data packets sent per second over the monitoring interval.

KB-In The number of kilobytes received per second over the monitoring interval.

KB-Out The number of kilobytes sent per second over the monitoring interval.

Physical disks

This subsection displays a list of hot physical disks. The maximum number of physical disks displayed is the number of hot physical disks being monitored as specified with the -d flag. A smaller number of physical disks will be displayed if other subsections are also being displayed. Pressing the d key turns off this subsection. Pressing the d key again shows a one-line report summary of the activity for all physical disks. Both reports display the following fields:

Disk	The name of the physical disk.
Busy%	Indicates the percentage of time the physical disk was active (bandwidth utilization for the drive).
KBPS	The number of kilobytes read and written per second over the monitoring interval. This field is the sum of KB-Read and KB-Writ.
TPS	The number of transfers per second that were issued to the physical disk. A transfer is an I/O request to the physical disk. Multiple logical requests can be combined into a single I/O request to the disk. A transfer is of indeterminate size.
KB-Read	The number of kilobytes read per second from the physical disk.
KB-Writ	The number of kilobytes written per second to the physical disk.

WLM classes

This subsection displays a list of hot Workload Management (WLM) classes. The maximum number of WLM classes displayed is the number of hot WLM classes being monitored, as specified with the -w flag. A smaller number of classes will be displayed if other subsections are also being displayed. Pressing the w key turns off this subsection. The following fields are displayed for each class:

% CPU Utilization	The average CPU utilization of the WLM class over the monitoring interval.
% Mem Utilization	The average memory utilization of the WLM class over the monitoring interval.
% Blk I/O	The average percent of Block I/O of the WLM class over the monitoring interval.

Processes

This subsection displays a list of hot processes. The maximum number of processes displayed is the number of hot processes being monitored as specified with the -p flag. A smaller number of processes will be displayed if other subsections are also being displayed. Pressing the p key turns off this subsection. The processes are sorted by their CPU usage over the monitoring interval. The following fields are displayed for each process:

Name	The name of the executable program executing in the process. The name is stripped of any path name and

argument information and truncated to nine characters in length.

Process ID The process ID of the process.

% CPU Utilization The average CPU utilization of the process over the monitoring interval. The first time a process is shown, this value is the average CPU utilization over the lifetime of the process.

Paging Space Used The size of the paging space allocated to this process. This can be considered an expression of the footprint of the process but does not include the memory used to keep the executable program and any shared libraries it may depend on.

Process Owner The user name of the user who owns the process (if the WLM section is off).

WLM Class The Workload Management class to which the process belongs (if the WLM section is on).

Events/queues

Displays the per-second frequency of selected system-global events over the monitoring interval, and the average size of the thread run and wait queues:

Cswitch The number of context switches.

Syscalls The total number of system calls.

Reads The number of read system calls.

Writes The number of write system calls.

Forks The number of fork system calls.

Execs The number of exec system calls.

Runqueue The average number of threads that were ready to run but were waiting for a processor to become available.

Waitqueue The average number of threads that were waiting for paging to complete.

File/TTY

Displays the per-second frequency of selected file and TTY statistics.

Readch The number of bytes read per second through the read system call over the monitoring interval.

Writech The number of bytes written per second through the write system call over the monitoring interval.

Rawin	The number of raw bytes read per second from TTYs over the monitoring interval.
Ttyout	The number of bytes written to TTYs per second over the monitoring interval.
Igets	The number of calls per second to the inode lookup routines over the monitoring interval.
Namei	The number of calls per second to the path name lookup routines over the monitoring interval.
Dirblk	The number of directory blocks scanned per second by the directory search routine over the monitoring interval.

Paging

Displays the per-second frequency of paging statistics.

Faults	Total number of page faults taken per second over the monitoring interval. This includes page faults that do not cause paging activity.
Steals	Physical memory 4 K frames stolen per second by the virtual memory manager over the monitoring interval.
PgspIn	Number of 4 K pages read from paging space per second over the monitoring interval.
PgspOut	Number of 4 K pages written to paging space per second over the monitoring interval.
PageIn	Number of 4 K pages read per second over the monitoring interval. This includes paging activity associated with reading from file systems. Subtract PgspIn from this value to get the number of 4 K pages read from file systems per second over the monitoring interval.
PageOut	Number of 4 K pages written per second over the monitoring interval. This includes paging activity associated with writing to file systems. Subtract PgspOut from this value to get the number of 4 K pages written to file systems per second over the monitoring interval.
Sios	The number of I/O requests per second issued by the virtual memory manager over the monitoring interval.

Memory

Displays the real memory size and the distribution of memory in use.

Real,MB	The size of real memory in megabytes.

% Comp The percentage of real memory currently allocated to computational page frames. Computational page frames are generally those that are backed by paging space.

% Noncomp The percentage of real memory currently allocated to non-computational frames. Non-computational page frames are generally those that are backed by file space, either data files, executable files, or shared library files.

% Client The percentage of real memory currently allocated to cache remotely mounted files.

Paging space

Displays the size and utilization of paging space.

Size,MB The sum of all paging spaces on the system, in megabytes.

% Used The percentage of total paging space currently in use.

% Free The percentage of total paging space currently free.

NFS

Displays the NFS statistics in calls per second.

- Server V2 calls/sec
- Client V2 calls/sec
- Server V3 calls/sec
- Client V3 calls/sec

Figure 8-1 shows a sample output of the overall system statistics screen.

```
Topas Monitor for host:    server3            EVENTS/QUEUES    FILE/TTY
Mon Dec  5 14:28:19 2005   Interval:  2       Cswitch      85  Readch        57
                                              Syscall    1271  Writech      289
Kernel     0.3   |#                        |  Reads         1  Rawin          0
User       0.1   |#                        |  Writes        1  Ttyout       119
Wait       0.0   |                         |  Forks         0  Igets          0
Idle      99.6   |#########################|  Execs         0  Namei          0
                                              Runqueue    0.0  Dirblk         0
Network   KBPS   I-Pack  O-Pack   KB-In  KB-Out  Waitqueue   0.0
en2        0.7      3.0     1.0     0.4     0.3
en1        0.0      0.0     0.0     0.0     0.0  PAGING           MEMORY
en0        0.0      0.0     0.0     0.0     0.0  Faults        0  Real,MB     3792
                                              Steals        0  % Comp      15.6
Disk     Busy%      KBPS     TPS KB-Read KB-Writ  PgspIn        0  % Noncomp    1.7
hdisk5     0.0      0.0     0.0     0.0     0.0  PgspOut       0  % Client     2.0
hdisk0     0.0      0.0     0.0     0.0     0.0  PageIn        0
hdisk4     0.0      0.0     0.0     0.0     0.0  PageOut       0  PAGING SPACE
                                              Sios          0  Size,MB      512
Name            PID  CPU%  PgSp Owner                          % Used       1.2
dtgreet      131162   0.1   1.4 root           NFS (calls/sec)  % Free      98.7
topas        381070   0.0   1.2 root           ServerV2      0
topas        315640   0.0   1.1 root           ClientV2      0    Press:
gil           65568   0.0   0.1 root           ServerV3      0    "h" for help
rgsr          98458   0.0   0.0 root           ClientV3      0    "q" to quit
```

Figure 8-1 Overall system statistics screen

Except for the variable Processes subsection, you can sort all of the subsections by any column by moving the cursor to the top of the desired column. All of the variable subsections, except the Processes subsection, have the following views:

- List of top resource users
- One-line report presenting the sum of the activity

For example, the one-line-report view might show just the total disk or network throughput. For the CPU subsection, you can select either the list of busy processors or the global CPU utilization, as shown in the previous example.

List of busiest processes screen

To view the screen that lists the busiest processes, use the -P flag of the **topas** command. This screen is similar to the Processes subsection of the overall system statistics screen, but with additional detail. You can sort this screen by any of the columns by moving the cursor to the top of the desired column.

Figure 8-2 on page 325 shows an example of the output of the busiest processes screen.

```
Topas Monitor for host:    server3     Interval:   2    Mon Dec  5 14:34:41 2005

                                    DATA   TEXT  PAGE                PGFAULTS
USER          PID    PPID PRI NI    RES    RES SPACE    TIME CPU%   I/O   OTH COMMAND
root       131162  106618  60 20    368     18   368   13:44  0.1     0     0 dtgreet
root       381070  368854  58 41    306     34   306    0:00  0.0     0     0 topas
root       389370  340048  58 41    280     34   280    0:00  0.0     0     0 topas
root        65568       0  37 41     29      0    29    1:03  0.0     0     0 gil
root       110846  356566  68 24    233     84   233    0:00  0.0     0     0 xterm
root       245904       1  60 20     53      0    53    0:17  0.0     0     0 nfsd
root       127072  135264  60 20    775    336   775    0:14  0.0     0     0 X
root       274588       1  60 20     51      0    51    0:12  0.0     0     0 rpc.lock
root        61470       0  36 41     12      0    12    0:01  0.0     0     0 netm
root       143440       1  60 20    129      2   129    0:53  0.0     0     0 syncd
root       327842  221296  60 20    531     80   531    0:03  0.0     0     0 IBM.CSMA
root        94346  278730  60 20    140     56   140    0:01  0.0     0     0 ksh
root        45078       0  60 41     12      0    12    0:01  0.0     0     0 xmgc
root            1       0  60 20    186     10   186    0:01  0.0     0     0 init
root        90230  221296  60 20    156     15   156    0:00  0.0     0     0 muxatmd
root        32784       0  16 41     12      0    12    0:00  0.0     0     0 memgrdd
root        98458       0  60 20     12      0    12    0:00  0.0     0     0 rgsr
root       102524  221296  60 20    263     28   263    0:00  0.0     0     0 IBM.Serv
root       106618  135264  60 20     63     39   123    0:00  0.0     0     0 dtlogin
root        28686       0  16 41     12      0    12    0:00  0.0     0     0 memp_rbd
```

Figure 8-2 Busiest processes screen

Disk metrics screen

To view the screen that lists the disk metrics, use the -D flag of the **topas** command. This screen reports disk service times, disk queuing metrics, and disk throughput. The following metrics are reported:

AQD Average number of requests waiting to be sent to disk.

AQW Average queue wait per request reported in milliseconds.

ART Indicates the average time to receive a response from the hosting server for the read request sent. The suffix indicates the unit of time. The default time unit is milliseconds.

AWT Indicates the average time to receive a response from the hosting server for the write request sent. The suffix indicates the unit of time. The default time unit is milliseconds.

MRT Indicates the maximum time to receive a response from the hosting server for the read request sent. The suffix indicates the unit of time. The default time unit is milliseconds.

MWT Indicates the maximum time to receive a response from the hosting server for the write request sent. The suffix indicates the unit of time. The default time unit is milliseconds.

Figure 8-3 shows an example of the output of the disk metrics screen.

```
Topas Monitor for host:    server3     Interval:   2    Mon Dec  5 17:26:52 2005
================================================================================
Disk       Busy%  KBPS     TPS   KB-R  ART   MRT   KB-W  AWT   MWT   AQW    AQD
hdisk0      14.0   2.8K   25.0    0.0  0.0   0.0   2.8K  6.3   9.6   0.0    0.0
hdisk1      12.0   2.8K   21.5    0.0  0.0   0.0   2.8K  6.0  17.4   0.0    0.0
hdisk4       0.0   0.0     0.0    0.0  0.0   0.0   0.0   0.0   0.0   0.0    0.0
hdisk5       0.0   0.0     0.0    0.0  0.0   0.0   0.0   0.0   0.0   0.0    0.0
hdisk2       0.0   0.0     0.0    0.0  0.0   0.0   0.0   0.0   0.0   0.0    0.0
cd0          0.0   0.0     0.0    0.0  0.0   0.0   0.0   0.0   0.0   0.0    0.0
hdisk3       0.0   0.0     0.0    0.0  0.0   0.0   0.0   0.0   0.0   0.0    0.0
```

Figure 8-3 Disk metrics screen

General topas subcommands

While the **topas** command is running, it accepts one-character subcommands. Each time the monitoring interval elapses, the program checks for one of the subcommands shown in Table 8-10 and responds to the action required.

Table 8-10 The topas command subcommands

Subcommand	Description
a	Shows all of the variable subsections being monitored (CPU, network, disk, WLM, and process). Pressing the a key always returns the **topas** command to the initial main display.
c	Toggles the CPU subsection between the cumulative report, off, and a list of the busiest CPUs. The number of busiest CPUs displayed will depend upon the space available on the screen.
C	The uppercase C key activates the Cross-Partition panel, or if currently active, resets the panel to display the global summary, dedicated, and shared sections.
d	Toggles the disk subsection between a list of busiest disks, off, and the report on the total disk activity of the system. The number of busiest disks displayed will depend upon the space available on the screen.
D	The uppercase D key replaces the current display with the Disk Metric display. This display offers additional information about disk access times and disk queuing. When the D key is pressed again, it toggles back to the default main screen.
f	Moving the cursor over a WLM class and pressing the f key displays the list of top processes in the class at the bottom of the WLM screen. This key is valid only when **topas** is in the full-screen WLM display mode (by using the W key or the -W flag).

Subcommand	Description
h	Show the help screen.
H	Shows the help screen for the local panel, if available.
L	The uppercase L key replaces the current display with the logical partition display; LPAR, Micro-Partitioning, and simultaneous multithreading metrics similar to what **lparstat** and **mpstat** provide are displayed.
n	Toggles the network interfaces subsection between a list of busiest interfaces, off, and the report on the total network activity of the system. The number of busiest interfaces displayed will depend upon the space available on the screen.
p	Toggles the hot processes subsection on and off. The number of busiest processes displayed will depend upon the space available on the screen.
q	Quits the program.
r	Refreshes the display.
w	Toggles the Workload Management (WLM) classes subsection on and off. The number of busiest WLM classes displayed will depend upon the space available on the screen.
W	The uppercase W key replaces the default display with the full-screen WLM class display. This display provides more detailed information about WLM classes and processes assigned to classes. When the W key is pressed again, it toggles back to the default main display.
Arrow and Tab keys	Subsections from the main display, such as the CPU, Network, Disk, WLM Classes, and the full-screen WLM and Process displays can be sorted by different criteria. Positioning the cursor over a column activates sorting on that column. The entries are always sorted from highest to lowest value. The cursor can be moved by using the Tab key or the arrow keys. Sorting is only valid for 128 disks and 16 network adapters.

8.5.4 The netstat command

When you are experiencing communication problems, the **netstat** command can be a very useful tool to help you identify the problem.

Interface availability, addresses, and statistics

First of all, you should verify the status of all configured interfaces using the **netstat -i** command, as shown in Example 8-1. Interface en0 is up, while interface en1 is down. By default, the **netstat** command tries to resolve the IP addresses to a host name. To display the IP address, you can additionally use the n flag, so you can use the **netstat** command to find both the MAC address and IP address for your interfaces.

Example 8-1 Using netstat -i to determine the status of all interfaces

```
# netstat -i
Name  Mtu   Network     Address            Ipkts Ierrs    Opkts Oerrs  Coll
en0   1500  link#2      0.2.55.2f.a7.e    14322381     0 14957286 19956     0
en0   1500  9.3.5       server2.itsc.aust 14322381     0 14957286 19956     0
en1*  1500  link#3      0.2.55.2f.a7.f           0     0        1     1     0
en1*  1500  9.1.1       alm-bd-as.almaden        0     0        1     1     0
lo0   16896 link#1                           12017     0    12950     0     0
lo0   16896 127         localhost            12017     0    12950     0     0
lo0   16896 ::1                              12017     0    12950     0     0
# netstat -in
Name  Mtu   Network     Address            Ipkts Ierrs    Opkts Oerrs  Coll
en0   1500  link#2      0.2.55.2f.a7.e    14321816     0 14957179 19956     0
en0   1500  9.3.5       9.3.5.195         14321816     0 14957179 19956     0
en1*  1500  link#3      0.2.55.2f.a7.f           0     0        1     1     0
en1*  1500  9.1.1       9.1.1.1                  0     0        1     1     0
lo0   16896 link#1                           12009     0    12942     0     0
lo0   16896 127         127.0.0.1            12009     0    12942     0     0
lo0   16896 ::1                              12009     0    12942     0     0
```

If the values for Ierrs and Oerrs are not zero, verify the network interfaces and cables or increase the size of the corresponding buffers. On Ethernet, the collision field is not supported and will always display 0 (zero).

Routing table

Every time an interface is configured on your system, a route for the corresponding subnetwork is added to the local routing table. If you want your system to communicate with remote systems located on networks that are not directly attached, you have to include in your routing table either a specific route for that subnetwork or a default route. Make sure that the gateway you specify is alive and resides on a directly connected network. A routing table from your system can be displayed using the **netstat -rn** command, as shown in Example 8-2 on page 329. From this example, you can see that the network 111.111.0.0 is accessible via gateway 9.3.5.254. For all other remote routes, the system will use default gateway 9.3.5.41.

Example 8-2 Displaying routing table

```
# netstat -rn
Routing tables
Destination      Gateway           Flags   Refs     Use  If   Exp  Groups

Route Tree for Protocol Family 2 (Internet):
default          9.3.5.41          UG        4     64879 en0     -      -
9.1.1.0          9.1.1.1           UHSb      0         0 en1     -      -   =>
9.1.1/24         9.1.1.1           U         0         0 en1     -      -
9.1.1.1          127.0.0.1         UGHS      0         0 lo0     -      -
9.1.1.255        9.1.1.1           UHSb      0         0 en1     -      -
9.3.5.0          9.3.5.195         UHSb      0         0 en0     -      -   =>
9.3.5/24         9.3.5.195         U         4  14699083 en0     -      -
9.3.5.195        127.0.0.1         UGHS      6      1846 lo0     -      -
9.3.5.255        9.3.5.195         UHSb      0         1 en0     -      -
111.111/16       9.3.5.254         UG        0         0 en0     -      -
127/8            127.0.0.1         U         5     10313 lo0     -      -

Route Tree for Protocol Family 24 (Internet v6):
::1              ::1               UH        0         0 lo0     -      -
```

You can use the **netstat -C** command to display route costs if you have multiple routes having different costs to the same destination. In Example 8-3, you can see that the system has four different routes, each using a different gateway and having different costs to reach network 111.111.0.0.

Example 8-3 Using netstat -C to display routes having different costs

```
# netstat -Cn
Routing tables
Destination      Gateway           Flags     Wt  Policy  If   Cost Config_Cost

Route Tree for Protocol Family 2 (Internet):
default          9.3.5.41          UG         1    -    en0      0    0
9.1.1.0          9.1.1.1           UHSb       1    -    en1      0    0 =>
9.1.1/24         9.1.1.1           U          1    -    en1      0    0
9.1.1.1          127.0.0.1         UGHS       1    -    lo0      0    0
9.1.1.255        9.1.1.1           UHSb       1    -    en1      0    0
9.3.5.0          9.3.5.195         UHSb       1    -    en0      0    0 =>
9.3.5/24         9.3.5.195         U          1    -    en0      0    0
9.3.5.195        127.0.0.1         UGHS       1    -    lo0      0    0
9.3.5.255        9.3.5.195         UHSb       1    -    en0      0    0
111.111/16       9.3.5.254         UG         1    -    en0      0    0 =>
111.111/16       9.3.5.253         UG         1    -    en0      5    5 =>
111.111/16       9.3.5.252         UG         1    -    en0     10   10 =>
111.111/16       9.3.5.251         UG         1    -    en0     15   15
127/8            127.0.0.1         U          1    -    lo0      0    0
222.222.222/24   9.1.3.5           UG         1    -    en0      0    0
```

```
Route Tree for Protocol Family 24 (Internet v6):
::1              ::1               UH          1     -     lo0        0    0
```

MTU size

The size of largest packet that can be sent over a network is named the maximum transfer unit (MTU). All devices on the same physical or logical (VLAN) network should use the same MTU size.

You can display the MTU size using the **netstat -in** command, as shown in Example 8-1 on page 328.

Memory buffers

Memory management routines keeps statistics about the way the kernel handles memory buffers (mbuf) for communication purposes. Each processor has its own mbuf pool.

You can displays these statistics using the **netstat -m** command, as shown in Example 8-4.

Example 8-4 Using the netstat -m command to display mbuf statistics

```
Kernel malloc statistics:

******* CPU 0 *******
By size           inuse     calls failed   delayed    free   hiwat   freed
32                   32        45      0         0      96    4852       0
64                   49     56964      0         1     143    4852       0
128                 229     15220      0        13     347    2426       0
256                 191   8802219      0         9    2321    4852       0
512                3245  39247629      0      1706    4003    6065   18871
1024                127      1499      0        35      25    2426       0
2048               2056     20668      0      2765    2048    3639     714
4096                 68       125      0         4     953    1213     746
8192                520      7635      0       505     668     606    2274
16384               512      3057      0       412     303     303     630
32768                 0       144      0        24       1     151       0
65536                 1       349      0        65      20     151       0
131072                0         6      0         0      86     121       0


******* CPU 1 *******
By size           inuse     calls failed   delayed    free   hiwat   freed
64                    2     35430      0         0      62    4852       0
128                   1      3136      0         1      95    2426       0
.
.
```

```
.
(lines omitted)
.
.
.
Streams mblk statistic failures:
0 high priority mblk failures
0 medium priority mblk failures
0 low priority mblk failures
```

If the number of failures is not zero, you can modify tunables thewall or maxmbuf.

Device driver statistics

You can use the **netstat -v** command to display device driver information, as shown in Example 8-5. The data will be generated using the **netstat** command.

Example 8-5 Using the netstat -v command

```
# netstat -v ent0|more
-------------------------------------------------------------
ETHERNET STATISTICS (ent0) :
Device Type: 2-Port 10/100/1000 Base-TX PCI-X Adapter (14108902)
Hardware Address: 00:02:55:2f:a7:0e
Elapsed Time: 2 days 3 hours 44 minutes 33 seconds

Transmit Statistics:                          Receive Statistics:
--------------------                          -------------------
Packets: 14928988                             Packets: 14347559
Bytes: 2150983710                             Bytes: 1895521071
Interrupts: 0                                 Interrupts: 2691358
Transmit Errors: 0                            Receive Errors: 0
Packets Dropped: 10547                        Packets Dropped: 0
                                              Bad Packets: 0
Max Packets on S/W Transmit Queue: 8656
S/W Transmit Queue Overflow: 43999
Current S/W+H/W Transmit Queue Length: 1

Broadcast Packets: 939                        Broadcast Packets: 435771
Multicast Packets: 0                          Multicast Packets: 34
No Carrier Sense: 0                           CRC Errors: 0
DMA Underrun: 0                               DMA Overrun: 0
Lost CTS Errors: 0                            Alignment Errors: 0
Max Collision Errors: 0                       No Resource Errors: 0
Late Collision Errors: 0                      Receive Collision Errors: 0
Deferred: 0                                   Packet Too Short Errors: 0
SQE Test: 0                                   Packet Too Long Errors: 0
Timeout Errors: 0                             Packets Discarded by Adapter: 0
Single Collision Count: 0                     Receiver Start Count: 0
```

```
Multiple Collision Count: 0
Current HW Transmit Queue Length: 1

General Statistics:
-------------------
No mbuf Errors: 0
Adapter Reset Count: 0
Adapter Data Rate: 200
Driver Flags: Up Broadcast Running
        Simplex 64BitSupport ChecksumOffload
        PrivateSegment LargeSend DataRateSet

2-Port 10/100/1000 Base-TX PCI-X Adapter (14108902) Specific Statistics:
--------------------------------------------------------------------
Link Status : Up
Media Speed Selected: Auto negotiation
Media Speed Running: 100 Mbps Full Duplex
PCI Mode: PCI-X (100-133)
PCI Bus Width: 64-bit
Latency Timer: 144
Cache Line Size: 128
Jumbo Frames: Disabled
TCP Segmentation Offload: Enabled
TCP Segmentation Offload Packets Transmitted: 10649
TCP Segmentation Offload Packet Errors: 0
Transmit and Receive Flow Control Status: Disabled
Transmit and Receive Flow Control Threshold (High): 45056
Transmit and Receive Flow Control Threshold (Low): 24576
Transmit and Receive Storage Allocation (TX/RX): 16/48
```

The output of this command will include very useful information, such as:

Device Type Description of the adapter type and possible media speeds.

Hardware Address MAC address of the adapter.

Elapsed Time Time elapsed since the statistics were reset.

Packets The number of packets transmitted successfully.

Transmit Errors The number of output errors. This is a counter for unsuccessful transmissions due to hardware/network errors.

Packets Dropped The number of packets accepted by the device driver to be transmitted, but were not transmitted for any reason.

S/W Transmit Queue Overflow
The total number of outgoing packets that have overflowed the software transmit queue.

No Carrier Sense The number of unsuccessful transmissions due to the no carrier sense error.

Current HW Transmit Queue Length
The number of outgoing packets that currently exist on the hardware transmit queue.

No Resource Errors The number of incoming packets dropped by the hardware due to no receive buffers on the adapter being available.

No mbuf Errors The number of times that mbufs were not available to the device driver.

Adapter Reset Count
The number of times that the adapter has been re-initialized.

Link Status The current state of the interface.

Media Speed Selected
The speed at which the adapter has been configured to connect.

Media Speed Running
The current speed at which the adapter is connected.

Jumbo Frames Specifies if jumbo frames are enabled or not.

Protocol statistics

You can use the **netstat -s** command to display statistics for all protocols or the **netstat -p** command to display statistics for a specific protocol, as shown in Example 8-6. Each piece of data displayed is relevant to the corresponding protocol.

Example 8-6 Displaying protocol statistics

```
# netstat -s
icmp:
        5286 calls to icmp_error
        0 errors not generated because old message was icmp
        Output histogram:
                echo reply: 161
                destination unreachable: 5286
        0 messages with bad code fields
        0 messages < minimum length
        0 bad checksums
        0 messages with bad length
        Input histogram:
                echo reply: 13340245
                destination unreachable: 5277
                echo: 161
```

```
        161 message responses generated
igmp:
        34 messages received
        0 messages received with too few bytes
        0 messages received with bad checksum
        0 membership queries received
        0 membership queries received with invalid field(s)
        34 membership reports received
        0 membership reports received with invalid field(s)
        34 membership reports received for groups to which we belong
        4 membership reports sent
tcp:
        83954 packets sent
                75656 data packets (27181497 bytes)
                558 data packets (945603 bytes) retransmitted
                6237 ack-only packets (4732 delayed)
                0 URG only packets
.
.
.
(lines omitted)
.
.
.
udp:
        178123 datagrams received
        0 incomplete headers
        0 bad data length fields
        0 bad checksums
        5287 dropped due to no socket
        154419 broadcast/multicast datagrams dropped due to no socket
        0 socket buffer overflows
        18417 delivered
        23689 datagrams output
ip:
        14076261 total packets received
        0 bad header checksums
        0 with size smaller than minimum
        0 with data size < data length
        0 with header length < data size
        0 with data length < header length
        0 with bad options
        0 with incorrect version number
        644016 fragments received
.
.
.
(lines ommited)
.
.
.
ipv6:
        0 total packets received
        0 with size smaller than minimum
```

```
        0 with data size < data length
.
.
.
(lines omitted)
.
.
.
icmpv6:
        0 calls to icmp6_error
        0 errors not generated because old message was icmpv6
        Output histogram:
                unreachable: 0
                packets too big: 0
                time exceeded: 0
.
.
.
(lines omitted)
.
.
.
# netstat -p ip
ip:
        14077033 total packets received
        0 bad header checksums
        0 with size smaller than minimum
        0 with data size < data length
        0 with header length < data size
        0 with data length < header length
        0 with bad options
        0 with incorrect version number
        644016 fragments received
        0 fragments dropped (dup or out of space)
        0 fragments dropped after timeout
        181998 packets reassembled ok
        13609226 packets for this host
.
.
.
(lines omitted)
.
.
.
```

Sockets

To display information about the sockets used on your system, use the **netstat -a** command, as shown in Example 8-7.

You can find useful information, such as protocol used, IP address, or socket status.

Example 8-7 Using the netstat -a command

```
Active Internet connections (including servers)
Proto Recv-Q Send-Q  Local Address          Foreign Address        (state)
tcp4       0      0  *.daytime              *.*                    LISTEN
tcp        0      0  *.ftp                  *.*                    LISTEN
tcp        0      0  *.telnet               *.*                    LISTEN
tcp4       0      0  *.smtp                 *.*                    LISTEN
tcp4       0      0  *.time                 *.*                    LISTEN
tcp4       0      0  *.sunrpc               *.*                    LISTEN
tcp4       0      0  *.smux                 *.*                    LISTEN
tcp        0      0  *.exec                 *.*                    LISTEN
tcp        0      0  *.login                *.*                    LISTEN
tcp        0      0  *.shell                *.*                    LISTEN
tcp4       0      0  *.printer              *.*                    LISTEN
tcp4       0      0  *.rmc                  *.*                    LISTEN
tcp4       0      0  *.sco_prin             *.*                    LISTEN
tcp4       0      0  *.sco_s5_p             *.*                    LISTEN
tcp4       0      0  *.filenet-             *.*                    LISTEN
tcp4       0      0  *.filenet-             *.*                    LISTEN
tcp4       0      0  *.filenet-             *.*                    LISTEN
tcp4       0      0  *.wsmserve             *.*                    LISTEN
tcp4       0      0  server2.itsc.aus.33060 kcyb72b.x11            ESTABLISHED
tcp4       0      0  server2.itsc.aus.filen server2.itsc.aus.filen ESTABLISHED
tcp4       0      0  server2.itsc.aus.filen server2.itsc.aus.filen ESTABLISHED
tcp4       0      0  server2.itsc.aus.filen server2.itsc.aus.filen ESTABLISHED
tcp4       0      0  server2.itsc.aus.filen server2.itsc.aus.filen ESTABLISHED
tcp4       0      0  server2.itsc.aus.filen server2.itsc.aus.filen ESTABLISHED
tcp4       0      0  server2.itsc.aus.filen server2.itsc.aus.filen ESTABLISHED
tcp4       0      0  localhost.smux         localhost.filenet-     ESTABLISHED
tcp4       0      0  localhost.filenet-     localhost.smux         ESTABLISHED
tcp4       0      0  server2.itsc.aus.telne tlm06.itsc.austi.carda ESTABLISHED
tcp4       0     17  server2.itsc.aus.telne tlm06.itsc.austi.mini- ESTABLISHED
tcp4       0      0  server2.itsc.aus.telne kcyb72b.discp-cl       ESTABLISHED
tcp4       0      0  server2.itsc.aus.telne kcyb72b.smart-di       ESTABLISHED
tcp4       0      0  server2.itsc.aus.telne esmsrv.itsc.aust.ms-cl ESTABLISHED
udp4       0      0  *.daytime              *.*
udp4       0      0  *.time                 *.*
udp4       0      0  *.sunrpc               *.*
udp4       0      0  *.snmp                 *.*
udp4       0      0  *.syslog               *.*
udp4       0      0  *.ntalk                *.*
```

```
udp4       0      0  *.rmc                *.*
udp4       0      0  *.xmquery            *.*
udp4       0      0  *.filenet-           *.*
udp4       0      0  *.32798              *.*
udp4       0      0  *.32801              *.*

Active UNIX domain sockets
SADR/PCB         Type   Recv-Q Send-Q      Inode            Conn
Refs           Nextref      Addr
f1000d0001d77400 stream     0     0             0 f1000d000198be80
0                0
f1000d0002021b00
f1000d00002dc000 dgram      0     0 f1000c00211673f8            0
0                0 /dev/.SRC-unix/SRCvFql7a
f1000d0000476980
.
.
.
(lines ommited)
.
.
.
```

To display detailed information about the sockets opened on your system, use the **netstat -an** command.

8.5.5 The iostat command

The **iostat** command is used to report CPU statistics, asynchronous input/output (AIO) statistics, and input/output statistics for the entire system, adapters, TTY devices, disks, and CD-ROMs.

We recommend using the **iostat** command in the following situations:

- Your system is functioning normally and you want to establish a baseline for system performance.
- Your system is experiencing performance problems.
- You make hardware or software changes to the disk subsystem.
- You change the attributes of volume groups, logical volumes, and file systems.
- You make changes to the operating system.
- You make changes to your application.

Using iostat to determine disk utilization

The **iostat** command can be used to determine if a physical disk has become a performance bottleneck. Information regarding disk activity provided by the **iostat** command includes:

% tm_act The percentage of time the physical disk was active.

KBPS The total amount of data transferred (read or write) on that disk.

tps The number of IO transfer requests per second for the physical disk.

Kb_read The total amount of data read from the disk.

Kb_wrtn The total amount of data written to the disk.

If the %tm_act field shows that the disk is very busy and the Kbps field shows a constant large amount of data being transferred (you should consider spreading the data across multiple disk drives). The command can support dynamic configuration changes. If a configuration change is detected, the **iostat** report issues a warning and refreshes the latest system configuration.

In Example 8-8, we use the **iostat -T -d 1 60** command to monitor disk activity for 60 seconds starting at 11:40:21.

Example 8-8 Using the iostat command to investigate disk bottlenecks

```
# iostat -T -d  1 60

System configuration: lcpu=4 drives=9 paths=8 vdisks=0

Disks:         % tm_act     Kbps      tps    Kb_read   Kb_wrtn  time
hdisk2          100.0     10740.0    2685.0      10740         0  11:40:21
hdisk7            0.0         0.0     0.0           0         0  11:40:21
hdisk3            0.0         0.0     0.0           0         0  11:40:21
hdisk4            0.0         0.0     0.0           0         0  11:40:21
hdisk0           97.0     10960.0    2740.0      10960         0  11:40:21
hdisk1           96.0     10716.0    2679.0      10716         0  11:40:21

Disks:         % tm_act     Kbps      tps    Kb_read   Kb_wrtn  time
hdisk5            0.0         0.0     0.0           0         0  11:40:21
hdisk6            0.0         0.0     0.0           0         0  11:40:21
cd0               0.0         0.0     0.0           0         0  11:40:21
hdisk2          100.0     10076.0    2520.0      10076         0  11:40:22
hdisk7            0.0         0.0     0.0           0         0  11:40:22
hdisk3            0.0         0.0     0.0           0         0  11:40:22
hdisk4            0.0         0.0     0.0           0         0  11:40:22
hdisk0           99.0     10352.0    2588.0      10352         0  11:40:22
hdisk1           99.0     10108.0    2527.0      10108         0  11:40:22
```

```
hdisk5          0.0       0.0       0.0          0         0  11:40:22
hdisk6          0.0       0.0       0.0          0         0  11:40:22
cd0             0.0       0.0       0.0          0         0  11:40:22
hdisk2         99.0   10928.0    2732.0      10928         0  11:40:23
hdisk7          0.0       0.0       0.0          0         0  11:40:23
hdisk3          0.0       0.0       0.0          0         0  11:40:23
hdisk4          0.0       0.0       0.0          0         0  11:40:23
hdisk0         98.0   11112.0    2778.0      11112         0  11:40:23
hdisk1         99.0   10956.0    2738.0      10956         0  11:40:23
hdisk5          0.0       0.0       0.0          0         0  11:40:23
hdisk6          0.0       0.0       0.0          0         0  11:40:23
cd0             0.0       0.0       0.0          0         0  11:40:23
hdisk2         97.0   10560.0    2640.0      10560         0  11:40:24
hdisk7          0.0       0.0       0.0          0         0  11:40:24
hdisk3          0.0       0.0       0.0          0         0  11:40:24

Disks:        % tm_act     Kbps      tps    Kb_read   Kb_wrtn  time
hdisk4          0.0       0.0       0.0          0         0  11:40:24
hdisk0        100.0   10208.0    2552.0      10208         0  11:40:24
hdisk1         98.0   10600.0    2651.0      10600         0  11:40:24
hdisk5          0.0       0.0       0.0          0         0  11:40:24
hdisk6          0.0       0.0       0.0          0         0  11:40:24
cd0             0.0       0.0       0.0          0         0  11:40:24
.
.
.
(lines omitted)
.
.
.
```

If you want to display more detailed statistics about a specific disk, you can use the **iostat -D** command. In Example 8-9, we artificially created disk activity on hdisk0 and then created ten disk performance reports every two seconds.

Example 8-9 Detailed monitoring of disk activity

```
# dd if=/dev/hdisk0 of=/dev/null &
[1]     409844
# iostat -D hdisk0 2 10

System configuration: lcpu=4 drives=7 paths=6 vdisks=0

hdisk0         xfer:  %tm_act      bps      tps      bread      bwrtn
                        72.5     15.0M   3673.5      15.0M        0.0
               read:      rps  avgserv  minserv  maxserv   timeouts      fails
                       3673.5      0.2      0.2      1.3          0          0
              write:      wps  avgserv  minserv  maxserv   timeouts      fails
                         0.0      0.0      0.0      0.0          0          0
```

```
              queue:  avgtime  mintime  maxtime  avgwqsz    avgsqsz     sqfull
                         0.0      0.0      0.0      0.0        0.2          0
------------------------------------------------------------------------------
-
hdisk0         xfer:  %tm_act      bps      tps      bread      bwrtn
                        78.5     15.1M   3680.5      15.1M        0.0
               read:      rps  avgserv  minserv  maxserv   timeouts      fails
                      3681.0      0.2      0.2      5.6          0          0
              write:      wps  avgserv  minserv  maxserv   timeouts      fails
                     9223372036854775808.0      0.0      0.0      0.0
0          0
              queue:  avgtime  mintime  maxtime  avgwqsz    avgsqsz     sqfull
                         0.0      0.0      0.0      0.0        0.2          0
------------------------------------------------------------------------------
-
hdisk0         xfer:  %tm_act      bps      tps      bread      bwrtn
                        77.0     15.1M   3688.0      15.1M        0.0
               read:      rps  avgserv  minserv  maxserv   timeouts      fails
                      3688.0      0.2      0.2      5.6          0          0
              write:      wps  avgserv  minserv  maxserv   timeouts      fails
                         0.0      0.0      0.0      0.0          0          0
              queue:  avgtime  mintime  maxtime  avgwqsz    avgsqsz     sqfull
                         0.0      0.0      0.0      0.0        0.2          0
------------------------------------------------------------------------------
-
hdisk0         xfer:  %tm_act      bps      tps      bread      bwrtn
                        73.0     14.9M   3646.0      14.9M        0.0
               read:      rps  avgserv  minserv  maxserv   timeouts      fails
                      3646.0      0.2      0.2     11.3          0          0
              write:      wps  avgserv  minserv  maxserv   timeouts      fails
                         0.0      0.0      0.0      0.0          0          0
              queue:  avgtime  mintime  maxtime  avgwqsz    avgsqsz     sqfull
                         0.0      0.0      0.0      0.0        0.2          0
------------------------------------------------------------------------------
.
.
.
(lines omitted)
.
.
.
```

If you are using multi-path input-output (MPIO) enabled devices, you can use the **iostat -m** command to display statistics corresponding to each path.

Using iostat to determine CPU utilization

The **iostat** command can be used to monitor and provide statistics about CPU activity that can be useful for identifying potential CPU problems. Information regarding CPU activity provided by **iostat** includes:

tin The total number of characters read by the system for all ttys.

tout The total number of characters written by the system to all ttys.

%user The percentage of CPU utilization while executing in user mode.

%sys The percentage of CPU utilization while executing in kernel mode.

%idle The percentage of time that the CPU or CPUs were idle and the system did not have an outstanding disk I/O request.

%iowait The percentage of time that the CPU or CPUs were idle during which the system had an outstanding disk I/O request.

%physc The percentage of physical processors consumed, displayed only if the partition is running with shared processor.

%entc The percentage of entitled capacity consumed, displayed only if the partition is running with shared processor.

In Example 8-10, we use the **iostat -T -t 1 60** command to monitor CPU activity for 60 seconds starting at 11:40:21.

Example 8-10 Using iostat to investigate CPU activity

```
# iostat -T -t 1 60

System configuration: lcpu=4

tty:      tin         tout    avg-cpu: % user % sys % idle % iowait  time
          2.0        936.0             17.1  32.8    6.9     43.2  11:40:21
          5.0        1532.0             16.6  31.3    6.8     45.3  11:40:22
          0.0        746.0             17.3  32.6    7.9     42.2  11:40:23
          0.0        745.0             16.8  31.9    7.2     44.1  11:40:24
.
.
.
(lines omitted)
.
```

```
.
.
```

Using iostat to determine AIO utilization

You can use the **iostat -A** command (Example 8-11) to obtain asynchronous I/O (AIO) statistics, including:

avgc Average global non-fast path AIO request count per second for the specified interval.

avfc Average global AIO fast path request count per second for the specified interval

maxg Maximum global non-fast path AIO request count since the last time this value was fetched

maxf Maximum fast path request count since the last time this value was fetched

maxr Maximum AIO requests allowed. This is the AIO device maxreqs attribute.

Example 8-11 Using the iostat -A command to monitor AIO activity

```
# dd if=/dev/hdisk0 of=/dev/null &
[1]     311456
# dd if=/dev/hdisk1 of=/dev/null &
[2]     274580
# dd if=/dev/hdisk4 of=/dev/null &
[3]     315546
# dd if=/dev/hdisk5 of=/dev/null &
[4]     323742
# iostat -A 1 10

System configuration: lcpu=4 drives=9 paths=8 vdisks=0

aio: avgc avfc maxg maif maxr avg-cpu: % user % sys % idle % iowait
        0    0    0    0 4096           23.9  42.9   10.8     22.5

Disks:        % tm_act     Kbps      tps    Kb_read   Kb_wrtn
hdisk5          63.2     11128.3     2781.1      11796         0
hdisk2           0.0         0.0     0.0           0         0
hdisk4          58.5     11064.2     2765.1      11728         0
hdisk3           0.0         0.0     0.0           0         0
hdisk7           0.0         0.0     0.0           0         0
hdisk1          61.3     10660.4     2666.0      11300         0
hdisk0          51.9     10920.8     2731.1      11576         0
hdisk6           0.0         0.0     0.0           0         0
cd0              0.0         0.0     0.0           0         0
```

```
aio: avgc avfc maxg maif maxr avg-cpu: % user % sys % idle % iowait
        0    0    0    0 4096            23.8  42.7   11.5     22.0

Disks:        % tm_act     Kbps      tps    Kb_read   Kb_wrtn
hdisk5          59.0     11636.0     2910.0      11636         0
hdisk2           0.0       0.0       0.0          0         0
hdisk4          68.0     11588.0     2897.0      11588         0
hdisk3           0.0       0.0       0.0          0         0
hdisk7           0.0       0.0       0.0          0         0
hdisk1          54.0     11448.0     2862.0      11448         0
hdisk0          67.0     11564.0     2891.0      11564         0
hdisk6           0.0       0.0       0.0          0         0
cd0              0.0       0.0       0.0          0         0

aio: avgc avfc maxg maif maxr avg-cpu: % user % sys % idle % iowait
        0    0    0    0 4096            23.6  42.8   11.1     22.5

Disks:        % tm_act     Kbps      tps    Kb_read   Kb_wrtn
hdisk5          71.0     11724.0     2930.0      11724         0
hdisk2           0.0       0.0       0.0          0         0
hdisk4          68.0     11616.0     2904.0      11616         0
hdisk3           0.0       0.0       0.0          0         0
hdisk7           0.0       0.0       0.0          0         0
hdisk1          68.0     11212.0     2803.0      11212         0
hdisk0          66.0     11352.0     2838.0      11352         0
hdisk6           0.0       0.0       0.0          0         0
cd0              0.0       0.0       0.0          0         0
.
.
.
(lines omitted)
.
.
.
```

If you want to display a list of all the mounted file systems and the associated queue numbers along with their request counts, use the **iostat -AQ** command, as shown in Example 8-12 on page 344.

Example 8-12 Using the iostat -AQ command

```
# iostat -AQ

System configuration: lcpu=4

aio: avgc avfc maxg maif maxr avg-cpu: % user % sys % idle % iowait
        0    0    0    0 4096            16.3  29.5   38.6     15.6

Queue#          Count           Filesystems
129             0               /
130             0               /usr
132             0               /var
133             0               /tmp
136             0               /home
137             0               /proc
138             0               /opt
```

Using iostat to determine adapter utilization

The **iostat** command can be used to determine adapter activity. Information regarding adapter activity provided by the **iostat** command includes:

KBPS The total amount of data transferred (read or write) to the adapter.

tps The number of IO transfer requests per second for the adapter.

Kb_read The total amount of data read from the adapter.

Kb_wrtn The total amount of data written to the adapter.

For virtual adapters, this command will also display:

KBPS The total amount of data transferred (read or write) to the adapter.

tps The number of IO transfer requests per second for the adapter.

bkread The total number of blocks per second sent from the hosting server to the adapter.

bkwrtn The total number of blocks per second sent from the adapter to the hosting server.

partition-id The partition ID of the hosting server, which serves the request sent by the adapter.

In Example 8-13, we artificially created disk activity on hdisk0, hdisk1, hdisk4, and hdisk5, and then created ten adapter performance reports every second.

Example 8-13 Using the iostat command to investigate adapter performance

```
# dd if=/dev/hdisk0 of=/dev/null &
[2]     380936
# dd if=/dev/hdisk1 of=/dev/null &
[3]     344184
# dd if=/dev/hdisk4 of=/dev/null &
[4]     405516
# dd if=/dev/hdisk5 of=/dev/null &
[5]     430240
# iostat -a 1 10|more

System configuration: lcpu=4 drives=7 paths=6 vdisks=0

tty:      tin         tout    avg-cpu: % user % sys % idle % iowait
          3.0         73.0               22.7  39.9    9.7     27.7

Adapter:                   Kbps      tps    Kb_read   Kb_wrtn
sisioa1                  22224.0     5556.0      22224         0

Disks:              % tm_act     Kbps      tps    Kb_read   Kb_wrtn
hdisk5                58.0     11096.0     2774.0      11096         0
hdisk4                62.0     11128.0     2782.0      11128         0

Adapter:                   Kbps      tps    Kb_read   Kb_wrtn
sisioa0                  22256.0     5564.0      22256         0

Disks:              % tm_act     Kbps      tps    Kb_read   Kb_wrtn
hdisk0                94.0     11344.0     2837.0      11344         0
hdisk1                62.0     10912.0     2727.0      10912         0
hdisk2                 0.0       0.0       0.0          0        0
hdisk3                 0.0       0.0       0.0          0        0

Adapter:                   Kbps      tps    Kb_read   Kb_wrtn
ide0                       0.0       0.0          0         0

Disks:              % tm_act     Kbps      tps    Kb_read   Kb_wrtn
cd0                    0.0       0.0       0.0          0        0

tty:      tin         tout    avg-cpu: % user % sys % idle % iowait
          0.0       1306.0               22.6  39.9    8.9     28.5
```

If you want to display more details about adapter activity, you can use the **iostat -a -D** command, as shown in Example 8-14 on page 346.

Example 8-14   Detailed monitoring of adapter activity

```
# iostat -D -a|more

System configuration: lcpu=4 drives=7 paths=6 vdisks=0

Adapter:
sisioa1              xfer:      bps      tps    bread      bwrtn
                              61.6K     15.0    61.6K       0.0

Disks:
hdisk5               xfer:  %tm_act      bps      tps      bread      bwrtn
                               0.2    30.5K     7.5      30.5K       0.0
                     read:      rps  avgserv  minserv  maxserv   timeouts      fails
                               7.5      0.2      0.2     13.3          0          0
                    write:      wps  avgserv  minserv  maxserv   timeouts      fails
                               0.0      0.0      0.0      0.0          0          0
                    queue:  avgtime  mintime  maxtime  avgwqsz    avgsqsz     sqfull
                               0.0      0.0      0.8      0.0        0.0          0
hdisk4               xfer:  %tm_act      bps      tps      bread      bwrtn
                               0.2    31.0K     7.6      31.0K       0.0
                     read:      rps  avgserv  minserv  maxserv   timeouts      fails
                               7.6      0.2      0.2     11.4          0          0
                    write:      wps  avgserv  minserv  maxserv   timeouts      fails
                               0.0      0.0      0.0      0.0          0          0
                    queue:  avgtime  mintime  maxtime  avgwqsz    avgsqsz     sqfull
                               0.0      0.0      1.2      0.0        0.0          0

Adapter:
sisioa0              xfer:      bps      tps    bread      bwrtn
                             220.4K     53.7   220.0K      409.1

Disks:
hdisk0               xfer:  %tm_act      bps      tps      bread      bwrtn
                               1.1   189.1K    46.2     189.0K     154.5
                     read:      rps  avgserv  minserv  maxserv   timeouts      fails
                              46.1      0.3      0.2     90.0          0          0
                    write:      wps  avgserv  minserv  maxserv   timeouts      fails
                               0.0      6.9      0.8     25.4          0          0
                    queue:  avgtime  mintime  maxtime  avgwqsz    avgsqsz     sqfull
                               0.0      0.0    232.8      0.0        0.0        679
hdisk1               xfer:  %tm_act      bps      tps      bread      bwrtn
                               0.2    31.3K     7.6      31.0K     254.6
                     read:      rps  avgserv  minserv  maxserv   timeouts      fails
                               7.6      0.2      0.2     17.9          0          0
                    write:      wps  avgserv  minserv  maxserv   timeouts      fails
                               0.0      7.2      1.1     29.8          0          0
                    queue:  avgtime  mintime  maxtime  avgwqsz    avgsqsz     sqfull
                               0.0      0.0     86.4      0.0        0.0        486
hdisk2               xfer:  %tm_act      bps      tps      bread      bwrtn
                               0.0      0.0      0.0       0.0       0.0
                     read:      rps  avgserv  minserv  maxserv   timeouts      fails
                               0.0      0.0      0.0      0.0          0          0
                    write:      wps  avgserv  minserv  maxserv   timeouts      fails
```

```
                    0.0      0.0      0.0      0.0          0          0
          queue:  avgtime  mintime  maxtime  avgwqsz    avgsqsz     sqfull
                    0.0      0.0      0.0      0.0        0.0          0
hdisk3     xfer:  %tm_act      bps      tps      bread      bwrtn
                    0.0      0.0      0.0        0.0        0.0
            read:      rps  avgserv  minserv  maxserv   timeouts      fails
                    0.0      0.0      0.0      0.0          0          0
          write:       wps  avgserv  minserv  maxserv   timeouts      fails
                    0.0      0.0      0.0      0.0          0          0
```

8.5.6 The proctools commands

The /proc file system provides a mechanism to control processes. It also gives access to information about the current state of processes and threads, but in binary form. The name of each entry in the /proc file system is a decimal number corresponding to the process ID. These entries are subdirectories and the owner of each is determined by the user ID of the process. Access to the process state is provided by additional files contained within each subdirectory.

The proctools commands (/proc commands) provide ASCII reports based on some of the available information. Most of the commands take a list of process IDs or /proc/ProcessID strings as input. The shell expansion /proc/* can therefore be used to specify all processes in the system.

Each of the proctools commands gathers information from /proc for the specified processes and displays it to the user. The information gathered by the commands from /proc is a snapshot of the current state of processes, and therefore can vary at any instant except for stopped processes. The proctools commands are:

procfiles Reports information about all file descriptors opened by processes.

proctree Prints the process tree containing the specified process IDs or users.

procsig Lists the signal actions defined by processes.

procstack Prints the hexadecimal addresses and symbolic names for all the threads in the process.

procrun Starts a process that has stopped on the PR_REQUESTED event.

procmap Prints the address space map of processes.

procflags Prints the /proc tracing flags, the pending and held signals, and other /proc status information for each thread in the specified processes.

proccred	Prints the credentials (effective, real, saved user IDs, and group IDs) of processes.
procldd	Lists the objects loaded by processes, including shared objects explicitly attached using dlopen().
procwait	Waits for all of the specified processes to terminate.
procwdx	Prints the current working directory of processes.
procstop	Stops processes on the PR_REQUESTED event.

8.5.7 The procmon tool

You can use the **procmon** tool on systems running AIX 5L Version 5.3 or later. The **procmon** tool allows you to view and manage the processes running on a system. The **procmon** tool has a graphical interface and displays a table of process metrics that you can sort on the different fields that are provided. The default number of processes listed in the table is 20, but you can change the value in the Table Properties panel from the main menu. Only the top processes based on the sorting metric are displayed and the default sorting key is CPU consumption.

The default value of the refresh rate for the table of process metrics is five seconds, but you can change the refresh rate by either using the Table Properties panel in the main menu or by clicking the **Refresh** button.

By default, the **procmon** tool displays the following:

- How long a process has been running
- How much CPU resources the processes are using
- Whether processes are being penalized by the system
- How much memory the processes are using
- How much I/O a process is performing
- The priority and nice values of a process
- Who has created a particular process

You can choose other metrics to display, and filter the processes that are displayed and perform performance commands on processes.

> **Note:** The **procmon** tool is a Performance Workbench plug-in, so you can only launch the procmon tool from within the Performance Workbench framework. You must install the bos.perf.gtools fileset by either using the **smitty** tool or the **installp** command. You can then access the Performance Workbench from the /opt/perfwb directory or run the **procmon** script from the /opt/perfwb/procmon directory.

The global statistics area of the procmon tool

The global statistics area displays the amount of CPU and memory that is being used by the system. You can refresh the statistics data by either clicking the **Refresh** button in the menu bar or by activating the automatic refresh option through the menu bar. Figure 8-4 shows a sample window of this area. To save the statistics information, you can export the table to any of the following file formats:

- XML
- HTML
- CSV

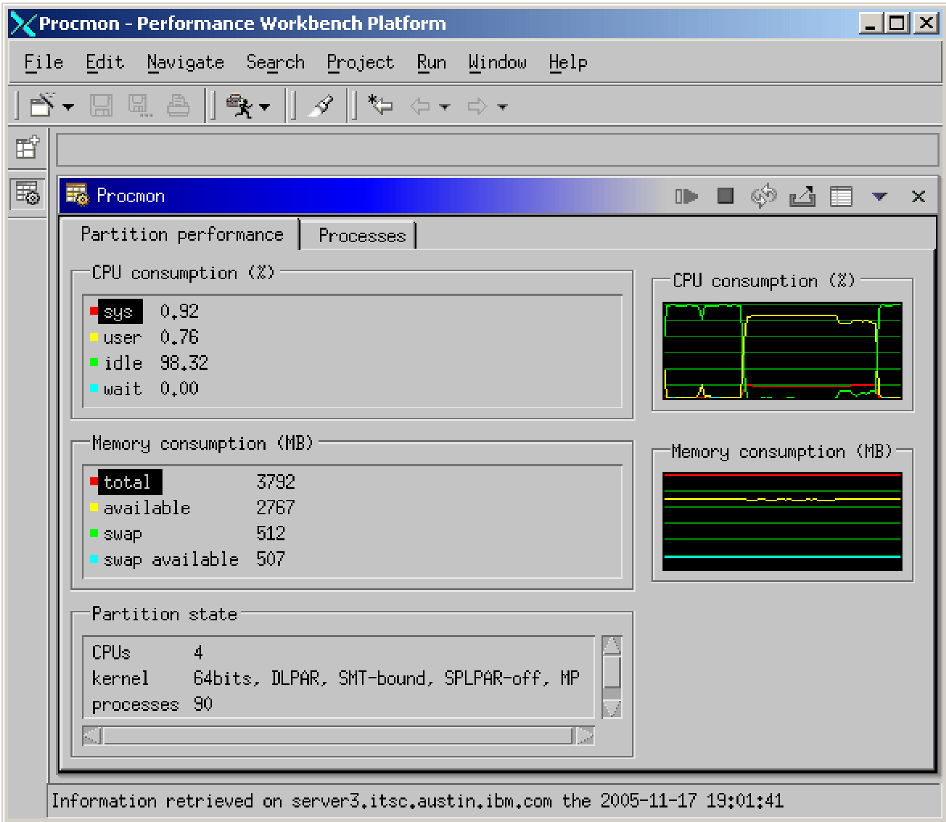


Figure 8-4 The global statistics area of the procmon tool

The process table of the procmon tool

The process table is the main component of the **procmon** tool. The process table displays the various processes that are running on the system, ordered and filtered according to the user configuration. The default value of the number of processes listed in the process table is 20, but you can change this value from the Table Properties panel from the main menu.

The yellow arrow key in the column header indicates the sort key for the process table. The arrow points either up or down, depending on whether the sort order is ascending or descending, respectively. You can change the sort key by clicking any of the column headers. Figure 8-5 shows a sample window of the process table.

You can customize the process table, modify the information about the various processes, and run commands on the displayed processes.

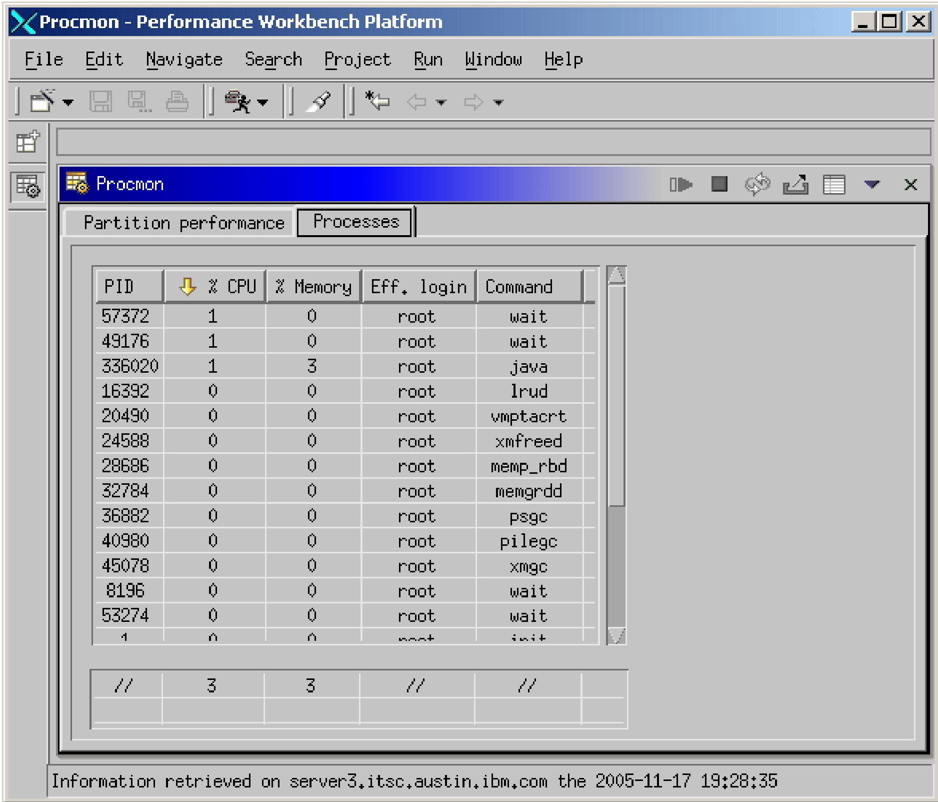


Figure 8-5 The processes table of the procmon tool

By default, the **procmon** tool displays the metrics listed in Table 8-11.

Table 8-11 Default metrics listed on the process table of the procmon tool

Metric	Description
PID	Process identifier
CPUPER	Percentage of CPU used per process since the last refresh
PRM	Percent real memory usage
ELOGIN	Effective login of the process user
COMMAND	Short name of the process launched

You can choose to display other metrics. The full list of metrics are in Table 8-12.

Table 8-12 List of all metrics available for display on the process table

Metric	Description
PPID	Parent process identifier
NICE	Nice value for the process
DRSS	Data resident set size
TRSS	Text resident set size
STARTTIME	Time when command started
PRI	Priority of the process
EUID	Effective user identifier
RUID	Real user identifier
EGID	Effective group identifier
RGID	Real group identifier
THCOUNT	Number of threads used
CLASSID	Identifier of the class that pertains to the WLM process
CLASSNAME	Name of the class that pertains to the WLM process
TOTDISKIO	Disk I/O for that process
NVCSW	N voluntary context switches

Metric	Description
NIVCSW	N involuntary context switches
MINFLT	Minor page faults
MAJFLT	Major page faults
INBLK	Input blocks
OUBLK	Output blocks
MSGSEND	Messages sent
MSGRECV	Messages received
EGROUP	Effective group name
RGROUP	Real group name

You can use either the table properties or preference to display the metrics in which you are interested. If you choose to change the table properties, the new configuration values are set for the current session only. If you change the preferences, the new configuration values are set for the next session of the **procmon** tool.

There are two types of values listed in the process table:

- Real values
- Delta values

Real values are retrieved from the kernel and displayed in the process table. An example of a real value is the PID, PPID, or TTY.

Delta values are values that are computed from the last-stored measurements. An example of a delta value is the CPU percent for each process, which is computed using the values measured between refreshes.

Below the process table, there is another table that displays the sum of the values for each column of the process table. For example, this table might provide a good idea of the percentage of total CPU used by the top 20 CPU-consuming processes.

You can refresh the data by either clicking the **Refresh** button in the menu bar or by activating the automatic refresh option through the menu bar.

Performing commands on processes

You can run the following commands on the processes you select in the process table:

- **svmon**
- **renice**
- **kill**
- The following proctools commands:
  - **procfiles**
  - **proctree**
  - **procsig**
  - **procstack**
  - **procrun**
  - **procmap**
  - **procflags**
  - **proccred**
  - **procldd**

To run any of the above commands on one or more processes, select the processes in the process table and right click your mouse, and select Detailed information or Modify and then select the command you want to run. A new window opens, which displays the command output while the command is running. You can interrupt the command by clicking the **STOP** button.

You can also invoke this commands from the command line, for example:

- To show the process tree for the current shell use the **proctree** command:

```
# ps
    PID    TTY  TIME CMD
 409744  pts/5  0:00 -ksh
 479424  pts/5  0:00 ps
# proctree 409744
155780    /usr/sbin/srcmstr
   204930    /usr/sbin/inetd
      471292    telnetd -a
         409744    -ksh
            426154    proctree 409744
#
```

- To show the file descriptors used by a running **find** process:

```
# tty
/dev/pts/5
# find / -type f > /tmp/files 2>/tmp/errors &
[1]     405622
# procfiles -n 405622
405622 : find / -type f
  Current rlimit: 2000 file descriptors
   0: S_IFCHR mode:00 dev:10,4 ino:1182 uid:0 gid:0 rdev:3
      O_RDWR  name:/dev/pts/5
   1: S_IFREG mode:0200 dev:10,7 ino:21 uid:0 gid:0 rdev:0
      O_WRONLY size:648723  name:/tmp/files
   2: S_IFREG mode:0200 dev:10,7 ino:25 uid:0 gid:0 rdev:0
      O_WRONLY size:0  name:/tmp/errors
   3: S_IFREG mode:0444 dev:10,5 ino:11305 uid:0 gid:0 rde
      O_RDONLY size:2208  name:/usr/lib/nls/msg/en_US/find
.
.
.
(lines ommited)
.
.
#
```

  Standard input (stdin, file descriptor 0) is assigned to the current terminal (/dev/pts/5), standard output (stdout, file descriptor 1) is assigned to file /tmp/files, and standard error (stderr, file descriptor 2) is assigned to file /tmp/errors. All the file descriptors used are displayed and the command prompt is returned.

- To wait for a process to finish and display the status, use the **procwait** command:

```
# find / -type f > /dev/null 2>&1 &
[1]     102614
# procwait -v 102614
102614 : terminated, exit status 0
[1] +  Done                    find / -type f > /dev/null 2>&1
&
#
```

8.6 Tuning using the /etc/tunables files

The /etc/tunables directory centralizes the tunable files. The tunable files contain the tunable parameters, bundled in one or more sections named *stanzas*.

A stanza is started by a line containing the stanza name followed by a colon (:). There is no marking for the end of a stanza. It simply continues until another stanza starts. Each stanza contains a set of parameter/value pairs, one per line. The values are surrounded by double quotes (") and an equal sign (=) separates the parameter name format value. The parameter/value pair must belong to a stanza; it has no meaning outside of a stanza. Everything with a number sign (#) is considered a comment and ignored.

A tunable file uses the following syntax:

```
schedo:
        %usDelta = "100"                # DEFAULT VALUE
        affinity_lim = "7"              # DEFAULT VALUE
        allowMCMmigrate = "0"           # DEFAULT VALUE
        big_tick_size = "1"             # DEFAULT VALUE
no:
        arpqsize = "12"                 # DEFAULT VALUE
        arpt_killc = "20"               # DEFAULT VALUE
        arptab_bsiz = "7"               # DEFAULT VALUE
```

The tunables files currently support seven different stanzas: one for each of the tunable commands (**schedo**, **vmo**, **ioo**, **raso**, **no**, and **nfso**), plus a special info stanza.

Three files under /etc/tunables have special names and meaning:

nextboot This file is automatically applied at boot time. The **bosboot** command also gets the value of **bosboot** type tunables from this file. It contains all the tunable command parameter settings that have been made permanent.

lastboot This file is automatically generated at boot time. It contains the full set of tunable parameters, with their values after the last boot. The default values are marked with # DEFAULT VALUE.

lastboot.log This should be the only file in /etc/tunables that is not in the stanza format described here. It is automatically generated at boot time, and contains the logging of the creation of the lastboot file. All parameter changes are logged in this file, and also the failed changes.

The six command stanzas contain tunable parameters managed by the corresponding command (see the **man** pages for the complete parameter lists):

schedo Manages CPU scheduler tunable parameters.

vmo Manages Virtual Memory Manager tunable parameters.

ioo Manages Input/Output tunable parameters.

raso Manages Reliability, Availability, and Serviceability parameters.

no Manages network tuning parameters.

nfso Manages Network File System (NFS) tuning parameters.

The **vmo** command is the replacement for the **vmtune** command used in previous versions of AIX.

The value can either be a numerical value or the literal words DEFAULT, which is interpreted as this tunable's default value, and STATIC, which indicates a Static variable that is never restored. It is possible that some stanza contains values for non-existent parameters (in the case where a tunable file was copied from a system running an older version of AIX 5L and one or more tunables do not exist anymore). When you use the **tunrestore** or the **tuncheck** commands, warnings about such parameters are shown.

The info stanzas used to store information about the purpose of the tunable file and the level of AIX 5L on which it was validated. Any parameter is acceptable in this stanza; however, some fields have a special meaning.

The following example shows a sample info stanza:

```
info:
        Logfile_checksum = "3444306214"
        Description = "Full set of tunable parameters after last boot"
        AIX_level = "5.3.0.30"
        Kernel_type = "MP"
        Last_validation = "2005-11-11 11:57:53 CST (current, reboot)"
```

Table 8-13 explains the different fields from the stanza.

Table 8-13 Description of the info stanza

Parameter	Value
Description	A character string describing the tunable file. SMIT displays this field in the file selection box.
AIX_level	AIX 5L version. This field is automatically updated by **tunsave** and **tuncheck** (on success only).
Kernel_type	“MP”: A multiprocessor kernel. “MP64”: A 64-bit multiprocessor kernel.
Last_validation	The date this file was validated for the last time, and the type of validation: “**current**” The file has been validated against the current context. “**reboot**” The file has been validated against the nextboot context. This field is automatically updated by **tunsave** and **tuncheck** (on success only).
Logfile_checksum	The checksum of the lastboot.log file matching this tunables file. This field is present only in the lastboot file.

The commands provided by AIX 5L to work with the tunables files are:

tunsave Saves current tunable parameter values to a file (backup).

tunrestore Restores tunable parameter values from a file.

tuncheck Validates a tunable file.

tunchange Updates one or more tunable stanzas in a file.

tundefault Reset all tunable parameters to their default value.

For more information about the tunables commands, consult the product documentation pages.

8.7 Documenting a system configuration

As a system administrator, you are expect to document and be familiar with the configuration of your system, including hardware configuration. It is good practice to obtain and document various settings for all devices contained in your system.

In Example 8-15 on page 358, we use the **lsattr** command to display for device `ent0` useful attributes such as media speed, transmit and receive queue size.

Example 8-15 Listing device attributes

```
# lsattr -El ent0
alt_addr       0x000000000000    Alternate ethernet address                     True
busintr        149               Bus interrupt level                            False
busmem         0xf8120000        Bus memory address                             False
chksum_offload yes               Enable hardware transmit and receive checksum  True
compat_mode    no                Gigabit Backward compatability                 True
copy_bytes     2048              Copy packet if this many or less bytes         True
delay_open     no                Enable delay of open until link state is known True
failback       yes               Enable auto failback to primary                True
failback_delay 15                Failback to primary delay timer                True
failover       disable           Enable failover mode                           True
flow_ctrl      yes               Enable Transmit and Receive Flow Control       True
intr_priority  3                 Interrupt priority                             False
intr_rate      10000             Max rate of interrupts generated by adapter    True
jumbo_frames   no                Transmit jumbo frames                          True
large_send     yes               Enable hardware TX TCP resegmentation          True
media_speed    Auto_Negotiation Media speed                                     True
rom_mem        0xf80c0000        ROM memory address                             False
rx_hog         1000              Max rcv buffers processed per rcv interrupt    True
rxbuf_pool_sz  2048              Rcv buffer pool, make 2X rxdesc_que_sz         True
rxdesc_que_sz  1024              Rcv descriptor queue size                      True
slih_hog       10                Max Interrupt events processed per interrupt   True
tx_que_sz      8192              Software transmit queue size                   True
txdesc_que_sz  512               TX descriptor queue size                       True
use_alt_addr   no                Enable alternate ethernet address              True
```

You can obtain information about the status of your devices and details and about the way they are connected to the system using the **lsdev** command.

In Example 8-16, we use the **lsdev** command to display the status and location codes for all disk drive devices.

Example 8-16 Using the lsdev command

```
# lsdev -Cc disk
hdisk0 Available 05-08-00-3,0 16 Bit LVD SCSI Disk Drive
hdisk1 Available 05-08-00-4,0 16 Bit LVD SCSI Disk Drive
hdisk2 Available 05-08-00-5,0 16 Bit LVD SCSI Disk Drive
hdisk3 Available 05-08-00-8,0 16 Bit LVD SCSI Disk Drive
hdisk4 Available 09-08-00-3,0 16 Bit LVD SCSI Disk Drive
hdisk5 Available 09-08-00-4,0 16 Bit LVD SCSI Disk Drive
hdisk6 Available 09-08-00-5,0 16 Bit LVD SCSI Disk Drive
hdisk7 Available 09-08-00-8,0 16 Bit LVD SCSI Disk Drive
```

When you are running on systems that support dynamic reconfiguration, it is especially useful to have detailed information about the dynamically reconfigurable slots, such as location, capabilities, frequency, or voltage. This information is valuable for both planning and troubleshooting.

In Example 8-17, we use the **lsslot** command to display characteristics and capabilities of hot plug PCI slots.

Example 8-17 Using the lsslot command

```
# lsslot -c pci
# Slot                      Description                          Device(s)
U787B.001.DNW108F-P1-C1  PCI-X capable, 64 bit, 133MHz slot  ent2
U787B.001.DNW108F-P1-C2  PCI-X capable, 64 bit, 133MHz slot  Empty
U787B.001.DNW108F-P1-C3  PCI-X capable, 64 bit, 133MHz slot  Empty
U787B.001.DNW108F-P1-C4  PCI-X capable, 64 bit, 133MHz slot  sisioa0
U787B.001.DNW108F-P1-C5  PCI-X capable, 64 bit, 133MHz slot  pci7 lai0
```

As a system administrator, you will be required to upgrade the microcode level of your system.

In Example 8-18, we use the **lsmcode** command to display microcode levels from permanent and temporary sides.

Example 8-18 Using the lsmcode command

```
# lsmcode -c
The current permanent system firmware image is SF235_160
The current temporary system firmware image is SF235_160
The system is currently booted from the temporary firmware image.
```

The system maintains an internal vital product data (VPD) database providing, for all hardware components of your system, very useful details, such as field replaceable unit (FRU) number, serial number, engineering change (EC) level, or other device-specific information. You can display this database using the **lscfg -vp** command.

In Example 8-19, we use the **lscfg** command to display stanzas that contain the system machine type and serial number for our system.

Example 8-19 Using the lscfg command

```
# lscfg -vp|grep -ip cabinet
      System VPD:
        Product Specific.(RT).......VSYS
        Flag Field..................XXSV
        Brand.......................P0
        System Info Specific.(YL)...U9113.550.105E9DE
        Machine/Cabinet Serial No...105E9DE
        Machine Type and Model......9113-550
        Product Specific.(SU).......0004AC08AFC3
        Version.....................ipzSeries
      Physical Location: U9113.550.105E9DE
```

```
  CEC:
    Product Specific.(RT).......VCEN
    Flag Field..................XXEV
    Brand.......................PO
    System Info Specific.(YL)...U787B.001.DNW108F
    Machine/Cabinet Serial No...DNW108F
    Machine Type and Model......787B-001
    Controlling CEC ID..........9113-550 105E9DE
    Rack Serial Number..........0000000000000000
    Feature Code/Marketing ID...787B-001
    Version.....................ipzSeries
  Physical Location: U787B.001.DNW108F
```

You can also use the **prtconf** command to display relevant information about the hardware and software of your system, as shown in Example 8-20.

Example 8-20 Using the prtconf command

```
System Model: IBM,9113-550
Machine Serial Number: 105E9DE
Processor Type: PowerPC_POWER5
Number Of Processors: 2
Processor Clock Speed: 1654 MHz
CPU Type: 64-bit
Kernel Type: 64-bit
LPAR Info: 1 10-5E9DE
Memory Size: 3792 MB
Good Memory Size: 3792 MB
Platform Firmware level: Not Available
Firmware Version: IBM,SF235_160
Console Login: enable
Auto Restart: false
Full Core: false

Network Information
        Host Name: server3
        IP Address: 9.3.5.196
        Sub Netmask: 255.255.255.0
        Gateway: 9.3.5.41
        Name Server: 9.3.4.2
        Domain Name: itsc.austin.ibm.com

Paging Space Information
        Total Paging Space: 512MB
        Percent Used: 1%

Volume Groups Information
```

```
===============================================================================
rootvg:
PV_NAME           PV STATE          TOTAL PPs   FREE PPs    FREE DISTRIBUTION
hdisk0            active            546         524
109..101..96..109..109
hdisk1            active            546         523
110..89..106..109..109
===============================================================================

dumpvg:
PV_NAME           PV STATE          TOTAL PPs   FREE PPs    FREE DISTRIBUTION
hdisk2            active            4375        4076
875..864..875..875..587
===============================================================================

test2vg:
PV_NAME           PV STATE          TOTAL PPs   FREE PPs    FREE DISTRIBUTION
hdisk3            active            136         133         28..24..27..27..27
===============================================================================

test1vg:
PV_NAME           PV STATE          TOTAL PPs   FREE PPs    FREE DISTRIBUTION
hdisk6            snapshotpv        136         118         28..17..25..27..21
hdisk7            active            136         118         28..20..25..27..18
===============================================================================

newvg:
PV_NAME           PV STATE          TOTAL PPs   FREE PPs    FREE DISTRIBUTION
hdisk6            active            136         118         28..17..25..27..21
===============================================================================

INSTALLED RESOURCE LIST

The following resources are installed on the machine.
+/- = Added or deleted from Resource List.
*   = Diagnostic support not available.

  Model Architecture: chrp
  Model Implementation: Multiple Processor, PCI bus

+ sys0                                                System Object
+ sysplanar0                                          System Planar
* vio0                                                Virtual I/O Bus
* ent4            U9113.550.10478DE-V1-C3-T1          Virtual I/O Ethernet
Adapter (l-lan)
* ent3            U9113.550.10478DE-V1-C2-T1          Virtual I/O Ethernet
Adapter (l-lan)
* vsa0            U9113.550.10478DE-V1-C0             LPAR Virtual Serial
Adapter
```

```
* vty0            U9113.550.10478DE-V1-C0-L0        Asynchronous Terminal
* pci1            U787B.001.DNW0974-P1              PCI Bus
* pci10           U787B.001.DNW0974-P1              PCI Bus
+ ent2           U787B.001.DNW0974-P1-C1-T1        10/100/1000 Base-TX PCI-X
Adapter (14106902)
* pci8            U787B.001.DNW0974-P1              PCI Bus
+ ent0            U787B.001.DNW0974-P1-T9           2-Port 10/100/1000
Base-TX PCI-X Adapter (14108902)
+ ent1            U787B.001.DNW0974-P1-T10          2-Port 10/100/1000
Base-TX PCI-X Adapter (14108902)
* pci9            U787B.001.DNW0974-P1              PCI Bus
+ sisioa1         U787B.001.DNW0974-P1              PCI-X Dual Channel U320
SCSI RAID Adapter
+ scsi2           U787B.001.DNW0974-P1-T14          Ultra320 SCSI RAID
Adapter Physical bus
+ rmt0            U787B.001.DNW0974-P1-T14-L0-L0    LVD SCSI Tape Drive
(80000 MB)
+ hdisk4          U787B.001.DNW0974-P1-T14-L3-L0    16 Bit LVD SCSI Disk
Drive (73400 MB)
+ hdisk5          U787B.001.DNW0974-P1-T14-L4-L0    16 Bit LVD SCSI Disk
Drive (73400 MB)
+ hdisk6          U787B.001.DNW0974-P1-T14-L5-L0    16 Bit LVD SCSI Disk
Drive (73400 MB)
+ hdisk7          U787B.001.DNW0974-P1-T14-L8-L0    16 Bit LVD SCSI Disk
Drive (73400 MB)
+ ses1            U787B.001.DNW0974-P1-T14-L15-L0   SCSI Enclosure Services
Device
+ scsi3           U787B.001.DNW0974-P1-T15          Ultra320 SCSI RAID
Adapter Physical bus
+ sisraid1        U787B.001.DNW0974-P1-C7-T1        Ultra320 SCSI RAID
Adapter Logical bus
* pci11           U787B.001.DNW0974-P1              PCI Bus
* pci0            U787B.001.DNW0974-P1              PCI Bus
* pci4            U787B.001.DNW0974-P1              PCI Bus
* pci2            U787B.001.DNW0974-P1              PCI Bus
+ usbhc0          U787B.001.DNW0974-P1              USB Host Controller
(33103500)
+ usbhc1          U787B.001.DNW0974-P1              USB Host Controller
(33103500)
* pci3            U787B.001.DNW0974-P1              PCI Bus
* ide0            U787B.001.DNW0974-P1-T16          ATA/IDE Controller Device
+ cd0             U787B.001.DNW0974-P4-D2           IDE DVD-ROM Drive
* pci5            U787B.001.DNW0974-P1              PCI Bus
+ sisioa0         U787B.001.DNW0974-P1-C4           PCI-X Dual Channel U320
SCSI RAID Adapter
+ scsi0           U787B.001.DNW0974-P1-C4-T1        Ultra320 SCSI RAID
Adapter Physical bus
+ hdisk0          U787B.001.DNW0974-P1-C4-T1-L3-L0  16 Bit LVD SCSI Disk
Drive (73400 MB)
```

```
+ hdisk1          U787B.001.DNW0974-P1-C4-T1-L4-L0   16 Bit LVD SCSI Disk
Drive (73400 MB)
+ hdisk2          U787B.001.DNW0974-P1-C4-T1-L5-L0   16 Bit LVD SCSI Disk
Drive (73400 MB)
+ hdisk3          U787B.001.DNW0974-P1-C4-T1-L8-L0   16 Bit LVD SCSI Disk
Drive (73400 MB)
+ ses0            U787B.001.DNW0974-P1-C4-T1-L15-L0  SCSI Enclosure Services
Device
+ scsi1           U787B.001.DNW0974-P1-C4-T2         Ultra320 SCSI RAID
Adapter Physical bus
+ sisraid0        U787B.001.DNW0974-P1-C4-T3         Ultra320 SCSI RAID
Adapter Logical bus
* pci6            U787B.001.DNW0974-P1               PCI Bus
* pci7            U787B.001.DNW0974-P1-C5            PCI Bus
+ lai0            U787B.001.DNW0974-P1-C5-T1         GXT135P Graphics Adapter
+ L2cache0                                           L2 Cache
+ mem0                                               Memory
+ proc0                                              Processor
+ proc2                                              Processor
* kbd0            U787B.001.DNW0974-P1-T7-L1-L1      USB keyboard
* mouse0          U787B.001.DNW0974-P1-T7-L1-L3      USB mouse
```

8.8 Controlling resource use with Reliable Scalable Cluster Technology (RSCT)

RSCT is a set of software components that together provide a comprehensive clustering environment for AIX 5L and Linux. RSCT is the infrastructure used by a variety of IBM products to provide clusters with improved system availability, scalability, and ease of use. RSCT can also be used on stand-alone systems.

The basic RSCT components are:

- The Resource Monitoring and Control (RMC) subsystem. This is the scalable, reliable backbone of RSCT. It runs on a single machine or on each node (operating system image) of a cluster and provides a common abstraction for the resources of the individual system or the cluster of nodes. You can use RMC for single system monitoring, or for monitoring nodes in a cluster. In a cluster, however, RMC provides global access to subsystems and resources throughout the cluster, thus providing a single monitoring/management infrastructure for clusters.
- The RSCT core resource managers. A resource manager is a software layer between a resource (a hardware or software entity that provides services to some other component) and RMC. A resource manager maps programmatic abstractions in RMC into the actual calls and commands of a resource.

- The RSCT cluster security services, which provide the security infrastructure that enables RSCT components to authenticate the identity of other parties.
- The Topology Services subsystem, which, on some cluster configurations, provides node/network failure detection.
- The Group Services subsystem, which, on some cluster configurations, provides cross node/process coordination.

All the same type of resources are defined into resource classes. The resource class sets the common characteristics that instances of the resource class can have, while the resource itself contains the specific particular characteristic value.

To display the classes available on your machine, issue the **lsrsrc** command:

```
# lsrsrc
class_name
"IBM.Association"
"IBM.ATMDevice"
"IBM.AuditLog"
"IBM.AuditLogTemplate"
"IBM.Condition"
"IBM.EthernetDevice"
"IBM.EventResponse"
"IBM.FDDIDevice"
"IBM.Host"
"IBM.FileSystem"
"IBM.PagingDevice"
"IBM.PhysicalVolume"
"IBM.Processor"
"IBM.Program"
"IBM.TokenRingDevice"
"IBM.Sensor"
"IBM.Sfp"
"IBM.ServiceEvent"
"IBM.ManagementServer"
"IBM.NetworkInterface"
"IBM.HostPublic"
"IBM.DRM"
"IBM.WLM"
"IBM.LPAR"
"IBM.LPCommands"
```

8.9 Workload Manager

Workload Manager (WLM) allows the system administrator to divide resources between jobs. WLM, part of the BOS, provides isolation between user communities with very different system behaviors. This can prevent effective starvation of workloads with certain characteristics, such as interactive or low CPU usage jobs, by workloads with other characteristics, such as batch or high memory usage jobs. CPU time, memory, and I/O bandwidth are managed separately; therefore, different styles of applications can be managed.

AIX 5L WLM delivers the basic ability to give system administrators more control over how scheduler, Virtual Memory Manager (VMM), and device driver calls allocate CPU, physical memory, and I/O bandwidth to class-based user, group, application path, process type, or application tags. It allows a hierarchy of classes to be specified, processes to be automatically assigned to classes by their characteristics, and manual assignment of processes to classes. Classes can be superclasses or subclasses. WLM self-adjusts when there are no jobs in a class or when a class does not use all the resources that are allocated for it. The resources will automatically be distributed to other classes to match the policies of the system administrator.

Attention: Efficient use of WLM requires extensive knowledge of existing system processes and performance. If the system administrator configures WLM with extreme or inaccurate values, performance will be significantly degraded.

Figure 8-6 on page 366 shows an example setup with WLM.

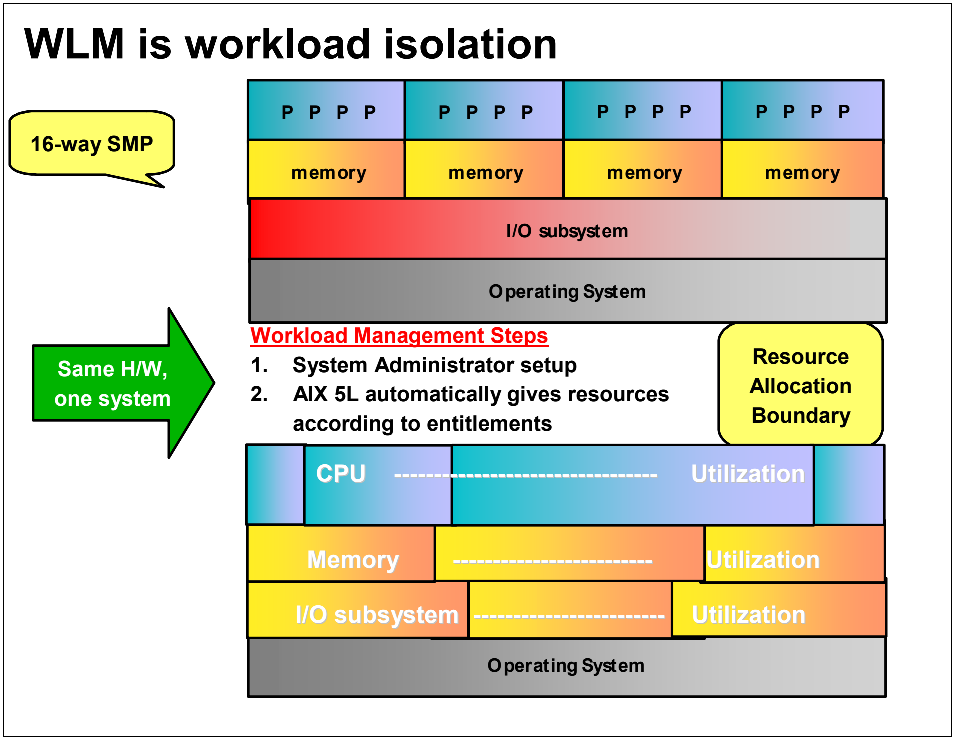


Figure 8-6 Example of WLM implementation

8.10 Partition Load Manager

Partition Load Manager (PLM) for AIX 5L is designed to automate the administration of memory and CPU resources across logical partitions within a single central electronics complex (CEC). To improve resource usage, PLM automates the migration of these resources between partitions based on partition load and priorities; partitions with a high demand will receive resources donated by or taken from partitions with a low demand. A user-defined policy governs how resources are moved. PLM will not contradict the partition definitions in the HMC. On the contrary, it adds additional flexibility on top of the micro-partitioning capability provided by the POWER Hypervisor.

The discussion of the benefits of partitioning in terms of network, disk, and CPU resource isolation are well covered in other publications. LPARs are the major tool in server consolidation. But how can you manage them? PLM is a solution.

PLM is part of the Advanced POWER Virtualization feature. It is supported on both dedicated and shared processor partitions of IBM Systems p5 servers running AIX 5L Version 5.3 or AIX 5L Version 5.2 (ML4) or later.

8.10.1 PLM operating modes

PLM can be started in one of two modes:

- Monitoring mode
- Management mode

In monitoring mode, PLM reports provide a number of statistics on resource usage in the managed partitions.

In management mode, PLM will initiate dynamic reconfiguration operations in order to match system resources with partition workload in accordance with the defined policy.

8.10.2 Monitoring mode

The PLM command for monitoring partition state is **xlpstat**. The syntax of the command is:

```
xlpstat  [ -r ] { -p | -f } filename [ interval ] [ count ]
```

The -p switch specifies that the list of managed partitions will be retrieved from the given policy file that is used when the PLM server is started. Alternatively you can provide a list of managed partitions in a text file, one partition per line, and use the -f flag to specify this file. The **xlpstat** command will query the status of the listed partitions. The output of this command does not distinguish between those partitions that are actively managed by PLM and that are not.

The -r switch prints the output in raw mode, which is easier to parse by scripting languages:

```
# xlpstat -p 2_groups
                   CPU                        MEM
        ------------------------    ------------------
  STAT  TYP    CUR    PCT   LOAD     CUR    PCT  PGSTL  HOST
group2:
  up    S     0.5    4.00   0.10     512  75.17      0  plmserver
  up    S     0.50  85.45   0.44     512  99.17    129  vio_client2
group1:
  up    D     1.00  95.09   0.19     512  99.23    129  app_server
  up    D     1.00   0.39   0.09     512  74.73      0  db_server
```

The display shows the state of each managed partition on a separate line (the partitions are grouped into PLM groups). In the above example, there are two groups.

The STAT column
: Indicates whether the partition is up or down. In the above example, all partitions are up.

The TYP column
: Shows whether the partition uses shared processor (S), dedicated processors (D), or if the **xlpstat** command cannot query the partition and the state is unknown (this column is shown as U, and is usually a sign of connection problems). The partitions in group 2 are shared and those in group 1 are dedicated.

The next six columns
: Split into two groups of three, one for CPU usage, the other for memory usage. The CUR column gives the current entitlement for CPU and memory, and the PCT column gives the percent utilization. The LOAD column indicates CPU load as measured by PLM, and the PGSTL column indicates the memory load measured with the page steal rate.

The HOST column
: Gives the name of the managed partition.

8.10.3 Management mode

PLM uses a client/server model, shown in Figure 8-7 on page 369, to monitor and manage partition resources. The clients act as agents on each of the managed partitions. The PLM server configures each of the agents (clients), setting the thresholds at which the server should be notified. The agents monitor the partition's resource usage and notify the PLM server whenever PLM-set thresholds are passed (under or overutilized). Based on a user-defined resource management policy, the PLM server invokes dynamic reconfiguration (DR) operations through the HMC to move resources from a spare pool to a partition or between partitions.

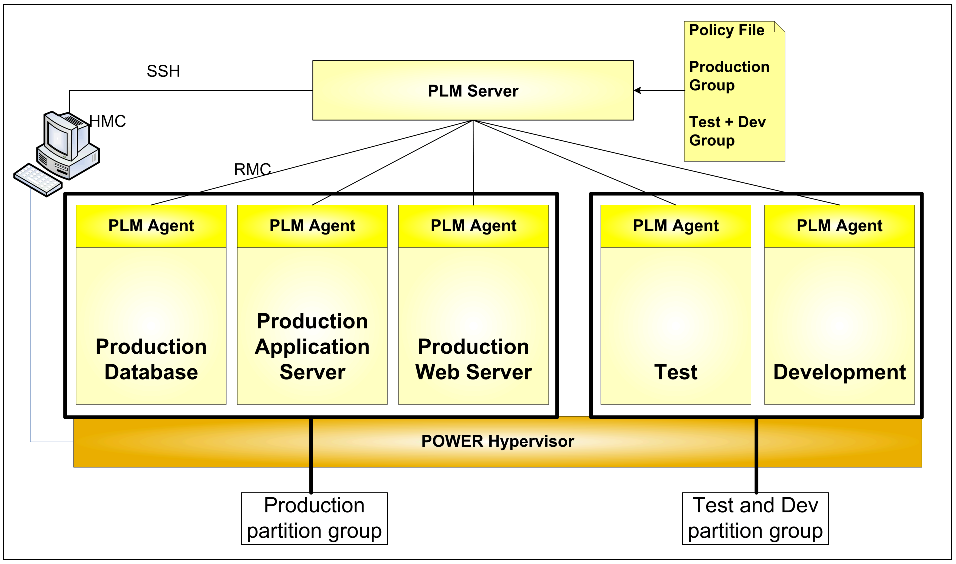


Figure 8-7 PLM architecture

PLM allows for groups of partitions. Resources within a group are managed independently. In Figure 8-7, two partition groups are shown, one for the production partitions, and the other for test and development.

Note: The following are the key points regarding PLM server management:

- The PLM server may reside either in a partition on the same server as the partitions being managed or on a different machine. When the PLM server runs in a partition, it is capable of managing its own partition.
- Multiple PLM servers may be run on one AIX 5L system.
- Different PLM groups on a given server may be managed by different PLM servers.
- A partition can have at most one PLM manager.
- It is not required that all partitions in a system be managed.
- One PLM server can manage partitions within only one managed CEC.
- It is not possible to have shared processor and dedicated processor partitions in the same PLM partition group.
- Resources are constrained to a group: A partition in one PLM group will never be given resources from another partition in another group.
- There should be at least two active partitions in a partition group.

Since each partition is monitored locally and the agents only communicate with the PLM server when an event occurs, PLM consumes a negligible amount of system and network resources.

8.10.4 Resource management policies

The resource management policy, which defines the managed partitions, their entitlements, and the thresholds in a policy file, also organizes the partitions into groups. The policy file defines guaranteed and variable CPU and memory resources to be allocated to the partitions and is loaded when PLM server is started. The PLM server is an application that runs on any AIX 5L system either in a logical partition or on another system and is the focal point of cross partition load management. For a detailed discussion on PLM policies as well as on how to install and configure PLM, see *Advanced POWER Virtualization on IBM System p5*, SG24-7940.

Every node managed by PLM must be defined in the policy file along with the values of several attributes, such as:

- Optional maximum, minimum, and guaranteed resource values
- The relative priority or weight of the partition
- Upper and lower load thresholds for resource event notification

PLM uses the Resource Monitoring and Control (RMC) subsystem for network communication. It is also used to communicate with the HMC to gather system information and to execute commands PLM requires to configure an SSH connection. PLM obtains partition load data from the Resource Management Agent via RMC, and LPAR configuration from the HMC.

If a partition has excess resources relative to the demand, or needs additional processors or memory, the PLM server moves resources from one LPAR to another by sending reconfiguration commands to the HMC via SSL.

The different partition states and load thresholds are shown in Figure 8-8 on page 371. For each resource, there is an upper and lower load threshold. Every time a threshold is crossed, PLM receives an RMC event. When the load on the resource is above the upper threshold, PLM considers the partition in need of additional resources; the partition is said to be a requestor. When the load on the resource is below the lower threshold, the partition becomes a potential donor. Normally, resources are only removed from donors when another partition enters the requestor state for the same resource. When the load on the resource is between the two thresholds, PLM considers that the resources available are adequate.

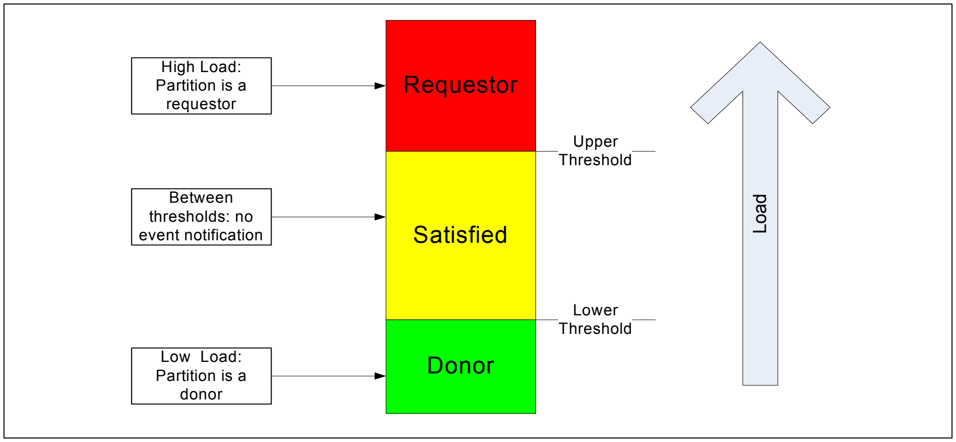


Figure 8-8 Resource utilization thresholds

The policy file, once loaded, is static; a partition's priority does not change upon the arrival of high priority work. The priority of partitions can only be changed by loading a new policy. Policy files can be changed on-the-fly, without stopping PLM.

Resource allocation

Part of a policy definition is the relative priority of each of the partitions in a group. This is done using a shares mechanism similar to that used in the AIX 5L Workload Manager (WLM). The greater the number of shares allocated to a partition, the higher its priority. To prevent some partitions from being starved, PLM modulates the partition priority with its current resource amounts. The mechanism is described below.

When PLM is notified that a partition has entered the requestor state, it will look for resources in the following order:

- Free pool of unallocated resources.
- A resource donor.
- A partition with fewer shares for the requested resource but has more resources than what is guaranteed as specified by the value of its *guaranteed* configurable.

If there are resources available in the free pool, they will be given to the requestor. If there are no resources in the free pool, the list of resource donors is checked. If there is a resource donor, the resource is moved from a donor to the requester. The amount of resource moved is the minimum of the delta values for the two partitions, or the amount that would give them equal priority as specified by the policy. If there are no resource donors, the list of partitions with more resources than what is guaranteed is checked.

Determining which node is more or less deserving of resources is done by comparing how much of any resource a partition owns relative to its priority as specified by the number of shares (weight). PLM calculates a ranking of partitions, including the requesting partition, from the list of partitions with excessive resources. A partition's priorities are defined as the following ratio:

$$\text{priority} = \frac{(\text{current\_amount} - \text{guaranteed\_amount})}{\text{shares}}$$

A lower value for this ratio represents a higher priority; partitions with a lower value of priority can take resources from partitions with a higher value.

Figure 8-9 shows an overview of the process for CPU resources in three capped partitions. Partition 3, under load, is a requestor. There are no free resources in the free pool or available donor partitions. PLM looks for partitions with excess resources (more resources than their guarantee). Both the other partitions in the group have excess resources. Partition 1 has the highest excess-to-shares ratio of all three partitions and resources will be moved from partition 1 to partition 3.

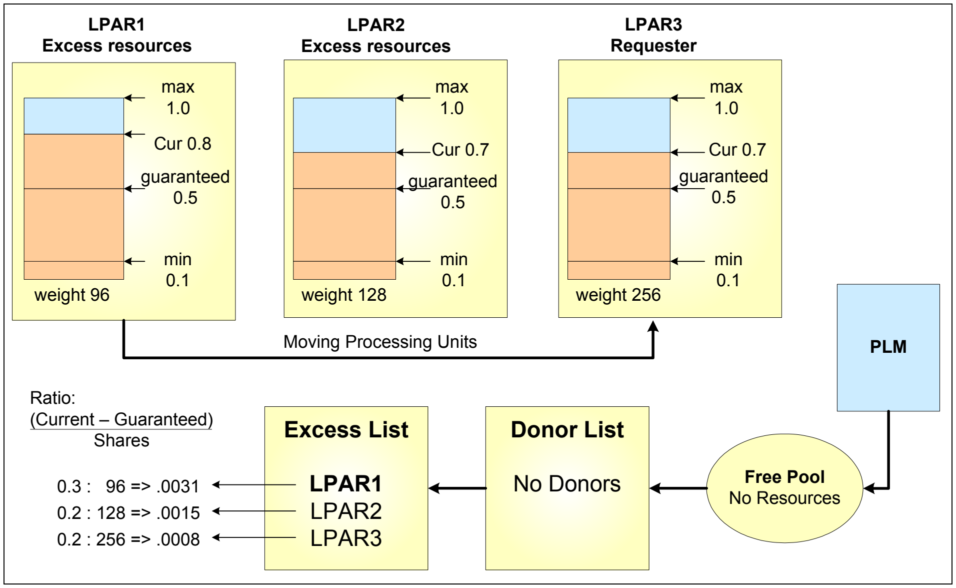


Figure 8-9 PLM resource distribution

If the request for a resource cannot be honored, it is queued and re-evaluated when resources become available.

Resource allocation constraints

There are number of restrictions that must be respected when specifying PLM policies:

- The minimum, guaranteed, and maximum values for the resources must satisfy the following relationship: minimum <= guaranteed <= maximum.
- If the minimum, guaranteed, and maximum values all have the same value or the group maximum is set to 0, then PLM will not manage the resource.
- Independently of the priority, PLM will not let a partition fall below its minimum or rise above its maximum limit for each resource.
- The range of the PLM maximums and minimums should be a subset of the range of the maximums and minimums set on the HMC; if not, then the intersection of the PLM and HMC values is used.
- If you do not specify any values for the PLM maximums and minimums, they default to the values on the HMC.

8.10.5 Memory management

PLM manages memory by moving logical memory blocks (LMBs) across partitions. The size of the LMB depends on the amount of memory installed in the CEC. It varies between 16 and 256 MB. The size of the LMB can be modified with the Advanced System Management Interface (ASMI) on the HMC.

To determine when there is demand for memory, PLM uses two metrics:

- Utilization percentage (ratio of memory in use to the amount of memory configured).
- The page replacement rate.

AIX 5L will make use of all the memory made available to it. It will not move pages out of memory unless it needs to bring in other pages from disk. This means that even if there is excess memory, AIX 5L will use it and it will be reported as used by the AIX 5L tools even though there are no applications using it. Because of this, partitions will rarely become donors.

8.10.6 Processor management

For dedicated processor partitions, PLM moves physical processors, one at a time, from partitions that are not utilizing them or that have a higher excess weight, to partitions that have demand for them. This enables dedicated processor partitions to better utilize their resources, for example, smoothing the transition from end-of-day transactions to the nightly batch jobs.

For shared processor partitions, PLM manages the entitled capacity and the number of virtual processors (VPs). When a partition has requested more processor capacity, PLM will increase the entitled capacity for the requesting partition if additional processor capacity is available. PLM can increase the number of virtual processors to increase the partition's potential to consume processor resources under high load conditions for both capped and uncapped partitions. Conversely, PLM will also decrease entitled capacity and the number of virtual processors under low-load conditions, to more efficiently utilize the underlying physical processors.

Note: The virtual processor folding optimization introduced in AIX 5L Version 5.3 ML3 renders the management of the virtual processor count by PLM unnecessary in most situations but removing virtual processors is more efficient than VP folding so in some circumstances management of virtual processors by PLM may be appropriate.

9

Problem determination and resolution

This chapter describes different problem determination tools, and when to use them. Examples show how to interpret the output of those commands and take the right action.

© Copyright IBM Corp. 2006. All rights reserved.

9.1 Problem determination and resolution

This section provides information to assist you in solving problems with the network, hardware, or operating system.

9.1.1 Network problems

You will find important aspects of network support and troubleshooting on systems in this section, including modifying network adapters and interfaces. Note that this does not present the full scope of network support (since networks can be very complex environments), but provides a useful starting point for support professionals.

The ping command

The **ping** command is useful for:

- Determining the status of the network and various foreign hosts
- Tracking and isolating hardware and software problems
- Testing, measuring, and managing networks

The **ping** command sends an Internet Control Message Protocol (ICMP) ECHO_REQUEST to obtain an ICMP ECHO_RESPONSE from a host or gateway in the network. If the host is operational and in the network, it responds to the echo request. Each echo request contains an Internet Protocol (IP) and ICMP header, followed by a timeval structure, and enough bytes to fill out the packet.

By default, the **ping** command sends one datagram per second and prints one line of output for every response received. The **ping** command calculates round-trip times and packet loss statistics, and displays a brief summary on completion. The **ping** command completes when the program times out or on receipt of a SIGINT signal (Ctrl-C).

The only mandatory parameter for the **ping** is either a valid host name or Internet address.

> **Note:** Because of the load that continuous echo requests can place on the system, repeated requests should be used primarily for problem isolation.

The following examples show some uses of the **ping** command:

- To check the network connection to a host and specify five echo requests to send, use the following command:

```
# ping -c 5 server2
PING server2.itsc.austin.ibm.com: (9.3.5.195): 56 data bytes
64 bytes from 9.3.5.195: icmp_seq=0 ttl=255 time=0 ms
64 bytes from 9.3.5.195: icmp_seq=1 ttl=255 time=0 ms
64 bytes from 9.3.5.195: icmp_seq=2 ttl=255 time=0 ms
64 bytes from 9.3.5.195: icmp_seq=3 ttl=255 time=0 ms
64 bytes from 9.3.5.195: icmp_seq=4 ttl=255 time=0 ms

----server2.itsc.austin.ibm.com PING Statistics----
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 0/0/0 ms
#
```

- To display the route buffer on the returned packets, run:

```
# ping -R server2
PING server2.itsc.austin.ibm.com: (9.3.5.195): 56 data bytes
64 bytes from 9.3.5.195: icmp_seq=0 ttl=255 time=0 ms
RR:     server2.itsc.austin.ibm.com (9.3.5.195)
        server3.itsc.austin.ibm.com (9.3.5.196)
64 bytes from 9.3.5.195: icmp_seq=1 ttl=255 time=0 ms   (same route)
64 bytes from 9.3.5.195: icmp_seq=2 ttl=255 time=0 ms   (same route)
64 bytes from 9.3.5.195: icmp_seq=3 ttl=255 time=0 ms   (same route)
64 bytes from 9.3.5.195: icmp_seq=4 ttl=255 time=0 ms   (same route)
64 bytes from 9.3.5.195: icmp_seq=5 ttl=255 time=0 ms   (same route)
^C
----server2.itsc.austin.ibm.com PING Statistics----
6 packets transmitted, 6 packets received, 0% packet loss
round-trip min/avg/max = 0/0/0 ms
#
```

Note: The IP header is only large enough for nine routes. Also, many hosts and gateways ignore this option.

If you cannot reach other computers on the same subnetwork with the **ping** command, look for problems on your system's network configuration. The **arp** and **ifconfig** commands can help you to isolate the problem.

The arp command

The **arp** command displays and modifies the Internet-to-physical address (MAC address) translation tables used by the Address Resolution Protocol. The **arp** command displays the current ARP entry for the host specified by the HostName variable. The host can be specified by name or number, using Internet dotted decimal notation.

Note: You can use the **arp** command to isolate problems on only the local subnetwork.

For example, when trying to ping a system with IP 9.3.5.193, we get this output:

```
# ping 9.3.5.193
PING 9.3.5.193: (9.3.5.193): 56 data bytes
^C
----9.3.5.193 PING Statistics----
7 packets transmitted, 0 packets received, 100% packet loss
#
```

But, when pinging a system with IP 9.3.5.196, we get this output:

```
# ping 9.3.5.196
PING 9.3.5.196: (9.3.5.196): 56 data bytes
64 bytes from 9.3.5.196: icmp_seq=0 ttl=255 time=0 ms
64 bytes from 9.3.5.196: icmp_seq=1 ttl=255 time=0 ms
64 bytes from 9.3.5.196: icmp_seq=2 ttl=255 time=0 ms
64 bytes from 9.3.5.196: icmp_seq=3 ttl=255 time=0 ms
64 bytes from 9.3.5.196: icmp_seq=4 ttl=255 time=0 ms
^C
----9.3.5.196 PING Statistics----
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 0/0/0 ms
#
```

Looking at the arp table, we find (after filtering the output through the **grep** command):

```
# arp -a | grep 9.3.5.19
  bcmmint.itsc.austin.ibm.com (9.3.5.193) at (incomplete)
server3.itsc.austin.ibm.com (9.3.5.196) at 0:2:55:d3:dd:0 [ethernet] stored in
 bucket 46
```

The physical address for a system with IP 9.3.5.193 could not be resolved; you should look for problems on that system. If your system is not able to resolve the physical address from computers on the same subnetwork, you should look at your cable connections. The **ifconfig** command is also useful for checking the status of the network interfaces.

The ifconfig command

The **ifconfig** command can be used in network problem determination tasks to show or change the status of the interfaces or redefine IP addresses, as shown in the following examples:

- To show the status of interface en2, type:

```
# ifconfig en2
en2:
flags=5e080863,c0<UP,BROADCAST,NOTRAILERS,RUNNING,SIMPLEX,MULTICAST,GROUPRT
,64BIT,CHECKSUM_OFFLOAD,PSEG,CHAIN>
       inet 9.3.5.196 netmask 0xffffff00 broadcast 9.3.5.255
        tcp_sendspace 131072 tcp_recvspace 65536
#
```

  The interface en2 is enabled (UP).

- To show only those interfaces that are down, type:

```
# ifconfig -a -d
en1:
flags=5e080862,c0<BROADCAST,NOTRAILERS,RUNNING,SIMPLEX,MULTICAST,GROUPRT,64
BIT,CHECKSUM_OFFLOAD,PSEG,CHAIN>
       inet 2.2.2.2 netmask 0xffffff00 broadcast 2.2.2.255
        tcp_sendspace 131072 tcp_recvspace 65536
#
```

  This output shows that interface en1 is disabled (DOWN). If you have problems reaching computers on the subnet on which this interface is configured, run the **errpt** command and look and see if any error has been reported for the interface (for example, a duplicate IP address in the network); run the **diag** command to run diagnostics over the interface.

  To enable interface en1 and mark it as active, type the command:

```
# ifconfig en1 up
```

  If the interfaces do not have problems, they are in an active state, and your system cannot reach computers on the same subnetwork, you should check that the interface's subnet mask is the correct one. To change the subnet mask to 255.255.255.252 for en1 interface, type the command:

```
# ifconfig en1 netmask 255.255.255.252 up
```

Network routing configuration can be also a source of communication problems; the **traceroute** command helps you to trace routes.

The traceroute command

The **traceroute** command attempts to trace the route an IP packet follows to an Internet host by launching UDP probe packets with a small, maximum time-to-live value, then listening for an ICMP TIME_EXCEEDED response from gateways along the way. Probes are started with a time-to-live value of one hop, which is increased one hop at a time until an ICMP PORT_UNREACHABLE message is returned. The ICMP PORT_UNREACHABLE message indicates either that the host has been located or the command has reached the maximum number of hops allowed for the trace.

> **Note:** The **traceroute** command is intended for use in network testing, measurement, and management. It should be used primarily for manual fault isolation. Because of the load it imposes in the network, the **traceroute** command should not be used during normal operations or from automated scripts.

The only mandatory parameter for the **traceroute** command is the destination host name or IP number. The **traceroute** command will determine the length of the probe packet based on the Maximum Transmission Unit (MTU) of the outgoing interface. The UDP probe packets are set to an unlikely value so as to prevent processing by the destination host.

For example, if you wanted to see which route information travels from the system with the IP address of 9.3.1.141 to the system with the IP address of 9.8.0.8, you would enter:

```
# traceroute 9.8.0.8
trying to get source for 9.8.0.8
source should be 9.3.1.141
traceroute to 9.8.0.8 (9.8.0.8) from 9.3.1.141 (9.3.1.141), 30 hops max
outgoing MTU = 1492
 1 itso.austin.ibm.com (9.3.1.74)  11 ms  2 ms  2 ms
 2  9.444.33.129 (9.444.33.129)  4 ms  4 ms  4 ms
 3 site1.austin.ibm.com (9.3.90.200)  7 ms
 4 site2.south.ibm.com (9.3.200.202)  8 ms  7 ms
 5 site3.austin.ibm.com (199.4.213.125)  17 ms  19 ms  17 ms
 6 9.88.1.174 (9.88.1.174)  74 ms  80 ms  71 ms
 7 9.8.0.8 (9.8.0.8)  327 ms  329 ms  327 ms
```

The netstat command

To view information about the local network routing configuration, run the **netstat** command, as shown in 8.5.4, “The netstat command” on page 327.

9.1.2 Hardware problems

In this section, we show how to interpret the output generated by the **errpt** command and to run diagnostics to find and troubleshoot hardware related problems.

The errpt command

The **errpt** command generates an error report from entries in an error log, but it does not perform error log analysis; for error analysis, use the **diag** command.

Consider the following sample error report generated by issuing the **errpt -a** command:

```
# errpt -a
---------------------------------------------------------------------------
LABEL:          LVM_SA_STALEPP
IDENTIFIER:     EAA3D429

Date/Time:       Tue Dec  6 23:24:37 CST 2005
Sequence Number: 441
Machine Id:      00C7CD9E4C00
Node Id:         lpar20
Class:           S
Type:            UNKN
Resource Name:   LVDD

Description
PHYSICAL PARTITION MARKED STALE

Detail Data
PHYSICAL VOLUME DEVICE MAJOR/MINOR
0000 0000 0000 0000
PHYSICAL PARTITION NUMBER (DECIMAL)
                   227
LOGICAL VOLUME DEVICE MAJOR/MINOR
8000 000A 0000 0005
SENSE DATA
00C7 CD9E 0000 4C00 0000 0107 706A 336C 0000 0000 0000 0000 0000 0000 0000 0000
---------------------------------------------------------------------------
.
.
.
(lines ommited)
.
.
.
#
```

Depending upon the type of errors, the **`errpt -a`** report may contain the following information:

LABEL Predefined name for the event.

IDENTIFIER Numerical identifier for the event.

Date/Time Date and time of the event.

Sequence Number Unique number for the event.

Machine ID Identification number of your system processor unit.

Node ID Mnemonic name of your system.

Class General source of the error. The possible error classes are:

- **H** Hardware.
- **S** Software.
- **O** Informational messages.
- **U** Undetermined.

Type Severity of the error that has occurred. The following types of errors are possible:

- **PEND** The loss of availability of a device or component is imminent.
- **PERF** The performance of the device or component has degraded to below an acceptable level.
- **PERM** A condition that could not be recovered from. Error types with this value are usually the most severe errors and are more likely to mean that you have a defective hardware device or software module. Error types other than PERM usually do not indicate a defect, but they are recorded so that they can be analyzed by the diagnostics programs.
- **TEMP** A condition that was recovered from after a number of unsuccessful attempts. This error type is also used to record informational entries, such as data transfer statistics for DASD devices.
- **UNKN** It is not possible to determine the severity of the error.
- **INFO** The error log entry is informational and was not the result of an error.

Resource Name Name of the resource that has detected the error. For software errors. this is the name of a software component or an executable program. For hardware errors, this is the name of a device or system component. It does not indicate that the component is faulty or needs replacement. Instead, it is used to determine the appropriate diagnostic modules to be used to analyze the error.

Resource Class General class of the resource that detected the failure (for example, a device class of disk).

Resource Type Type of the resource that detected the failure.

Location Code Path to the device. There may be up to four fields, which refer to drawer, slot, connector, and port, respectively.

VPD Vital product data. The contents of this field, if any, vary. Error log entries for devices typically return information concerning the device manufacturer, serial number, Engineering Change levels, and Read Only Storage levels.

Description Summary of the error.

Probable Cause List of some of the possible sources of the error.

User Causes List of possible reasons for errors due to user mistakes. An improperly inserted disk and external devices (such as modems and printers) that are not turned on are examples of user-caused errors.

Actions Description of recommended actions for correcting a user-caused error.

Install Causes List of possible reasons for errors due to incorrect installation or configuration procedures. Examples of this type of error include hardware and software mismatches, incorrect installation of cables or cable connections becoming loose, and improperly configured systems.

Actions Description of recommended actions for correcting an installation-caused error.

Failure Causes List of possible defects in hardware or software.

Actions Description of recommended actions for correcting the failure. For hardware errors, this will lead to running the diagnostic programs.

Detailed Data Failure data that is unique for each error log entry, such as device sense data.

Consider the following sample error output from the **errpt** command when run without flags:

```
# errpt
IDENTIFIER TIMESTAMP  T C RESOURCE_NAME  DESCRIPTION
A6DF45AA   1207112405 I O RMCdaemon      The daemon is started.
2BFA76F6   1205155605 T S SYSPROC        SYSTEM SHUTDOWN BY USER
9DBCFDEE   1207112305 T O errdemon       ERROR LOGGING TURNED ON
192AC071   1205155505 T O errdemon       ERROR LOGGING TURNED OFF
291D64C3   1202154105 I H sysplanar0     platform_dump indicator event
BFE4C025   1202154105 P H sysplanar0     UNDETERMINED ERROR
291D64C3   1202154105 I H sysplanar0     platform_dump indicator event
291D64C3   1202145005 I H sysplanar0     platform_dump indicator event
EAA3D429   1202144505 U S LVDD           PHYSICAL PARTITION MARKED STALE
BFE4C025   1202144505 P H sysplanar0     UNDETERMINED ERROR
F6A86ED5   1202144105 P S ent7           SERVICE CALL FAILED
F6A86ED5   1202144105 P S ent6           SERVICE CALL FAILED
#
```

- An error-class value of H and an error-type value of PERM (identifier BFE4C025 in this example) indicates that the system encountered a hardware problem and could not recover from it. Diagnostic information might be associated with this type of error.
- An error-class value of H and an error-type value of PEND indicates that a piece of hardware may become unavailable soon due to numerous errors detected by the system.
- An error-class value of S and an error-type value of PERM (identifier F6A86ED5 in this example) indicate that the system encountered a problem with software and could not recover from it.
- An error-class value of S and an error-type value of TEMP indicates that the system encountered a problem with software. After several attempts, the system was able to recover from the problem.
- An error class value of O indicates that an informational message has been logged (identifiers A6DF45AA, 9DBCFDEE, and 192AC071 in this example).

When you suspect there is a hardware problem, the **diag** command assists you in finding it.

The diag command

The **diag** command is the starting point to run a wide choice of tasks and service aids. The **diag** uses the error log to diagnose hardware problems.

To correctly diagnose new system problems, the system deletes hardware-related entries older than 90 days from the error log. The system also deletes software-related entries 30 days after they are logged.

Note: If you remove hardware error entries less than 90 days old, you can limit the effectiveness of the error log analysis.

For example, to run problem diagnostics, type the following:

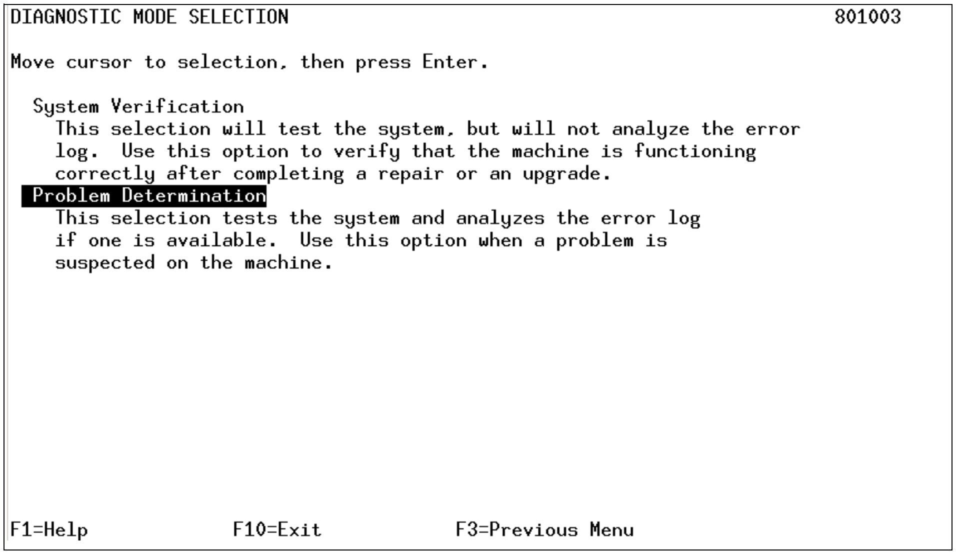
```
# diag
```

On the DIAGNOSTIC OPERATING INSTRUCTIONS screen, press Enter to continue; the menu shown on Figure 9-1 is displayed.

```
FUNCTION SELECTION                                                      801002


Move cursor to selection, then press Enter.

  Diagnostic Routines
    This selection will test the machine hardware. Wrap plugs and
    other advanced functions will not be used.
  Advanced Diagnostics Routines
    This selection will test the machine hardware. Wrap plugs and
    other advanced functions will be used.
  Task Selection (Diagnostics, Advanced Diagnostics, Service Aids, etc.)
    This selection will list the tasks supported by these procedures.
    Once a task is selected, a resource menu may be presented showing
    all resources supported by the task.
  Resource Selection
    This selection will list the resources in the system that are supported
    by these procedures. Once a resource is selected, a task menu will
    be presented showing all tasks that can be run on the resource(s).




F1=Help              F10=Exit             F3=Previous Menu
```

Figure 9-1 Function selection menu

If you select Diagnostic Routines and press Enter, then the DIAGNOSTIC MODE SELECTION menu shown in Figure 9-2 is displayed.



```
DIAGNOSTIC MODE SELECTION                                          801003

Move cursor to selection, then press Enter.

  System Verification
    This selection will test the system, but will not analyze the error
    log.  Use this option to verify that the machine is functioning
    correctly after completing a repair or an upgrade.
  Problem Determination
    This selection tests the system and analyzes the error log
    if one is available.  Use this option when a problem is
    suspected on the machine.

F1=Help               F10=Exit               F3=Previous Menu
```

Figure 9-2 Diagnostic mode selection menu

If you select Problem Determination, tests on the devices are run. Once the tests finish, the system may show you:

- The PREVIOUS DIAGNOSTICS RESULTS, which means one or more errors have already been reported.
- The ADDITIONAL INFORMATION screen, which means that the error log analysis has detected something that may need your attention.
- The DIAGNOSTIC SELECTION menu shown in Figure 9-3 on page 387.

```
DIAGNOSTIC SELECTION                                                    801006

From the list below, select any number of resources by moving
the cursor to the resource and pressing 'Enter'.
To cancel the selection, press 'Enter' again.
To list the supported tasks for the resource highlighted, press 'List'.

Once all selections have been made, press 'Commit'.
To avoid selecting a resource, press 'Previous Menu'.


[TOP]
  All Resources
      This selection will select all resources that can be tested.
  sysplanar0                             System Planar
                 U787B.001.DNW108F-
  ent2             P1-C1-T1              10/100/1000 Base-TX PCI-X Adapter
                                         (14106902)
  ent0             P1-T9                 2-Port 10/100/1000 Base-TX PCI-X Adapter
                                         (14108902)
[MORE...21]

F1=Help             F4=List             F7=Commit           F10=Exit
F3=Previous Menu
```

Figure 9-3 Diagnostic selection

On the DIAGNOSTIC SELECTION screen, a list of devices is shown. You can select any resource by moving the cursor to the resource and pressing Enter; a plus symbol (+) indicates that a resource has been selected. You can cancel the selection of the resource by selecting it and pressing Enter again. Resources preceded by an M are missing, and those preceded by a P have at least one missing path.

Once all the selections have been made, press the Commit key (F7 or ESC-7). The diagnostics will run over the resources selected.

9.2 Reasons to monitor root mail

AIX 5L has different mechanisms to keep the system administrator informed if something is going wrong with the system.

The following methods can be used to find error messages:

- Check the local root mailbox.
- Check the syslog output files. See 8.4, "The system log configuration" on page 303 for additional information.
- Check the **alog** output files.

Most of the processes will send a mail to the root account with detailed information.

To check root's mail, execute the **mail** command:

```
# mail
Mail [5.2 UCB] [AIX 5.X]  Type ? for help.
"/var/spool/mail/root": 1 message 1 new
>N  1 root          Tue Mar 17 17:11  24/872  "diagela message from localhost"
```

To read the diagela message, press the number in front of the mail (in this example, 1):

```
? 1
Message  1:
From root Wed Nov 09 17:11:48 2005
Date: Tue, 09 Nov 2005 17:11:48 -0600
From: root
To: root
Subject: diagela message from localhost

A PROBLEM WAS DETECTED ON Wed Nov 09 17:11:48 CST 2005                801014

The Service Request Number(s)/Probable Cause(s)
(causes are listed in descending order of probability):

  110000AC: Power/Cooling subsystem Unrecovered Error, general. Refer to the
            system service documentation for more information.
           Error log information:
                Date: Wed Nov 09 17:11:45 CST 2005
                Sequence number: 11
                Label: SCAN_ERROR_CHRP
    Priority: L  FRU: ACMODUL Location:
    U787B.001.DNW108F

?
```

To exit the **mail** command, type q.

For more information about the mail functionality, consult 12.1, "Overview of mail system" on page 582.

The Diagnostics Automatic Error Log Analysis (diagela) provides the capability to do error log analysis whenever a permanent hardware error is logged. No automatic testing is done. If diagnostics determines that the error requires a service action, it sends a message to your console and to all system groups. The message contains the SRN, or a corrective action. Diagela is enabled by default at BOS installation time.

The following example shows another diagela message:

```
From root Tue Nov 22 15:18:46 2005
Date: Tue, 22 Nov 2005 15:18:46 -0600
From: root
To: root
Subject: diagela message from server3

A PROBLEM WAS DETECTED ON Tue Nov 22 15:18:46 CST 2005
801014

The Service Request Number(s)/Probable Cause(s)
(causes are listed in descending order of probability):

  57D-129: Error log analysis indicates a SCSI bus problem.
           Error log information:
                 Date: Tue Nov 22 15:18:39 CST 2005
                 Sequence number: 80
                 Label: DISK_ERR3
    n/a              FRU: n/a                  SCSI bus problem: cables,
                                               terminators or other SCSI
                                               devices
    U787B.001.DNW108F-P1-T14-L8-L0
    hdisk7           FRU: 00P3833              16 Bit LVD SCSI Disk Drive
                                               (73400 MB)
    U787B.001.DNW108F-P1-T14-L8-L0
    sisioa1          FRU: 80P5530              PCI-X Dual Channel U320 SCSI
                                               RAID Adapter
    U787B.001.DNW108F-P1
    n/a              FRU: n/a                  Software
    U787B.001.DNW108F-P1-T14-L8-L0
```

From the content of this mail, hdisk7 has a problem on the U320 SCSI RAID adapter on server3.

Another example of a command that uses mail messages is **crontab**. Because **crontab** is running in the background without user interaction, it sends a mail with the output of the scheduler.

A sample **crontab** output message can be similar to:

```
From daemon Wed Nov 23 11:07:01 2005
Date: Wed, 23 Nov 2005 11:07:01 -0600
From: daemon
To: root
Subject: Output from cron job /usr/sbin/nonexistingprogram, root@server3, exit
status 127

Cron Environment:
 SHELL =

PATH=/usr/bin:/etc:/usr/sbin:/usr/ucb:/usr/bin/X11:/sbin:/usr/java14/jre/bin:/u
sr/java14/bin
 CRONDIR=/var/spool/cron/crontabs
 ATDIR=/var/spool/cron/atjobs
 LOGNAME=root
 HOME=/

Your "cron" job executed on server3 on Wed Nov 23 11:07:00 CST 2005
/usr/sbin/nonexistingprogram

produced the following output:

sh: /usr/sbin/nonexistingprogram:  not found.

*****************************************************************
        cron: The previous message is the standard output
        and standard error of one of the cron commands.
```

The cause of this error message is the non-existence of the /usr/sbin/nonexistingprogram.

Other software packages, especially security related ones, have the ability to specify the administrator. For example, in case of a security breach, illegal file permission change, or unauthorized passwd-file access, the system administrator receives a message.

9.3 System dump facility

Your system generates a system dump when a severe error occurs. System dumps can also be user-initiated by users with root user authority. A system dump creates a picture of your system's memory contents. System administrators and programmers can generate a dump and analyze its contents when debugging new applications.

9.3.1 Configure a dump device

When you install the operating system, the dump device is automatically configured for you. By default, the primary device is /dev/hd6, which is a paging logical volume, and the secondary device is /dev/sysdumpnull.

> **Note:** If your system has 4 GB or more of memory, then the default dump device is /dev/lg_dumplv, and is a dedicated dump device.

The dump device can be configured to either tape or a logical volume on the hard disk to store the system dump. A primary dump device is a dedicated dump device, while a secondary dump device is shared.

> **Note:** If you use a paging device for dump devices, only use hd6, the primary paging space. If you use a removable device, such as a tape or DVD, be aware that the dump does not span volumes; thus, the dump must fit on a single volume.

The **sysdumpdev** command changes the primary or secondary dump device designation in a system that is running, as shown in the following examples:

- To list the current dump destination, use the following command:

```
# sysdumpdev -l
primary             /dev/hd6
secondary           /dev/sysdumpnull
copy directory      /var/adm/ras
forced copy flag    TRUE
always allow dump   FALSE
dump compression    ON
#
```

- The following example shows the command to change the primary dump device from /dev/hd6 to the logical volume /dev/dumpdev:

```
# sysdumpdev -P -p /dev/dumpdev
primary             /dev/dumpdev
secondary           /dev/sysdumpnull
copy directory      /var/adm/ras
forced copy flag    TRUE
always allow dump   TRUE
dump compression    ON
#
```

> **Note:** You can use a dump logical volume outside the root volume group, if it is not a permanent dump device, for example, if the -P flag is not specified. However, if you choose a paging space, you cannot copy the dump device unless it is in rootvg. During that time, the system must copy the dump device; only rootvg is active before paging is started.

- To show statistical information about the previous dump, type the command:

```
# sysdumpdev -L
0453-039

Device name:         /dev/hd6
Major device number: 10
Minor device number: 2
Size:                65765888 bytes
Uncompressed Size:   643336319 bytes
Date/Time:           Mon Nov 21 10:43:09 CST 2005
Dump status:         0
dump completed successfully
Dump copy filename: /var/adm/ras/vmcore.3.Z
#
```

Increase the size of a dump device

Refer to the following section to determine the appropriate size for your dump logical volume and to increase the size of either a logical volume or a paging space logical volume.

Determining the size of a dump device

The size required for a dump is not a constant value, because the system does not dump paging space; only data that resides in real memory can be dumped. Paging space logical volumes will generally hold the system dump. However, because an incomplete dump may not be usable, follow the procedure below to make sure that you have enough dump space.

When a system dump occurs, all of the kernel segment that resides in real memory is dumped (the kernel segment is segment 0). Memory resident user data (such as u-blocks) are also dumped.

The minimum size for the dump space can best be determined using the **`sysdumpdev -e`** command. This gives an estimated dump size taking into account the memory currently in use by the system. If dumps are being compressed, then the estimate shown is for the compressed size of the dump, not the original size. In general, compressed dump size estimates will be much higher than the actual size. This occurs because of the unpredictably of the compression algorithm's efficiency. You should still ensure your dump device is large enough to hold the estimated size in order to avoid losing dump data.

For example, enter:

```
# sysdumpdev -e
0453-041 Estimated dump size in bytes: 120586240
#
```

The size of the dump device should be at least 120586249 bytes or 115 MB.

For example, if you need to increase the size of the primary dump device, follow the next steps:

1. Enter the **sysdumpdev** command to list the dump devices:

   ```
   # sysdumpdev -l
   primary              /dev/hd6
   secondary            /dev/sysdumpnull
   ...
   #
   ```

2. Determine the logical volume type of the primary dump device:

   ```
   # lslv  hd6
   LOGICAL VOLUME:     hd6                    VOLUME GROUP:   rootvg
   LV IDENTIFIER:      00c5e9de00004c0000000107 7c447899.2 PERMISSION:
   read/write
   VG STATE:           active/complete        LV STATE:       opened/syncd
   TYPE:               paging                 WRITE VERIFY:   off
   MAX LPs:            512                    PP SIZE:        128 megabyte(s)
   ```

3. Increase the size of a dump device:

   a. If you have confirmed that your dump device is a paging space (as in this example), refer to 5.10, “Paging space” on page 189.

   b. If you have confirmed that your dump device type is sysdump, use the **extendlv** command to increase the space available. See 6.5, “Summary of the LVM commands” on page 268.

9.3.2 Start a system dump

The dump can either be system initiated or user initiated.

System initiated dump

If your system stops with an 888 number flashing in the operator panel display, the system has generated a dump and saved it to a primary dump device.

Understanding flashing 888 error messages

An 888 sequence in operator panel display indicates that either a hardware or software problem has been detected and a diagnostic message is ready to be read.

Perform the following steps to record the information contained in the 888 sequence message:

1. Wait until the 888 sequence displays.
2. Record, in sequence, every code displayed after the 888. On systems with a 3-digit or a 4-digit operator panel, you may need to press the reset button to view the additional digits after the 888. Some systems use an advance button to perform this task, or the increment and decrement buttons (see Figure 9-5 on page 398).
3. Stop recording when the 888 reappears.

A value of 102 for the first code recorded indicates that a software or hardware occurs during system execution of an application, as shown on Figure 9-4. The value of mmm indicates the cause of halt and crash code (see Table 9-4 on page 396), and the value of ddd indicates the dump status and dump code

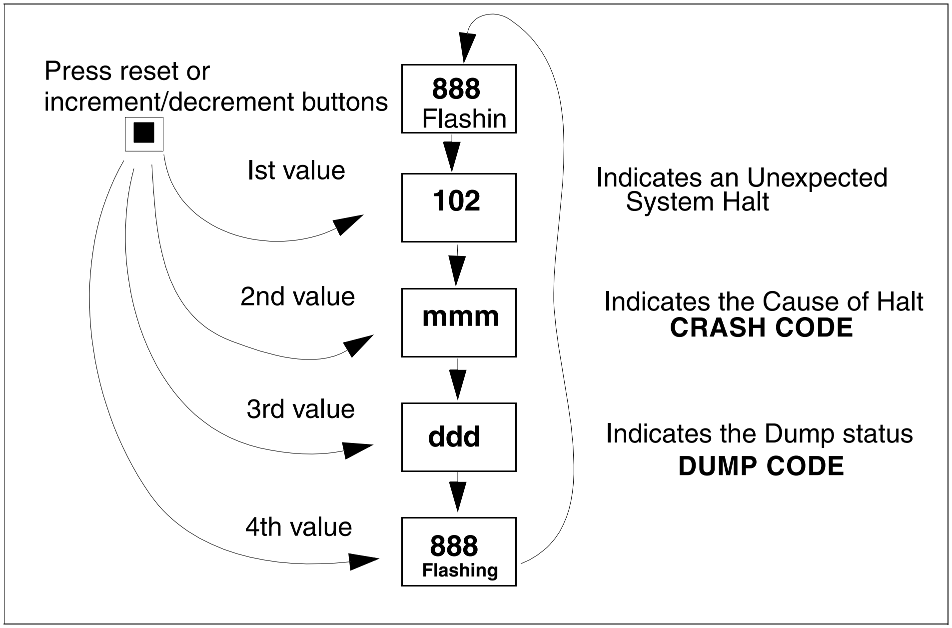


Figure 9-4 Unexpected system halt - Three-digit display string

Crash codes

The crash codes that follow are part a Type 102 message. These crash codes are grouped into three categories:

Category1 Dump analysis is the appropriate first action in problem determination. Begin the problem determination with software support (Table 9-1).

Category 2 Dump analysis most likely will not aid in problem determination. Begin the problem determination process with hardware support (Table 9-2).

Category 3 Both software and hardware support may be needed in problem determination (Table 9-3 on page 396).

Table 9-1 Category 1 crash progress codes

Progress code	Description/Application
300	Data storage interrupt for the processor
32x	Data storage interrupt because of an IO exception from IOCC
38x	Data storage interrupt because of an I/O exception from SLA
400	Instruction storage interrupt
700	Program interrupt

Table 9-2 Category 2 crash progress codes

Progress code	Description/Application
200	Machine check due to memory bus error (RAS/CAS parity)
201	Machine check due to memory time out
202	Machine check due to memory card failure
203	Machine check due to address exception: address out of range
204	Machine check due to attempted store into R
205	Machine check due to uncorrectable error correction code, due to address parity
206	Machine check due to uncorrectable error correction code
207	Machine check due to undefined error
208	Machine check due to an L2 uncorrectable ECC
500	External interrupt because of a scrub memory error

Progress code	Description/Application
501	External interrupt because of an unidentified error
51x	External interrupt because of a DMA memory bus error
52x	External interrupt
53x	External interrupt because of IOCC bus timeout
54x	External interrupt because of IOCC keyboard check
800	Floating point unavailable

Table 9-3 Category 3 crash progress codes

Progress code	Description/Application
000	Unexpected system interrupt.
558	There is not enough memory to continue the IPL.

Table 9-4 shows the system dump codes.

Table 9-4 System dump codes

Code	Description
0c0	Dump completed successfully.
0c2	Dump started.
0c3	The dump is inhibited.
0c4	Dump completed unsuccessfully. Not enough space available on dump device. Partial dump available.
0c5	Dump failed to start. Unexpected error occurred when attempting to write to dump device.
0c7	Network dump in progress.
0c8	Dump disabled. System configuration does not include a dump device.
0c9	System initiated dump started.

When the system dump completes, the system either halts or reboots, depending upon the setting of the auto restart attribute of sys0:

```
# lsattr -El sys0 -a autorestart
autorestart true Automatically REBOOT system after a crash True
#
```

You can change this setting by typing the following command:

```
# chdev -l sys0 -a autorestart=false
sys0 changed
# lsattr -El sys0 -a autorestart
autorestart false Automatically REBOOT system after a crash True
#
```

User initiated dump

There are several ways for a user to invoke a system dump, depending upon the system condition. The following sections explain them.

Starting a dump from the command line

If you initiate the system dump from the command line, use the **sysdumpstart** command with a -p flag to write to the primary device or a -s flag to write to the secondary device.

Starting a dump from the physical control panel

If an HMC is not connected to your system, you can initiate a system dump by using the physical control panel.

> **Note:** The physical control panel is your initial interface with the server. You can use the physical control panel to perform functions such as IPL, power on, and power off. Control panel functions range in complexity from functions that display status (such as IPL speed) to low-level service functions that only service representatives must access.

Figure 9-5 shows a type of physical control panel, and Table 9-5 has the description of each one of its components.

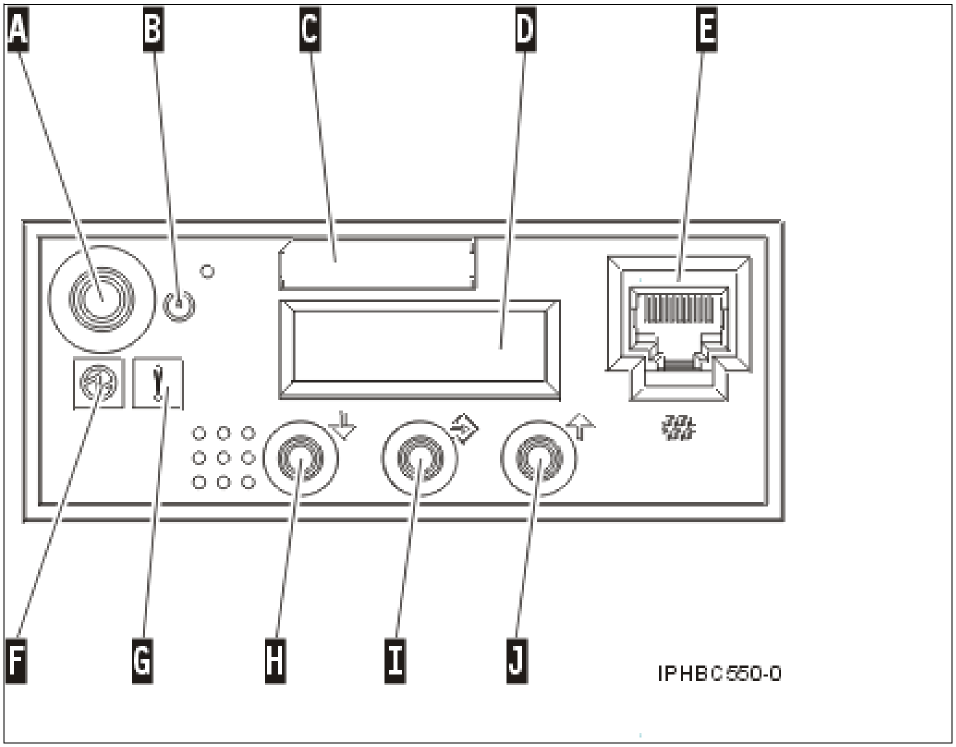


Figure 9-5 View of the physical control panel

Table 9-5 describes the components of the physical control panel.

Table 9-5 Components of the physical control panel

Component	Description
A	Power button
B	ON/OFF power symbol
C	Type and serial number label
D	Function/data display
E	Serial port 1 connector
F	Power on light ► A blinking light indicates standby power to the unit. ► A constant light indicates full system power to the unit.

Component	Description
G	System attention light
H	Decrement button
I	Enter button
J	Increment button

To generate a system dump using the physical control panel, use the following procedure:

1. Set the physical control panel to manual mode:

   a. Use the Increment button to scroll to function 02:

   ```
   0 2 _ _ _ _ _ _ _ _ _ _ _ _ _ _
   _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
   ```

   b. Press the Enter button to start function 02.

   c. Press the Enter button again to move to the second character on the function 02 menu. The current system operating mode is displayed with a pointer:

   ```
   0 2 _ _ B _ _ N < _ _ _ _ _ _ _
   _ _ _ _ _ _ _ _ _ _ _ _ P _ _ _
   ```

   d. Use the Increment button to scroll through the system operating modes and select M for manual:

   ```
   0 2 _ _ B _ _ M < _ _ _ _ _ _ _
   _ _ _ _ _ _ _ _ _ _ _ _ P _ _ _
   ```

   e. Press the Enter button to select the operating system mode.

   f. Press the Enter button again to exit function 02.

   Your system is now in manual mode.

2. Use function 22 (partition dump) to dump a partition's operating system data:

   a. Use the Increment or Decrement buttons to scroll to function 22:

   ```
   2 2 _ _ _ _ _ _ _ _ _ _ _ _ _ _
   _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
   ```

   b. Press the Enter button to start function 22:

   ```
   2 2 _ _ _ _ 0 0 _ _ _ _ _ _ _ _
   _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
   ```

   c. The panel displays the partition dump debounce SRC:

   ```
   A 1 0 0 3 0 2 2 _ _ _ _ _ _ _ _
   _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
   ```

d. Use the Increment or Decrement buttons to scroll to function 22:

```
2 2 _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
```

e. Press the Enter button to start function 22:

```
2 2 _ _ _ _ 0 0 _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
```

The system dump initiates.

Starting a dump using special key sequences

You can initiate a dump using the following special key sequences:

- Ctrl-Alt-NumPad1 to write to the primary dump device.
- Ctrl-Alt-NumPad2 to write to the secondary dump device.

> **Note:** By default, the system will not begin a dump by pressing the special key sequences. To enable dumps via the use of the special key sequences, run the following command before trying to use them:
>
> ```
> # sysdumpdev -K
> ```

Starting a dump from the HMC

To initiate a system dump from the HMC, do the following procedure:

1. On the HMC console, choose the partition you want to dump and press the right mouse button; a pop-up menu appears, as shown in Figure 9-6 on page 401.
2. Select **Restart Partition** and click the left mouse button; a new screen appears, as shown on Figure 9-7 on page 401.
3. Select **Dump** and click the **OK** button to start the dump.

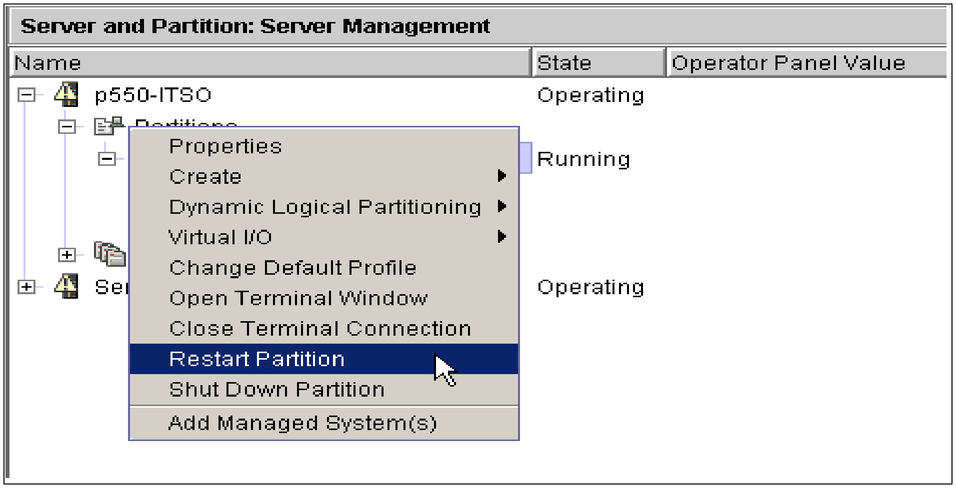


Figure 9-6 Restart Partition option from the pop-up menu

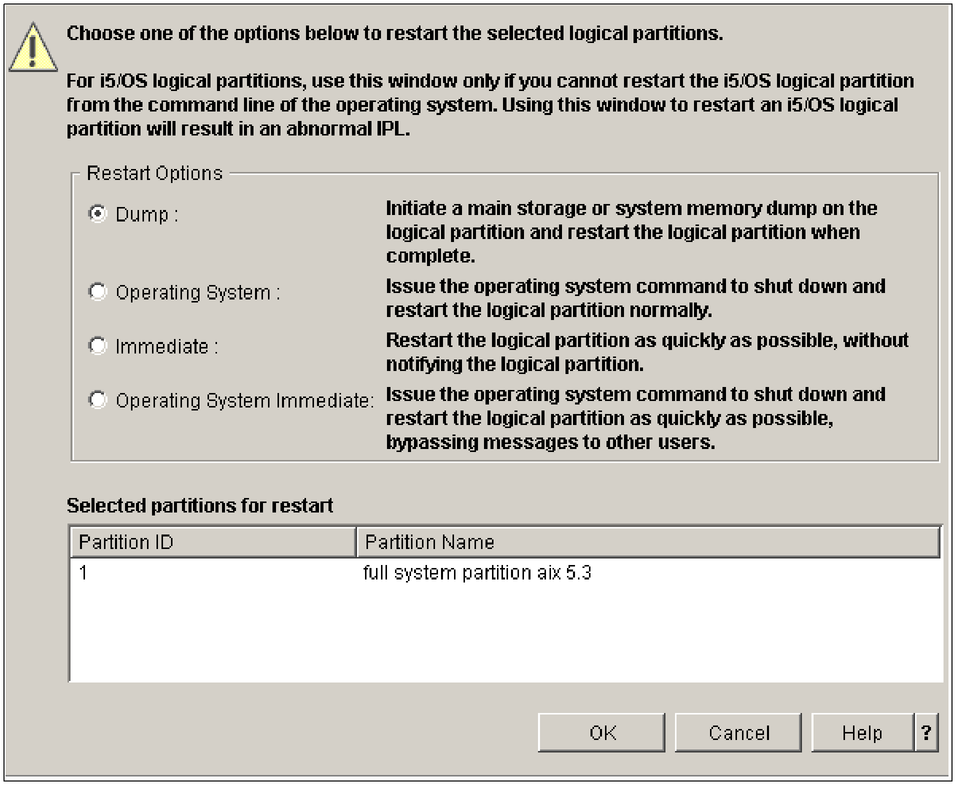


Figure 9-7 Restart options available for a partition, including initiate a dump

9.3.3 Copy a system dump

The **pax** command allows you to copy, create, and modify files that are greater than 2 GB in size, such as system dumps, from one location to another. This can be useful in migrating dumps, as the **tar** and **cpio** commands (formerly used for copying dumps in previous versions of AIX) cannot handle manipulating files that are larger than 2 GB in size. The **pax** command can also view and modify files in tar and cpio format.

The syntax for the **pax** command has many flags. Table 9-6 shows the commonly used flags for the **pax** command. For a complete list of options, refer to the **pax** man page.

Table 9-6 Commonly used flags for the pax command

Flag	Description
-a	Appends files to the end of an archive.
-c	Matches all file or archive members except those specified by the Pattern parameter.
-d	Causes directories being copied, archived, or extracted, to match only the directory itself and not the contents of the directory.
-f *Archive*	Specifies the path of an archive file to be used instead of standard input (when the -w flag is not specified) or standard output (when the -w flag is specified but the -r flag is not).
-i	Renames files or archives interactively.
-k	Prevents the **pax** command from writing over existing files.
-r	Reads an archive file from the standard input.
-v	Writes information about the process. If neither the -r or -w flags are specified, the -v flag produces a verbose table of contents; otherwise, archive member path names are written to standard error.
-w	Writes files to the standard output in the specified archive format.
-x *Format*	Specifies the output archive format. In copy mode, if no format is provided, **pax** uses the pax format by default.

The **pax** command can output archives in one of three formats by using the -x flag:

pax The default interchange format. This format can support file sizes larger than 2 GB in size.

cpio The extended cpio interchange format.

ustar The extended tar interchange format.

Note: If the -x flag is not specified during an archive operation, the **pax** command will output archives in tar format by default.

For example, to view the contents of the tar file /tmp/test.tar, run the command:

```
# pax -vf /tmp/test.tar
USTAR format archive
-rw-r--r--  0 root     system    344834 Nov 22 16:39 ./services
-rw-rw-r--  0 root     system      1836 Nov 22 18:37 ./hosts
#
```

To create a **pax** command archive on tape that contains two files, run the following command:

```
# pax -x pax -wvf /dev/rmt0 /var/adm/ras/cfglog /var/adm/ras/nimlog
/var/adm/ras/cfglog
/var/adm/ras/nimlog
#
```

To un-tar the tar file /tmp/test.tar to the current directory, run the command:

```
# pax -rvf /tmp/test.tar
USTAR format archive
./services
./hosts
#
```

To copy the file run.pax to the /tmp directory, run the command:

```
# pax -rw run.pax /tmp
```

9.3.4 Compile and copy system information onto media

The **snap** command is used to gather configuration information of the system. It provides a convenient method of sending the **lslpp** and **errpt** output to your service support center. It gathers the information and compresses the information to a pax file. The file may then be written to a device, such as tape or DVD, or transmitted to a remote system. The information gathered with the **snap** command might be required to identify and resolve system problems.

Note: Root user authority is required to execute the **snap** command.

Table 9-7 shows commonly used flags for the **snap** command.

Table 9-7 Commonly used flags for the snap command

Flag	Description
-a	Gathers all system configuration information. This option requires approximately 8 MB of temporary disk space.
-c	Creates a compressed pax image (snap.pax.Z file) of all files in the /tmp/ibmsupt directory tree or other named output directory.
-D	Gathers dump and /unix information. The primary dump device is used.
-d *Directory*	Identifies the optional **snap** command output directory (/tmp/ibmsupt is default).
-f	Gathers file system information.
-g	Gathers the output of the **lslpp -hac** command, which is required to recreate the exact operating system environments. Writes output to the /tmp/ibmsupt/general/lslpp.hac file. Also collects general system information and writes the output to the /tmp/ibmsupt/general/general.snap file.
-k	Gathers kernel information.
-L	Gathers LVM information.
-o *OutputDevice*	Copies the compressed image onto diskette or tape.
-r	Removes the **snap** command output from the /tmp/ibmsupt directory.
-v *Component*	Displays the output of the commands executed by the **snap** command. Use this flag to view the specified name or group of files.

The default directory for the output from the **snap** command is /tmp/ibmsupt. If you want to name an optional directory, use the -d option with the path of the desired output directory. Approximately 8 MB of temporary disk space is required when executing all of the **snap** command options. The cleanup option -r should be used to remove the information saved by the **snap** command and to retrieve disk space.

For example, to copy general system information, including file system, kernel parameters and dump information to rmt0, type the command:

```
# /usr/sbin/snap -gfkD -o /dev/rmt0
```

Note: If you intend to use a tape to send a snap image to IBM for software support, the tape must be one of the following formats:

- 8 mm, 2.3 GB capacity
- 8 mm, 5.0 GB capacity
- 4 mm, 4.0 GB capacity

Using other formats prevents or delays IBM software support from being able to examine the contents.

Before executing the **snap -c** or **snap -o** commands, any additional information required by your service support center should be copied to the /tmp/ibmsupt directory. For example, you may be asked by your service support center to provide a test case that demonstrates the problem. The test case should be copied to the /tmp/ibmsupt/testcase directory. When the -c or -o option of the **snap** command is executed, the test case will be included.

9.3.5 Analyzing system dumps

The **kdb** command allows you to examine a system dump or a running kernel. The **kdb** command interprets and formats control structures in the system and provides miscellaneous functions for examining a dump.

The syntax for the **kdb** command is:

```
kdb [flags] [ SystemImageFile [ KernelFile [KernelModule ... ]]]
```

The SystemImageFile parameter specifies the file that contains the system image. The value can indicate a system dump, the name of a dump device, or the /dev/pmem special file. The default SystemImageFile is /dev/pmem.

The KernelFile parameter specifies the AIX 5L kernel that kdb will use to resolve kernel symbol definitions. A kernel file must be available. When examining a system dump, it is imperative that the kernel file be the same as the kernel that was used to take the system dump. The default for the KernelFile is /unix.

The KernelModule parameters specify the file names of any additional kernel modules that the **kdb** command uses to resolve symbol definitions not found in the kernel file itself.

For example, to invoke the **kdb** command using a dump file named /var/adm/ras/vmcore.0 and the kernel file named /unix, type:

```
# kdb /var/adm/ras/vmcore.0 /unix
```

The **kdb** program returns a (0)> prompt and waits for entry of a subcommand.

> **Note:** The kernel file is used by the **kdb** command to resolve symbol names from the dump file. It is imperative that the kernel file specified on the command line is the kernel file that was running at the time the system dump was created.

9.4 Using the alog command

The **alog** command reads from standard input, and writes to standard output and a log file simultaneously. The log file is a circular fixed-size log.

All boot messages are collected in a boot log file, because at boot time there is not console available. Boot information is usually collected in /var/adm/ras/bootlog. It is a very good idea to check the bootlog file when you are investigating boot problems. The file will contain output generated by the **cfgmgr** command and rc.boot.

You can use the **alog -L** command to display all log files that are defined for your system.

To change the size of the boot log, use the command:

```
# echo " boot log resizing" | alog -t boot -s 8192
```

To display the boot log, use the command:

```
# alog -t boot -o|more
```

In Example 9-1, we show an excerpt of the bootlog.

Example 9-1 Last lines of bootlog

```
Configuration time: 12 seconds

Saving Base Customize Data to boot disk
Starting the sync daemon
Starting the error daemon
System initialization completed.
Starting Multi-user Initialization
 Performing auto-varyon of Volume Groups
 Activating all paging spaces
0517-075 swapon: Paging device /dev/hd6 is already active.

The current volume is: /dev/hd1
Primary superblock is valid.
```

```
The current volume is: /dev/hd10opt
Primary superblock is valid.
 Performing all automatic mounts
Multi-user initialization completed
 boot log resizing
```

In Example 9-2, we show an excerpt of the bootlog.

Example 9-2 Last lines of bosinst log

```
Erasing drives
Preparing target disks.
hdisk0 changed
rootvg
Making boot logical volume.
hd5
Making paging logical volumes.
hd6
Making logical volumes.
hd8
hd4
hd2
hd9var
hd3
hd1
hd10opt
Forming the jfs log.
Making file systems.
File system created successfully.
130864 kilobytes total disk space.
File system created successfully.
130864 kilobytes total disk space.
File system created successfully.
130864 kilobytes total disk space.
File system created successfully.
130864 kilobytes total disk space.
File system created successfully.
130864 kilobytes total disk space.
File system created successfully.
130864 kilobytes total disk space.
Mounting file systems.
Restoring base operating system
Initializing disk environment.

Over mounting /.
Copying Cu* to disk.
Installing additional software.
mkitab: ident entry found in /etc/inittab
Please wait...
```

```
Some locale or message software did not install.
See /var/adm/ras/devinst.log for details.
lft0 changed
Initializing trusted computing base.
Initializing dump device.
Platform dump is not supported on this system.
primary              /dev/hd6
secondary            /dev/sysdumpnull
copy directory       /var/adm/ras
forced copy flag     TRUE
always allow dump    FALSE
dump compression     ON
Creating boot image.

bosboot: Boot image is 23795 512 byte blocks.

                               Running Customization
```

9.5 Determine appropriate actions for user problems

Several check commands (**grpck**, **usrck**, **pwdck**, **sysck**, and **tcbck**) and list commands (**lsuser** and **lsgroup**) are available for use by root (or anyone in the security group). These commands can help you determine and fix some problem users environment.

9.5.1 The usrck command

The **usrck** command verifies the correctness of the user definitions in the user database files, by checking the definitions for ALL the users or for the users specified by the User parameter. If more than one user is specified, there must be a space between the names. You must select a flag to indicate whether the system should try to fix erroneous attributes.

The command first checks the entries in the /etc/passwd file. If you indicate that the system should fix errors, duplicate user names are reported and removed. Duplicate IDs are reported only because there is no system fix. If an entry has fewer than six colon-separated fields, the entry is reported, but not fixed. The **usrck** command next checks specific user attributes in other files.

The **usrck** command verifies that each user name listed in the /etc/passwd file has a stanza in the /etc/security/user, /etc/security/limits, and /etc/security/passwd files. The **usrck** command also verifies that each group name listed in the /etc/group file has a stanza in the /etc/security/group file. The

usrck command using the -y flag creates stanzas in the security files for the missing user and group names. Table 9-8 explains the commonly used flags.

usrck verifies the correctness of a user definition.

Table 9-8 Commonly used flags for the usrck command

Flag	Description
-n	Reports errors but does not fix them.
-p	Fixes errors but does not report them.
-t	Reports errors and asks if they should be fixed.
-y	Fixes errors and reports them.

The following are useful examples:

- To verify that all the users exist in the user database, and have any errors reported (but not fixed), enter:

```
# usrck  -n ALL
```

- To delete, from the user definitions, those users who are not in the user database files, and have any errors reported, enter:

```
# usrck  -y ALL
```

9.5.2 The grpck command

The **grpck** command verifies the correctness of the group definitions in the user database files by checking the definitions for *all* the groups or for the groups specified by the Group parameter. If more than one group is specified, there must be a space between the groups.

Note: The **grpck** command writes its messages to stderr.

You must select a flag to indicate whether the system should try to fix erroneous attributes. The following attributes are checked:

name Checks the uniqueness and composition of the group name. The group name must be a unique string of eight bytes or less. It cannot begin with a + (plus sign), a : (colon), a - (minus sign), or a ~ (tilde). It cannot contain a colon (:) in the string and cannot be the ALL or default keywords. No system fix is possible.

groupID Checks the uniqueness and composition of the group ID. The ID must not be null and must consist of decimal digits only. No system fix is possible.

users Checks the existence of the users listed in the group database files. If you indicate that the system should fix errors, it will delete all the users that are not found in the user database files.

adms Checks the existence of the users listed as group administrators in the group database files. If you indicate that the system should fix errors, it will delete all the administrators that are not found in the user database files.

admin Checks for a valid admin attribute for each group in the /etc/security/group file. No system fix is available.

Table 9-9 shows a summary of the flags for the **grpck** command.

Table 9-9 Commonly used flags for the grpck command

Flag	Description
-n	Reports errors but does not fix them.
-p	Fixes errors but does not report them.
-t	Reports errors and asks if they should be fixed.
-y	Fixes errors and reports them.

The following are useful examples:

- To verify that all the group members and administrators exist in the user database, and have any errors reported (but not fixed), enter:

```
# grpck  -n ALL
```

- To verify that all the group members and administrators exist in the user database and to have errors fixed, but not reported, enter:

```
# grpck  -p ALL
```

- To verify the uniqueness of the group name and group ID defined for the install group, enter *one* of the following commands:

```
# grpck  -n install
# grpck  -t install
# grpck  -y install
```

The **grpck** command does not correct the group names and IDs. Therefore, the -n, -t, and -y flags report problems with group names and group IDs, but do not correct them.

9.5.3 The pwdck command

The **pwdck** command verifies the correctness of the password information in the user database files by checking the definitions for *all* the users or for the users specified by the User parameter. If more than one user is specified, there must be a space between the names.

> **Note:** The **pwdck** command writes its messages to stderr.

You must select a flag to indicate whether the system should try to fix erroneous attributes. The following attributes are checked for locally defined users in the /etc/passwd file:

entry Ensures that each entry is readable and that it contains at least two : (colons). If you indicate that the system should fix errors, the entire entry is discarded.

passwd Ensures that the password field is an ! (exclamation point). If you indicate that the system should fix errors, it transfers the information in the password field to the /etc/security/passwd file, updates the lastupdate attribute in the /etc/security/passwd file, and then replaces the password field in the /etc/passwd file with an !. In general, passwords are required if the minalpha, minother, or minlen password restriction is set to a nonzero value in the /etc/security/user file.

user Ensures that the user name is a unique string of 8 bytes or less. It cannot begin with a + (plus sign), a : (colon), a - (minus sign), or a ~ (tilde). It cannot contain a : (colon) in the string and cannot be the ALL, default, or * keywords. If you indicate that the system should fix errors, it removes this user's entry line from the /etc/passwd file. If the user name starts with a + or a - symbol, the user is not locally defined, and checks are not performed.

Table 9-10 summarizes the different flags for the **pwdck** command.

Table 9-10 Commonly used flags for the pwdck command

Flag	Description
-n	Reports errors but does not fix them.
-p	Fixes errors but does not report them.
-t	Reports errors and asks if they should be fixed.
-y	Fixes errors and reports them.

Examples

- To verify that all local users have valid passwords, enter:

  ```
  # pwdck  -y ALL
  ```

  This reports errors, and fixes them.

- To ensure that user gpsilva has a valid stanza in the /etc/security/passwd file, enter:

  ```
  # pwdck  -y gpsilva
  ```

9.5.4 The sysck command

The **sysck** command checks file definitions against the extracted files from the installation and update media and updates the Software Vital Product Data (SWVPD) database. The **sysck** command does not recognize the following special characters in file names: `, ', \, ", ^, ( ), |, {}, [], <>, and : . If a file name contains one of these characters, the **sysck** command fails.

The **sysck** command is primarily used during the installation and update of software products.

When invoked with the -i flag, the **sysck** command checks the attributes of an extracted file with its file definitions, updates the SWVPD, and attempts to fix some errors if they exist.

The File parameter is the name of the stanza file that contains the file definitions. An example of such a file is the /etc/security/sysck.cfg file, although the **syschk** command does not use this file. The **sysck** command checks the size, links, symlinks, owner, group, and mode attributes of a file for which the type attribute is set to FILE. When invoked with the -v flag as well as the -i flag, the **sysck** command also checks the checksum value of a file.

The **sysck** command updates the file name, product name, type, checksum, and size of each file in the SWVPD database.

To fix errors, the **sysck** command resets the attribute of the installed or updated file to the defined value in the File stanza file.

When invoked with the -u flag, the **sysck** command removes the entry from the SWVPD database for each file that is part of the software product ProductName. The **sysck** command also deletes any hard links and symbolic links for each file, as defined in the SWVPD database.

Table 9-11 gives an overview of the flags for the **sysck** command.

Table 9-11 Commonly used flags for the sysck command

Flag	Description
-f File	Specifies the name of the stanza file that contains the file definitions.
-i	Checks for the correct installation of a software product's files. Updates the SWVPD database with the file definitions, and attempts to fix some errors if found.
-N	Specifies that the SWVPD database should not be updated.
-O {r|s|u}	Specifies which part of the SWVPD is to be updated, as follows: ► r: Specifies the root part of the SWVPD. ► s: Specifies the /usr/share part of the SWVPD. ► u: Specifies the /usr part of the SWVPD (default).
-R RootPath	Use RootPath as root instead of "/".
-s SaveFile	Takes a snapshot of what is currently in the VPD and saves it in stanza format to the file specified by SaveFile. Called with the -u option. No action is taken in the database with this flag. Must be used with the -f option.
-u	Deletes file entries from the SWVPD and deletes hard links and symbolic links.
-v	Verifies that the checksum is correct.
ProductName	Specifies the installable software product or option that is being checked.

The following are several useful examples

- A product that uses the **installp** command to install has an inventory file in its image. To add the definitions to the inventory database and check permissions, links, checksums, and so on, enter:

```
# sysck  -i  -f smart.rte.inventory smart.rte
```

where smart.rte.inventory would look like the following:

```
/usr/bin/smart.exec:
   class = apply,inventory,smart.rte
   owner = bin
   group = bin
   mode = 555
   type = FILE
   size = 2744
   checksum = "047203"
```

- To remove any links to files for a product that has been removed from the system and remove the files from the inventory database, enter:

```
# sysck  -u  -f smart.rte.inventory smart.rte
```

9.5.5 The lsgroup and lsuser commands

These commands are used internally by SMIT, but you can also use them directly. Direct use may be more convenient when you want to place their output in a file.

The commands are:

```
# lsgroup -f ALL >> /tmp/check
# lsuser    -f ALL >> /tmp/check
```

In the examples shown here, these commands create the file /tmp/check and write their output into it. There is too much output for direct display on the screen, so the output would normally be directed to a file. These commands display most of the control information about users and groups. These commands may be used by any user, but much more information is displayed when they are used by root (or any member of the security group).

The **lsuser** command is directly useful when used by root for a specific user:

```
# lsuser gpsilva
gpsilva id=203 pgrp=staff groups=staff home=/home/gpsilva shell=/usr/bin/ksh
login=true su=true rlogin=true daemon=true admin=false sugroups=ALL admgroups=
tpath=nosak ttys=ALL expires=0 auth1=SYSTEM auth2=NONE umask=22 registry=files
SYSTEM=compat logintimes= loginretries=0 pwdwarntime=0 account_locked=false
minage=0 maxage=0 maxexpired=-1 minalpha=0 minother=0 mindiff=0 maxrepeats=8
minlen=0 histexpire=0 histsize=0 pwdchecks= dictionlist= fsize=2097151 cpu=-1
data=262144 stack=65536 core=2097151 rss=65536 nofiles=2000
time_last_login=1133304706 tty_last_login=ftp host_last_login=kcyb72b
unsuccessful_login_count=0 roles=
```

This command will display several lines containing control information for user gpsilva. When used with the ALL operand, information is displayed for all users in the system. Several formatting options are available. You could write local programs to extract and display locally-important information obtained from these commands.

9.5.6 The user limits

The /etc/security/limits file is an ASCII file that contains stanzas that specify the process resource limits for each user. These limits are set by individual attributes within a stanza.

Each stanza is identified by a user name followed by a colon and contains attributes in the Attribute=Value form. Each attribute is ended by a new-line character, and each stanza is ended by an additional new-line character. If you do not define an attribute for a user, the system applies default values.

When you create a user with the **mkuser** command, the system adds a stanza for the user to the /etc/security/limits file. Once the stanza exists, you can use the **chuser** command to change the user's limits. To display the current limits for a user, use the **lsuser** command. To remove users and their stanzas, use the **rmuser** command.

9.6 Identifying hardware problems

The error logging facility records hardware and software failures in the error log for informational purposes or for fault detection and corrective action. 8.3, “The system error log” on page 298 describes errorlog in more detail. The other tool where the hardware problems are logged is in the mail box for the root user. The root mail for error messages is described on 9.2, “Reasons to monitor root mail” on page 387.

9.7 Replacing hot plug devices

Some of the devices installed on your systems have the ability to be installed, removed, or replaced without shutting down the system. Devices that support these operations include PCI adapters, SCSI devices, and some RAID devices. Not all PCI adapters support the hot plug task. To determine whether a specific device supports hot plug operations, consult the hardware product documentation.

In order to replace a hot plug device, you have to go to the Hot Plug Task menu that can be accessed from the diagnostics tools utility.

Depending on the environment and the software packages installed, selecting this task displays some of the following subtasks:

PCI hot plug manager The PCI hot plug manager (HPM) task is a SMIT menu that allows you to identify, add, remove, or replace PCI adapters that are hot pluggable.

RAID hot plug devices This task allows the user to identify or remove a RAID device in a system unit that uses a SCSI Enclosure Services (SES) device.

SCSI and SCSI RAID hot plug manager
This task was known as *SCSI Device Identification and Removal* or *Identify and Remove Resources* in previous releases. This task allows the user to identify, add, remove, and replace a SCSI device in a system unit that uses a SCSI Enclosure Services (SES) device.

You can use the **lsslot** command to display dynamically reconfigurable slots and their characteristics.

In Example 9-3, we use the **lsslot** command to display the number, location, and capabilities of hot plug PCI slots.

Example 9-3 Using the lsslot command

```
# lsslot -c pci
# Slot                      Description                          Device(s)
U787B.001.DNW0974-P1-C1  PCI-X capable, 64 bit, 133MHz slot  ent2
U787B.001.DNW0974-P1-C2  PCI-X capable, 64 bit, 133MHz slot  Empty
U787B.001.DNW0974-P1-C3  PCI-X capable, 64 bit, 133MHz slot  Empty
U787B.001.DNW0974-P1-C4  PCI-X capable, 64 bit, 133MHz slot  sisioa0
U787B.001.DNW0974-P1-C5  PCI-X capable, 64 bit, 133MHz slot  pci7 lai0
```

Before replacing a hot plug adapter or disk, you should unconfigure all other devices or interfaces that are dependent on the physical device you want to remove.

In Example 9-4 on page 417, we show the error message displayed when trying to remove a device that has not been unconfigured. The **lsdev** command confirms that the device is in status available.

Example 9-4   Error message displayed when trying to replace a configured device sisia0

```
COMMAND STATUS

Command: failed       stdout: yes          stderr: no

Before command completion, additional instructions may appear below.


The visual indicator for the specified PCI slot has
been set to the identify state. Press Enter to continue
or enter x to exit.


The specified slot contains device(s) that are currently
configured. Unconfigure the following device(s) and try again.

sisioa0

# lsdev -C|grep sis
gxme0      Defined                 Graphics Data Transfer Assist Subsystem
sisioa0    Available 05-08         PCI-X Dual Channel U320 SCSI RAID Adapter
sisioa1    Available 09-08         PCI-X Dual Channel U320 SCSI RAID Adapter
sisraid0   Available 05-08-ff      Ultra320 SCSI RAID Adapter Logical bus
sisraid1   Available 09-08-ff      Ultra320 SCSI RAID Adapter Logical bus
```

If you are running in a multiple partitions environment, you should verify if the device you are trying to remove or replace has been marked as required for a partition. If this is the case, the partition should be stopped prior to device replacement.

In the following scenario, we will replace the ent2 Ethernet PCI adapter. From the output of the **lsslot** command displayed in Example 9-3 on page 416, we can find the physical location of the adapter.

We remove the corresponding interface en2, as shown in Example 9-5.

Example 9-5   Deleting en2 interface

```
# ifconfig -a
en0:
flags=5e080863,c0<UP,BROADCAST,NOTRAILERS,RUNNING,SIMPLEX,MULTICAST,GROUPRT,64B
IT,CHECKSUM_OFFLOAD,PSEG,CHAIN>
        inet 9.3.5.195 netmask 0xffffff00 broadcast 9.3.5.255
         tcp_sendspace 131072 tcp_recvspace 65536
en1:
flags=5e080863,c0<UP,BROADCAST,NOTRAILERS,RUNNING,SIMPLEX,MULTICAST,GROUPRT,64B
IT,CHECKSUM_OFFLOAD,PSEG,CHAIN>
        inet 9.1.1.1 netmask 0xfffffff8 broadcast 9.1.1.7
```

```
        tcp_sendspace 131072 tcp_recvspace 65536
en2:
flags=5e080863,c0<UP,BROADCAST,NOTRAILERS,RUNNING,SIMPLEX,MULTICAST,GROUPRT,64B
IT,CHECKSUM_OFFLOAD,PSEG,CHAIN>
        inet 0.0.0.0 netmask 0x0
         tcp_sendspace 131072 tcp_recvspace 65536
lo0:
flags=e08084b<UP,BROADCAST,LOOPBACK,RUNNING,SIMPLEX,MULTICAST,GROUPRT,64BIT>
        inet 127.0.0.1 netmask 0xff000000 broadcast 127.255.255.255
        inet6 ::1/0
         tcp_sendspace 131072 tcp_recvspace 131072 rfc1323 1
# ifconfig en2 detach
# ifconfig -a
en0:
flags=5e080863,c0<UP,BROADCAST,NOTRAILERS,RUNNING,SIMPLEX,MULTICAST,GROUPRT,64B
IT,CHECKSUM_OFFLOAD,PSEG,CHAIN>
        inet 9.3.5.195 netmask 0xffffff00 broadcast 9.3.5.255
         tcp_sendspace 131072 tcp_recvspace 65536
en1:
flags=5e080863,c0<UP,BROADCAST,NOTRAILERS,RUNNING,SIMPLEX,MULTICAST,GROUPRT,64B
IT,CHECKSUM_OFFLOAD,PSEG,CHAIN>
        inet 9.1.1.1 netmask 0xfffffff8 broadcast 9.1.1.7
         tcp_sendspace 131072 tcp_recvspace 65536
lo0:
flags=e08084b<UP,BROADCAST,LOOPBACK,RUNNING,SIMPLEX,MULTICAST,GROUPRT,64BIT>
        inet 127.0.0.1 netmask 0xff000000 broadcast 127.255.255.255
        inet6 ::1/0
         tcp_sendspace 131072 tcp_recvspace 131072 rfc1323 1
```

The Hot Plug Task can be started with either the SMIT or diagnostics (DIAG) tools menu. In this scenario, we use the DIAG tool.

1. To start the DIAG tool, enter the **diag** command at the command prompt. Press Enter when presented with the screen in Figure 9-8 on page 419.

```
DIAGNOSTIC OPERATING INSTRUCTIONS  VERSION 5.3.0.30                        801001


LICENSED MATERIAL and LICENSED INTERNAL CODE - PROPERTY OF IBM
(C) COPYRIGHTS BY IBM AND BY OTHERS 1982, 2004.
ALL RIGHTS RESERVED.

These programs contain diagnostics, service aids, and tasks for
the system.  These procedures should be used whenever problems
with the system occur which have not been corrected by any
software application procedures available.

In general, the procedures will run automatically.  However,
sometimes you will be required to select options, inform the
system when to continue, and do simple tasks.

Several keys are used to control the procedures:
- The Enter key continues the procedure or performs an action.
- The Backspace key allows keying errors to be corrected.
- The cursor keys are used to select an option.

Press the F3 key to exit or press Enter to continue.
```

Figure 9-8 Initial Diagnostic screen

2. From the Function Selection menu, choose the Task Selection (Diagnostics, Advanced Diagnostics, Service Aids, etc.) option. Figure 9-9 shows the Diagnostics, Advanced Diagnostics, Service Aids, etc. screen.

```
FUNCTION SELECTION                                                         801002


Move cursor to selection, then press Enter.

  Diagnostic Routines
    This selection will test the machine hardware. Wrap plugs and
    other advanced functions will not be used.
  Advanced Diagnostics Routines
    This selection will test the machine hardware. Wrap plugs and
    other advanced functions will be used.
  Task Selection (Diagnostics, Advanced Diagnostics, Service Aids, etc.)
    This selection will list the tasks supported by these procedures.
    Once a task is selected, a resource menu may be presented showing
    all resources supported by the task.
  Resource Selection
    This selection will list the resources in the system that are supported
    by these procedures. Once a resource is selected, a task menu will
    be presented showing all tasks that can be run on the resource(s).




F1=Help              F10=Exit             F3=Previous Menu
```

Figure 9-9 Diagnostic Function Selection screen

3. Once in the Task Selection List menu, scroll down until and select the Hot Plug Task menu option. Figure 9-10 shows the Task Selection List with the Hot Plug Task menu option highlighted.

```
TASKS SELECTION LIST                                                    801004

From the list below, select a task by moving the cursor to
the task and pressing 'Enter'.
To list the resources for the task highlighted, press 'List'.

[MORE...24]
  Format Media
  Gather System Information
  Hot Plug Task
  Identify and Attention Indicators
  Local Area Network Analyzer
  Log Repair Action
  Microcode Tasks
  RAID Array Manager
  SSA Service Aids
    This selection provides tools for diagnosing and resolving
    problems on SSA attached devices.
  Update Disk Based Diagnostics
[BOTTOM]

F1=Help              F4=List              F10=Exit              Enter
F3=Previous Menu
```

Figure 9-10 The Task Selection List menu option

When using the diag menu, you will be given the choice of choosing the options available for your hardware environment. In this case, we have three options listed

a. The PCI Hot Plug Manager menu

b. The RAID Hot Plug Devices menu

c. The SCSI and SCSI RAID Hot Plug Manager menu

Choose the PCI Hot Plug Manager menu option. Figure 9-11 shows the HPT menu with PCI Hot Plug Manager highlighted.

```
Hot Plug Task                                                              801004

Move cursor to desired item and press Enter.

  PCI Hot Plug Manager
  RAID Hot Plug Devices
  SCSI and SCSI RAID Hot Plug Manager











F1=Help             F4=List             F10=Exit            Enter
F3=Previous Menu
```

Figure 9-11 The Hot Plug Task menu

4. From the PCI Hot Plug Manager menu, select the Unconfigure A device. Select the appropriate adapter to unconfigure. In this scenario, the adapter ent2 is to be unconfigured. Figure 9-12 shows the Unconfigure A device menu with the ent2 device highlighted. For this scenario, we can keep the device definition in the database. Once the ent2 device has been placed into a *defined* state, return to the PCI Hot Plug Manager menu.

```
                          Unconfigure a Device

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                    [Entry Fields]
* Device Name                                    [ent2]                +
  Unconfigure any Child Devices                   no                   +
  KEEP definition in database                     yes                  +










F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 9-12 Unconfigure A device

5. From the PCI Hot Plug Manager menu, select the Replace/Remove a PCI Hot Plug Adapter. Select the appropriate adapter to replace/remove. In this scenario, the adapter ent2 is to be replaced. Figure 9-13 on page 423 shows the PCI Hot Plug Manager menu with the Replace/Remove option highlighted.

```
                              PCI Hot Plug Manager

Move cursor to desired item and press Enter.

  List PCI Hot Plug Slots
  Add a PCI Hot Plug Adapter
  Replace/Remove a PCI Hot Plug Adapter
  Identify a PCI Hot Plug Slot
  Unconfigure a Device
  Configure a Defined Device
  Install/Configure Devices Added After IPL

  +--------------------------------------------------------------------------+
  |                  Replace/Remove a PCI Hot Plug Adapter                   |
  |                                                                          |
  | Move cursor to desired item and press Enter.                             |
  |                                                                          |
  |   # Slot                     Description                        Device(s)|
  |   U7879.001.DQD185T-P1-C3  PCI-X capable, 64 bit, 133MHz slot  ent2      |
  |                                                                          |
  | F1=Help                  F2=Refresh                F3=Cancel             |
  | F8=Image                 F10=Exit                  Enter=Do              |
F1| /=Find                   n=Find Next                                     |
F9+--------------------------------------------------------------------------+
```

Figure 9-13 PCI Hot Plug Manager Replace/Remove menu

6. Once the ent2 adapter is selected, the Replace/Remove a PCI Hot Plug Adapter menu will be displayed. The adapter can be either replaced or removed. In this scenario, we will replace the adapter. Select replace by using the using the Tab or F4 key.

   Once the replace option has been selected, the PCI slot will be put into a state that allows the PCI adapter to be removed. A blinking attention light will identify the slot that contains the adapter that has been selected for replacement.

   To begin the replacement procedure, press the Enter key.

Figure 9-14 shows the highlighted replace option.

```
                    Replace/Remove a PCI Hot Plug Adapter

Type or select a value for the entry field.
Press Enter AFTER making all desired changes.

                                                     [Entry Fields]
  Operation                                         replace              +












F1=Help              F2=Refresh           F3=Cancel            F4=List
F5=Reset             F6=Command           F7=Edit              F8=Image
F9=Shell             F10=Exit             Enter=Do
```

Figure 9-14 Replace/Remove a PCI Hot Plug Adapter menu

7. Figure 9-15 on page 425 shows the completed adapter replacement procedure. During this phase of the replacement, you will be asked to:

   a. Verify that the adapter should be replaced by pressing the Enter key. The visual indicator will be set to the identify state. You may exit by pressing the x key followed by the Enter key.

   b. Exchange the PCI adapter. The visual indicator will be set to the action state. To continue, press the Enter key. To exit, press the x key followed by the Enter key. The exchange process requires the PCI Blind Swap cassette to be removed from the PCI slot, the PCI Ethernet adapter to be exchanged with a replacement PCI ethernet adapter of the same FRU, and then assembled into the PCI Blind Swap Canister.

      Note: If you choose to exit at this stage, the PCI slot will be left in the removed state.

   c. Return the PCI Blind Swap cassette to the PCI slot. To continue, press the Enter key. The adapter has now been replaced and is ready to be reconfigured.

   Figure 9-15 on page 425 shows the three steps and the successful completion of the replacement task.

```
                              COMMAND STATUS

Command: OK            stdout: yes           stderr: no

Before command completion, additional instructions may appear below.


The visual indicator for the specified PCI slot has
been set to the identify state. Press Enter to continue
or enter x to exit.


The visual indicator for the specified PCI slot has
been set to the action state. Replace the PCI card
in the identified slot and press Enter to continue.
Enter x to exit. Exiting now leaves the PCI slot
in the removed state.


Replace Operation Complete.





F1=Help             F2=Refresh          F3=Cancel           F6=Command
F8=Image            F9=Shell            F10=Exit            /=Find
n=Find Next
```

Figure 9-15 PCI adapter replacement

> **Tip:** If the Ethernet adapter interface was not unconfigured (see step 4 previous), then the procedure will fail at this point. Unconfigure the Ethernet adapter interface and begin the procedure from step 5.

8. Once the PCI Ethernet adapter has been replaced, the device must be configured for AIX 5L. From the PCI Hot Plug Manager menu, choose Configure a Defined Device, select the ent2 Ethernet device, and press the Enter key. The ent2 device adapter will now be configured.

Figure 9-16 shows the Configure a Defined Device menu.

```
                         Configure a Defined Device

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                     [Entry Fields]
* Device Name                                     [ent2]                +









F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 9-16 Configure A Defined Device

9. The ent2 device is now ready for use, and may be reconfigured with an IP address. A repair action should be logged in the operating system error log against the ent2 device. The status is shown in Figure 9-17 on page 427.

```
                           PCI Hot Plug Manager

Move cursor to desired item and press Enter.

  List PCI Hot Plug Slots
  Add a PCI Hot Plug Adapter
  Replace/Remove a PCI Hot Plug Adapter
  Identify a PCI Hot Plug Slot
  Unconfigure a Device
  Configure a Defined Device
  Install/Configure Devices Added After IPL


   Replace/Remove a PCI Hot Plug Adapter

   Move cursor to desired item and press Enter.

     # Slot                  Description                          Device(s)
     U7879.001.DQD185T-P1-C3  PCI-X capable, 64 bit, 133MHz slot  ent2

   F1=Help                 F2=Refresh               F3=Cancel
   F8=Image                F10=Exit                 Enter=Do
F1 /=Find                  n=Find Next
F9
```

Figure 9-17 PCI Hot Plug Manager Replace/Remove menu

10. Once the ent2 adapter is selected, the Replace/Remove a PCI Hot Plug Adapter menu will be displayed. The adapter can be either replaced or removed. In this scenario, we will replace the adapter. Select replace by using the using the Tab or F4 key.

    Once the replace option has been selected, the PCI slot will be put into a state that allows the PCI adapter to be removed. A blinking attention light will identify the slot that contains the adapter that has been selected for replacement.

    To begin the replacement procedure, press the Enter key.

Figure 9-18 shows the highlighted replace option.

```
                    Replace/Remove a PCI Hot Plug Adapter

Type or select a value for the entry field.
Press Enter AFTER making all desired changes.

                                                      [Entry Fields]
  Operation                                          replace              +













F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 9-18 Replace/Remove a PCI Hot Plug Adapter menu

11. Figure 9-19 on page 429 shows the completed adapter replacement procedure. During this phase of the replacement, you will be asked to:

    a. Verify that the adapter should be replaced by pressing the Enter key. The visual indicator will be set to the identify state. You may exit by pressing the x key followed by the Enter key.

    b. Exchange the PCI adapter. The visual indicator will be set to the action state. To continue, press the Enter key. To exit, press the x key followed by the Enter key. The exchange process requires the PCI Blind Swap cassette to be removed from the PCI slot, the PCI Ethernet adapter to be exchanged with a replacement PCI Ethernet adapter of the same FRU, and then assembled into the PCI Blind Swap Canister.

    > **Note:** If you choose to exit at this stage, the PCI slot will be left in the removed state.

    c. Return the PCI Blind Swap Canister to the PCI slot. To continue, press the Enter key. The adapter has now been replaced and is ready to be reconfigured.

Figure 9-19 shows the three steps and the successful completion of the replacement task.

```
                              COMMAND STATUS

Command: OK             stdout: yes           stderr: no

Before command completion, additional instructions may appear below.


The visual indicator for the specified PCI slot has
been set to the identify state. Press Enter to continue
or enter x to exit.


The visual indicator for the specified PCI slot has
been set to the action state. Replace the PCI card
in the identified slot and press Enter to continue.
Enter x to exit. Exiting now leaves the PCI slot
in the removed state.


Replace Operation Complete.





F1=Help            F2=Refresh         F3=Cancel          F6=Command
F8=Image           F9=Shell           F10=Exit           /=Find
n=Find Next
```

Figure 9-19 PCI adapter replacement

Tip: If the Ethernet adapter interface was not unconfigured (see step 4 previous), then the procedure will fail at this point. Unconfigure the Ethernet adapter interface and begin the procedure from step 5.

12. Once the PCI Ethernet adapter has been replaced, the device must be configure to AIX 5L. From the PCI Hot Plug Manager menu, choose Configure a Defined Device, select the ent2 Ethernet device, and press the Enter key. The ent2 device adapter will now be configured.

Figure 9-20 shows the Configure a Defined Device menu.

```
                         Configure a Defined Device

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                      [Entry Fields]
* Device Name                                      [ent2]               +










F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 9-20 Configure A Defined Device

13. The ent2 device is now ready for use, and may be reconfigured with an IP address via the **smitty chinet** fast path. A repair action should be logged in the AIX 5L error report against the `ent2` device. This will assist other system administrators that may use this server by showing that the error logged in the error report has been resolved. To enter a repair action use the diag → Task Selection → Log Repair Action and choose the ent2 device.

9.8 Failed disk replacement

Most of the problems a system administrator encounters are related to disk drives. As a consequence, you must be aware of some of the procedures that can be used for disk replacement. A disk can be replaced for one of the several reasons:

- The disk has failed.
- The disk started to report IO errors and you want to replace it to prevent a complete failure.
- The disk does not satisfy/meet your requirements for, for example, size or speed.

We try to cover the most common scenarios of failing disks in the following sections.

9.8.1 Scenario 1

If the disk you are going to replace is mirrored, we recommend following these steps:

1. Remove copies of all logical volumes that were residing on that disk using either the **rmlvcopy** command or **unmirrorvg** command.
2. Remove the disk from the volume group using the **reducevg** command.
3. Remove the disk definition using the **rmdev** command.
4. Physically remove the disk. If the disk is not hot-swappable, you may be required to reboot the system.
5. Make the replacement disk available. If the disk is hot-swappable, you can run **cfgmgr**; otherwise, you may need to reboot the system.
6. Include the newly added disk into the volume group using the **extendvg** command.
7. Recreate and synchronize the copies for all logical volumes using either **mklvcopy** or **mirrorvg**.

9.8.2 Scenario 2

If the disk you are going to replace is not mirrored and is still functional, we recommend following these steps:

1. Make the replacement disk available. If the disk is hot-swappable, you can run **cfgmgr**; otherwise, you may need to reboot the system.
2. Include the newly added disk into the volume group using the **extendvg** command.
3. Migrate all partitions from the failing disk to the new disk using either the **migratepv** command or the **migratelp** command. If the disks are part of the rootvg, you should consider the following:
   - If the disk to be replaced contains a copy of the BLV, you have to clear it using the **chpv -c** command.
   - A new BLV image must be created on the new disk using the **bosboot** command.
   - The bootlist must be updated to reflect these changes using the **bootlist** command.

– If the disk to be replaced contains a paging space or a primary dump device, you should disable them. After the **migratepv** command completes, you should reactivate them.

4. Remove the failing disk from the volume group using the **reducevg** command.
5. Remove the disk definition using the **rmdev** command.

9.8.3 Scenario 3

If the disk is not mirrored, has failed completely, and there are other disks available in the volume group, we recommend following these steps:

1. Identify all logical volumes that have at least one partition located on the failed disk.
2. Close the logical volumes and unmount all the corresponding file systems using the **umount** command.
3. Remove the file systems and logical volumes using the **rmfs** command.
4. Remove the failing disk from the volume group using the **reducevg** command.
5. Remove the disk definition using the **rmdev** command.
6. Physically remove the disk. If the disk is not hot-swappable, you may be required to reboot the system.
7. Make the replacement disk available. If the disk is hot-swappable, you can run **cfgmgr**; otherwise, you may need to reboot the system.
8. Include the newly added disk into the volume group using the **extendvg** command.
9. Recreate all the logical volumes and the corresponding file systems using the **mklv** command and the **crfs** command.
10. If you have a backup of your data, restore your data from backup.

9.8.4 Scenario 4

If the disk is not mirrored, has failed completely, there are no other disks available in the volume group (the volume group contained only one physical volume or all the physical volumes failed simultaneously), and the volume group is not rootvg, we recommend the following steps:

1. Export the volume group definition from the system using the **exportvg** command.
2. Ensure that /etc/filesystems does not contain any incorrect stanzas.
3. Remove the disk definition using the **rmdev** command.

4. Physically remove the disk. If the disk is not hot-swappable, you may be required to reboot the system.
5. Make the replacement disk available. If the disk is hot-swappable, you can run **cfgmgr**; otherwise, you may need to reboot the system.
6. If you have a volume group backup, restore it using the **restvg** command.
7. If you do not have volume group backup, recreate the volume group, all the logical volumes, and the corresponding file systems using the **mkvg** command, the **mklv** command, and the **crfs** command.
8. If you have a backup of your data, restore your data from backup.

9.8.5 Scenario 5

If the disk is not mirrored, has failed completely, there are not other disks available in the volume group (the volume group contained only one physical volume or all physical volumes failed simultaneously), and the volume group is rootvg, we recommend following these steps:

1. Replace the failing disk.
2. Boot the system in maintenance mode.
3. Restore the system from an mksysb image.

9.9 Access rootvg in maintenance mode

The maintenance mode of AIX 5L is designed to assist you when your system is unable to boot. For example, you use the following procedure to reset the root password, in case you forget it.

The procedure enables you to get a system prompt so that you may attempt to recover data from the system or perform corrective actions that will enable the system to boot from the hard disk. You can use the available log files to analyze the probable cause of the possible failures.

To execute the following procedure, you need to understand how to boot from an installation media. 3.3, “Base Operating System installation” on page 31 gives an example. Follow the steps as described in 3.3.2, “Sample AIX 5L installation procedure” on page 32, until step 13, when the main menu of the BOS installation and maintenance is shown.

From step 13, you have to choose the following menu choices:

1. On the main menu, choose option 3 (Start Maintenance Mode for System Recovery) and press Enter. Figure 9-21 shows the maintenance menu.

```
                    Maintenance

 Type the number of your choice and press Enter.

 >>> 1 Access a Root Volume Group
     2 Copy a System Dump to Removable Media
     3 Access Advanced Maintenance Functions
     4 Erase Disks
     5 Install from a System Backup










     88  Help ?
     99  Previous Menu

 >>> Choice [1]:
```

Figure 9-21 Maintenance menu

2. Select option 1 (Access a Root Volume Group) and press Enter. A screen similar to the one in Figure 9-22 on page 435 is shown.

```
                  Warning:

    If you choose to access a root volume group, you will not be able to return
    to the Base Operating System Installation menus without rebooting.


    Type the number of your choice and press Enter.

        0 Continue










        88  Help ?
    >>> 99  Previous Menu

    >>> Choice [99]:
```
Figure 9-22   Warning screen

3. Take note of the warning. If you want to return to the previous menu, enter 99. Otherwise, enter 0 to confirm. A screen similar to Figure 9-23 is shown.

```
                          Access a Root Volume Group

  Type the number for a volume group to display the logical volume information
  and press Enter.

     1)   Volume Group 00c478de00004c0000000107b393746c contains these disks:
            hdisk0   70006 05-08-00-3,0









     Choice:
```

Figure 9-23 List of found volume group(s)

4. Select the volume group whose logical volume information you want to display. For your information, rootvg has hd5, the BLV. Enter the number of the volume group and press Enter. A screen similar to Figure 9-24 on page 437 is shown.

```
                                Volume Group Information

      ------------------------------------------------------------------------------
         Volume Group ID 00c478de00004c0000000107a1137f36 includes the following
         logical volumes:

              hd5         hd6         hd8         hd4         hd2      hd9var
              hd3         hd1     hd10opt      fslv00      fslv01
      ------------------------------------------------------------------------------






     Type the number of your choice and press Enter.

        1) Access this Volume Group and start a shell
        2) Access this Volume Group and start a shell before mounting filesystems

       99) Previous Menu

         Choice [99]:
```

Figure 9-24 List of logical volumes found on the selected volume group

5. Select one of the options from the Volume Group Information screen and press Enter. Each option does the following:

   Choice 1 Selecting this choice imports and activates the volume group and mounts the file systems for this root volume group before providing you with a shell and a system prompt.

   Choice 2 Selecting this choice imports and activates the volume group and provides you with a shell and system prompt before mounting the file systems for this root volume group.

   Choice 99 Entering 00 returns you to the Access a Root Volume Group menu.

After you choose either option 1 or 2, a shell and system prompt are displayed, as shown in Figure 9-25.

```
pmap 0 and inoext 0xa08 non-zero
pmap 0 and inoext 0x2c non-zero
logredo start at: 1132586042 sec and end at 1132586045 sec
Primary superblock is valid.
Checking the /usr filesystem.



The current volume is: /dev/hd2
Primary superblock is valid.
Saving special files and device configuration information.
Checking and mounting the /tmp filesystem.



The current volume is: /dev/hd3
Primary superblock is valid.
Checking and mounting the /var filesystem.



The current volume is: /dev/hd9var
Primary superblock is valid.
Filesystems mounted for maintenance work.
#
```

Figure 9-25 System maintenance prompt

6. Take the appropriate measures to recover data or take action to enable the system to boot normally.

   For example:

   – Execute the **fsck** command on file systems.

   – Edit /etc/security/password file for root password reset.

   – Execute the **bosboot** command to recreate BLV.

7. To exit the maintenance mode, issue the **shutdown -Fr** command.

> **Attention:** If you altered any data on disk, be sure to synchronize the data onto the disk. Issue the **sync;sync** command before rebooting.

9.10 Troubleshoot graphical problems

This section covers problem resolution related to display output.

9.10.1 System hangs when trying to start desktop

The following sections help you understand what prevents a graphical desktop from starting.

Full /home file system

Users will not be able to log in using the AIX 5L CDE when the /home file system is full. If /home is full, the AIX 5L CDE welcome window will accept the users's name and password, the display will then go blank and appear to hang, and then will return to the AIX 5L CDE welcome window. To log in and investigate, press the Options button on the AIX 5L CDE welcome window and select Command Line Login, or log in from a non-graphical display.

Name resolution problems

If name resolution configuration is wrong or your system cannot reach your DNS servers, users will experience log in and network problems. If you have problems with name resolution, the AIX 5L CDE welcome window will accept the user's name and password, the display then shows the message `Starting the Common Desktop Environment`, and it will stay there for several minutes, and then it will start a basic graphical environment.

Use the **nslookup** command to verify if your system can resolve names through a DNS server. For example, resolve the www.ibm.com name using the **nslookup** command:

```
# nslookup
Default Server:  dhcp001.itsc.austin.ibm.com
Address:  9.3.4.2

> www.ibm.com
Server:  dhcp001.itsc.austin.ibm.com
Address:  9.3.4.2

Non-authoritative answer:
Name:    livesite.events.ibm.com
Addresses:  129.42.21.99, 129.42.16.99, 129.42.17.99, 129.42.18.99
          129.42.19.99, 129.42.20.99
Aliases:  www.ibm.com

> exit
#
```

If the **nslookup** command does not show a similar output when trying to resolve a name or delays showing the greater-than prompt (>), it means that there is a problem trying to reach the DNS servers:

- Verify if your system has access to the network.
- Verify if your DNS servers are up and running and that you can access them for name resolution queries.
- Stop using a DNS server. Use the SMIT fast path:

  ```
  # smitty spnamerslv
  ```

  You have to provide a name for renaming the /etc/resolv.conf. You can also use the following command:

  ```
  # namerslv -e
  ```

Attention: When you stop using a DNS server, it affects all programs and services that use the name resolution service. Do not do this action unless you are aware of the impact of it on your system.

9.10.2 Troubleshoot error unable to open display

When using commands that use the graphical services, you may find that they send a message similar to this:

```
Error: Can't open display:
```

The program is trying to display output to a graphical environment and is not able to do it. Before a program can use a display, it must establish a connection to the X server driving the display.

Opening a display

To open a connection to the X server controlling a specified display, you have to set the display name or DISPLAY environment variable, which is a string with the following format:

```
HostName:Number.Screen
```

where:

HostName Specifies the name or the IP address of the host system where the display is physically attached. The host name should be followed by a : (colon).

Number Specifies the ID number of the display server on that host machine. The display number can be followed by a . (period).

Screen Specifies the number of the screen on that host server. Multiple screens can be connected to or controlled by a single X server.

For example, you can use the following command to set the DISPLAY environment variable to the values of screen 0, display 2 of the system named server3:

```
# export DISPLAY=server3:2.0
```

If you still get the `Can't open display` message after setting the DISPLAY environment variable and the X server is running, you must check if your system has permissions to access the X server on the machine on which it is running.

You use the **xhost** command to control who access the X server running on the current host machine. This command adds or deletes host names in the list of machines from which the X server accepts connections.

For example, run the command **xhost** on the system named server3 that is running the X server to grant access to your system (named server2) to connect to the X server:

```
# xhost +server2
```

To deny access to the X server to the system named server2, type the command:

```
# xhost -server2
```

Note: Ensure that you specify a host name with the **xhost +** command, because it disables access control for the X server. This allows you to grant access to specific hosts, which eases monitoring for potential attacks to the X server. If you do not specify a host name, access will be granted to all hosts which is a potential security risk.

9.10.3 Troubleshoot TTY display problems

You can find problems with TTY devices like, for example, when trying to clear the screen with the **clear** command and the screen remains the same, or when you run the **smit** command and you get an scrambled output like this:

```
# smitty
B@B5,Alqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqk6,x6cx7,mqqqqqqqq
q
qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj5,B6.Initializing SMIT ...5,
                                                  6,                        6c
7,

5,Alqqqqqqqqqqqqqqqqqqqq
qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqk6,x6cx7,mqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq
qqqqqqqqqqqqqqqqqqqqj5,B6.Processing data ...5, ?System Management" Move cursor
to desired item and press Enter.$"!aSoftware Installation and Maintenance%"
@BSoftware License Management&"Devices'"System Storage Management (Physical &
Logical Storage)("Security & Users)"Communications Applications and Services
*"Print Spooling+"Advanced Accounting,"Problem Determination-"Performance &
Resource Scheduling."System Environments/"Processes & Subsystems0
"Applications1"Installation Assistant2"Cluster Systems Management3"Using SMIT
(information only)5,
F9=Shell7,       F10=Exit            Enter=Do                    $!
```

It means that the TERM environment variable is not set to the correct value.

TERM values for different displays and terminals

Information about terminal capabilities is stored in the terminfo database. The value of the TERM environment variable identifies the specific terminal description in the terminfo database. This provides all information that a program needs for communicating effectively with the current TTY device. Table 9-12 shows some values for various terminals.

Table 9-12 Values for various terminals

Display/Terminal	Value
3161 ASCII Terminal	ibm3161
3163 ASCII Terminal	ibm3161
DEC VT100 (terminal)	vt100
DECVT220	vt220
3151 ASCII Terminal	ibm3151
AIXwindows	aixterm

For example, to set the value of the TERM environment variable to vt100, type the following command:

```
# export TERM=vt100
```

9.11 The perfpmr command

The **perfpmr** command consists of a set of utilities that build a test case containing the necessary information to assist in analyzing performance issues. It is primarily designed to assist IBM software support, but is also useful as a documentation tool for your system.

As the **perfpmr** command is updated frequently, it is not distributed on AIX 5L media. It can be downloaded from

ftp://ftp.software.ibm.com/aix/tools/perftools/perfpmr

Use the version that is appropriate for your AIX 5L level. For our case, the file that we need is distributed in:

ftp://ftp.software.ibm.com/aix/tools/perftools/perfpmr/perf53/perf53.tar.Z

9.11.1 perfpmr

The syntax of the **perfpmr** command is:

`perfpmr.sh [-PDgfnpsc][-F` *file*`][-x` *file*`][-d` *sec*`] monitor_seconds`

Table 9-13 gives a description of the parameters.

Table 9-13 Commonly used flags for the perfpmr.sh command

Flag	Description
-P	Preview only - show scripts to run and disk space needed.
-D	Run **perfpmr** the original way without a perfpmr.cfg file.
-g	Do not collect the **gennames** output.
-f	If **gennames** is run, specify **gennames -f**.
-n	Used if no **netstat** or **nfsstat** output is desired.
-p	Used if no **pprof** collection is desired while **monitor.sh** is running.
-s	Used if no **svmon** output is desired.
-c	Used if no configuration information is desired.
-F	File use file as the **perfpmr** cfg file; the default is perfpmr.cfg.

Flag	Description
-x	File only execute file found in the perfpmr installation directory.
-d	sec is time to wait before starting collection period; the default is delay_seconds 0.
-s	Used if **svmon** output is not required.

Parameters

The following parameter is of special interest.

monitor_seconds Collection period in seconds. The minimum period is 60 seconds.

Use the **perfpmr.sh 600** command for a standard collection period of 600 seconds.

9.11.2 Measurement and sampling

Unless you run the shell scripts separately, the **perfpmr.sh 600** command executes the following shell scripts to obtain a test case. You can also run these scripts on their own. Refer to “Running perfpmr” on page 450 for details.

config.sh Collects configuration information into a report called config.sum.

emstat.sh time Builds a report called emstat.int on emulated instructions. The time parameter must be greater than or equal to 60.

filemon.sh time Builds a report called filemon.sum on file I/O. The time parameter does not have any restrictions.

iostat.sh time Builds two reports on I/O statistics: a summary report called iostat.sum and an interval report called iostat.int. The time parameter must be greater than or equal to 60.

iptrace.sh time Builds a raw Internet Protocol (IP) trace report on network I/O called iptrace.raw. You can convert the iptrace.raw file to a readable **ipreport** file called iptrace.int using the **iptrace.sh -r** command. The time parameter does not have any restrictions.

monitor.sh time Invokes system performance monitors and collects interval and summary reports:

- **lsps.after** Contains **lsps -a** and **lsps -s** output after **monitor.sh** was run. Used to report on paging space use.

lsps.before Contains **lsps -a** and **lsps -s** output before **monitor.sh** was run. Used to report on paging space use.

nfsstat.int Contains **nfsstat -m** and **nfsstat -csnr** output before and after **monitor.sh** was run. Used to report on Network File System use and configuration.

monitor.int Contains samples by interval using **ps -efk** (showing active processes before and after **monitor.sh** was run). It also contains **sadc**, **sar -A**, **iostat**, **vmstat**, and **emstat** output.

monitor.sum Contains samples by summary using **ps -efk** (showing changes in **ps** output for active processes before and after **monitor.sh** was run). It also contains **sadc**, **sar -A**, **iostat**, **vmstat**, and **emstat** outputs.

pprof.trace.raw
Contains the raw trace for **pprof**.

psb.elfk Contains a modified **ps -elk** output before monitor.sh was run.

svmon.after Contains **svmon -G** and **svmon -Pns** output and top segments use by process with the **svmon -S** command after **monitor.sh** was run. Used to report on memory use.

svmon.before Contains **svmon -G** and **svmon -Pns** output and top segment use by process with the **svmon -S** command before **monitor.sh** was run. Used to report on memory use.

vmstati.after Contains **vmstat -i** output after **monitor.sh** was run. Used to report on I/O device interrupts.

vmstati.before Contains **vmstat -i** output before **monitor.sh** was run. Used to report on I/O device interrupts.

netstat.sh [-r] time

Builds a report on network configuration and use called netstat.int containing **tokstat -d** of the token-ring interfaces, **entstat -d** of the Ethernet interfaces, **netstat -in**, **netstat -m**, **netstat -rn**, **netstat -rs**, **netstat -s**, **netstat -D**, and **netstat -an** before and after **monitor.sh** was run. You can reset the Ethernet and token-ring statistics and re-run this report by running **netstat.sh -r 60**. The time parameter must be greater than or equal to 60.

nfsstat.sh time

Builds a report on NFS configuration and use called netstat.int containing **nfsstat -m**, and **nfsstat -csnr** before and after **nfsstat.sh** was run. The time parameter must be greater than or equal to 60.

pprof.sh time

Builds a file called pprof.trace.raw that can be formatted with the **pprof.sh -r** command.The time parameter does not have any restrictions.

ps.sh time

Builds reports on process status (ps). **ps.sh** creates the following files:

psa.elfk A **ps -elfk** listing after **ps.sh** was run.

psb.elfk A **ps -elfk** listing before **ps.sh** was run.

ps.int Active processes before and after ps.sh was run.

ps.sum A summary report of the changes between when **ps.sh** started and finished. This is useful for determining what processes are consuming resources.

The time parameter must be greater than or equal to 60.

sar.sh time

Builds reports on **sar**. **sar.sh** creates the following files:

sar.int Output of commands **sadc 10 7** and **sar -A**.

sar.sum A **sar** summary over the period **sar.sh** was run.

The time parameter must be greater than or equal to 60.

tcpdump.sh int.time

The int. parameter is the name of the interface; for example, tr0 is token-ring. Creates a raw **trace** file of a TCP/IP dump called tcpdump.raw. To produce a readable tcpdump.int file, use the **tcpdump.sh -r** command. The time parameter does not have any restrictions.

tprof.sh time Creates a **tprof** summary report called tprof.sum. Used for analyzing memory use of processes and threads. You can also specify a program to profile by specifying the **tprof.sh -p program 60** command, which enables you to profile the executable-called program for 60 seconds. The time parameter does not have any restrictions.

trace.sh time Creates the raw trace files (trace*) from which an ASCII trace report can be generated using the **trcrpt** command or by running **trace.sh -r**. This command creates a file called trace.int that contains the readable trace. Used for analyzing performance problems. The time parameter does not have any restrictions.

vmstat.sh time Builds reports on **vmstat**: a **vmstat** interval report called vmstat.int and a **vmstat** summary report called vmstat.sum. The time parameter must be greater than or equal to 60.

Due to the volume of data the **trace** command collects, the **trace** command will only run for five seconds (by default), so it is possible that it will not be running when the performance problems occur on your system, especially if performance problems occur for short periods. In this case, it would be advisable to run the **trace** command by itself for a period of 15 seconds when the problem is present. The **trace.sh 15** command runs a trace for 15 seconds.

An system can produce a test case of 135 MB, with 100 MB just for the traces. This size can vary considerably depending on system load. If you run the trace on the same system with the same workload for 15 seconds, then you could expect the trace files to be approximately 300 MB in size.

One raw trace file per CPU is produced. The files are called trace.raw-0, trace.raw-1, and so forth for each CPU. An additional raw trace file called trace.raw is also generated. This is a master file that has information that ties in the other CPU-specific traces. To merge the trace files together to form one raw trace file, run the following commands:

```
# trcrpt -C all -r trace.raw > trace.r
# rm trace.raw*
```

9.11.3 Building and submitting a test case

You may be asked by IBM to supply a test case for a performance problem or you may wish to run the **perfpmr.sh** command for your own requirements (for example, to produce a base line for detecting future performance problems). In either case, **perfpmr.sh** is the tool to collect performance data. Even if your performance problem is attributed to one component of your system, such as the network, **perfpmr.sh** is still the way to send a test case because it contains other information that is required for problem determination. Additional information for problem determination may be requested by IBM software support.

> **Note:** IBM releases Maintenance Levels for AIX. These are a collection of Program Temporary Fixes (PTFs) used to upgrade the operating system to the latest level, but remaining within your current release. Often these, along with the current version of micro-code for the disks and adapters, have performance enhancement fixes. You may therefore wish to load these.

There are five stages to building and sending a test case. These steps must be completed when you are logged in as root. The steps are listed as follows:

- Prepare to download **perfpmr**.
- Download **perfpmr**.
- Install **perfpmr**.
- Run **perfpmr**.
- Upload the test case.

Preparing for perfpmr

These filesets should be installed before running **perfpmr.sh**:

- bos.acct
- bos.sysmgt.trace
- perfagent.tools
- bos.net.tcp.server
- bos.adt.include
- bos.adt.samples

Downloading perfpmr

The **perfpmr** is downloadable from:

`ftp://ftp.software.ibm.com/aix/tools/perftools/perfpmr`

Using a browser, download the version that is applicable to your version of AIX 5L. The file size should be under 1 MB.

> **Important:** Always download a new copy of **perfpmr** in case of changes. Do not use an existing pre-downloaded copy.

If you have downloaded **perfpmr** to a PC, transfer it to the system in binary mode using **ftp**, placing it in an empty directory.

Installing perfpmr

Uncompress and extract the file with the **tar** command. The directory contains:

- Install
- PROBLEM.INFO
- README
- config.sh
- emstat.sh
- filemon.sh
- getdate
- getevars
- iostat.sh
- iptrace.sh
- lsc
- memfill
- monitor.sh
- netstat.sh
- nfsstat.sh
- perfpmr.cfg
- perfpmr.sh
- pprof.sh
- ps.sh
- pstat.sh
- sar.sh
- setpri
- setsched
- svmon
- tcpdump.sh
- tprof.sh
- trace.sh
- vmstat.sh

In the directory you will notice files ending in .sh. These are shell scripts that may be run separately. Normally, these shell scripts are run automatically by running **perfpmr.sh**. Read the README file to find any additional steps that may be applicable to your system.

Install **perfpmr** by running **./Install**. This will replace the following files in the /usr/bin directory with symbolic links to the files in the directory where you installed the **perfpmr** command:

- config.sh
- curt
- emstat.sh
- filemon.sh
- getevars

- hd_pbuf_cnt.sh
- iostat.sh
- iptrace.sh
- lsc
- monitor.sh
- netstat.sh
- nfsstat.sh
- perfpmr.sh
- pprof.sh
- ps.sh
- sar.sh
- setpri
- tcpdump.sh
- tprof.sh
- trace.sh
- utld
- vmstat.sh

The output of the installation procedure will be similar to Example 9-6.

Example 9-6 perfpmr installation screen

```
# ./Install

(C) COPYRIGHT International Business Machines Corp., 2000


    PERFPMR Installation started...



    PERFPMR Installation completed.
```

Running perfpmr

There are two scenarios to consider when running the **perfpmr** command.

- If your system is performing poorly for long periods of time and you can predict when it runs slow, then you can run **./perfpmr.sh 600**.
- In some situations, a system may perform normally but will run slow at various times of the day. If you run **perfpmr.sh 600**, then there is a chance that **perfpmr** might not have captured the performance slowdown. In this case, you could run the scripts manually when the system is slow and use a longer time-out period, for example, a **trace.sh 15** command will perform a trace for 15 seconds instead of the default five seconds. We would still need a **perfpmr.sh 600** to be initially run before running individual scripts. This will ensure that all of the data and configuration have been captured.

Attention: If you are using HACMP, then you may want to extend the Dead Man Switch (DMS) time-out or shut down HACMP prior to collecting **perfpmr** data to avoid accidental failovers.

After executing **perfpmr.sh**, it creates the files in Table 9-14.

Table 9-14 Overview of the files created by perfpmr.sh

config.sum	crontab_l	devtree.out
errpt_a	etc_security_limits	filemon.sum
genkex.out	genkld.out	gennames.out
getevars.out	iptrace.raw	lsps.after
lsps.before	lsrset.out	monitor.int
monitor.sum	netstat.int	nfsstat.int
perfpmr.int	pprof.trace.raw	psa.elfk
psb.elfk	psemo.after	psemo.before
svmon.after	svmon.before	tcpdump.raw
tprof.csyms	tprof.ctrc	tprof.out
tprof.sum	trace.crash.inode	trace.fmt
trace.inode	trace.j2.inode	trace.maj_min2lv
trace.nm	trace.raw	trace.raw-0
trace.raw-1	trace.raw-10	trace.raw-11
trace.raw-12	trace.raw-13	trace.raw-14
trace.raw-15	trace.raw-2	trace.raw-3
trace.raw-4	trace.raw-5	trace.raw-6
trace.raw-7	trace.raw-8	trace.raw-9
trace.syms	tunables_lastboot	tunables_lastboot.log
tunables_nextboot	vfs.kdb	vmstat_v.after
vmstat_v.before	vmstati.after	vmstati.before
vnode.kdb	w.int	

Tip: After you have installed the **perfpmr** command, you can run it at any time to make sure that all of the files described above are captured. By doing this, you can be confident that you will get a full test case.

Uploading the test case

The directory also contains a file called PROBLEM.INFO that must be completed. Bundle the files together using the **tar** command and upload the file to IBM as documented in the README files.

9.11.4 Examples for perfpmr

Example 9-7 is an example of running **perfpmr.sh 600**.

Example 9-7 Running perfpmr.sh

```
# perfpmr.sh 600

(C) COPYRIGHT International Business Machines Corp., 2000,2001,2002,2003

16:54:32-11/29/05 :     perfpmr.sh begin
    PERFPMR: hostname: server2
    PERFPMR: perfpmr.sh Version 530 2005/10/19
    PERFPMR: current directory: /home/gpsilva
    PERFPMR: perfpmr tool directory: /home/gpsilva
    PERFPMR: Parameters passed to perfpmr.sh: 600
    PERFPMR: Data collection started in foreground (renice -n -20)

     TRACE.SH: Starting trace for 5 seconds
/bin/trace -r PURR -k 492,10e,254,116,117  -f -n -C all -d -L 20000000 -T
200000
00 -ao trace.raw
     TRACE.SH: Data collection started
     TRACE.SH: Data collection stopped
     TRACE.SH: Trace stopped
     TRACE.SH: Collecting gennames data
     TRACE.SH: Trcnm data is in file trace.nm
     TRACE.SH: /etc/trcfmt saved in file trace.fmt
     TRACE.SH: Binary trace data is in file trace.raw

     TRACE.SH: Enabling locktrace
lock tracing enabled for all classes
     TRACE.SH: Starting trace for 5 seconds
/bin/trace -r PURR -j 106,10C,10E,112,113,134,139,465,46D,606,607,608,609
-f -n -C all -d -L 20000000 -T 20000000 -ao trace.raw.lock
     TRACE.SH: Data collection started
     TRACE.SH: Data collection stopped
     TRACE.SH: Trace stopped
```

```
     TRACE.SH: Disabling locktrace
lock tracing disabled for all classes
     TRACE.SH: Binary trace data is in file trace.raw.lock

     MONITOR: Capturing initial lsps, svmon, and vmstat data
     MONITOR: Starting system monitors for 600 seconds.
     MONITOR: Waiting for measurement period to end....
MONITOR: Capturing final lsps, svmon, and vmstat data
     MONITOR: Generating reports....
     MONITOR: Network reports are in netstat.int and nfsstat.int
     MONITOR: Monitor reports are in monitor.int and monitor.sum

     IPTRACE: Starting iptrace for 10 seconds....
0513-059 The iptrace Subsystem has been started. Subsystem PID is 389334.
0513-044 The iptrace Subsystem was requested to stop.
     IPTRACE: iptrace collected....
     IPTRACE: Binary iptrace data is in file iptrace.raw

     TCPDUMP: Starting tcpdump for 10 seconds....
     TCPDUMP: tcpdump collected....
     TCPDUMP: Binary tcpdump data is in file tcpdump.raw

     FILEMON: Starting filesystem monitor for 60 seconds....
     FILEMON: tracing started
     FILEMON: tracing stopped
     FILEMON: Generating report....

     TPROF: Starting tprof for 60 seconds....
TPROF: Sample data collected....
     TPROF: Generating reports in background (renice -n 20)
     TPROF: Tprof report is in tprof.sum
17:07:43-11/29/05 :     config.sh begin

     CONFIG.SH: Generating SW/HW configuration
17:07:43-11/29/05 :     copying ODM files
17:07:44-11/29/05 :     ipcs -Sa
17:07:44-11/29/05 :     lspv
0516-320 : Physical volume 00c478de09caf37f0000000000000000 is not assigned
to a volume group.
0516-320 : Physical volume 00c478de49630c6a0000000000000000 is not assigned
to a volume group.
0516-320 : Physical volume 00c478de00655246000000000000000 is not assigned
to a volume group.
17:07:45-11/29/05 :     lsvg -l
17:07:45-11/29/05 :     lslv lv
17:07:48-11/29/05 :     lsattr -E -l dev
17:07:49-11/29/05 :     df
17:07:49-11/29/05 :     netstat -in -rn -D -an -c
17:07:50-11/29/05 :     getmempool.sh
```

```
17:07:51-11/29/05 :     getj2mem.sh
17:07:51-11/29/05 :     genkld
17:07:51-11/29/05 :     genkex
17:07:51-11/29/05 :     getevars
17:07:51-11/29/05 :     errpt
17:07:51-11/29/05 :     emgr -l

There is no efix data on this system.
17:07:51-11/29/05 :     lslpp -ch
17:07:51-11/29/05 :     instfix -ic
17:07:52-11/29/05 :     lscfg -vp
17:07:53-11/29/05 :     xm -u |kdb
17:07:53-11/29/05 :     echo vnode|kdb
17:07:56-11/29/05 :     echo vfs|kdb
17:07:57-11/29/05 :     echo dmpdt_chrp -i
17:07:58-11/29/05 :     sysdumpdev -l, -e
     CONFIG.SH: Report is in file config.sum
17:07:58-11/29/05 :     config.sh completed

    PERFPMR: Data collection complete.

    PERFPMR: Data files can be archived and gzipped using:
            perfpmr.sh -z filename [-o "dirs"]
         where
            filename is the name of the archive file.
         An example of a typical archive filename:
            /tmp/NNNNN.bNNN.cNNN.perfpmr.pax.gz
         where NNNNN is the pmr#, .bNNN is the pmr branch #,
         and .cNNN is the country code
            -o "dirs":
         dirs is a list of directories enclosed in quotes. If -o is
         not specified, all files in current directory are archived.
         You must be in the directory which contains the list
         of directories when using the -z and -o flags
         After pax gzip file has been created, ftp the file to:
             testcase.software.ibm.com in /aix/toibm
         Login as user id: ftp

17:07:58-11/29/05 :     perfpmr.sh completed
```

Tip: It is useful to run the **`perfpmr`** command when your system is under load and performing normally. This gives you a baseline to determine future performance problems.

You should run the **`perfpmr`** command again when:

- Your system is experiencing performance problems.
- You make hardware changes to the system.
- You make any changes to your network configuration.
- You make changes to the AIX 5L Operating System, such as when you install upgrades or tune AIX 5L.
- You make changes to your application.

9.12 Manage a support call with IBM

To manage a support call with IBM, you have to follow these steps:

- Determine the business impact of your problem.
- Describe your problem and gather background information.
- Submit your problem to IBM Support.
- Update your request.

9.12.1 Determine the business impact of your problem

When you report a problem to IBM, you are asked to supply a severity level. Therefore, you need to understand and assess the business impact of the problem you are reporting.

Use Table 9-15 to select the corresponding severity level.

Table 9-15 Severity level versus business impact

Severity level	Business impact
Severity 1	*Critical* business impact: You are unable to use the program, resulting in a critical impact on operations. This condition requires an immediate solution.
Severity 2	*Significant* business impact: The program is usable but is severely limited.
Severity 3	*Some* business impact: The program is usable with less significant features (not critical to operations) unavailable.
Severity 4	*Minimal* business impact: The problem causes little impact on operations, or a reasonable circumvention to the problem has been implemented.

9.12.2 Describe your problem and gather background information

When explaining a problem to IBM, be as specific as possible. Include all relevant background information so that IBM Support specialists can help you solve the problem efficiently.

To save time, know the answers to these questions:

- On what machine did the problem occur? Keep the serial numbers ready.
- What software versions were you running when the problem occurred?
- Do you have logs, traces, and messages that are related to the problem symptoms?
- Can the problem be re-created? If so, what steps led to the failure?
- Have any changes been made to the system? For example, hardware, operating system, networking software, and so on.
- Are you currently using a workaround for this problem? If so, be prepared to explain it when you report it.

9.12.3 Submit your problem to IBM support

You can submit your problem in two ways:

- Online:
  - Software problems: *Electronic Service Request* (ESR) on `http://www.ibm.com/software/support/esr/support_contracts`

  - Hardware problems: *Electronic Service Call* (ESC+) on `http://www.ibm.com/support/esc/signin.jsp`

- By phone: For the phone number to call in your country, go to the contacts page on the web (`http://www.ibm.com/planetwide/`) and select the name of your country.

9.12.4 Updating your request

You can always update your support request by using the online forms or by phone, as mentioned above. Refer to the problem number the support specialist gave you. A Problem Management Record (PMR) is created for any problems reported to the IBM Support Center. A PMR is an online software record used to keep track of software problems reported by customers.

An advantage of updating the request online is that you can easily attach files to the PMR, for example, screen captures, logs, and so on.

Use the same routine to close the call or to alter the severity level.

10

Backup and recovery

This chapter covers the backup and restore procedures and the common tasks that are carried out to ensure that you have backed up the proper information and that your backup media is reliable.

© Copyright IBM Corp. 2006. All rights reserved.

10.1 The mksysb command

The **mksysb** command creates a bootable image of all mounted file systems on the rootvg volume group. You can use this backup command to restore a system to its original state.

The tape format includes a BOS boot image, a BOS install image, and a dummy table of contents (TOC) followed by the system backup (root volume group) image. The root volume group image is in backup-file format, starting with the data files and then any optional map files.

User-defined paging spaces, unmounted file systems, and raw devices are not backed up.

10.1.1 The data layout of a mksysb tape

The layout of a mksysb tape is shown in Figure 10-1.

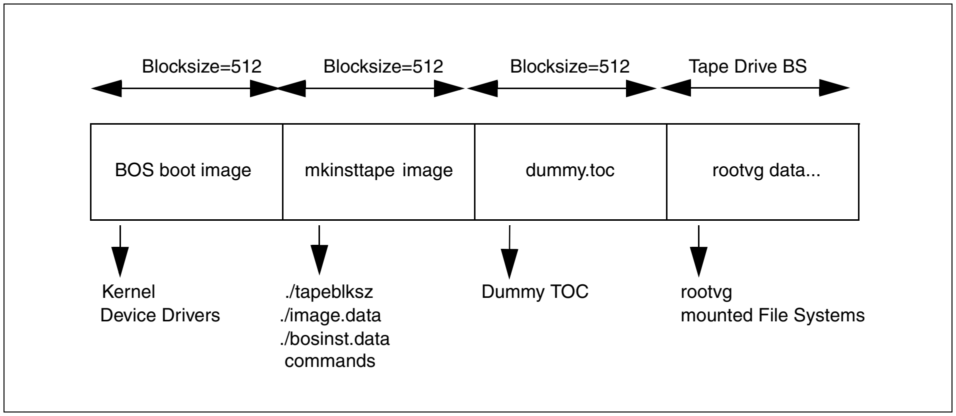


Figure 10-1 Layout of a mksysb tape

The BOS boot image contains a copy of the system's kernel and device drivers needed to boot from the mksysb tape. It is created by the **bosboot** command.

There are three important files in the mkinsttape image (the second section of the **mksysb** tape): ./tapeblksz, ./bosinst.data, and ./image.data. The ./tapeblksz file contains the block size the tape drive was set to when the **mksysb** command was run.

The ./bosinst.data file allows you to specify the requirements at the target system and how the user interacts with the target system. This file contains the customized BOS install procedures and dictates how the BOS install program will behave. You can customize this file before issuing the **mksysb** command or use a procedure to customize this file after the image backup is done.

The ./image.data file contains information describing the image installed during the BOS installation process. This information includes the sizes, names, maps, and mount points of logical volumes and file systems in the rootvg. You can customize this file before using the **mksysb** command, or run **mksysb -i** to generate a new ./image.data file on the tape during a backup. The **mkszfile** command generates the ./image.data file. The ./image.data file is arranged in stanza format. Each stanza contains one or more fields. The most important fields are:

SHRINK When set to YES, causes the system to create the smallest file systems required to contain all the data in the file system.

BOSINST_FILE Provides the full path name of a file or command to execute after the BOS install completes.

EXACT_FIT When set to YES, causes the system to place logical volumes on the disk according to the physical partition maps that were generated with the -m flag of the **mksysb** command.

The dummy table of contents (TOC) is used so that the **mksysb** tape contains the same number of images as a BOS install tape.

The rootvg data area contains all data in the rootvg volume group backed up by the **mksysb** command. The **mksysb** command uses the **backup** command to save the contents of mounted JFS data in rootvg, excluding raw data.

10.1.2 Excluding file systems from a backup

When you need to make a **mksysb** backup of a system, and you want to exclude some data file systems from the system, you need to edit the /etc/exclude.rootvg file.

The **mksysb** processing excludes files using the **grep** format, and the filelist that the exclusions are matched against are relative to the root directory.

If, for example, you want to exclude the file system and /tmp from your **mksysb** backup, add the following:

```
^./tmp/
```

Make sure there are no empty lines in this file. You can list the contents of the file as follows:

```
# cat /etc/exclude.rootvg
^./tmp/
#
```

Then run the **mksysb** command using the -e flag to exclude the contents of the exclude.rootvg file as follows:

```
# mksysb -e /dev/rmt0

Creating information file (/image.data) for rootvg.

Creating tape boot image..............
bosboot: Boot image is 29316 512 byte blocks.


Creating list of files to back up.
Backing up 2679 files.....
2679 of 2679 files (100%)
0512-038 mksysb: Backup Completed Successfully.

bosboot: Boot image is 29316 512 byte blocks.
#
```

10.1.3 How to create a bootable system backup

The **mksysb** command creates a bootable image of the rootvg file system either in a file system directory or onto a bootable tape, and is used to restore a system after a system failure or for system cloning.

To use SMIT to create a bootable system backup, follow the steps below:

1. Run the SMIT mksysb fast path **smitty mksysb**. In the Back Up the System menu, select Backup DEVICE or FILE field. This is where you would select your backup device. If you press F4, it will give you a list of available backup devices. Choose the device you want, and then press Enter, as shown in Figure 10-2 on page 463, where /dev/rmt0 is selected as the backup device.

```
                              Back Up the System

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[TOP]                                                    [Entry Fields]
    WARNING:  Execution of the mksysb command will
              result in the loss of all material
              previously stored on the selected
              output medium. This command backs
              up only rootvg volume group.

* Backup DEVICE or FILE                               [/dev/rmt0]             +/
  Create MAP files?                                    no                     +
  EXCLUDE files?                                       no                     +
  List files as they are backed up?                    no                     +
  Verify readability if tape device?                   no                     +
  Generate new /image.data file?                       yes                    +
  EXPAND /tmp if needed?                               no                     +
[MORE...4]

F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 10-2 Backup Up the System configuration window

2. The COMMAND STATUS screen is now displayed. Figure 10-3 is a screen capture of what information is being displayed during the backup process.

```
                                COMMAND STATUS

Command: running       stdout: yes          stderr: no

Before command completion, additional instructions may appear below.


Creating information file (/image.data) for rootvg.

Creating tape boot image..
```

Figure 10-3 Command status output from a running back up

3. The system has now created a bootable system backup, as shown in Figure 10-4.

```
                              COMMAND STATUS

Command: OK            stdout: yes          stderr: no

Before command completion, additional instructions may appear below.

Creating information file (/image.data) for rootvg.

Creating tape boot image................
bosboot: Boot image is 29316 512 byte blocks.
.

Creating list of files to back up.
Backing up 27894 files............................
27894 of 27894 files (100%)
0512-038 mksysb: Backup Completed Successfully.

bosboot: Boot image is 29316 512 byte blocks.


F1=Help             F2=Refresh          F3=Cancel           F6=Command
F8=Image            F9=Shell            F10=Exit            /=Find
n=Find Next
```

Figure 10-4 Command status output from a successful backup

10.1.4 Using mksysb to back up a user volume group

You can run the **mksysb** command on rootvg. You cannot run the **mksysb** command against a user volume group. If you want to back up a user volume group, you must use the **savevg**, **tar**, **cpio**, or **backup** commands.

10.1.5 List content of a mksysb image

If you want to verify the content of an mksysb image, execute the following instructions:

1. Run the SMIT lsmksysb fast path:

   ```
   smitty lsmksysb
   ```

   Select the DEVICE or FILE field. This is where you would select your device that contains the mksysb image. If you press F4, it will give you a list of available devices. Choose the device you want, and then press Enter, as shown in Figure 10-5 on page 465, where /dev/rmt0 is selected as the tape device.

```
                          List Files in a System Image

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                          [Entry Fields]
* DEVICE or FILE                                        [/dev/rmt0]            +/
  Number of BLOCKS to read in a single input            []                      #
     (Leave blank to use a system default)
  Verify BLOCK size if tape device?                      no                    +










F1=Help              F2=Refresh           F3=Cancel            F4=List
F5=Reset             F6=Command           F7=Edit              F8=Image
F9=Shell             F10=Exit             Enter=Do
```

Figure 10-5 lsmksysb SMIT fast path configuration

2. An output similar to Figure 10-6 is shown when the process finishes.

```
                                COMMAND STATUS

Command: OK            stdout: yes           stderr: no

Before command completion, additional instructions may appear below.

[TOP]
New volume on /dev/rmt0.1:
Cluster size is 51200 bytes (100 blocks).
The volume number is 1.
The backup date is: Mon Nov 14 18:32:16 CST 2005
Files are backed up by name.
The user is root.
        6011 ./bosinst.data
        8051 ./image.data
      129035 ./tmp/vgdata/rootvg/backup.data
           0 ./opt
           0 ./opt/lost+found
           0 ./opt/freeware
[MORE...32585]

F1=Help              F2=Refresh           F3=Cancel            F6=Command
F8=Image             F9=Shell             F10=Exit             /=Find
n=Find Next
```

Figure 10-6 Sample output of the SMIT lsmksysb

Attention: Note the relative path structure: The directory structure is composed in a relative way, indicated by the . (dot). When restoring single files from the mksysb image, it is important to maintain the same structure.

10.1.6 Restore a mksysb image

An mksysb image enables you to restore the system image onto target systems that might not contain the same hardware devices or adapters, require the same kernel (uniprocessor or microprocessor), or be the same hardware platform (rs6k, rspc, or chrp) as the source system.

You have several possibilities to restore the mksysb image:

- If restoring on exactly the same machine, you can boot directly from the mksysb media and restore from there.
- If restoring on a different type of machine, you use the cloning function, as described in 3.6, “Cloning an AIX 5L system” on page 57.
- If you do not want to interfere with the production environment, you can use the **alt_mksysb** command, as described in 3.5, “Alternate disk installation” on page 56.

If you want to restore only several files from the mksysb image, execute the following steps:

1. Run the SMIT **restmksysb** fast path:

   ```
   smitty restmksysb
   ```

   Select Restore DEVICE or FILE field. This is where you would select your restore device. If you press F4, it will give you a list of available restore devices. Choose the device you want, and then press Enter, as shown in Figure 10-7 on page 467, where /dev/rmt0 is selected as the restore device. In the example below, the /etc/hosts file is specified to be restored.

Important: Note the relative path structure for the hosts file: The directory structure is composed in a relative way; without the . (dot), the file will not be restored. See 10.1.5, “List content of a mksysb image” on page 464 for more details.

```
                         Restore Files in a System Image

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                         [Entry Fields]
* Restore DEVICE or FILE                               [/dev/rmt0]             +/
  FILES to restore                                     [./etc/hosts]
     (Leave blank to restore entire archive)
  Number of BLOCKS to read in a single input           []                       #
     (Leave blank to use a system default)
  Verify BLOCK size if tape device?                     no                      +
  Restore extended attributes?                          yes                     +










F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 10-7 Restore Files in a System Image within SMIT

2. The result is similar to the screen shown in Figure 10-8.

```
                                 COMMAND STATUS

Command: OK             stdout: yes           stderr: no

Before command completion, additional instructions may appear below.

New volume on /dev/rmt0.1:
Cluster size is 51200 bytes (100 blocks).
The volume number is 1.
The backup date is: Mon Nov 14 18:32:16 CST 2005
Files are backed up by name.
The user is root.
x         1836 ./etc/hosts
The total size is 1836 bytes.
The number of restored files is 1.






F1=Help             F2=Refresh          F3=Cancel           F6=Command
F8=Image            F9=Shell            F10=Exit            /=Find
n=Find Next
```

Figure 10-8 Sample output of the restore of one file from mksysb

If you want to perform the same steps on the command line, you need to use the **tctl** command with the fsf 3 parameter, followed by the **restore** command.

10.2 Managing tape backup media

AIX 5L provides you with a number of commands to successfully back up your system to ensure availability of data at all times. These commands vary, depending on the type of information you are trying to save.

For example, in order to save rootvg, you can use different approaches, depending on what are you trying to save by using the **mksysb** command. You can alternatively save individual files by using the **tar**, **cpio**, **dd**, and **backup** commands. You can also save entire user volume groups by using the **savevg** command.

Figure 10-9 on page 469 summarizes the methods used to back up system and user data.

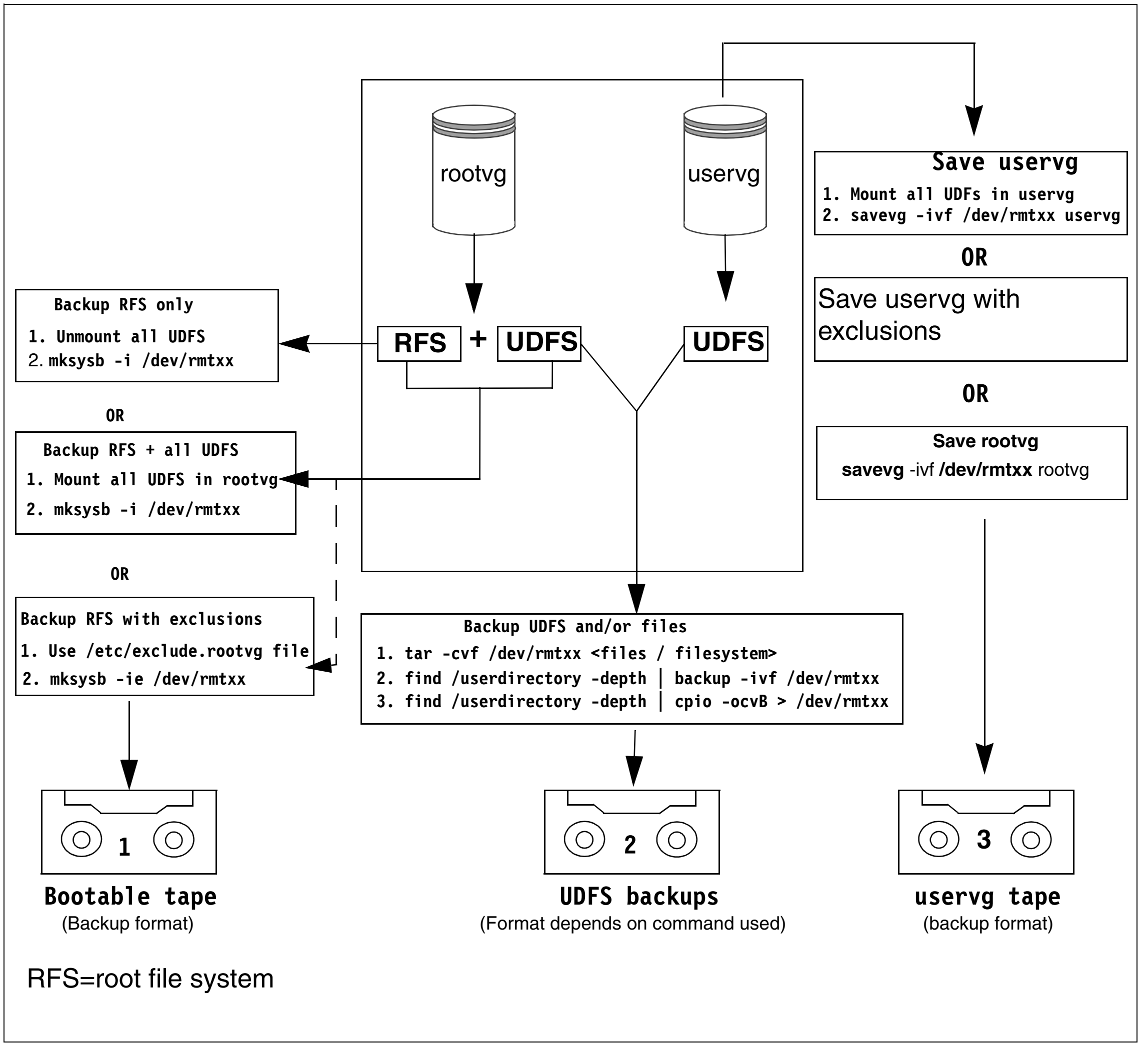


Figure 10-9 Flowchart for saving information

10.2.1 The tctl command

The **tctl** command sends subcommands to a streaming tape device, like /dev/rmt0. This command can manipulate tapes easily for data restoration. The general syntax of the **tctl** command is as follows:

`tctl [ -f` *Device* `] [ eof | weof | fsf | bsf | fsr | bsr | rewind | offline | rewoffl | erase | retension | reset | status ] [` *Count* `]`

If you do not specify the *Device* variable with the -f flag, the TAPE environment variable is used.

Table 10-1 describes subcommands to manipulate media, to add or restore data, after and before marks.

Table 10-1 Commonly used subcommands for the tctl command

Subcommand	Description
rewind	Rewinds the tape.
rewoffl or offline	Rewinds the tape and takes the tape drive offline. This will unload or eject the tape when appropriate. The tape must be reinserted, or another tape must be loaded before the device can be used again.
status	Displays status information about the specified tape device.
reset	Sends a bus device reset (BDR) to the tape device. The BDR will only be sent if the device cannot be opened and is not busy. Once complete, the system will return the cursor to a prompt without any notification.
fsf *Count*	Moves the tape forward by the number of file marks specified by the *Count* parameter, and positions it on the end-of-tape (EOT) side of the file mark.
bsf *Count*	Moves the tape backward by the number of file marks specified by the *Count* parameter, and positions it on the beginning-of-tape (BOT) side of the file mark. If the bsf subcommand attempts to move the tape past the beginning, the tape will rewind and the **tctl** command returns EIO.

10.3 Backup strategies

Before implementing your backup strategy, you have to choose the best one for your environment. As all of the strategies have their advantages, you have to match them with your requirements, based on how much new data you generate, the number of media sets you want to use, and the backup and restore window you rely on.

There are three types of backup methods:

- Full backup
- Differential backup
- Incremental backup

10.3.1 Full backup

All the files are put on media during a full backup.

The advantages are:

- Restoring from a full backup lowers the tape handling activities compared to an incremental or differential backup.
- The content on the media is from within the same backup window (rollback and redundancy).

The disadvantages are:

- The backup window is wide.
- If the data does not change, every full backup media set contains the same data.

10.3.2 Differential backup

The differential backup strategy first looks for the modification time of a file and compares it with the last full backup time. In other words, only modified files are backed up, but only if they changed after the latest full backup. Differential backups are cumulative; once a file has been modified, it will be included in every differential backup until the next full backup.

The advantages are:

- To restore, the latest full backup and only the latest differential backup media sets are needed.
- The backup window is smaller than a full backup.

The disadvantages are:

- If data changes a lot between full backups, the data to back up tends to grow a bit over time, using more media.

10.3.3 Incremental backup

Incremental backups are similar to differential backups in that both back up only modified files. However, incremental backup checks the difference between the modification time of a file and the last backup time (either being full or incremental backup). If the modification date is more recent than the last backup date, the file is backed up.

The advantages are:

- The backup window is smaller than a full backup.
- Only the differences from a previous backup will be written on media.

The disadvantages are:

- To restore, the latest full backup and all the subsequent incremental backup media sets following that full backup are needed.
- To restore a single file, tape handling operations are intensive.
- A damaged or lost media in the incremental set can mean disaster: The modifications of those files on that media may be lost forever.

> **Note:** Both incremental and differential backups are used in conjunction with regularly scheduled full backups.

10.3.4 Backup example

Customer XYZ takes a weekly full backup every Saturday on tape. It takes two hours to back up all the data, which is too long to do a daily full backup. Customer XYZ therefore schedules a daily incremental backup at midnight.

Which tape sets are required when a disaster strikes Tuesday around noon?

The required tape sets are:

- Saturday full backup
- Sunday midnight incremental backup
- Monday midnight incremental backup

If everything restores fine, only the data changed after Monday midnight until the crash are lost.

Customer XYZ changes to the differential backup strategy, maintaining the current backup schedules. The following tape sets are required:

- Saturday full backup
- Monday midnight differential backup

The number of media sets required to recover decreases.

10.4 Related backup and restore commands

The following sections describes various different backup and restore commands. They can be divided into volume group, file system, or file backup and restore tools.

10.4.1 The savevg command

The **savevg** command finds and backs up all files belonging to a specified volume group. The command has the following syntax:

```
savevg [ -e ] [ -f Device ] [ -i | -m ] [ -p ] [ -v ] [ -X ] VGName
```

The commonly used flags are shown in Table 10-2.

Table 10-2 Commonly used flags for the savevg command

Flag	Description
-e	Excludes files specified in the /etc/exclude.vgname file from being backed up by this command.
-f *Device*	Specifies the device or file name on which the image is to be stored. The default is the /dev/rmt0 device.
-i	Creates the data file by calling the **mkvgdata** command.
-m	Creates the data file with map files.
-p	Disables software packing of the files as they are backed up.
-X	Specifies to automatically expand the /tmp file system if necessary.

Figure 10-10 shows the screen that appears after running the **smit savevg** command.

```
                       Back Up a Volume Group to Tape/File

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[TOP]                                                 [Entry Fields]
    WARNING:  Execution of the savevg command will
              result in the loss of all material
              previously stored on the selected
              output medium.

* Backup DEVICE or FILE                              []                      +/
* VOLUME GROUP to back up                            []                      +
  List files as they are backed up?                   no                     +
  Generate new vg.data file?                          yes                    +
  Create MAP files?                                   no                     +
  EXCLUDE files?                                      no                     +
  EXPAND /tmp if needed?                              no                     +
  Disable software packing of backup?                 no                     +
[MORE...5]

F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 10-10 Back Up a Volume Group from SMIT

The **savevg** command uses a data file created by the **mkvgdata** command. The data file created is as follows:

/tmp/vgdata/vgname/vgname.data

The vgname.data file contains information about a user volume group. The **savevg** command uses this file to create a backup image that can be used by the **restvg** command to re-create the user volume group.

Excluding data using the savevg command

The **savevg** command with the -r flag is used to back up only a user-volume group's logical volume structure information. The data needed to list backup properties is also backed up. The -r flag runs the **mkvgdata** command for the volume group specified to create a vgname.data file. The -r flag backs up only the vgname.data file, any map files, and the backup.data file. The backup image that is created is used with the **restvg -r** command option to create only the volume group, logical volumes, and file system information contained in the file, without restoring any data. For example, to back up only the paul user volume group's structure information to the /vg_backup/paul_vg_data file, type the following:

```
# savevg -r -f /vg_backup/paul_vg_data paul
```

10.4.2 The restvg command

The **restvg** command restores the user volume group and all its containers and files. The command has the following syntax:

```
restvg [ -f Device ] [ -s ] [-n ] DiskName ... ]
```

The commonly used flags are shown in Table 10-3.

Table 10-3 Commonly used flags for the restvg command

Flag	Description
-n	Specifies that the existing MAP files are ignored.
-f *Device*	Specifies the device or file name on which the image is to be stored. The default is the /dev/rmt0 device.
-s	Specifies that the logical volumes be created at the minimum size possible to accommodate the file systems.

Figure 10-11 shows the screen that appears after running the **smit restvg** command.

```
                           Remake a Volume Group

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[TOP]                                                  [Entry Fields]
* Restore DEVICE or FILE                              [ ]                      +/
  SHRINK the filesystems?                              no                      +
  Recreate logical volumes and filesystems only?       no                      +
  PHYSICAL VOLUME names                               []                       +
     (Leave blank to use the PHYSICAL VOLUMES listed
      in the vgname.data file in the backup image)
  Use existing MAP files?                              yes                     +
  Physical partition SIZE in megabytes                []                       +#
     (Leave blank to have the SIZE determined
      based on disk size)
  Number of BLOCKS to read in a single input          []                        #
     (Leave blank to use a system default)
  Alternate vg.data file                              []                        /
[MORE...2]

F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 10-11 Remake a Volume Group from SMIT

Excluding data using the restvg command

If your rootvg image and savevg image are small enough to fit on to one CD, you can save them both by using the -l (stacklist) and -z (customization_script) flags. The -l flag gives a list of images to copy to the CD. The -z flag lets you create a script to restore **savevg** backups. For example, if you make a copy of a non-rootvg volume group ahead of time, and then write a script that calls the **restvg** command, your non-rootvg volume group would be restored to hdisk2 at the end of the installation of rootvg, as shown by the following command:

```
# restvg -d /SPOT/installp/ppc/savevg_image hdisk2
```

This procedure is recommended only if you know you want to restore the non-rootvg volume group every time you install. Otherwise, you might just want to store it on the CD/DVD, then use **restvg** to restore it after reboot. The **restvg** command can restore from CD or DVD if the name of the image is savevg_image. If you save the non-rootvg backup on a CD or DVD with a different file name, you can insert that CD or DVD and use the full path to the file name as the device for the **restvg** command.

10.4.3 The backup command

The **backup** command backs up files and file systems. The command has the following syntax:

```
backup -i [ -p [ -e RegularExpression ] ] [ -f Device ] [ -l Number ] [ -o ]
[ -q ] [ -v ] file1 [file2]...
```

The commonly used flags are shown in Table 10-4.

Table 10-4 Commonly used flags for the backup command

Flag	Descriptions
-f *Device*	Specifies the output device.
-i	Specifies that files be read from standard input and archived by file name.
-m	Creates the data file with map files.
-q	Indicates that the removable medium is ready to use.
-v	Causes the **backup** command to display additional information about the backup.
-u	Updates the /etc/dumpdates file with the raw device name of the file system and the time, date, and level of the backup. You must specify the -u flag if you are making incremental backups. The -u flag applies only to backups by i-node.

Excluding data using backup command

To exclude data that you do not want to back up from a specific path (directory or file system), use the association between some commands (for example, **find** and **print**) and send the result to the **backup** command.

The following are some useful examples:

- To back up all the files and subdirectories in the /home directory using full path names, enter:

  ```
  # find /home -print | backup  -i -f /dev/rmt0
  ```

  The -i flag specifies that files will be read from standard input and archived by file name. The **find** command generates a list of all the files in the /home directory. The files in this list are full path names. The | (pipe symbol) causes this list to be read from standard input by the **backup** command. The -f flag directs the **backup** command to write the files to the /dev/rmt0 tape device. Because the files are archived using full path names, they will be written to the same paths when restored.

- To back up all the files and subdirectories in the /home/mike directory using relative path names, enter:

```
# cd /home
# find . -print | backup  -i -v -q
```

  Each file name in the list generated by the **find** command is preceded by ./ (dot, slash). Because the files are backed up using relative path names, they will be written to the current directory when restored. The -v flag causes the **backup** command to display additional information about the backup. The files are written to the default backup device /dev/fd0.

- To back up the / (root) file system, enter:

```
# backup  -0 -u -f /dev/rmt0 /
```

  The 0 level specifies that all the files in the / (root) file system be backed up. The -u flag causes the **backup** command to update the /etc/dumpdates file for this backup.

- To back up all the files in the / (root) file system that have been modified since the last level 0 backup, enter:

```
# backup  -1 -u -f /dev/rmt0 /
```

  If the /etc/dumpdates file does not have an entry for a level 0 backup of the / (root) system, all the files in the file system are backed up.

10.4.4 The restore command

The **restore** command extracts files from archives created with the **backup** command. The command has the following syntax:

```
restore -x -T [ v q ] [ -f Device ] [ File ... ]
```

The commonly used flags are shown in Table 10-5.

Table 10-5 Commonly used flags for the restore command

Flag	Description
-f *Device*	Specifies the device or file name on which the image is to be stored. The default is the /dev/rmt0 device.
-T	Displays information about the backup archive.
-q	Indicates that the removable medium is ready to use.
-v	Causes the **restore** command to display additional information about the backup.
-x	Restores individually named files specified by the file parameter.

Excluding data using the restore command

To exclude data that you do not want to restore from a specific path (directory or file system), use the association between some commands (for example, **find** and **print**) and send the result to the **restore** command.

The following are some useful examples:

- To restore an entire file-system archive, type:

  ```
  # restore -rvqf /dev/rmt0
  ```

  This command restores the entire file system archived on the tape device, /dev/rmt0, into the current directory. This example assumes you are in the root directory of the file system to be restored. If the archive is part of a set of incremental file-system archives, the archives should be restored in increasing backup-level order beginning with level 0 (for example, 0, 1, and 2).

- To restore a specific directory and the contents of that directory from a file-name archive, type:

  ```
  # restore -xdvqf /dev/rmt0 /home/mike/tools
  ```

  The -x flag tells **restore** to extract files by their file name. The -d tells **restore** to extract all the files and subdirectories in the /home/mike/tools directory. File and directory names are displayed as they are restored when using the -T flag. If the directories do not exist, they are created.

- To restore a specific directory and the contents of that directory from a file-system archive, type:

  ```
  # restore -xvqf /dev/rmt0 /home/mike/tools
  ```

  This command extracts files by file name. File and directory names must be specified as they are displayed when using the -T flag. If the directories do not exist, they are created.

10.4.5 The tar command

The **tar** command manipulates archives by writing files to, or retrieving files from, an archive storage medium. The command has the following syntax:

```
tar { -c | -t | -x } [ -B ] [ -v ] [ -f Archive] [ File | Directory ...
```

The commonly used flags are shown in Table 10-6.

Table 10-6 Commonly used flags for the tar command

Flag	Description
-c	Creates a new archive and writes the files specified by one or more file parameters to the beginning of the archive.
-t	Lists the files in the order in which they appear in the archive.
-B	Forces input and output blocking to 20 blocks per record.
-f *Archive*	Uses the Archive variable as the archive to be read or written.
N	Block size
-v	Lists the name of each file as it is processed.
-x	Restores individually named files specified by the File parameter.

Excluding data using the tar command

To exclude data that you do not want to back up using the **tar** command on a specific path (directory or file system), use the association between some commands (for example, **ls** and **list files**) and send the result to the **tar** command.

The following are some useful examples:

- To write the file1 and file2 files to a new archive on the default tape drive, enter:

```
# tar  -c file1 file2
```

- To extract all files in the /tmp directory from the archive file on the /dev/rmt2 tape device and use the time of extraction as the modification time, enter:

```
# tar -xm -f/dev/rmt2 /tmp
```

- To archive a list of all C files that is listed in the file through the InputList argument of the -L option, enter:

```
# tar -cvf fl.tar -L fl_list
```

Where fl_list is a file consisting a list of all .c files in it. This can be obtained as follows:

```
# ls *.c > fl_list
```

3. To archive a list of all C files by setting a variable using the -L option, enter:

```
# ls *.c > fl_list
# fl=fl_list
# tar -cvf var.tar -L $fl
```

10.4.6 The cpio command

The **cpio** command copies files into and out from a cpio archive. The cpio archive may span multiple volumes. The -i, -o, and -p flags select the action to be performed. The command has the following syntax:

```
cpio -i [ -B ] [ -c ] [ -d ] [ -m ] [ -u ] [ -v ] [ -o] ] [ Patterns ...]
```

The commonly used flags are shown in Table 10-7.

Table 10-7 Commonly used flags for the cpio command

Flag	Description
-B	The default buffer size is 512 bytes when neither this or the -C option is used. But when the -B flag is used, the buffer size is set to a 5120 bytes block for the Input/Output operations
-c	Reads or writes header information in ASCII character form for system interoperability and portability. The -c option is mutually exclusive with -H and -6. Either the -c or -H option can be used when the target and destination machines are different types.
-d	Creates directories as needed.
-m	Retain previous file modification time. The modification time and access time of a restored file is set to the modification time of the file when it was backed up. Modification time of directories is not retained.
-u	Copies unconditionally. An older file now replaces a newer file with the same name.
-v	Lists file names.
-o	Reads file path names from standard input and copies these files to standard output.

Excluding data using the cpio command

To exclude data that you do not want to back up using the **cpio** command on a specific path (directory or file system), use the association between some commands (for example, **ls** and **find**) and send the result to the **cpio** command.

The following are some useful examples:

To copy files onto diskette, enter:

```
# cpio -ov path /dev/fd0
```

To copy files in the current directory onto diskette, enter:

```
# ls *.c | cpio -ov >/dev/fd0
```

This copies all the files in the current directory whose names end with .c

To copy the current directory and all subdirectories onto diskette, enter:

```
# find . -print | cpio -ov >/dev/fd0
```

This saves the directory tree that starts with the current directory (.) and includes all of its subdirectories and files. Do this faster by entering:

10.4.7 The pax command

The **pax** command extracts and writes member files of archive files, writes lists of the member files of archives, and copies directory hierarchies. The -r and -w flags specify the type of archive operation.

The commonly used flags are shown in Table 10-8.

Table 10-8 Commonly used flags for the pax command

Flag	Description
-a	Appends files to the end of an archive.
-f *Archive*	Specifies the path of an archive file to be used instead of standard input (when the -w flag is not specified) or standard output (when the -w flag is specified but the -r flag is not).
-r	Reads an archive file from the standard input.
-v	Writes information about the process. If neither the -r or -w flags are specified, the -v flag produces a verbose table of contents; otherwise, archive member path names are written to standard error.
-w	Writes files to the standard output in the specified archive format.
-x *Format*	Specifies the output archive format.

Excluding data using the pax command

To exclude data that you do not want to back up using the **pax** command on a specific path (directory or file system), use the association between some commands (for example, **ls** and **find**) and send the result to the **pax** command.

The following are some useful examples:

1. To copy the contents of the current directory to the tape drive, enter:

   ```
   # pax -wf /dev/rmt0
   ```

2. To archive the file xyz as XYZ and display the successful substitution, enter:

```
# pax -wvf/dev/fd0 -s /xyz/XYZ/p xyz
```

or:

```
# pax -wvf/dev/fd0 -s/x/X/gp xyz
```

3. To read a file from a standard input and dump it to a data stream file with a specified size, enter:

```
# dd if=/dev/hd6 bs=36b count=480 | pax -wf /dev/fd0 -o
datastream=_filename_,datastr_size=_size_
```

4. To ignore the path name from the archive in pax format during extraction, enter:

```
# pax -rvf pax.ar -o delete=path
```

10.4.8 The mkcd command

The **mkcd** command creates a system backup image (mksysb) to CD-Recordable (CD-R) or DVD-Recordable (DVD-R, DVD-RAM) from the system rootvg or from a previously created **mksysb** image. It also creates a volume group backup image (savevg) to CD-R from a user-specified volume group or from a previously created **savevg** image.

For DVD media, system backups made with the **mkcd** command have a limitation in that they expect the media to be 4.7 GB or larger per side. The **mkcd** command will not process the next volume until it writes over 4 GB on the current volume, so the use of smaller media would result in corruption when going beyond the media's capacity.

With the **mkcd** command, you can create bootable and non-bootable CDs in Rock Ridge (ISO9660) or UDF (Universal Disk Format) format.

See the -L flag for details about creating DVD-sized images. What applies to CDs also applies to DVDs, except where noted.

If you need to create multi-volume CDs because the volume group image does not fit on one CD, the **mkcd** command gives instructions for CD replacement and removal until all the volumes have been created.

The most commonly used flags are shown in Table 10-9.

Table 10-9 Commonly used flags for the mkcd command

Flag	Descriptions
-d cd_device	Indicates the CD-R, DVD-R or DVD-RAM device (/dev/cd1, for example). This flag is required unless you use the -S flag.
-r directory	Indicates the existing directory structure to burn onto a CD or DVD. This makes a CD image that is a copy of the given directory structure.
-m mksysb_image	Specifies a previously created mksysb image. If you do not give the -m flag, the **mkcd** command calls the **mksysb** command. (See the -M flag for more information about where the mksysb image is placed.)
-s savevg_image	Indicates a previously created savevg image.
-v savevg_volume_group	Denotes the volume group to be backed up using the **savevg** command
-C cd_fs_dir	Specifies the file system used to create the CD file system structure, which must have at least 645 MB of available disk space (up to 4.38 GB for DVD sized images). The CD image will only consume as much room as necessary to contain all the data on the CD.
-M mksysb_target	States the directory or file system where the mksysb or savevg image is stored if a previously created backup is not given with the -m or -s flags. If the -M flag is not used and a mksysb or savevg image is not provided, the **mkcd** command verifies that /mkcd/mksysb_image exists. If the directory does not exist, then the **mkcd** command creates a separate file system, /mkcd/mksysb_image, where the mksysb or savevg images are temporarily stored. The command creates the file system in the volume group given with the -V flag, or in rootvg if that flag is not used.
-I cd_image_dir	Specifies the directory or file system where the final CD images are stored before writing to the CD-R, DVD-R or DVD-RAM device. If this flag is not used, the **mkcd** command uses the /mkcd/cd_images directory if it already exists. If not, the command creates the /mkcd/cd_images file system in the volume group given with the -V flag, or in rootvg if that flag is not used.

Flag	Descriptions
-V cdfs_volume_group	Indicates the volume group used when creating the file systems needed for the **mkcd** command. If the -V flag is not given and a file system is needed but not there (because it was not supplied with other flags), then rootvg is the default volume group for creating the file systems. If the **mkcd** command creates the file systems in the backup volume group, those file systems are not included as part of the backup image. The file systems created by the **mkcd** command are removed upon the command's completion.
-p pkg_source_dir	Names the directory or device that contains device and kernel package images. The device can only be a CD device (for example, /dev/cd0). If you use the same CD-R, DVD-R, or DVD-RAM device that you gave with the -d flag, the product CD media must be inserted into the CD-R drive first. The **mkcd** command then prompts you to insert the writable CD before the actual CD creation.

The following are some useful examples:

1. To generate a bootable system backup to the CD-R device named /dev/cd1, enter:

```
# mkcd -d /dev/cd1
```

2. To generate a system backup to the DVD-R or DVD-RAM device named /dev/cd1, enter:

```
# mkcd -d /dev/cd1 -L
```

3. To generate a non-bootable volume group backup of the volume group myvg to /dev/cd1, enter:

```
# mkcd -d /dev/cd1 -v myvg
```

4. To generate a non-bootable system backup, but stop **mkcd** before the CD is created and save the final images to the /mydata/my_cd file system, and create the other mkcd file systems in myvg, enter:

```
# mkcd -B -I /mydata/my_cd -V myvg -S
```

10.4.9 The gzip and gunzip commands

The **gzip** command is a compression utility used to compress files, and the **gunzip** command to expand files. They have been adopted by the GNU project. They are included in the AIX 5L Toolbox for Linux Applications.

Their usage is as follows:

```
gzip [-cdfhlrt] [-S suffix] [file ...]
```

The commonly used flags are shown in Table 10-10.

Table 10-10 Commonly used flags for the gzip and gunzip commands

Flag	Description
-c	Writes on standard output. Keeps original files unchanged.
-d	Decompresses.
-f	Forces overwrite of output file and compress links.
-h	Gives help.
-l	Lists compressed file contents.
-r	Operates recursively on directories.
-t	Tests compressed file integrity.
-S	Choose the suffix on compressed files.
-v	Verbose mode.

To exclude data that you do not want to compress or decompress using the **gzip** command or the **gunzip** command on a specific path (directory or file system), use the association between some commands (for example, **ls** and **find**) and send the result to the **gzip** or **gunzip** commands.

The following are some useful examples:

To compress files:

```
# gzip -c file1
```

To decompress files:

```
# gzip -d file1.gz
```

To test a compressed files:

```
# gzip -t file1
```

10.5 Verify the content of a backup media

Verifying the content of a backup media is a very important step in your backup strategy. It is a good practice to verify the readability to eliminate troubles at recovery time, to avoid tape incompatibilities, damaged media, or missing files.

When changing backup strategy (for example: additional file systems, executing new backup scripts), you verify the list of files. The command to create the backup has an equivalent command to restore or verify the content. See 10.4.3, "The backup command" on page 477 for more information. You can also use the **tcopy** command, which is normally used to duplicate tapes from one media format to another. With only a source tape parameter specified, the **tcopy** command prints information about the size of records and tape files.

If a backup media has difficulties while reading the tape, execute the following steps:

1. Ensure that the media is not damaged. If possible, try another (new) media.
2. Verify that you have the latest device drivers installed for your backup device.
3. Check that the backup device is turned on (applicable on external devices).
4. Change the block_size parameter or the tape streamer to 0 (auto detect).
5. Try the media on another server.

The following examples show the usage of the **tcopy** command:

- To verify an mksysb image integrity:

  The **tcopy** command confirms the layout of the mksysb image:

```
# tcopy /dev/rmt0
tcopy: Tape File: 1; Records: 1 to 29316; Size: 512.
tcopy: File: 1; End of File after: 29316 Records, 15009792 Bytes.
tcopy: Tape File: 2; Records: 1 to 10300; Size: 512.
tcopy: File: 2; End of File after: 10300 Records, 5273600 Bytes.
tcopy: Tape File: 3; Record: 1; Size 512.
tcopy: File: 3; End of File after: 1 Records, 512 Bytes.
tcopy: Tape File: 4; Records: 1 to 798150; Size: 1024.
tcopy: File: 4; End of File after: 798150 Records, 817305600 Bytes.
tcopy: The end of the tape is reached.
tcopy: The total tape length is 837589504 bytes.
#
```

- To verify a **tar** archive:

  The **tcopy** command will only see one block of data. You receive an output similar to the following example:

```
# tcopy /dev/rmt0
tcopy: Tape File: 1; Records: 1 to 477; Size: 2097152.
tcopy: Tape File: 1; Record: 478; Size 976896.
tcopy: File: 1; End of File after: 478 Records, 1001318400 Bytes.
tcopy: The end of the tape is reached.
tcopy: The total tape length is 1001318400 bytes.
#
```

10.6 Disaster Recovery plans

This section discusses Disaster Recovery for two ways, enterprise environment and for local servers, where AIX 5L resides.

Worldwide, businesses continually increase their dependence on IT systems for routine businesses process. The business process that directly rely on information systems and the supporting IT infrastructure often require high levels of availability and recovery in the case of an unplanned outage. As a result, the process of business continuity planning must intimately relate business process to the traditional process of IT disaster recovery. The common backup commands provided on AIX 5L can be used to save data on reliable devices and media. A good disaster recovery includes IBM Tivoli Storage Manager software, where all environments, servers, devices ,and user data can be part of the backup strategy.

Creating a Disaster Recovery Plan is not enough — it must be carefully and regularly tested to ensure that it works and that the people responsible for executing it know how to execute it. The overall business structure of an enterprise will remain relatively stable over a period of time. A Disaster Recovery Plan is a vital element for an enterprise to describe how the continuity of the business processes will be preserved in case of a disaster. The technical details and the human resources of a business requirement typically change more frequently. An update process for the Disaster Recovery Plan is necessary, so its functionality and effectiveness is preserved, TSM can provide a lot of possibilities to implement a secure Disaster Recovery Plan.

There are several publications about disaster recovery, such as *Disaster Recovery Strategies with Tivoli Storage Management*, SG24-6844.

HACMP for AIX 5L is an application solution that can link up to eight IBM @server p5 servers or SP nodes into highly available clusters. With the enhanced scalability feature, up to 16 SP nodes can be linked. Clustering servers or nodes enables parallel access to their data, which can help provide the redundancy and fault resilience required for business critical applications. HACMP includes graphical user interface-based tools to help install, configure, and manage your clusters in a highly productive manner.

HACMP is flexible in configuration and use. Uniprocessors, symmetric multiprocessors (SMPs), and SP nodes can all participate in highly available clusters. Micro Channel® and PCI-based systems are supported under AIX 5L. You can mix and match system sizes and performance levels as well as network adapters and disk subsystems to satisfy your application, network, and disk performance needs.

HACMP clusters can be configured in several modes for different types of processing requirements. Concurrent access mode suits environments where all of the processors must work on the same workload and share the same data at the same time. In a mutual takeover mode, the processors share the workload and back each other up. Idle standby allows one node to back up any of the other nodes in the cluster.

Whichever mode you choose, HACMP provides data access and backup plans to help optimize application execution and scalability while helping to guard against costly unplanned outages and down time. HACMP also enables server clusters to be configured for application recovery/restart to provide a measure of fault resilience for your business critical applications through redundancy.

Understanding HACMP is a lesson in highly available systems. If you do not want to commit all the resources required for an HACMP installation, you can still eliminate many of the potential exposures for system downtime by adding redundancy to disk drives, adapter cards, network connections, and by implementing software RAS features, such as disk mirroring and system monitoring, as described in an HACMP installation.

11

Daily management

This chapter discusses user administration, which consists of creating and removing user accounts, defining and changing user attributes, and working with the user administration related files. This chapter also discusses monitoring and managing processes, file and directory permissions and ownership local and global variables, and the **cron** daemon and the **crontab** command.

© Copyright IBM Corp. 2006. All rights reserved.

11.1 User administration overview

Users are the primary agents on the system. Each user is required to log in to the system. The user supplies the user name of an account and a password if the account has one (on a secure system, all accounts either have passwords or are invalidated). If the password is correct, the user is logged in to that account; the user acquires the access rights and privileges of the account. The /etc/passwd and /etc/security/passwd files maintain user passwords.

Groups are collections of users who can share access permissions for protected resources. A group has an ID, and is composed of members and administrators. The creator of the group is usually the first administrator. There are three types of groups:

User group User groups should be made for people who need to share files on the system, such as people who work in the same department or people who are working on the same project. In general, create as few user groups as possible.

System administrator groups System administrator groups correspond to the SYSTEM group. SYSTEM group membership allows an administrator to perform some system maintenance tasks without having to operate with root authority.

System-defined groups There are several system-defined groups. The STAFF group is the default group for all non administrative users created in the system. You can change the default group by using the **chsec** command to edit the /usr/lib/security/mkuser.default file. The SECURITY group is a system-defined group having limited privileges for performing security administration.

An attribute is a characteristic of a user or a group that defines the type of functions that a user or a group can perform. These can be extraordinary privileges, restrictions, and processing environments assigned to a user. Their attributes control their access rights, environment, how they are authenticated, and how, when, and where their accounts can be accessed. These attributes are created from default values when a user is created through the **mkuser** command. They can be altered by using the **chuser** command.

Some users and groups can be defined as administrative. These users and groups can be created and modified only by the root user.

11.1.1 User administration related commands

The following are a few of the important commands used for user administration:

mkuser	Creates a new user account.
passwd	Creates or changes the password of a user.
chuser	Changes user attributes.
lsuser	Displays user account attributes.
rmuser	Removes a user account.
chsec	Changes the attributes in the security stanza files.
login	Initiates a user session.
who	Identifies the users currently logged in.
dtconfig	Enables or disables the desktop autostart feature.

An explanation of these commands can be found in 11.2, "User administration tasks" on page 503 and also in the following sections.

11.1.2 User administration related files

The following files are referenced while doing user administration:

/etc/security/environ	Contains the environment attributes for users.
/etc/security/lastlog	Contains the last login attributes for users.
/etc/security/limits	Contains process resource limits for users.
/etc/security/user	Contains extended attributes for users.
/usr/lib/security/mkuser.default	Contains the default attributes for new users.
/usr/lib/security/mkuser.sys	Customizes new user accounts.
/etc/passwd	Contains the basic attributes of users.
/etc/security/passwd	Contains password information.
/etc/security/login.cfg	Contains system default login parameters.
/etc/utmp	Contains a record of users logged into the system.
/var/adm/wtmp	Contains connect-time accounting records.
/etc/security/failedlogin	Records all failed login attempts.

/etc/motd	Contains the message to be displayed every time a user logs in to the system.
/etc/environment	Specifies the basic environment for all processes.
/etc/profile	Specifies additional environment settings for all users.
$HOME/.profile	Specifies environment settings for a specific user.
/etc/group	Contains the basic attributes of groups.
/etc/security/group	Contains the extended attributes of groups.

/etc/security/environ

The /etc/security/environ file is an ASCII file that contains stanzas with the environment attributes for users. Each stanza is identified by a user name and contains attributes in the Attribute=Value form with a comma separating the attributes. Each line is ended by a new-line character, and each stanza is ended by an additional new-line character. If environment attributes are not defined, the system uses default values.

The **mkuser** command creates a user stanza in this file. The initialization of the attributes depends upon their values in the /usr/lib/security/mkuser.default file. The **chuser** command can change these attributes, and the **lsuser** command can display them. The **rmuser** command removes the entire record for a user.

A basic /etc/security/environ file is shown in the following example, which has no environment attributes defined. Therefore, the system is using default values.

```
# pg /etc/security/environ
default:
root:
daemon:
bin:
sys:
adm:
uucp:
guest:
```

/etc/security/lastlog

The /etc/security/lastlog file is an ASCII file that contains stanzas with the last login attributes for users. Each stanza is identified by a user name and contains attributes in the Attribute=Value form. Each attribute is ended by a new-line character, and each stanza is ended by an additional new-line character. Two stanzas for users (root and test) are shown in the following example:

```
root:
        time_last_login = 1134081482
        tty_last_login = /dev/pts/6
```

```
        host_last_login = server2.itsc.austin.ibm.com
        unsuccessful_login_count = 0

test:
        time_last_unsuccessful_login = 1132588833
        tty_last_unsuccessful_login = ftp
        host_last_unsuccessful_login = workgroup
        unsuccessful_login_count = 0
        time_last_login = 1133188676
        tty_last_login = ftp
        host_last_login = win2kmodel
```

The **mkuser** command creates a user stanza in the lastlog file. The attributes of this user stanza are initially empty. The field values are set by the **login** command as a result of logging in to the system. The **lsuser** command displays the values of these attributes; the **rmuser** command removes the user stanza from this file along with the user account.

/etc/security/limits

The /etc/security/limits file is an ASCII file that contains stanzas that specify the process resource limits for each user. These limits are set by individual attributes within a stanza.

Each stanza is identified by a user name followed by a colon and contains attributes in the Attribute=Value form. Each attribute is ended by a new-line character, and each stanza is ended by an additional new-line character. If you do not define an attribute for a user, the system applies the default values.

The default attributes, and attributes for the user smith, are shown in the following example:

```
default:
        fsize = 2097151
        core = 2097151
        cpu = -1
        data = 262144
        rss = 65536
        stack = 65536
        nofiles = 2000

smith:
        fsize = 3007151
        data = 332144
        data_hard = 3400000
```

When you create a user with the **mkuser** command, the system adds a stanza for the user to the /etc/security/limits file. Once the stanza exists, you can use the **chuser** command to change the user's limits. Use a value of -1 to set a resource to unlimited. To display the current limits for a user, use the **lsuser** command. To remove users and their stanzas, use the **rmuser** command.

/etc/security/user

The /etc/security/user file contains extended user attributes. This is an ASCII file that contains stanzas for user attributes. The **mkuser** command creates a stanza in this file for each new user and initializes its attributes with the default attributes defined in the /usr/lib/security/mkuser.default file.

Each stanza in the /etc/security/user file is identified by a user name, followed by a colon (:), and contains comma-separated attributes in the Attribute=Value form. If an attribute is not defined for a user, either the default stanza or the default value for the attribute is used. A default stanza applies to all of the stanzas that follow, but does not apply to the stanzas preceding it.

Each attribute is ended by a new-line character, and each stanza is ended by an additional new-line character.

The **mkuser** command creates an entry for each new user in the /etc/security/user file and initializes its attributes with the attributes defined in the /usr/lib/security/mkuser.default file. To change attribute values, use the **chuser** command. To display the attributes and their values, use the **lsuser** command. To remove a user, use the **rmuser** command.

Password controls

The /etc/security/user file contains many attributes that allow you to control how users must manage their passwords. These attributes include:

histsize Defines the number of previous passwords a user cannot reuse. The value is a decimal integer string. The default is 0.

histexpire Defines the period of time (in weeks) that a user cannot reuse a password. The value is a decimal integer string. The default is 0, indicating that no time limit is set.

maxage Defines the maximum age (in weeks) of a password. The password must be changed by this time. The value is a decimal integer string. The default is a value of 0, indicating no maximum age.

maxexpired Defines the maximum time (in weeks) beyond the maxage value that a user can change an expired password. After this defined time, only an administrative user can change the password. The value is a decimal integer string. The default is -1, indicating no restriction is set. If the maxexpired attribute is 0, the password expires when the maxage value is met. If the maxage attribute is 0, the maxexpired attribute is ignored.

maxrepeats Defines the maximum number of times a character can be repeated in a new password. Since a value of 0 is meaningless, the default value of 8 indicates that there is no maximum number. The value is a decimal integer string.

minage Defines the minimum age (in weeks) a password must be before it can be changed. The value is a decimal integer string. The default is a value of 0, indicating no minimum age.

minalpha Defines the minimum number of alphabetic characters that must be in a new password. The value is a decimal integer string. The default is a value of 0, indicating no minimum number.

minother Defines the minimum number of non-alphabetic characters that must be in a new password. The value is a decimal integer string. The default is a value of 0, indicating no minimum number.

minlen Defines the minimum length of a password. The value is a decimal integer string. The default is a value of 0, indicating no minimum length. The maximum value allowed is 8. The value of minlen is determined by the minalpha value added to the minother value. If the result of this addition is greater than the minlen attribute, the value is set to the result.

mindiff Defines the minimum number of characters required in a new password that were not in the old password. The value is a decimal integer string. The default is a value of 0, indicating no minimum number.

/usr/lib/security/mkuser.default

The /usr/lib/security/mkuser.default file contains the default attributes for new users. It is an ASCII file that contains user stanzas. These stanzas have the default values of the attributes for the users created by the **`mkuser`** command. Each attribute has the Attribute=Value form. If an attribute has a value of $USER, the **`mkuser`** command substitutes the name of the user. The end of each attribute pair and stanza is marked by a new-line character.

There are two stanzas, user and admin, that can contain all defined attributes except the id and admin attributes. The **`mkuser`** command generates a unique ID attribute. The admin attribute depends on whether the -a flag is used with the **`mkuser`** command.

For a list of possible user attributes, look at the documentation of the **`chuser`** command (**`man chuser`**).

The following example shows a typical stanza in /usr/lib/security/mkuser.default:

```
# pg /usr/lib/security/mkuser.default
user:
        pgrp = staff
        groups = staff
        shell = /usr/bin/ksh
        home = /home/$USER

admin:
        pgrp = system
        groups = system
        shell = /usr/bin/ksh
        home = /home/$USER
```

/usr/lib/security/mkuser.sys

The /usr/lib/security/mkuser.sys is a shell script that customizes a new user account. The **mkuser** command calls **mkuser.sys** after it has created and initialized the new account. The script creates the home directory, the primary group, and a copy of the appropriate profile for the user's shell. This file can be customized to tailor the creation of new users.

/etc/passwd

The /etc/passwd file contains basic user attributes. This is an ASCII file that contains an entry for each user. Each entry defines the basic attributes applied to a user.

When you use the **mkuser** command to add a user to your system, the command updates the /etc/passwd file.

An entry in the /etc/passwd file has the following form with all attributes separated by a colon(:).

```
Name:Password:UserID:PrincipleGroup:Gecos:HomeDirectory:Shell
```

Password attributes can contain an asterisk (*), indicating an incorrect password or an exclamation point (!), indicating that the password is in the /etc/security/passwd file. Under normal conditions, the field contains an exclamation point (!). If the field has an asterisk (*) and a password is required for user authentication, the user cannot log in.

The shell attribute specifies the initial program or shell (login shell) that is started after a user invokes the **login** command or **su** command. The Korn shell is the default login shell and is backwardly compatible with the Bourne shell. If a user does not have a defined shell, /usr/bin/sh, the Bourne shell, is used. The Bourne shell is a subset of the Korn shell.

The **mkuser** command adds new entries to the /etc/passwd file and fills in the attribute values as defined in the /usr/lib/security/mkuser.default file. The Password attribute is always initialized to an asterisk (*), which is an invalid password. You can set the password with the **passwd** or **pwdadm** commands. When the password is changed, an exclamation point (!) is added to the /etc/passwd file, indicating that the encrypted password is in the /etc/security/passwd file.

Use the **chuser** command to change all user attributes except Password. The **chfn** command and the **chsh** command change the Gecos attribute and Shell attribute, respectively. To display all the attributes in this file, use the **lsuser** command. To remove a user and all the user's attributes, use the **rmuser** command.

The contents of /etc/passwd file in the following example shows that the Password attributes for two users (john and bob) are ! and *, respectively, which implies that user bob cannot login, as it has an invalid password:

```
# cat /etc/passwd
root:!:0:0::/:/usr/bin/ksh
daemon:!:1:1::/etc:
bin:!:2:2::/bin:
sys:!:3:3::/usr/sys:
adm:!:4:4::/var/adm:
uucp:!:5:5::/usr/lib/uucp:
guest:!:100:100::/home/guest:
nobody:!:4294967294:4294967294::/:
lpd:!:9:4294967294::/:
lp:*:11:11::/var/spool/lp:/bin/false
john:!:204:1::/home/john:/usr/bin/ksh
bob:*:205:1::/home/bob:/usr/bin/ksh
```

/etc/security/passwd

The /etc/security/passwd file is an ASCII file that contains stanzas with password information. Each stanza is identified by a user name followed by a colon (:) and contains attributes in the form Attribute=Value. Each attribute is ended with a new-line character, and each stanza is ended with an additional new-line character.

Although each user name must be in the /etc/passwd file, not all user names are necessarily listed in the /etc/security/passwd file. A user who has an invalid password (*) in the /etc/passwd file will have no entry in the /etc/security/passwd file. A typical file would have contents similar to that shown in the following example:

```
root:
        password = cHbMSxw6Ze2PM
        lastupdate = 1134082556
        flags =

daemon:
        password = *

bin:
        password = *

sys:
        password = *

adm:
        password = *
... (lines omitted) ...
```

/etc/security/login.cfg

The /etc/security/login.cfg file is an ASCII file that contains stanzas of configuration information for login and user authentication. Each stanza has a name, followed by a : (colon). Attributes are in the form Attribute=Value. Each attribute ends with a new-line character, and each stanza ends with an additional new-line character. There are two types of stanzas.

port stanza Defines the login characteristics of ports.

user configuration stanza
Defines programs that change user attributes (usw).

The following example shows the content of the /etc/security/login.cfg file:

```
default:
        sak_enabled = false
        logintimes =
        logindisable = 0
        logininterval = 0
        loginreenable = 0
        logindelay = 0
usw:
        shells =
/bin/sh,/bin/bsh,/bin/csh,/bin/ksh,/bin/tsh,/bin/ksh93,/usr/bin
```

```
/sh,/usr/bin/bsh,/usr/bin/csh,/usr/bin/ksh,/usr/bin/tsh,/usr/bin/ksh93,/usr/bin
/
rksh,/usr/bin/rksh93,/usr/sbin/uucp/uucico,/usr/sbin/sliplogin,/usr/sbin/snappd
        maxlogins = 32767
        logintimeout = 60
        auth_type = STD_AUTH
```

/etc/utmp, /var/adm/wtmp, and /etc/security/failedlogin

The utmp file, the wtmp file, and the failedlogin file contain records with user and accounting information. When a user successfully logs in, the login program writes entries in two files:

- The /etc/utmp file, which contains a record of users logged into the system. The command **who -a** processes the /etc/utmp file, and if this file is corrupted or missing, no output is generated from the **who** command.
- The /var/adm/wtmp file (if it exists), which contains connect-time accounting records.

On an invalid login attempt, due to an incorrect login name or password, the login program makes an entry in the /etc/security/failedlogin file, which contains a record of unsuccessful login attempts.

/etc/motd

The /etc/motd file is an ASCII file and contains the message of the day. It is displayed every time a user logs in to the system. It is a convenient way for the system administrator to communicate to all users information about such items as system shutdown, installed software version numbers, or current system news. To change the message of the day, simply edit that file. It is usually writable only by root.

A typical /etc/motd file looks like:

```
*******************************************************************************
*                                                                             *
*                                                                             *
*  Welcome to AIX Version 5.3!                                                *
*                                                                             *
*                                                                             *
*  Please see the README file in /usr/lpp/bos for information pertinent to    *
*  this release of the AIX Operating System.                                  *
*                                                                             *
*                                                                             *
*******************************************************************************
```

/etc/environment

The /etc/environment file contains variables specifying the basic environment for all processes. When a new process begins, the exec subroutine makes an array of strings available that have the form Name=Value. This array of strings is called the environment. Each name defined by one of the strings is called an environment variable or shell variable. Environment variables are examined when a command starts running.

The /etc/environment file is not a shell script. It should only contain data in the Name=Value format, and should not contain shell commands. Trying to run commands from this file may cause failure of the initialization process.

When you log in, the system sets environment variables from the environment file before reading your login profile, .profile. The following are a few variables that make up part of the basic environment:

HOME The full path name of the user login or HOME directory. The login program sets this to the directory specified in the /etc/passwd file.

LANG The locale name currently in effect. The LANG variable is set in the /etc/environment file at installation time.

NLSPATH The full path name for message catalogs.

PATH The sequence of directories that commands such as **sh**, **time**, **nice**, and **nohup** search when looking for a command whose path name is incomplete. The directory names are separated by colons.

LPDEST The printer to use when a print-related command does not specify a destination printer.

TERM The terminal type.

EDITOR The default editor to be used by various commands that perform editing functions, such as **crontab**.

TZ The time zone information.

Attention: Changing the time zone only affects processes that begin after the change is made. The init process only reads /etc/environment at startup; therefore, init and its child processes will not be aware of a change to TZ until the system is rebooted.

/etc/profile and $HOME/.profile

The /etc/profile file is the first file that the operating system uses at login time. It controls system-wide default variables, such as:

- Export variables
- File creation mask (umask)

- Terminal types
- Mail messages to indicate when new mail has arrived

The system administrator configures the file for all users on the system. Only the system administrator can change this file.

Commands and variables to be included in /etc/profile should be appropriate for *all* users of the system. An example of a command that you may want all users to run when they log in is the **mail** command.

The $HOME/.profile file is the second file that the operating system uses at login time. It enables you to customize your individual working environment. The .profile file overrides commands and variables set in the /etc/profile file. Use it to control personal settings such as:

- Shells to open
- Environment variables (for example, search path variables)
- Default editor
- Default printer
- Prompt appearance
- Keyboard sound

11.2 User administration tasks

User administration creates users, defines or changes their attributes, and defines the security environment for the users. These topics are discussed in the following sections.

11.2.1 Adding a new user account

The **mkuser** command creates a new user account. It takes a user name as a parameter that must be a unique string (whose length is administrator-configurable via the **chdev** command up to a maximum of 255 characters). By default, the **mkuser** command creates a standard user account. To create an administrative user account, specify the -a flag.

The **mkuser** command does not create password information for a user. It initializes the user's Password attribute in the /etc/passwd file with an * (asterisk). This disables the new account until the **passwd** command is used to add authentication information to the /etc/security/passwd file for this account.

- To create the smith account with smith as an administrator, enter:

  `# mkuser -a smith`

  You must be the root user to create smith as an administrative user.

- To create the smith user account and set the **su** attribute to a value of false, enter:

  `# mkuser su=false smith`

- To create a user account, smith, with the default values in the /usr/lib/security/mkuser.default file, enter:

  `# mkuser smith`

  Alternatively, you can use SMIT:

  a. Run **smitty mkuser** to access the menu shown in Figure 11-1.

  b. Type `smith` for the field User NAME.

  c. Press the Enter key to create the user.

  d. When SMIT returns an OK prompt, press the F10 key to return to the command prompt.

.

```
                                 Add a User

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[TOP]                                                  [Entry Fields]
* User NAME                                           [smith]
  User ID                                             []                     #
  ADMINISTRATIVE USER?                                 false                +
  Primary GROUP                                       []                    +
  Group SET                                           []                    +
  ADMINISTRATIVE GROUPS                               []                    +
  ROLES                                               []                    +
  Another user can SU TO USER?                         true                 +
  SU GROUPS                                           [ALL]                 +
  HOME directory                                      []
  Initial PROGRAM                                     []
  User INFORMATION                                    []
  EXPIRATION date (MMDDhhmmyy)                        [0]
[MORE...37]

F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 11-1 Adding a user with SMIT

11.2.2 Creating or changing a user password

The **passwd** command is run to create or change a user's password. It creates an encrypted passwd entry in /etc/security/passwd file and changes the user's Password attribute in the /etc/passwd file from * to ! (exclamation).

- To change the full name of user smith in the /etc/passwd file, enter:

  ```
  # passwd -f smith
  ```

  The **passwd** command displays the name stored in the Gecos attribute of the /etc/passwd file for the user smith. For example, the **passwd** command could display the message shown in the following example.

  ```
  # passwd -f smith
   smith's current gecos:
                  "Mr J.Smith"
   Change (yes) or (no)? > n
   Gecos information not changed.
  ```

  If you enter a Y or yes, the **passwd** command prompts you for the new name. The **passwd** command records the name you enter in the Gecos field of the /etc/passwd file.

- To change your password, enter:

  ```
  # passwd
  ```

  The **passwd** command prompts you for your old password. If it exists, and after you enter the old password, the command prompts you twice for the new password. If the old password does not exist, the command does not prompt you for the old password

  Alternatively, you can use SMIT:

  a. Running **smitty passwd** will prompt you with the menu shown in Figure 11-2 on page 506.

  b. Type `smith` for the field User NAME.

  c. Press Enter, and you will be prompted to enter the new password (twice) (user smith does not have a password), as shown in Figure 11-3 on page 506.

  d. Enter the new password and press the Enter key.

  e. When SMIT returns an OK prompt, press the F10 key to return to the command prompt.

```
                          Change a User's Password

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                      [Entry Fields]
  User NAME                                          [smith]              +









F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 11-2 Changing a user password

```
Changing password for "smith"
smith's New password:
Enter the new password again:
```

Figure 11-3 Entering a user password

11.2.3 Changing user attributes

The **chuser** command changes attributes of a user. The user name must already exist.

> **Note:** Do not use the **chuser** command if you have a Network Information Service (NIS) database installed on your system.

Only the root user or users with UserAdmin authorization can use the **chuser** command to perform the following tasks:

- Make a user an administrative user by setting the admin attribute to true.
- Change any attributes of an administrative user.
- Add a user to an administrative group.

The following examples show the use of the **chuser** command with various flags:

- To enable user smith to access this system remotely, enter:

  ```
  # chuser rlogin=true smith
  ```

- To change the expiration date for the smith user account to 8 a.m., 1 December, 1998, enter:

  ```
  #chuser expires=1201080098 smith
  ```

- To add smith to the group program, enter:

  ```
  #chuser groups=program smith
  ```

  Alternatively, you can go through the SMIT hierarchy:

  a. Running **smitty chuser** will prompt you with the menu shown in Figure 11-4 on page 508.

  b. Type `smith` for the field User NAME.

  c. Use the Arrows key to highlight the Primary GROUP field and type program in it.

  d. Press Enter.

  e. When SMIT returns an OK prompt, press the F10 key to return to the command prompt.

```
                     Change / Show Characteristics of a User

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[TOP]                                                  [Entry Fields]
  User NAME                                           smith
  User ID                                            [218]              #
  ADMINISTRATIVE USER?                                false             +
  Primary GROUP                                      [program]          +
  Group SET                                          [staff]            +
  ADMINISTRATIVE GROUPS                              []                 +
  ROLES                                              []                 +
  Another user can SU TO USER?                        true              +
  SU GROUPS                                          [ALL]              +
  HOME directory                                     [/home/smith]
  Initial PROGRAM                                    [/usr/bin/ksh]
  User INFORMATION                                   []
  EXPIRATION date (MMDDhhmmyy)                       [0]
[MORE...37]

F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 11-4 Changing user characteristics

11.2.4 Displaying user attributes

The **lsuser** command displays the user account attributes. You can use this command to list all attributes of all the users or all the attributes of specific users except their passwords. Since there is no default parameter, you must enter the ALL keyword to see the attributes of all the users. By default, the **lsuser** command displays all user attributes. To view selected attributes, use the -a List flag. If one or more attributes cannot be read, the **lsuser** command lists as much information as possible.

> **Note:** If you have a Network Information Service (NIS) database installed on your system, some user information may not appear when you use the **lsuser** command.

By default, the **lsuser** command lists each user's attributes on one line. It displays attribute information as Attribute=Value definitions, each separated by a blank space. To list the user attributes in stanza format, use the -f flag. To list the information as colon-separated records, use the -c flag.

The following examples show the use of the **lsuser** command with various flags.

- To display the user ID and group-related information for the root account in stanza form, enter:

```
# lsuser -f -a id pgrp home root
root:
        id=0
        pgrp=system
        home=/
```

- To display the user ID, groups, and home directory of user smith in colon format, enter:

```
# lsuser -c -a id home groups smith
```

- To display all the attributes of user smith in the default format, enter:

```
# lsuser smith
```

  All attribute information appears with each attribute separated by a blank space.

- To display all the attributes of all the users, enter:

```
# lsuser ALL
```

  For each user, all attribute information appears with each attribute separated by a blank space.

- Alternatively, you can use SMIT:

  a. Run **smitty users**, which will prompt you with the menu shown in Figure 11-5 on page 510.

  b. Type smith for the field User NAME and press Enter. This will display the screen shown in Figure 11-6 on page 510.

  c. When SMIT returns an OK prompt, press the F10 key to return to the command prompt.

```
                                    Users

Move cursor to desired item and press Enter.

  Add a User
  Change a User's Password
  Change / Show Characteristics of a User
  Lock / Unlock a User's Account
  Reset User's Failed Login Count
  Remove a User
  List All Users









F1=Help             F2=Refresh          F3=Cancel           F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 11-5 SMIT users command

```
                    Change / Show Characteristics of a User

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[TOP]                                                   [Entry Fields]
* User NAME                                             smith
  User ID                                              [218]              #
  ADMINISTRATIVE USER?                                  false             +
  Primary GROUP                                        [staff]            +
  Group SET                                            [staff]            +
  ADMINISTRATIVE GROUPS                                []                 +
  ROLES                                                []                 +
  Another user can SU TO USER?                          true              +
  SU GROUPS                                            [ALL]              +
  HOME directory                                       [/home/smith]
  Initial PROGRAM                                      [/usr/bin/ksh]
  User INFORMATION                                     []
  EXPIRATION date (MMDDhhmmyy)                         [0]
[MORE...37]

F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 11-6 Listing user characteristics

11.2.5 Removing a user account

The **rmuser** command removes a user account. This command removes a user's attributes without removing the user's home directory and files. The user name must already exist. If the -p flag is specified, the **rmuser** command also removes passwords and other user authentication information from the /etc/security/passwd file.

Only the root user or a user with UserAdmin authorization can remove administrative users.

- The following example shows the use of the **rmuser** command to remove a user account smith and its attributes from the local system:

  ```
  # rmuser smith
  ```

- To remove the user smith account and all its attributes, including passwords and other user authentication information in the /etc/security/passwd file, use the following command:

  ```
  # rmuser -p smith
  ```

  Alternatively, you can use the SMIT hierarchy:

  a. Running **smitty rmuser** will prompt you with the menu shown in Figure 11-7 on page 512.

  b. Type `smith` for the field User NAME.

  c. Press the Enter key.

  d. When SMIT returns an OK prompt, press the F10 key to return to the command prompt.

```
                          Remove a User from the System

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                     [Entry Fields]
* User NAME                                         [smith ]              +
  Remove AUTHENTICATION information?                 yes                  +










F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 11-7 Removing a user

11.2.6 Changing security attributes of user

The **chsec** command changes the attributes stored in the security configuration stanza files. It has the following syntax:

```
chsec [ -fFile] [ -s Stanza] [ -a Attribute = Value ... ]
```

The following security configuration stanza files have attributes that you can specify with the Attribute = Value parameter:

- /etc/security/environ
- /etc/security/group
- /etc/security/lastlog
- /etc/security/limits
- /etc/security/login.cfg
- /usr/lib/security/mkuser.default
- /etc/security/passwd
- /etc/security/portlog
- /etc/security/user

When modifying attributes in the /etc/security/environ, /etc/security/lastlog, /etc/security/limits, /etc/security/passwd, and /etc/security/user files, the stanza name specified by the Stanza parameter must either be a valid user name or default.

When modifying attributes in the /etc/security/group file, the stanza name specified by the Stanza parameter must either be a valid group name or default.

When modifying attributes in the /usr/lib/security/mkuser.default file, the Stanza parameter must be either admin or user.

When modifying attributes in the /etc/security/portlog file, the Stanza parameter must be a valid port name. When modifying attributes in the /etc/security/login.cfg file, the Stanza parameter must either be a valid port name, a method name, or the usw attribute.

When modifying attributes in the /etc/security/login.cfg or /etc/security/portlog files in a stanza that does not already exist, the stanza is automatically created by the **chsec** command.

> **Note:** You cannot modify the password attribute of the /etc/security/passwd file using the **chsec** command. Instead, use the **passwd** command.

The following examples show the usage of **chsec** command to change security stanzas in various files:

- To change the /dev/tty0 port to automatically lock if five unsuccessful login attempts occur within 60 seconds, enter:

```
# chsec -f /etc/security/login.cfg -s /dev/tty0 -a logindisable=5 -a
logininterval=60
```

- To unlock the /dev/tty0 port after it has been locked by the system, enter:

```
# chsec -f /etc/security/portlog -s /dev/tty0 -a locktime=0
```

- To allow logins from 8:00 a.m. until 5:00 p.m. for all users, enter:

```
# chsec -f /etc/security/user -s default -a logintimes=:0800-1700
```

- To change the CPU time limit of user smith to one hour (3600 seconds), enter:

```
# chsec -f /etc/security/limits -s smith -a cpu=3600
```

11.2.7 Displaying currently logged users

The **who** command displays information about all users currently on the local system. The following information is displayed: login name, TTY, and the date and time of login. Entering **who am i** or **who am I** displays your login name, TTY, and the date and time you logged in. If the user is logged in from a remote machine, then the host name of that machine is displayed as well. The **who** command can also display the elapsed time since line activity occurred, the process ID of the command interpreter (shell), logins, log offs, restarts, and changes to the system clock, as well as other processes generated by the initialization process.

> **Note:** The /etc/utmp file contains a record of users logged into the system. The command **who -a** processes the /etc/utmp file, and if this file is corrupted or missing, no output is generated from the **who** command.

The following examples show the usage of the **who** command with various flags.

- The following example shows the command to display information about all the users who are logged on to the system:

```
# who
root        pts/0       Nov 30 18:38     (ibm-dn1ao2berub)
test1       pts/1       Dec 01 18:15     (tlm06.itsc.austin.ibm.com)
root        pts/2       Dec 01 19:41     (moller.austin.ibm.com)
root        pts/3       Dec 01 15:46     (win2kmodel)
root        pts/4       Dec 01 17:19     (esmsrv.itsc.austin.ibm.com)
root        pts/5       Nov 30 20:09     (moller.austin.ibm.com)
#
```

- The following example shows the command to display information about your user name:

```
# who am i
root        pts/2       Dec 01 19:41     (moller.austin.ibm.com)
#
```

- The following shows how to display the run-level of the local system node:

```
# who -r
   .        run-level 2 Nov 17 10:19      2    0    S
```

  A system's run level determines what services and resources are available to the users of the system. The number 2 in both places indicates that the system is in multiuser mode. The number 0 indicates that the system has been at this run level 0 number of times since last reboot and S indicates the previous run level.

- The following shows how to display any active process that was spawned by init:

```
# who -p
srcmstr         .       Nov 30 18:34     4:49    131284 id=srcmstr
cron            .       Nov 30 18:35     4:49    200816 id=cron
uprintfd        .       Nov 30 18:35     4:49    237716 id=uprintf
#
```

The entries in the first column are the User IDs of the processes. The numbers in the next to the last column are the Process IDs of the running processes, which are in the last column.

In the event that the /etc/utmp file becomes corrupt or lost, you can use the **ps** command to list processes and their associated users.

11.2.8 Preventing user logins

If the /etc/nologin file exists, the system accepts the user's name and password, but prevents the user from logging in and displays the contents of the /etc/nologin file. However, the system does allow the root user to log in. The /etc/nologin file is removed when you reboot the system. You can also allow users to log in again by deleting the file.

11.2.9 Changing a user's login shell

The **chsh** command changes a user's login shell attribute. The shell attribute defines the initial program that runs after a user logs in to the system. This attribute is specified in the /etc/passwd file. By default, the **chsh** command changes the login shell for the user who gives the command.

The **chsh** command is interactive. When you run the **chsh** command, the system displays a list of the available shells and the current value of the shell attribute, as shown in Figure 11-8 on page 516. In addition to the default shells (/usr/bin/ksh, /usr/bin/sh, /usr/bin/bsh and /usr/bin/csh), your system manager may have defined more. Then the system prompts you to change the shell. You must enter the full path name of an available shell.

If you have execute permission for the **chuser** command, you can change the login shell for another user.

```
# chsh
 Current available shells:
                /bin/sh
                /bin/bsh
                /bin/csh
                /bin/ksh
                /bin/tsh
                /usr/bin/sh
                /usr/bin/bsh
                /usr/bin/csh
                /usr/bin/ksh
                /usr/bin/tsh
                /usr/sbin/sliplogin
 root´s current login shell:
                /bin/ksh
 Change (yes) or (no)? > yes
 To?>/bin/csh
#
```

Figure 11-8 chsh command

11.2.10 Changing the shell prompt

The shell uses the following three prompt variables:

PS1 This variable determines your primary prompt.

PS2 The shell displays this variable via standard error after you have pressed Enter and thus started a new line, without having entered a complete command. This is called a secondary prompt.

PS3 The shell displays this variable via standard error to prompt you to select one of the choices that you specified with the `select` compound command. This is called the Select Command Prompt.

PS4 The shell displays this variable via standard error when the shell is ready to display a command during execution trace. This is called the Debug Prompt.

You can change any of your prompt characters by changing the value of its shell variable. The changes to your prompts last until you log off. To make your changes permanent, place them in your .env file.

- The following command shows how to display the current value of the PS1 variable:

```
# echo "prompt is $PS1"
prompt is $
```

- The following example shows the command to change the prompt to `Ready>`:

```
# export PS1="Ready> "
# echo "prompt is $PS1"
prompt is Ready>
```

- The following example shows the command to change the continuation prompt to `Enter more->`:

```
# export PS2="Enter more->"
# print Tod\
Enter more->ay is Tuesday
Today is Tuesday
```

- The following example shows the command to change the Select Command prompt to `Please enter a number`:

```
# export PS3="Please enter a number: "
# select i in foo bar1 bar2 bar3
> do command
> done
1> foo
2> bar1
3> bar2
4> bar4
Please enter a number:
```

- The following example shows the command to change the Debug prompt to line number +:

```
# export PS4='[$LINENO] + '
# set -x
# print $HOME
[1] + print /home/jim
/home/jim
```

11.3 Common login errors

The following section summarizes a few of the login Error Messages and their possible causes. Refer to the AIX 5L product documentation for more information.

3004-004 `You must "exec" login from the lowest login shell`. You attempted to log off the system while processes are still running in another shell.

3004-007	`You entered an invalid login name or password.` You tried to log in to a system that does not recognize your login or password.
3004-008	`Failed setting credentials.` Login failed.
3004-009	`Failed running login shell.` You tried to log in to a system that has a damaged login shell. The login shell does not exist.
3004-030	`You logged in using all uppercase characters.` You attempted to log in with Caps Lock on.
3004-031	`Password read timed out--possible noise on port.` You logged in but did not enter your password within a specified amount of time. Your password was not validated within a specified amount of time due to a failed network connection.
3004-302	`Your account has expired. Please see the system administrator.` Your password has expired.
3004-312	`All available login sessions are in use.` You tried to log in to a system that had all present sessions in use.
3004-687	`User does not exist.` You specified an invalid user name with the **lsuser**, **chuser**, **rmuser**, or **passwd** command.

11.4 Monitoring and managing processes

The following commands are used to examine specific system areas, either to supplement the results from **vmstat**, **iostat**, and **netstat**, or to provide more in-depth information.

11.4.1 Using the ps command

The **ps** command writes the current status of active processes and (if the -m flag is given) associated kernel threads to standard output. While the -m flag displays threads associated with processes using extra lines, you must use the -o flag with the THREAD field specifier to display extra thread-related columns.

The following is the syntax of the **ps** command:

```
ps [ -A ] [ -M ] [ -N ] [ -a ] [ -d ] [ -e ] [ -f ] [ -k ] [ -l ] [ -F format]
[ -o Format ] [ -c Clist ] [ -G Glist ] [ -g Glist ] [ -m ] [ -n NameList ] [
-p Plist ] [ -P ] [ -t Tlist ] [ -U Ulist ] [ -u Ulist ] [ -T pid ] [ -L
pidlist ] [ -X ]
```

Without flags, the **ps** command displays information about the current terminal. The -f, -o, and -l flags only determine how much information is provided about a process; they do not determine which processes are listed.

With the -o flag, the **ps** command examines memory or the paging area and determines what the command name and parameters were when the process was created. If the **ps** command cannot find this information, the command name stored in the kernel is displayed in square brackets.

- To display all processes, type the following at the login prompt:

```
# ps -e -f
```

- To list processes owned by specific users, type the following at the login prompt:

```
# ps -f -l -ujim,jane,su
```

- To display information about all processes and kernel threads, type the following at the login prompt:

```
# ps -emo THREAD
The output is similar to:
USER   PID  PPID   TID S  C PRI SC   WCHAN   FLAG   TTY BND  CMD
jane  1716 19292    - A 10  60  1       * 260801 pts/7   -  biod
   -     -     - 4863 S  0  60  0 599e9d8   8400     -   -  -
   -     -     - 5537 R 10  60  1 5999e18   2420     -   3  -
luke 19292 18524    - A  0  60  0 586ad84 200001 pts/7   -  -ksh
   -     -     - 7617 S  0  60  0 586ad84    400     -   -  -
luke 25864 31168    - A 11  65  0       - 200001 pts/7   -  -
   -     -     - 8993 R 11  65  0       -      0     -   -  -
```

The headings have the following meaning:

USER The login name of the process owner.

PID The process ID of the process.

PPID The process ID of the parent process.

TID The thread ID of the kernel thread.

S The state of the process or kernel thread:

A Active

R Running

S Sleeping

C CPU utilization of process or thread.

PRI The priority of the process or kernel thread.

SC The suspend count of the process or kernel thread.

WCHAN The event for which the process or kernel thread is waiting or sleeping.

TTY The controlling workstation for the process:

- **-** The process is not associated with a workstation.
- **?** Unknown.
- **Number** The TTY number. For example, the entry 2 indicates TTY2.

BND The logical processor number of the processor to which the kernel thread is bound (if any).

CMD Contains the command name.

- To list all the 64-bit processes, type:

```
# ps -M
```

11.4.2 Using the kill command

The **kill** command sends a signal (by default, the SIGTERM signal) to a running process. This default action normally stops processes.

The following is the syntax of the **kill** command:

```
kill [  -s { SignalName | SignalNumber } ] ProcessID ...
```

If you want to stop a process, specify the process ID (PID) in the ProcessID variable. A root user can stop any process with the **kill** command. If you are not a root user, you must have initiated the process you want to stop.

SignalName is recognized in a case-independent fashion, without the SIG prefix.

If the specified SignalNumber is 0, the **kill** command checks the validity of the specified PID.

The signal names are listed in the /usr/include/sys/signal.h file.

- To stop a given process, enter:

```
# kill 1095
```

This stops process 1095 by sending it the default SIGTERM signal. Note that process 1095 might not actually stop if you have made special arrangements to ignore or override the SIGTERM signal.

- To stop several processes that ignore the default signal, enter:

```
# kill -kill 2098 1569
```

This sends signal 9, the SIGKILL signal, to processes 2098 and 1569. The SIGKILL signal is a special signal that normally cannot be ignored or overridden.

- To stop all of your processes and log yourself off, enter:

  ```
  # kill -kill 0
  ```

  This sends signal 9, the SIGKILL signal, to all processes having a process group ID equal to the senders process group ID. Because the shell cannot ignore the SIGKILL signal, this also stops the login shell and logs you off.

- To stop all processes that you own, enter:

  ```
  #kill -9 -1
  ```

  This sends signal 9, the SIGKILL signal, to all processes owned by the effective user, even those started at other work stations and that belong to other process groups.

- To send a different signal code to a process, enter:

  ```
  # kill  -USR1  1103
  ```

  The name of the **kill** command is misleading because many signals, including SIGUSR1, do not stop processes. The action taken on SIGUSR1 is defined by the particular application you are running.

Note: To send signal 15, the SIGTERM signal with this form of the **kill** command, you must explicitly specify -15 or TERM.

11.4.3 Using the nice and renice commands

The **nice** and **renice** commands are used to change the priority of a process. The **nice** command runs another command at a different priority, while the **renice** command changes the priority of an already running process. The root user can increase or decrease the priority of any process. Other users can only decrease the priority of processes they own.

The following syntax is used by the **nice** command:

```
nice [  - Increment|  -n Increment ] Command [ Argument ... ]
```

The Command parameter is the name of any executable file on the system. If you do not specify an Increment value, the **nice** command defaults to an increment of 10. The higher the **nice** value, the lower the priority.

The **nice** value can range from 0 to 39, with 39 being the lowest priority. For example, if a command normally runs at a **nice** value of 20 (default value), specifying an increment of 5 runs the command at a **nice** value of 25, and the command runs slower. The **nice** command does not return an error message if you attempt to increase a command's priority without the appropriate authority. Instead, the command's priority is not changed, and the system starts the command as it normally would.

Users with root authority can specify a negative increment that would increase the priority of the process.

The following example shows the **nice** command running the **cc** command at a lower priority:

```
# nice -n 15 cc -c *.c
```

The syntax for the **renice** command is the following:

```
renice [  -n Increment ] [  -g |  -p |  -u ] ID ...
```

The parameter Increment specifies the number to add to the nice value of the process. The value of Increment can only be a decimal integer from -20 to 20. Positive increment values lead to a lower scheduling priority. Negative increment values require root privileges and lead to a higher priority.

The specified Increment changes the priority of a process in the following ways:

1 to 20 Runs the specified processes slower than the base priority.

0 Sets priority of the specified processes to the base scheduling priority.

-20 to -1 Runs the specified processes quicker than the base priority.

The flags have the following meaning:

-g Interprets all IDs as unsigned decimal integer process group IDs.

-p Interprets all IDs as unsigned integer process IDs. The -p flag is the default if you specify no other flags.

-u Interprets all IDs as user name or numerical user IDs.

- To alter the system scheduling priority so that process IDs 987 and 32 have lower scheduling priorities, enter:

  ```
  # renice -n 5 -p 987 32
  ```

- To alter the system scheduling priority so that group IDs 324 and 76 have higher scheduling priorities (if the user has the root privileges to do so), enter:

  ```
  # renice -n -4 -g 324 76
  ```

11.4.4 Using the fuser command

The **fuser** command lists the process numbers of local processes that use the local or remote files specified by the File parameter, as shown in the following syntax:

```
fuser [ -c | -d | -f ] [ -k | -K { SignalNumber | SignalName }] [ -u ] [ -x ] [
-V ]File ...
```

For block special devices, the command lists the processes that use any file on that device:

- To list the process numbers and user login names of processes using the /etc/filesystems file, enter:

  ```
  fuser -u /etc/filesystems
  ```

- To terminate all of the processes using a given file system, enter:

  ```
  fuser -k -x -u -c /dev/hd1
  or
  fuser -kxuc /home
  ```

  Either command lists the process number and user name, and then terminates each process that is using the /dev/hd1 (/home) file system. Only the root user can terminate processes that belong to another user. You might want to use this command if you are trying to unmount the /dev/hd1 file system and a process that is accessing the /dev/hd1 file system prevents this.

- To list all processes that are using a file that has been deleted from a given file system, enter:

  ```
  fuser -d /usr
  ```

- The **fuser** command is often the best way to determine what is still active in the file system. The **fuser** command will return the process IDs for all processes that have open references within a specified file system, as shown in the following example:

  ```
  # fuser -xc /tmp
  /tmp: 2910 3466 11654 26400
  ```

  The process having an open reference can be killed by using the **kill** command, and the unmount can be accomplished.

- If the file system is still busy and cannot be unmounted, this could be due to a kernel extension that is loaded but exists within the source file system. The **fuser** command will not show these kinds of references since a user process is not involved.
- A process is using a directory within the file system as its current working directory. The **fuser** command appends the letter “c” to the process IDs of all processes that are using a directory as their current working directory, and the -u flag identifies the owner of the process. It can be used with the **find** command, as shown in the following example:

```
# find /home -type d -exec fuser -u {} \;
/home:
/home/lost+found:
/home/guest:
/home/kenzie:     3548c(kenzie)
```

11.4.5 Using the topas command

The **topas** command is a performance monitoring tool that was introduced in AIX Version 4.3.3. In AIX 5L Version 5.3, it reports on CPU usage, NFS statistics, and per disk or adapter breakdown of network and disk usage. Figure 11-9 provides a sample **topas** command screen.

```
Topas Monitor for host:    server1               EVENTS/QUEUES    FILE/TTY
Fri Aug 31 11:52:48 2001   Interval:  2          Cswitch      35  Readch        78
                                                 Syscall      76  Writech     3252
Kernel     0.1   |                            |  Reads         1  Rawin          0
User       0.0   |                            |  Writes        3  Ttyout         0
Wait       0.0   |                            |  Forks         0  Igets          0
Idle      99.8   |############################|  Execs         0  Namei          0
                                                 Runqueue    0.0  Dirblk         0
Network   KBPS   I-Pack  O-Pack   KB-In  KB-Out  Waitqueue   0.0
tr0        3.5      3.0     3.0     0.2     3.3
lo0        0.0      0.0     0.0     0.0     0.0  PAGING           MEMORY
                                                 Faults        0  Real,MB      511
Disk     Busy%     KBPS     TPS KB-Read KB-Writ  Steals        0  % Comp      24.5
hdisk0     0.0      0.0     0.0     0.0     0.0  PgspIn        0  % Noncomp   19.8
hdisk1     0.0      0.0     0.0     0.0     0.0  PgspOut       0  % Client     0.5
                                                 PageIn        0
Name          PID CPU% PgSp Owner                PageOut       0  PAGING SPACE
topas       13284  0.1  0.8 root                 Sios          0  Size,MB      512
xterm       15076  0.0  0.9 root                                  % Used       1.2
dtexec      22776  0.0  0.7 root                 NFS (calls/sec)  % Free      98.7
dtscreen    21362  0.0  0.6 root                 ServerV2       0
syncd        5956  0.0  0.3 root                 ClientV2       0   Press:
X            5454  0.0  3.1 root                 ServerV3       0   "h" for help
dtsession    5198  0.0  2.7 root                 ClientV3       0   "q" to quit
```

Figure 11-9 topas command output

The general syntax of the **topas** command is as follows:

```
topas [ -d number_of_monitored_hot_disks ] [ -h ] [ -i
monitoring_interval_in_seconds ] [ -n
number_of_monitored_hot_network_interfaces ] [ -p
number_of_monitored_hot_processes ] [ -w number_of_monitored_hot_WLM classes ]
[ -c number_of_monitored_hot_CPUs ][ -I remote_polling_interval ] [ -U
username_owned_processes ] | [ -C [ -o field = value,... ] -D | -L |-P | -W ] [
-m]
```

Table 11-1 provides a list of some of the flags for **topas** and their descriptions.

Table 11-1 Commonly used flags for the topas command

Flag	Description
-d *NumberOfHotDisks*	Specifies the maximum number of disks shown. If this argument is omitted, a default of 5 is assumed. If a value of zero is specified, no disk information will be displayed.
-h	Displays help information.
-i	Sets the monitoring interval in seconds. The default is two seconds.
-n	Specifies the maximum number of network interfaces shown. If this argument is omitted, a default of 2 is assumed. If a value of zero is specified, no network information will be displayed.
-p	Specifies the maximum number of processes shown. If this argument is omitted, a default of 16 is assumed. If a value of zero is specified, no process information will be displayed. Retrieval of process information constitutes the majority of the **topas** overhead. If process information is not required, you should always use this option to specify that you do not want process information.

For example, to view the top 10 processes in use while not displaying any network interface statistics, in five second intervals, run the command:

```
# topas -i5 -n0 -p10
```

Within the **topas** command screen, there are a variety of subcommands that can change the output of what is being displayed. Table 11-2 has a list of subcommands you can use in the **topas** command screen.

Table 11-2 topas command screen subcommands

Subcommand	Description
a	Show all the variable subsections being monitored. Pressing the "a" key always returns **topas** to the main initial display.
c	Pressing the "c" key repeatedly toggles the CPU subsection between the cumulative report, off, and a list of busiest CPUs.
d	Pressing the "d" key repeatedly toggles the disk subsection between busiest disks list, off, and total disk activity for the system.
f	Moving the cursor over a WLM class and pressing "f" shows the list of top processes in the class on the bottom of the screen (WLM Display Only).
h	Toggles between the help screen and main display.
n	Pressing the "n" key repeatedly toggles the network interfaces subsection between busiest interfaces list, off, and total network activity.
p	Pressing the "p" key toggles the hot processes subsection on and off.
q	Quits the program.
r	Refreshes the screen.
w	Pressing the "w" key toggles the Workload Management (WLM) classes subsection on and off.
W	Toggle to the Full Screen WLM Class Display.

11.4.6 Using the svmon command

The **svmon** command captures and analyzes a snapshot of virtual memory. It displays information about the current state of memory. The displayed information does not constitute a true snapshot of memory, because the **svmon** command runs at user level with interrupts enabled.

The **svmon** command creates nine types of reports:

1. global
2. user

3. command
4. class
5. tier
6. process
7. segment
8. detailed segment
9. frame

The **svmon** command uses a slightly different syntax to generate these reports as shown in the following:

- Global Report

  ```
  svmon -G [ -i Interval [ NumIntervals ] ] [ -z ]
  ```

- User Report

  ```
  svmon -U [ LogName1...LogNameN ] [ -r ] [ -n | -s ] [ -w | -f -c ] [ -t
  Count ] [ -u | -p | -g | -v ] [ -i Interval [ NumIntervals ] ] [ -l ]
  [ -j ] [ -d ]
  [ -z ] [ -m ]
  ```

- Command Report

  ```
  svmon -C Command1...CommandN [ -r ] [ -n | -s ] [ -w | -f | -c ] [-t Count
  ]
  [ -u | -p | -g | -v ] [ -i Interval [ NumIntervals] ] [ -l ] [ -j ] [ -d ]
  [ -z ] [ -m ] [ -q [ s | L ] ]
  ```

- Workload Management Class Report

  ```
  svmon -W [ ClassName1...ClassNameN ] [ -e ] [ -r ] [ -n | -s ] [ -w | -f |
  -c ] [-t Count ] [ -u | -p | -g | -v ] [ -i Interval [ NumIntervals]]
  [ -l ] [ -d ] [ -z ] [ -m ] [ -q [ s | L ] ]
  ```

- Workload Management Tier Report

  ```
  svmon -T [ Tier1...TierN ] [ -a SupClassName ] [ -x ] [ -e ] [ -r ] [ -u |
  -p | -g | -v ] [ -n | -s ] [ -w | -f | -c ] [ -q [ s | L ] [ -t Count ]
  [ -i Interval [ NumIntervals ] ] [ -l ] [ -z ] [ -m ]
  ```

- Process Report

  ```
  svmon -P [ PID1... PIDN ] [ -r [ -n | -s ] [ -w | -f | -c ] [ -t Count ]
  [ -u | -p | -g | -v ] [ -i Interval [ NumIntervals] ] [ -l ] [ -j ] [ -z ]
  [ -m ]
  [ -q [ s | L ] ]
  ```

- Segment Report

```
svmon -S [ SID1...SIDN ] [ -r ] [ -n | -s ] [ -w | -f | -c ] [ -t Count ]
[ -u | -p | -g | -v ] [ -i Interval [ NumIntervals] ] [ -l ] [ -j ] [ -z ]
[ -m ] [ -q [ s | L ] ]
```

- Detailed Report

```
svmon -D SID1..SIDN [ -b ] [ -i Interval [ NumIntervals] ] [ -z ]
[ -q [ s | L ] ]
```

- Framed Report

```
svmon -F [ Frame1..FrameN ] [ -i Interval [ NumIntervals] ] [ -z ]
[ -q [ s | L ] ]
```

The details on these flags and the parameters used by the **svmon** command are available on the following Web site:

http://publib.boulder.ibm.com/infocenter/pseries/index.jsp

11.5 File and directory permissions and ownership

This section discusses various topics that explain how to assign and control access to AIX 5L files and directories.

11.5.1 Access control lists

Access control consists of protected information resources that specify who can be granted access to such resources. The operating system allows for need-to-know or discretionary security. The owner of an information resource can grant other users read or write access rights for that resource. A user who is granted access rights to a resource can transfer those rights to other users. This security allows for user-controlled information flow in the system; the owner of an information resource defines the access permissions to the object.

Users have user-based access only to the objects that they own. Typically, users receive either the group permissions or the default permissions for a resource. The major task in administering access control is to define the group memberships of users, because these memberships determine the users' access rights to the files that they do not own.

Access control lists (ACLs) increase the quality of file access controls by adding extended permissions that modify the base permissions assigned to individuals and groups. With extended permissions, you can permit or deny file access to specific individuals or groups without changing the base permissions.

Note: The access control list for a file cannot exceed one memory page (approximately 4096 bytes) in size.

To maintain access control lists, use the **aclget**, **acledit**, and the **aclput** commands.

The **chmod** command in numeric mode (with octal notations) can set base permissions and attributes. The **chmod** subroutine, which the command calls, disables extended permissions. If you use the numeric mode of the **chmod** command on a file that has an ACL, extended permissions are disabled. The symbolic mode of the **chmod** command does not disable extended permissions. For information about numeric and symbolic mode, refer to the **chmod** command.

Base permissions

Base permissions are the traditional file-access modes assigned to the file owner, file group, and other users. The access modes are: read (r), write (w), and execute/search (x).

In an access control list, base permissions are in the following format, with the Mode parameter expressed as rwx (with a hyphen (-) replacing each unspecified permission):

- base permissions:
- owner(name): Mode
- group(group): Mode
- others: Mode

Attributes

Three attributes can be added to an access control list:

- setuid (SUID): Set-user-ID mode bit. This attribute sets the effective and saved user IDs of the process to the owner ID of the file on execution.
- setgid (SGID): Set-group-ID mode bit. This attribute sets the effective and saved group IDs of the process to the group ID of the file on execution.
- savetext (SVTX): Saves the text in a text file format.

These attributes are added in the following format:

attributes: SUID, SGID, SVTX

Extended permissions

Extended permissions allow the owner of a file to define access to that file more precisely. Extended permissions modify the base file permissions (owner, group, and others) by permitting, denying, or specifying access modes for specific individuals, groups, or user and group combinations. Permissions are modified through the use of keywords.

The permit, deny, and specify keywords are defined as follows:

- permit: Grants the user or group the specified access to the file.
- deny: Restricts the user or group from using the specified access to the file.
- specify: Precisely defines the file access for the user or group.

If a user is denied a particular access by either a deny or a specify keyword, no other entry can override that access denial.

The enabled keyword must be specified in the ACL for the extended permissions to take effect. The default value is the disabled keyword.

In an ACL, extended permissions are in the following format:

```
extended permissions:
  enabled | disabled
    permit   Mode  UserInfo...:
    deny     Mode  UserInfo...:
    specify  Mode  UserInfo...:
```

Use a separate line for each permit, deny, or specify entry. The Mode parameter is expressed as rwx (with a hyphen (-) replacing each unspecified permission). The UserInfo parameter is expressed as u:UserName, or g:GroupName, or a comma-separated combination of u:UserName and g:GroupName.

> **Note:** If more than one user name is specified in an entry, that entry cannot be used in an access control decision, because a process has only one user ID.

The following is an example of an ACL:

```
attributes: SUID
base permissions:
      owner(frank):  rw-
      group(system): r-x
      others: ---
extended permissions:
      enabled
        permit  rw-  u:dhs
        deny    r--  u:chas, g:system
        specify r--  u:john, g:gateway, g:mail
```

```
    permit  rw-  g:account, g:finance
```

The parts of the ACL and their meanings are as follows:

- The first line indicates that the setuid bit is turned on.
- The next line, which introduces the base permissions, is optional.
- The next three lines specify the base permissions. The owner and group names in parentheses are for information only. Changing these names does not alter the file owner or file group. Only the **chown** command and the **chgrp** command can change these file attributes.
- The next line, which introduces the extended permissions, is optional.
- The next line indicates that the extended permissions that follow are enabled.
- The last four lines are the extended entries. The first extended entry grants the user read (r) and write (w) permission on the file.
- The second extended entry denies read (r) access to a user only when he is a member of the system group.
- The third extended entry specifies that as long as user john is a member of both the gateway group and the mail group, he has read (r) access. If user john is not a member of both groups, this extended permission does not apply.
- The last extended entry grants any user in both the account group and the finance group read (r) and write (w) permission.

Note: More than one extended entry can be applied to a process, with restrictive modes taking precedence over permissive modes.

Displaying Access Control Information (the aclget command)

To display the access control information of a file, use the **aclget** command. The information that you view includes attributes, base permissions, and extended permissions.

For example, to display the access control information for the status file, type:

```
# aclget status
```

Press Enter. The access control information that displays includes a list of attributes, base permissions, and extended permissions.

Setting Access Control Information (the aclput command)

To set the access control information for a file, use the **aclput** command.

> **Note:** The access control list for a file cannot exceed one memory page (approximately 4096 bytes) in size.

The following are useful examples:

1. To set the access control information for the status file with the access control information stored in the acldefs file, type:

   ```
   # aclput -i acldefs status
   ```

   Press Enter.

2. To set the access control information for the status file with the same information used for the plans file, type:

   ```
   # aclget plans | aclput status
   ```

   Press Enter.

Editing Access Control Information (acledit command)

To change the access control information of a file, use the **acledit** command. The command displays the current access control information and lets the file owner change it. Before making any changes permanent, the command asks if you want to proceed.

> **Note:** The EDITOR environment variable must be specified with a complete path name; otherwise, the **acledit** command will fail.

The access control information that displays includes a list of attributes, base permissions, and extended permissions.

The following provides an example:

1. To edit the access control information of the plans file, type:

   ```
   # acledit plans
   ```

   Press Enter.

11.5.2 The chmod command

The **chmod** command modifies the mode bits and the extended access control lists (ACLs) of the specified files or directories. The mode can be defined symbolically or numerically (absolute mode).

The **chmod** command has the following syntax:

To change file modes symbolically:

```
chmod [ -R ] [ -h ] [ -f ] [ [ u ] [ g ] [ o ] | [ a ] ] { { - | + | = } [ r ]
[ w ] [ x ] [ X ] [ s ] [ t ] } { File ... | Directory ... }
```

To change file modes numerically:

```
chmod [ -R ] [ -h ] [ -f ] PermissionCode { File ... | Directory ... }
```

where:

-f	Suppresses all error reporting except invalid permissions and usage statements.
-h	Suppresses a mode change for the file or directory pointed to by the encountered symbolic link.
-R	Descends only directories recursively, as specified by the pattern File...|Directory... . The -R flag changes the file mode bits of each directory and of all files matching the specified pattern.

Symbolic mode

To specify a mode in symbolic form, you must specify three sets of flags.

> **Note:** Do not separate flags with spaces.

The first set of flags specifies who is granted or denied the specified permissions, as follows:

u	File owner.
g	Group and extended ACL entries pertaining to the file's group.
o	All others.
a	User, group, and all others. The a flag has the same effect as specifying the ugo flags together. If none of these flags are specified, the default is the a flag and the file creation mask (umask) is applied.

The second set of flags specifies whether the permissions are to be removed, applied, or set:

-	Removes specified permissions.
+	Applies specified permissions.
=	Clears the selected permission field and sets it to the permission specified. If you do not specify a permission following =, the **chmod** command removes all permissions from the selected field.

The third set of flags specifies the permissions that are to be removed, applied, or set:

r	Read permission.
w	Write permission.
x	Execute permission for files; search permission for directories.
X	Execute permission for files if the current (unmodified) mode bits have at least one of the user, group, or other execute bits set. The X flag is ignored if the File parameter is specified and none of the execute bits are set in the current mode bits.

These flags set the search permissions for directories:

s	Set-user-ID-on-execution permission if the u flag is specified or implied. Set-group-ID-on-execution permission if the g flag is specified or implied.
t	For directories, indicates that only file owners can link or unlink files in the specified directory. For files, sets the save-text attribute.

Numeric or absolute mode

The **chmod** command also permits you to use octal notation for the mode. The numeric mode is the sum of one or more of the following values:

4000	Sets user ID on execution.
2000	Sets group ID on execution.
1000	Sets the link permission to directories or sets the save-text attribute for files.
0400	Permits read by owner.
0200	Permits write by owner.
0100	Permits execute or search by owner.
0040	Permits read by group.
0020	Permits write by group.

0010 Permits execute or search by group.

0004 Permits read by others.

0002 Permits write by others.

0001 Permits execute or search by others.

Note:

1. Specifying the mode numerically disables any extended ACLs.
2. Changing group access permissions symbolically also affects the extended ACL entries. The group entries in the ACL that are equal to the owning group of the file are denied any permission that is removed from the mode. Refer to "Access Control Lists" for more information.
3. You can specify multiple symbolic modes separated with commas. Operations are performed in the order they appear from left to right.
4. You must specify the mode symbolically or use an explicit 4-character octal with a leading zero (for example, 0755) when removing the set-group-ID-on-execution permission from directories.

- The following example shows how to make several permission changes at once:

  ```
  #chmod  go-w+x  mydir
  ```

  This denies group members and others the permission to create or delete files in mydir (go-w) and allows group members and others to search mydir or use it in a path name (go+x). This is equivalent to the command sequence:

  ```
  chmod  g-w  mydir
  chmod  o-w  mydir
  chmod  g+x  mydir
  chmod  o+x  mydir
  ```

- To permit only the owner to use a shell procedure as a command, do the following:

  ```
  # chmod  u=rwx,go= cmd
  ```

  This gives read, write, and execute permission to the user who owns the file (u=rwx). It also denies the group and others the permission to access cmd in any way (go=).

  If you have permission to execute the **cmd** shell command file, then you can run it by entering:

  ```
  # cmd
  ```

Note: Depending on the PATH shell variable, you may need to specify the full path to the **cmd** command.

- The following shows how to use Set-ID Modes:

  ```
  # chmod  ug+s  cmd
  ```

  When the **cmd** command is executed, the effective user and group IDs are set to those that own the **cmd** file. Only the effective IDs associated with the child process that runs the **cmd** command are changed. The effective IDs of the shell session remain unchanged.

  This feature allows you to permit access to restricted files. Suppose that the **cmd** program has the Set-User-ID Mode enabled and is owned by a user called dbms. The user dbms is not actually a person, but might be associated with a database management system. The user betty does not have permission to access any of dbms's data files. However, she does have permission to execute the **cmd** command. When she does so, her effective user ID is temporarily changed to dbms, so that the **cmd** program can access the data files owned by the user dbms.

  This way, the user betty can use the **cmd** command to access the data files, but she cannot accidentally damage them with the standard shell commands.

- To use the absolute mode form of the **chmod** command:

  ```
  # chmod  644  text
  ```

  This sets read and write permission for the owner, and it sets read-only mode for the group and others. This also removes all extended ACLs that might be associated with the file.

- To recursively descend directories and change file and directory permissions given the tree structure:

  ```
  ./dir1/fdir2/file1
  ./dir1/fdir2/file2
  ./dir1/file3
  ```

  the command:

  ```
  # chmod -R 777 f*
  ```

  will change permissions on:

  ```
  ./dir1/fdir2
  ./dir1/fdir2/file1
  ./dir1/fdir2/file2
  ./dir1/file3
  ```

11.5.3 The chown command

The following is the syntax of the **chown** command:

```
chown [  -f ] [ -h ] [  -R ] Owner [ :Group ] { File ... | Directory ... }
chown -R  [  -f ] [ -H | -L | -P ] Owner [ :Group ] { File ... | Directory ...
}
```

The **chown** command changes the owner of the file specified by the File parameter to the user specified by the Owner parameter. The value of the Owner parameter can be a user ID found in the /etc/passwd file. Optionally, a group can also be specified. The value of the Group parameter can be a group ID found in the /etc/group file. The **chown** command can also be used for non-local users within the context of NIS, for example.

Only the root user can change the owner of a file. You can change the group of a file only if you are a root user or if you own the file. If you own the file but are not a root user, you can change the group only to a group of which you are a member.

Although the -H, -L, and -P flags are mutually exclusive, specifying more than one is not considered an error. The last flag specified determines the behavior that the command will exhibit.

When a symbolic link is encountered and you have not specified the -h flag, the **chown** command changes the ownership of the file or directory pointed to by the link and not the ownership of the link itself.

If you specify the -H flag, the **chown** command has the opposite effect and changes the ownership of the link itself and not that of the file or directory pointed to by the link.

If you specify the -R flag, the **chown** command recursively descends the specified directories.

If you specify both the -h flag and the -R flag, the **chown** command descends the specified directories recursively, and when a symbolic link is encountered, the ownership of the link itself is changed and not that of the file or directory pointed to by the link.

The flags used by the **chown** command have the following meanings:

-f Suppresses all error messages except usage messages.

-h Changes the ownership of an encountered symbolic link and not that of the file or directory pointed to by the symbolic link.

-H If the -R option is specified and a symbolic link referencing a file of type directory is specified on the command line, the

	chown command shall change the user ID (and group ID, if specified) of the directory referenced by the symbolic link and all files in the file hierarchy below it.
-L	If the -R option is specified and a symbolic link referencing a file of type directory is specified on the command line or encountered during the traversal of a file hierarchy, the **chown** command shall change the user ID (and group ID, if specified) of the directory referenced by the symbolic link and all files in the file hierarchy below it.
-P	If the -R option is specified and a symbolic link is specified on the command line or encountered during the traversal of a file hierarchy, the **chown** command shall change the owner ID (and group ID, if specified) of the symbolic link if the system supports this operation. The **chown** command shall not follow the symbolic link to any other part of the file hierarchy.
-R	Descends directories recursively, changing the ownership for each file. When a symbolic link is encountered and the link points to a directory, the ownership of that directory is changed, but the directory is not further transversed. If the -h, -H, -L, or -P flags are not also specified, when a symbolic link is encountered and the link points to a directory, the group ownership of that directory is changed, but the directory is not traversed further.

- The following shows how to change the owner of the file program.c:

  ```
  chown jim program.c
  ```

  The user access permissions for program.c now apply to jim. As the owner, jim can use the **chmod** command to permit or deny other users access to program.c.

- To change the owner and group of all files in the directory /tmp/src to owner john and group build:

  ```
  #chown -R john:build /tmp/src
  ```

11.5.4 The chgrp command

The following is the syntax of the **chgrp** command:

```
chgrp [  -f ] [ -h ] [-R ] Group { File ... | Directory ... }
chgrp -R  [  -f ] [ -H | -L | -P ] Group { File... | Directory... }
```

The **chgrp** command changes the group associated with the specified file or directory to the specified group name or group ID number. When a symbolic link is encountered and you have not specified the -h or -P flags, the **chgrp** command changes the group ownership of the file or directory pointed to by the link and not the group ownership of the link itself.

Although the -H, -L and -P flags are mutually exclusive, specifying more than one is not considered an error. The last flag specified determines the behavior that the command will exhibit.

If you specify the -h flag, the **chgrp** command has the opposite effect and changes the group ownership of the link itself and not that of the file or directory pointed to by the link.

If you specify both the -h flag and the -R flag, the **chgrp** command descends the specified directories recursively, and when a symbolic link is encountered, the group ownership of the link itself is changed and not that of the file or directory pointed to by the link.

The flags used by the **chgrp** command have the following meanings:

-f Suppresses all error messages except usage messages.

-h Changes the group ownership of an encountered symbolic link and not that of the file or directory pointed to by the symbolic link.

-H If the -R option is specified and a symbolic link referencing a file of type directory is specified on the command line, **chgrp** shall change the group of the directory referenced by the symbolic link and all files in the file hierarchy below it.

-L If the -R option is specified and a symbolic link referencing a file of type directory is specified on the command line or encountered during the traversal of a file hierarchy, **chgrp** shall change the group of the directory referenced by the symbolic link and all files in the file hierarchy below it.

-P If the -R option is specified and a symbolic link is specified on the command line or encountered during the traversal of a file hierarchy, **chgrp** shall change the group ID of the symbolic link if the system supports this operation. The **chgrp** utility shall not follow the symbolic link to any other part of the file hierarchy.

-R Descends directories recursively, setting the specified group ID for each file. When a symbolic link is encountered and the link points to a directory, the group ownership of that directory is changed but the directory is not further traversed. If the -h, -H, -L, or -P flags are not also specified when a symbolic link is encountered and the link points to a directory, the group ownership of that directory is changed, but the directory is not traversed further.

- The following will change the group ownership of the file or directory named *proposals* to staff:

  ```
  #chgrp staff proposals
  ```

  The group access permissions for *proposals* now apply to the staff group.
- The following will change the group ownership of the directory named *proposals*, and of all the files and subdirectories under it, to staff:

  ```
  #chgrp -R staff proposals
  ```

  The group access permissions for *proposals* and for all the files and subdirectories under it now apply to the staff group.

11.6 Local and global variables

This section will describe local and global variables and how they are set, unset, and displayed.

11.6.1 Local variables

A variable name has local scope or block scope if it is declared in a block. A name with local scope can be used in that block and in blocks enclosed within that block, but the name must be declared before it is used. When the block is exited, the names declared in the block are no longer available.

When one block is nested inside another, the variables from the outer block are usually visible in the nested block. However, if the declaration of a variable in a nested block has the same name as a variable that is declared in an enclosing block, the declaration in the nested block hides the variable that was declared in the enclosing block. The original declaration is restored when program control returns to the outer block. This is called block visibility.

Name resolution in a local scope begins in the immediate scope in which the name is used and continues outward with each enclosing scope. The order in which scopes are searched during name resolution causes the phenomenon of information hiding. A declaration in an enclosing scope is hidden by a declaration of the same identifier in a nested scope.

11.6.2 Global variables

A variable name has global scope if its declaration appears outside of any block. A name with global scope and internal linkage is visible from the point where it is declared to the end of the translation unit.

11.6.3 How to set, unset, and show these variables

By default, all shell variables are global. A global variable can be used anywhere in a program or a script. However, if you use **typeset** (or **integer** or **readonly**) within a function, the newly declared variable will be a local variable. That is, you will be able to use that variable only within the function in which it is declared.

In the following script, variables star and glass are global. Variable star is global because it is explicitly declared in a region of the script that is not within a function. Variables glass is global because it is implicitly declared; that is, it is not created with typeset.

```
integer star=10 #star is global
function answer
{
integer drop=5 #drop is local to function answer
((drop = drop * 10))
print “drop = $drop”
((star = star * 100))
print “star = $star”]
function result
{typeset water #water is a string variable local to function result
water=”cold”
print “water = $water”
glass=”blue” #glass is global because it is not declared with typeset
}
```

The script begins execution at the next line:

```
answer
result
print “glass = $glass”
print “star = $star”
The output:
drop = 50
star = 1000
water = cold
glass = blue
star = 1000
```

unset Name ... Unsets the values and attributes of the variables given by the list of names Name. Unsetting a variable removes its special meaning as shown in the following example:

```
# x=3
# echo $x
3
# unset x
# echo $x
#
```

11.7 The cron daemon and crontab

The **cron** daemon runs shell commands at specified dates and times. An entry in the /etc/inittab file starts the **cron** daemon at system initialization. The default record for AIX 5L Version 5.3 is shown in Table 11-3.

Table 11-3 Default cron record in the /etc/inittab file

Field	Value
Identifier	cron
RunLevel	23456789
Action	respawn
Command	/usr/sbin/cron

Regularly scheduled commands can be specified according to instructions contained in the crontab files. You can submit your crontab file with the **crontab** command. Use the **at** command to submit commands that are to be run only once. You can also use the **batch** command to run a shell script invoking the **at** command.

11.7.1 The at command

The **at** command reads, from standard input, the names of commands to be run at a later time and allows you to specify when the commands should be run. The **at** command mails you all output from standard output and standard error for the scheduled commands, unless you redirect that output. It also writes the job number and the scheduled time to standard error.

Submit a job to be run at a later time using the format:

`at -f` *filename* `-t` *CCYYMMDDhhmmSS Increment*

where the values of the -t and -f flags are:

-f filename Uses the specified file as input rather than using standard input.

-t Submits the job to be run at the time specified by *CCYYMMDDhhmmSS*.

- **CC** Specifies the first two digits of the year (the century). If this is not specified, the default value is 19, if the value of YY is between 70 and 99. The default value is 20 if the value of YY is between 00 and 37.
- **YY** Specifies the second two digits of the year.

MM Specifies the month of the year (01 through 12).

DD Specifies the day of the month (01 through 31).

hh Specifies the hour of the day (00 through 23).

mm Specifies the minute of the hour (00 through 59).

SS Specifies the second of the minute (00 through 59). The default value is 00 is it is not specified.

filename Specifies the command file to be run at a later time.

The optional Increment parameter can be one of the following:

- A + (plus sign) followed by a number and one of the following words:
  - minute[s]
  - hour[s]
  - day[s]
  - week[s]
  - month[s]
  - year[s]
- The special word `next` followed by one of these words:
  - minute[s]
  - hour[s]
  - day[s]
  - week[s]
  - month[s]
  - year[s]

Instead of the -t flag, you can also use the following keywords:

- noon
- midnight
- now

Other flags and values can be found in the AIX 5L product documentation.

11.7.2 Location of the spool area directory for the at command

The spool area directory for the **at** command is /var/spool/cron/atjobs. After a job is scheduled to run, a file exists in the /var/spool/cron/atjobs directory and the **atq** command shows:

```
# ls /var/spool/cron/atjobs
root.1134169200.a
# atq
root.1134169200.a        Fri Dec  9 17:00:00 CST 2005
```

After the job is deleted, the /var/spool/cron/atjobs directory is empty and the **atq** command no longer shows anything:

```
# at -r root.1134169200.a
The root.1134169200.a at file is deleted.
# ls /var/spool/cron/atjobs
# atq
```

11.7.3 Location of crontab files

The crontab files are kept in the /var/spool/cron/crontabs directory. Each **cron** user has a crontab file with their user name as the file name in the /var/spool/cron/crontabs directory. The content of a sample /var/spool/cron/crontabs directory is shown in the following example:

```
# ls -l /var/spool/cron/crontabs
total 32
-rw-r--r--   1 adm      cron           2020 Jun 17 2004  adm
-rw-------   1 root     cron           1726 Nov 22 16:17 root
-rw-r--r--   1 sys      cron            853 Jun 17 2004  sys
-rw-r--r--   1 root     cron           1122 Nov 22 16:08 uucp
```

A sample of the /var/spool/cron/crontabs/root file is shown in the following example:

```
# crontab -l
... (lines omitted) ...
#0 3 * * * /usr/sbin/skulker
#45 2 * * 0 /usr/lib/spell/compress
#45 23 * * * ulimit 5000; /usr/lib/smdemon.cleanu > /dev/nul
0 11 * * * /usr/bin/errclear -d S,O 30
0 12 * * * /usr/bin/errclear -d H 90
0 15 * * *  /usr/lib/ras/dumpcheck >/dev/null 2>&1
```

11.7.4 Verifying job runs

You can examine each file in the /var/spool/cron/crontabs directory to verify what cron jobs are scheduled to run at a certain time. Together with the **atq** command, you can determine whether some jobs need to be rescheduled to better utilize the system resources. Both the users root and deploy have scheduled a job to run using the **at** command, as shown by the **atq** command in the following example:

```
# atq
root.1134169200.a        Fri Dec  9 17:00:00 CST 2005
deploy.1134187200.a      Fri Dec  9 22:00:00 CST 2005
#
```

The **cron** daemon also creates a log of its activities in the /var/adm/cron/log file. AIX 5L Version 5.3 enhances the amount of data logged to include the following information about individual jobs:

- The owner of the job run by the **cron** daemon.
- The time of execution of the job.
- The PID of the job.
- The actual command line that is run to accomplish the job.
- Whether the job has run successfully or not.

Either of the following display formats are used:

- User : CMD (actual command that is executed) : time when the job is executed : Cron Job with pid : Successful
- User : CMD (actual command that is executed) : time when the job is executed : Cron Job with pid : Failed

For example:

```
root : CMD ( /usr/lib/ras/dumpcheck >/dev/null 2>&1 ) : Tue Feb. 20 15:00:00 2001
Cron Job with pid: 20664 Successful
```

Every time **cron** runs a job (either from the crontab file, for the system-related jobs, or from the /var/spool/cron/crontab/userfile, for user-related processes), all its activity will be logged into the /var/adm/cron/log file in the mentioned format. You can inspect this data to verify jobs after they have been run.

The **cron** daemon also mails any command output or errors to the user who submitted the job, unless the **at** command or crontab entry redirects the standard output or standard error. The following crontab entry redirects standard output and standard error, so no mail is sent:

```
0 15 * * *  /usr/lib/ras/dumpcheck >/dev/null 2>&1
```

The following entry in root's crontab file runs the **mksysb** command and does not redirect its output, so **cron** mails the results to root after the command completes:

```
0 02 * * *  /usr/bin/mksysb -i /dev/rmt0
```

11.7.5 Crontab file record format

A crontab file contains entries for each **cron** job. Entries are separated by new-line characters. Each crontab file entry contains six fields separated by spaces or tabs in the following form:

```
minute  hour  day_of_month  month  weekday  command
```

These fields accept the following values:

minute	0 through 59
hour	0 through 23
day_of_month	1 through 31
month	1 through 12
weekday	0 through 6 for Sunday through Saturday
command	A shell command

You must specify a value for each field. Except for the command field, these fields can contain the following:

- A number in the specified range. To run a command in May, specify 5 in the month field.
- Two numbers separated by a dash to indicate an inclusive range. To run a cron job on Tuesday through Friday, place 2-5 in the weekday field.
- A list of numbers separated by commas. To run a command on the first and last day of January, you would specify 1,31 in the day_of_month field.
- An * (asterisk), meaning all allowed values. To run a job every hour, specify an asterisk in the hour field.

Blank lines and lines whose first non-blank character is # (number sign) are ignored. By default, the records containing /usr/sbin/skulker, /usr/lib/spell/compress, and /usr/lib/smdemon.cleanu in the command field are commented. This is shown in the example in 11.7.3, "Location of crontab files" on page 544.

11.7.6 Scheduling job runs

Consider the following single line script called program in the /appl directory:

```
#!/usr/bin/ksh
print program executed at `date +%H:%M` >> /data/output
```

Some examples of the crontab entries for scheduling this /appl/program to run are shown in Table 11-4.

Table 11-4 Sample crontab entries for scheduling a running of a program

Program run time	crontab entry
Every Tuesday at 02:00	`0 2 * * 2 /appl/program`
Every night at 02:00	`0 2 * * * /appl/program`
Every night at midnight	`0 0 * * * /appl/program`
On the 7th, 14th, and 21st day at 4:20 pm	`20 16 7,14,21 * * /appl/program`
Monday through Friday at 4:20 pm	`20 16 * * 1-5 /appl/program`
1st January, 1st July, and 1st December at 4:20 pm	`20 16 1 1,7,12 * /appl/program`

To schedule /appl/program to run every five minutes, you can either use the **crontab** command or the **at** command.

Technically, you can define all the values in the hour in five minute intervals starting at 0, 1, 2, 3, or 4, as one of the following crontab entries:

```
0,5,10,15,20,25,30,35,40,45,50,55 * * * * /appl/program
1,6,11,16,21,26,31,36,41,46,51,56 * * * * /appl/program
2,7,12,17,22,27,32,37,42,47,52,57 * * * * /appl/program
3,8,13,18,23,28,33,38,43,48,53,58 * * * * /appl/program
4,9,14,19,24,29,34,39,44,49,54,59 * * * * /appl/program
```

Alternatively, you can use the **at** command. For scheduling the /appl/program to run every five minutes, or some short intervals repeatedly, modify the /appl/program file by adding the **at** command as follows:

```
#!/usr/bin/ksh
at now + 5 minutes -f /appl/program > /dev/null 2>&1
print program executed at `date +%H:%M` >> /data/output
```

Start the first job with the command:

```
at now -f appl/program > /dev/null 2>&1
```

When the /appl/program runs for the first time, it schedules the next run five minutes later. This process repeats itself until you stop it by entering the **atq** command to get the scheduled job name and then the **at -r** command to remove the scheduled job.

11.7.7 Allowing access to the crontab command

The /var/adm/cron/cron.allow and /var/adm/cron/cron.deny files control which users can use the **crontab** command. A root user can create, edit, or delete these files. Entries in these files are user login names with one name to a line.

If the cron.allow file exists, only users whose login names appear in it can use the **crontab** command.

> **Note:** The root user name must appear in the cron.allow file if the file exists.

You can explicitly stop a user from using the **crontab** command by listing the user's login name in the cron.deny file. If only the cron.deny file exists, any user whose name does not appear in the file can use the **crontab** command.

A user cannot use the **crontab** command if one of the following is true:

- The cron.allow file and the cron.deny file do not exist (allows root user only).
- The cron.allow file exists but the user's login name is not listed in it.
- The cron.deny file exists and the user's login name is listed in it.

If neither the cron.allow or the cron.deny file exists, only someone with root user authority can submit a job with the **crontab** command.

To allow the people in charge of application deployment to schedule applications to be run, a /var/adm/cron/cron.allow file is created, as shown in the following example:

```
# cat > /var/adm/cron/cron.allow
root
deploy

Ctrl+d
#
```

There are also the /var/adm/cron/at.allow and /var/adm/cron/at.deny files that control which users can use the **at** command, using the same rules as the /var/adm/cron/cron.allow and /var/adm/cron/cron.deny files.

11.7.8 Creating and updating the crontab file

The user deploy uses the **crontab -e** command to create and update the crontab file. The **crontab** command invokes the editor. If the EDITOR environment variable exists, the command invokes the editor it specifies. Otherwise, the **crontab** command uses the **vi** editor. On saving and exiting from the editor, a message indicating the change is sent to the **cron** daemon. The user deploy's **crontab -e** session using the **vi** editor is shown in Figure 11-10.

```
30 13 * * * /appl/program
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"/tmp/crontabJKuIya"
```

Figure 11-10 crontab -e command using the vi editor

The content of the /var/spool/cron/crontabs directory and the content of the crontab file created by the user deploy is shown in the following example:

```
# ls -l /var/spool/cron/crontabs/
total 40
-rw-r--r--   1 adm      cron          2020 Jun 17 2004  adm
-rw-------   1 root     cron            26 Dec 08 17:41 deploy
-rw-------   1 root     cron          1726 Dec 08 17:28 root
-rw-r--r--   1 sys      cron           853 Jun 17 2004  sys
-rw-r--r--   1 root     cron          1122 Nov 22 16:08 uucp
# cat /var/spool/cron/crontabs/deploy
30 13 * * * /appl/program
#
```

The user deploy is scheduling /appl/program to run at 1:30 PM every day.

11.7.9 Verifying job schedules

As a general user without root authority, the user deploy is not authorized to list the content of the /var/spool/cron/crontabs/deploy file. However, the user can still verify his job schedules by using the **crontab -l** command, as shown in the following example:

```
$ whoami
deploy
$ cat /var/spool/cron/crontabs/deploy
cat: 0652-050 Cannot open /var/spool/cron/crontabs/deploy.
$ crontab -l
30 13 * * * /appl/program
$
```

11.7.10 Checking the time of the crontab file

Merely verifying the job schedule is not enough to prove that **cron** will schedule the command to run at that time. If you have submitted your crontab file past the time the command is scheduled to run, nothing would have been scheduled. Use the **crontab -v** command to check the crontab submission time, as shown in the following example:

```
$ crontab -l
30 13 * * * /appl/program
$ crontab -v
Crontab file: deploy         Submission time: Thu Dec  8 13:07:21 CST 2005
$ date
Thu Dec  8 17:47:34 CST 2005
$ cat /data/output
program executed at 13:30
$
```

The /appl/program is scheduled to run at 1:30 PM everyday. The crontab file is submitted at 1:07 PM. The time now is 5:47 PM. The command should have been scheduled to run. Checking the /data/output file confirms that the /appl/program has been run.

11.7.11 Removing the crontab file

The responsibility for scheduling jobs to run is now moved from the application deployment group to the operations group. The user name deploy in the /var/adm/cron.allow file has been replaced by another user name from operations. The user deploy then removes the /var/spool/cron/crontabs/deploy file with the **crontab -r** command. The /var/spool/cron/crontabs/deploy is simply deleted without any message. Until the operations people create their crontab file, the content of the /var/spool/cron/crontabs directory is reverted back to the state, as shown in 11.7.3, “Location of crontab files” on page 544.

> **Note:** Avoid running **crontab -r** when you are logged in as root. It removes the /var/spool/cron/crontabs/root file. This file usually contains the scheduling of housekeeping jobs, such as diagnostics of hardware errors. If you remove the file, you will have to restore the file from your backup.

11.7.12 Using crontab to append to a user’s cron file

The operations group has been given a user name ops. They do not want to start from scratch for scheduling jobs. Instead, they just want to reschedule the time of run for the jobs that the application deployment people run. The application deployment group has actually saved a backup copy of their crontab file, as shown in the following example:

```
$ crontab -l > deploy.schedule
$ cat deploy.schedule
30 13 * * * /appl/program
$
```

The operations people are given access to this deploy.schedule file. They can then create their crontab file using the **crontab file** command, as shown in the following example:

```
$ whoami
ops
$ crontab -l
0481-103 Cannot open a file in the /var/spool/cron/crontabs directory.
A file or directory in the path name does not exist.
$ crontab ~deploy/deploy.schedule
$ crontab -l
30 13 * * * /appl/program
$
```

After that, they will use the **crontab -e** command to edit the time to schedule the command to be run at a different time.

11.8 System Resource Controller administration

The System Resource Controller (SRC) provides a set of commands and subroutines to make it easier for the system manager and programmer to create and control subsystems. A subsystem is any program or process or set of programs or processes that is capable of operating independently or with a controlling system. A subsystem is designed as a unit to provide a designated function. A subserver is a program or process that belongs to a subsystem.

The SRC is designed to minimize the need for operator intervention. It provides a mechanism to control subsystem processes using a common command line and the C interface. This mechanism includes the following:

- Consistent user interface for start, stop, and status inquiries.
- Logging of the abnormal termination of subsystems.
- A notification program called at the abnormal system termination of related processes.
- Tracing of a subsystem, a group of subsystems, or a subserver.
- Support for control of operations on a remote system.
- Refreshing of a subsystem (such as after a configuration data change).

The SRC is useful if you want a common way to start, stop, and collect status information about processes.

11.8.1 Starting the SRC

The System Resource Controller (SRC) is started during system initialization with a record for the **/usr/sbin/srcmstr** daemon in the /etc/inittab file. The default /etc/inittab file already contains such a record, so starting the SRC may be unnecessary. You can, if needed, start the SRC from the command line, a profile, or a shell script, but there are several reasons for starting it during initialization:

- Starting the SRC from the /etc/inittab file allows the **init** command to restart the SRC should it stop for any reason.
- The SRC is designed to simplify and reduce the amount of operator intervention required to control subsystems. Starting the SRC from any source other than the /etc/inittab file would be counter-productive to that goal.
- The default /etc/inittab file contains a record for starting the print scheduling subsystem (qdaemon) with the **startsrc** command. Typical installations have other subsystems started with **startsrc** commands in the /etc/inittab file as well. Since the **startsrc** command requires the SRC to be running, removing

the **srcmstr** daemon from the /etc/inittab file would cause these **startsrc** commands to fail.

Refer to the manual page using the command **man srcmstr** for the configuration requirements to support remote SRC requests.

If the /etc/inittab file does not already contain a record for the **srcmstr** daemon, you can add one using the following procedure:

1. Make a record for the **srcmstr** daemon in the /etc/inittab file using the **mkitab** command. For example, to make a record identical to the one that appears in the default /etc/inittab file, enter:

   ```
   mkitab -i fbcheck srcmstr:2:respawn:/usr/sbin/srcmstr
   ```

   The -i fbcheck flag ensures that the record is inserted before all subsystems records.

2. Tell the **init** command to reprocess the /etc/inittab file by entering:

   ```
   telinit q
   ```

   When **init** revisits the /etc/inittab file, it processes the newly entered record for the **srcmstr** daemon and starts the SRC.

11.8.2 The telinit command

The **telinit** command directs the actions of the **init** process (process ID 1) by taking a one-character argument and signaling the **init** process to perform the appropriate action. In general, the **telinit** command sets the system at a specific run level. A run level is a software configuration that allows only a selected group of processes to exist. The following arguments serve as directives that the **telinit** command passes to the **init** process:

0-9 Tells the init process to put the system in one of the run levels 0-9.

S,s,M,m Tells the init process to enter the maintenance mode.

a,b,c Tells the init process to examine only those records in the /etc/inittab file with a, b, or c in the run level field.

Q,q Tells the init process to re-examine the entire /etc/inittab file.

N Sends a signal that stops processes from being respawned.

11.8.3 Restarting the SRC

Normally, you do not need to restart **srcmstr**. The default record in /etc/inittab for AIX 5L Version 5.3 is shown in Table 11-3 on page 542.

Table 11-5 Default srcmstr record in the /etc/inittab file

Field	Value
Identifier	srcmstr
RunLevel	23456789
Action	respawn
Command	/usr/sbin/srcmstr

If the **srcmstr** daemon terminates abnormally, the respawn action specified in the /etc/inittab restarts the **srcmstr** daemon. The **srcmstr** daemon then determines which SRC subsystems were active during the previous invocation. The daemon reestablishes communication with these subsystems (if it existed previously) and initializes a private kernel extension and the **srcd** daemon to monitor the subsystem processes. Note that the process ID is changed after **srcmstr** is terminated and restarted automatically, as shown in the following example:

```
# ps -ef | grep srcmstr
    root 131284      1    0   Nov 30      -  0:00 /usr/sbin/srcmstr
    root 413714 450604    0 17:58:54  pts/8  0:00 grep srcmstr
# kill -9 131284
# ps -ef | grep srcmstr
    root 131286      1    0 17:59:16      -  0:00 /usr/sbin/srcmstr
    root 430288 450604    0 17:59:22  pts/8  0:00 grep srcmstr
#
```

However, if you have edited the /etc/inittab file by adding the -r or -B flag to /usr/sbin/srcmstr, you must use the command **init -q** to reexamine /etc/inittab or reboot to make the new flags effective. The -r flag allows **srcmstr** to accept remote requests, and -B runs **srcmstr** in a pre-AIX Version 4.3.1 mode.

11.8.4 The startsrc command

The **startsrc** command sends the System Resource Controller (SRC) a request to start a subsystem or a group of subsystems or to pass on a packet to the subsystem that starts a subserver.

If a start subserver request is passed to the SRC, and the subsystem to which the subserver belongs is not currently active, the SRC starts the subsystem and transmits the start subserver request to the subsystem.

The flags for the **startsrc** command are shown in Table 11-6.

Table 11-6 Commonly used flags for the startsrc command

Flag	Description	Example
To start a subsystem		
-a *argument*	Specifies an argument string that is passed to the subsystem when the subsystem is executed.	**startsrc -s srctest -a "-D DEBUG"** This starts the srctest subsystem with "-D DEBUG" as two arguments to the subsystem.
-e *Environment*	Specifies an environment string that is placed in the subsystem environment when the subsystem is executed.	**startsrc -s srctest -e "TERM=dumb HOME=/tmp"** This starts the srctest subsystem with "TERM=dumb", "HOME=/tmp" in its environment to the subsystem.
-g *Group*	Specifies a group of subsystems to be started.	**startsrc -g nfs** This starts all the subsystems in the subsystem nfs group
-s *Subsystem*	Specifies a subsystem to be started.	**startsrc -s srctest** This stops the srctest subsystem.
To start either a subsystem or a subserver		
-h *Host*	Specifies the foreign host on which this start action is requested. The local user must be running as root. The remote system must be configured to accept remote System Resource Controller requests.	**startsrc -g nfs -h itsosmp** This starts all the subsystems in the nfs group on the itsosmp machine.
To start a subserver		
-t *Type*	Specifies that a subserver is to be started.	**startsrc -t tester** This sends a start subserver request to the subsystem that owns the tester subsystem.

Flag	Description	Example
-o *Object*	Specifies that a subserver object is to be passed to the subsystem as a character string. It is the subsystem's responsibility to determine the validity of the Object string.	**startsrc -o tester -p 1234** The subserver tester is passed as a character string to the subsystem with a PID of 1234.
-p *SubsystemPID*	Specifies a particular instance of the subsystem to which the start subserver request is to be passed.	**startsrc -t tester -p 1234** This starts the tester subserver that belongs to the srctest subsystem with a subsystem PID of 1234.

11.8.5 Refreshing a daemon

Use the **refresh** command to tell a System Resource Controller (SRC) resource, such as a subsystem or a group of subsystems, to refresh itself.

The prerequisites for using the **refresh** command are:

- The SRC must be running.
- The resource you want to refresh must not use the signals communications method.
- The resource you want to refresh must be programmed to respond to the refresh request.

The **refresh** command sends the System Resource Controller a subsystem refresh request that is forwarded to the subsystem. The refresh action is subsystem-dependent.

To start the Lotus® Domino® Go Web server, use the following command:

```
# startsrc -s httpd
```

To allow users to open a home page, index.html, in a new directory, /newdir, you have added a directory mapping in the /etc/httpd.conf file:

```
pass        /*        /newdir/*
```

To refresh the Web server, enter:

```
# refresh -s httpd
```

After this, the users will be able to access the new home page by entering the following URL in their Web browser:

http://*server_name*[:*port_number*]/newdir/index.html

11.8.6 The stopsrc command

The **stopsrc** command sends a request to the System Resource Controller (SRC) to stop a subsystem, a group of subsystems, or all subsystems. The **stopsrc** command sends the System Resource Controller a subsystem request packet that is forwarded to the subsystem for a stop subserver request.

In the absence of the -f (stop force) flag, a normal stop action is assumed. A normal stop requests that a subsystem or subserver complete all current processing, release resources when all application activity has been completed, and then end. No new requests for work should be accepted by the subsystem.

A forced stop requests that a subsystem or subserver end quickly, releasing all resources, but not wait for application activity to complete.

The flags for the **stopsrc** command are shown in Table 11-7.

Table 11-7 Commonly used flags for the stopsrc command

Flag	Description	Example
-a	Specifies that all subsystems are to be stopped.	**stopsrc -a** This stops all the active subsystems on the local machine.
-g Group	Specifies that a group of subservers is to be stopped.	**stopsrc -g tcpip** This stops all subsystems in the tcpip group.
-h Host	Specifies the foreign Host machine on which this stop action is requested. The local user must be running as root. The remote system must be configured to accept remote System Resource Controller requests.	**stopsrc -h zork -s srctest** This stops all instances of the srctest subsystem on the zork machine.
-o Object	Specifies that a subserver Object value is to be passed to the subsystem as a character string.	**startsrc -o tester -p 1234** The subserver tester is passed as a character string to the subsystem with a PID of 1234.

Flag	Description	Example
-p SubsystemPID	Specifies a particular instance of the subsystem to stop, or a particular instance of the subsystem to which the stop subserver request is to be passed.	**stopsrc -t tester -p 1234** This stops the tester subserver that belongs to the srctest subsystem with a subsystem PID of 1234.
-s Subsystem	Specifies a subsystem to be stopped.	**stopsrc -s srctest** This stops the srctest subsystem.
-t Type	Specifies that a subserver is to be stopped.	**stopsrc -t tester** This sends a stop subserver request to the subsystem that owns the tester subsystem.

11.9 Commands and processes

A command is a request to perform an operation or run a program. You use commands to tell the operating system what task you want it to perform. When commands are entered, they are deciphered by a command interpreter (also known as a shell), and that task is processed.

A program or command that is actually running on the computer is referred to as a *process*. The operating system can run many different processes at the same time.

A single user can run more than one process at a time. To check how many process a user can run, type the following command:

```
# lsattr -E -l sys0 -a maxuproc
maxuproc 128 Maximum number of PROCESSES allowed per user True
#
```

To change this value, in this example from 128 to 256, type the following command:

```
# chdev -l sys0 -a maxuproc=256
sys0 changed
#
```

Change takes effect immediately and is preserved over boot. If value is reduced, then it goes into effect only after a system boot.

Note: Changing the number of processes a user can run is a performance tuning topic, so use it carefully, since this is a safeguard to prevent users from creating too many processes.

The common types of processes are as follows:

- Foreground and background processes

  Processes that require a user to start them or to interact with them are called foreground processes. Processes that are run independently of a user are referred to as background processes. Programs and commands run as foreground processes by default.

- Daemon processes

  Daemons are processes that run unattended. They are constantly in the background and are available at all times. Daemons are usually started when the system starts, and they run until the system stops. A daemon process typically performs system services and is available at all times to more than one task or user. Daemon processes are started by the root user or root shell and can be stopped only by the root user. For example, the **qdaemon** process provides access to system resources such as printers. Another common daemon is the **sendmail** daemon.

- Zombie processes

  A zombie process is a dead process that is no longer executing but is still recognized in the process table (in other words, it has a PID number). It has no other system space allocated to it. Zombie processes have been killed or have exited and continue to exist in the process table until the parent process dies or the system is shut down and restarted. Zombie processes display as <defunct> when listed by the **ps** command.

11.9.1 Controlling foreground processes

You start a foreground process from a display station by either entering a program name or command name at the system prompt. After a foreground process has started, the process interacts with you at your display station until it is complete. No other interaction (for example, entering another command) can take place at the display station until the process is finished or you halt it.

Canceling a foreground process

If you start a foreground process and then decide that you do not want it to finish, you can cancel it by pressing the INTERRUPT key. This is usually Ctrl-C or Ctrl-Backspace.

Note: For the purposes of this redbook, we will assume Ctrl-C is the INTERRUPT key. You can check how this key is set on your system by running the **stty -a** command:

```
# stty -a
speed 9600 baud; 24 rows; 80 columns;
eucw 1:1:0:0, scrw 1:1:0:0:
intr = ^C; quit = ^\; erase = ^?; kill = ^U; eof = ^D; eol = ^@
eol2 = ^@; start = ^Q; stop = ^S; susp = ^Z; dsusp = ^Y; reprint = ^R
.
.
#
```

Most simple commands run so quickly that they finish before you have time to cancel them. The examples in this section, therefore, use a command that takes more than a few seconds to run: **find / -type f**. This command displays the path names for all files on your system. You do not need to study the **find** command in order to complete this section; it is used here simply to demonstrate how to work with processes.

In the following example, the **find** command starts a process. After the process starts, you can cancel it by pressing Ctrl-C:

```
# find / -type f
/var/adm/cron/at.deny
/var/adm/cron/cron.deny
/var/adm/cron/log
/var/adm/cron/queuedefs
/var/adm/ras/codepoint.cat
/var/adm/ras/BosMenus.log
/var/adm/ras/errtmplt
/var/adm/ras/bosinst.data
/var/adm/ras/devinst.log
/var/adm/ras/suma.log
/var/adm/ras/bosinstlog
^C#
```

Stopping a foreground process

It is possible for a process to be stopped but not have its process ID (PID) removed from the process table. You can stop a foreground process by pressing Ctrl-Z from the keyboard. This key sequence sends a STOP signal to the current process. The shell normally indicates that the process has been stopped and then displays a shell prompt.

In the following example, the **find** command starts a process. After the process starts, you can stop it by pressing Ctrl-Z:

```
# find / -type f
/var/adm/cron/at.deny
/var/adm/cron/cron.deny
/var/adm/cron/log
/var/adm/cron/queuedefs
/var/adm/ras/codepoint.cat
/var/adm/ras/BosMenus.log
/var/adm/ras/errtmplt
/var/adm/ras/bosinst.data
/var/adm/ras/devinst.log
[1] + Stopped (SIGTSTP)        find / -type f
#
```

The Ctrl-Z key sequence takes effect immediately, and is like an interrupt in that the shell discards pending output and unread input when you type the sequence.

Note: Ctrl-Z works in the Korn shell (ksh) and C shell (csh), but not in the Bourne shell (bsh).

Restarting a foreground stopped process

This procedure describes how to restart a process that has been stopped with Ctrl-Z.

The ps command

You can use the **ps** command to obtain which processes are running and display information about those processes. This command has several flags that enable you to specify which processes to list and what information to display about each process.

For example, to show all the processes running or stopped on your system, type the following command:

```
# ps -ef
```

You might want to pipe this command through a **grep** command to restrict the list to those processes most likely to be the one you want to restart. For example, if you want to restart the previous **find** command, you could execute the following sequence of commands:

1. Run the **ps** command and display only those processes that contain the word `find`:

```
# ps -ef | grep find
    root 589934 712768   0 10:22:26  pts/1  0:00 find / -type f
    root 610432 712768   0 10:24:32  pts/1  0:00 grep find
#
```

2. In the **ps** command output, find the process you want to restart and note its PID number. In the example, the PID is 589934.

3. To bring the process in to the foreground, run the **fg** command:

```
# fg 589934
/var/adm/ras/suma.log
/var/adm/ras/bosinstlog
/var/adm/ras/image.data
/var/adm/ras/bootlog
/var/adm/ras/conslog
/var/adm/ras/errlog
.
.
.
(limes omitted)
.
.
.
#
```

Note: To restart a stopped process, you must either be the user who started the process or have root user authority.

11.9.2 Controlling background processes

To run a process in the background, type the name of the command with the appropriate parameters and flags, followed by an ampersand (&):

```
CommandName&
```

When a process is running in the background, you can perform additional tasks by entering other commands at the command prompt.

Generally, background processes are most useful for commands that take a long time to run. However, because they increase the total amount of work the processor is doing, background processes can slow down the rest of the system.

Most processes direct their output to standard output (stdout), even when they run in the background. Unless redirected, stdout goes to the display device. Because the output from a background process can interfere with your other work on the system, it is usually good practice to redirect the output of a background process to a file or a printer. You can then look at the output whenever you are ready.

Removing a background process

If you decide, after starting a background process, that you do not want the process to finish, you can cancel the process with the **kill** command; you might want to do this if you realize that you have mistakenly put a process in the background or that a process is taking too long to run.

The general format for the **kill** command is as follows:

```
kill ProcessID
```

The **kill** command sends a signal (by default, the SIGTERM signal) to a running process. This default action normally stops processes.

> **Note:** The name of the **kill** command is misleading because many signals do not stop processes.

In the following example, you execute the **find** command to run in the background and to send its output to the dir.paths file. Then you decide to cancel the process:

1. Run the **find** command in the background:

   ```
   # find / -type f > dir.paths &
   [1]     344144
   ```

2. Run the **ps** command without parameters to list the PID numbers:

   ```
   # ps
       PID    TTY  TIME CMD
    323808  pts/1  0:00 ps
    344144  pts/1  0:00 find / -type f
    417896  pts/1  0:00 -ksh
   #
   ```

3. Identify the PID or job number (in this example the PID is 344144) and run the **kill** command to cancel the process:

   ```
   # kill 344144
   ```

4. Run the **ps** command to show the active processes:

```
# ps
    PID    TTY  TIME CMD
 188440  pts/1  0:00 ps
 417896  pts/1  0:00 -ksh
[1] + Terminated               find / -type f > dir.paths &
#
```

Note: To remove a process, you must have root user authority or be the user who started the process. The default signal to a process from the **kill** command is -15 (SIGTERM).

11.9.3 Job control in the Korn Shell or POSIX shell

The Korn shell, or POSIX shell, provides a facility to control command sequences, or jobs. When you execute the **set -m** special command, the Korn shell associates a job with each pipeline. It keeps a table of current jobs, printed by the jobs command, and assigns them small integer numbers.

When a job is started in the background with an ampersand (&), the shell prints a line that looks like the following:

```
[1] 1234
```

For example, running the **find** command in the background:

```
# find / -type f > dir.paths &
[1]     262174
#
```

This output indicates that the job, which was started in the background, was job number 1 (enclosed in square brackets). It also shows that the job has a process ID of 262174.

You can manipulate the state of this job: returning it to the foreground with the **fg** command, stopping it (for example, by pressing Ctrl-Z), or putting it again in the background with the **bg** command.

A job being run in the background stops if it tries to read from the terminal. Background jobs are normally allowed to produce output. You can disable this option by issuing the **stty tostop** command. If you set this terminal option, then background jobs stop when they try to produce output or read input.

You can refer to jobs in the Korn shell in several ways. A job is referenced by the process ID of any of its processes or in one of the following ways:

%Number Specifies the job with the given number

%String Specifies any job whose command line begins with the *String* variable.

%?String Specifies any job whose command line contains the *String* variable.

%% Specifies the current job.

%+ Equivalent to %%.

%- Specifies the previous job.

For example, stop a **find** process which is already running on the background, bring it to the foreground, stop it again, and return it to the background:

1. Run the **find** command on the background:

   ```
   # find / -type f > dir.paths &
   [1]     962718
   #
   ```

2. Stop the background process by sending it a STOP signal:

   ```
   # kill -18 962718
   [1] + Stopped (SIGTSTP)        find / -type f > dir.paths &
   #
   ```

3. Bring the stopped process to the foreground with the **fg** command:

   ```
   # fg %1
   find / -type f > dir.paths
   ```

4. Stop the foreground process by pressing Ctrl-Z:

   ```
   ^Z[1] + Stopped (SIGTSTP)        find / -type f > dir.paths &
   #
   ```

5. Return the stopped process to the background with the **bg** command:

   ```
   # bg %1
   [1]     find / -type f > dir.paths &
   #
   ```

The Korn shell or POSIX shell immediately recognizes changes in the process state. It normally informs you whenever a job becomes blocked so that no further progress is possible. The shell does this just before it prints a prompt so that it does not otherwise disturb your work.

If you try to leave the shell (either by typing exit or using the Ctrl-D key sequence) while jobs are stopped or running, the system warns you with the message `There are stopped (running) jobs`.

For example, run the **find** command in the background and then try to exit the current session:

```
# find / -type f > dir.paths &
[1]     962762
# exit
There are running jobs.
#
```

Use the **jobs** command to see which jobs are affected.

```
# jobs
[1] +  Running                  find / -type f > dir.paths &
#
```

If you immediately try to exit again, the shell terminates the stopped and running jobs without warning. To run processes in the background after logging off, use the **nohup** command.

11.9.4 The nohup command

The syntax for this command is:

```
nohup { -p pid | command [ arg ... ] [ & ] }
```

The **nohup** command runs the command specified by the command parameter and any related arg parameters, ignoring all hangup (SIGHUP) signals or modifies the process specified with the -p option to ignore all hangup (SIGHUP) signals.

To run a **nohup** command in the background, add an & (ampersand) to the end of the command.

For example, if you want to run the **find** command and leave it running after you log off, type:

```
# nohup find / -type f &
[1]     950484
# Sending nohup output to nohup.out.
```

The message `Sending output to nohup.out` informs you that the output from the **find** command is in the nohup.out file; if you want to save the output on another file, just redirect the output.

You can log off after you see these messages, even if the **find** command is still running.

11.10 Metacharacters and wild cards

Metacharacters and wild cards are useful when performing file name substitution; they provide a convenient short way to specify multiple file or directory names instead of typing each one of them. The wildcard characters are asterisk (*) and question mark (?). The metacharacters are open and close square brackets ([ ]), hyphen (-), and exclamation mark (!).

11.10.1 Matching patterns using the * wild card character

Use the asterisk (*) to match any (zero or more) sequence or string of characters. The (*) indicates any character, including non characters.

See the following examples:

1. Create the following files in an empty directory:

```
# mkdir wildcards
# cd wildcards
# touch 1test 2test afile1 afile2 bfile1 file file1 file10 file2 file3
```

2. List only the files that begin with the word *file* followed by anything else:

```
# ls file*
file    file1   file10  file2   file3
#
```

3. List the files that contain anywhere on its name the number 2:

```
# ls *2*
2test   afile2  file2
#
```

11.10.2 Matching patterns using the ? wild card character

Use the ? (question mark) to match any single character.

See the following examples:

1. List only the files that start with the word *file* and end with a single character:

```
# ls file?
file1  file2  file3
#
```

2. List only the files that start with the word *file* and end with any two characters:

```
# ls file??
file10
#
```

11.10.3 Matching patterns using [ ] shell metacharacters

Metacharacters offer another type of wildcard notation by enclosing the desired characters within square brackets: [ ]. It is like using the ?, as it will only match one single character, but it allows you to choose specific characters to be matched. The [ ] also allows you to specify a range of values using the hyphen (-).

See the following examples:

1. List the files that starts with any string followed by the word *file* and that end in 1 or 2:

```
# ls *file[12]
afile1  afile2  bfile1  file1   file2
#
```

2. List only the files that start with any number:

```
# ls [0123456789]*
1test  2test
#
```

3. List only the files that do not begin with an a:

```
# ls [!a]*
1test   2test   bfile1  file    file1   file10  file2   file3
#
```

11.10.4 Quoting in the Korn shell or POSIX shell

When you want the Korn shell or POSIX shell to read a character as a regular character, rather than with any normally associated meaning, you must quote it. To negate the special meaning of a metacharacter, use one of the quoting mechanisms in the following list.

Each metacharacter has a special meaning to the shell and, unless quoted, causes termination of a word. The following characters are considered metacharacters by the Korn shell or POSIX shell and must be quoted if they are to represent themselves:

Pipe	|
Ampersand	&
Semicolon	;
Less-than sign	<
Greater-than sign	>
Left parenthesis	(

Right parenthesis)

Dollar sign $

Backquote `

Single quotation mark '

Backslash \

Double-quotation marks "

New-line character

Space character

Tab character

The quoting mechanisms are the backslash (\), single quotation mark ('), and double quotation marks (").

Backslash

A backslash (\) that is not quoted preserves the literal value of the following character, with the exception of a new-line character. If a new-line character follows the backslash, the shell interprets this as line continuation.

Single quotation marks

Enclosing characters in single quotation marks ('') preserves the literal value of each character within the single quotation marks. A single quotation mark cannot occur within single quotation marks. A backslash cannot be used to escape a single quotation mark in a string that is set in single-quotation marks.

For example:

1. Display on the screen the message `Hello world`:

```
# echo 'Hello world'
Hello world
#
```

2. Display on the screen the message `Hello world` enclosed on single quotation marks:

```
# echo \''Hello world'\'
'Hello world'
#
```

Double quotation marks

Enclosing characters in double quotation marks (" ") preserves the literal value of all characters within the double quotation marks, with the exception of the dollar sign, backquote, and backslash characters, as follows:

Dollar sign ($)

The dollar sign retains its special meaning of introducing parameter expansion, a form of command substitution, and arithmetic expansion.

The input characters within the quoted string that are also enclosed between $(and the matching) will not be affected by the double quotation marks.

For example:

1. Define two variables:

```
# export H=Hello
# export W=world
#
```

2. Use quotation marks to display these two variables:

```
# echo "$H $W"
Hello world
#
```

3. Use an arithmetic expansion inside quotation marks:

```
# echo "$H $W $((8+2)) times"
Hello world 10 times
#
```

4. Use a command substitution inside quotation marks:

```
# echo "$H $W from host $(hostname)"
Hello world from host server2
#
```

Parameter and command substitution occurs inside double quotation marks, but not inside single quotation marks. For example:

1. Use backquotes inside double quotation marks:

```
# echo "Today is `date +%D`"
Today is 11/14/05
#
```

2. Use backquotes inside single quotation marks:

```
# echo 'Today is `date +%D`'
Today is `date +%D`
#
```

Backquote (`)

The backquote retains its special meaning of introducing the other form of command substitution. The portion of the quoted string, from the initial backquote and the characters up to the next backquote that is not preceded by a backslash, defines that command whose output replaces ` ... ` when the word is expanded.

For example:

- Use command substitution with backquotes inside quotation marks:

```
# echo "This host's name is: `hostname`"
This host's name is: server2
#
```

*Backslash *

The backslash retains its special meaning as an escape character only when followed by one of the following characters: $, `, ", \, or a newline character.

A double quotation mark must be preceded by a backslash to be included within double quotation marks. For example:

1. Display the message `Hello world`:

```
# echo Hello world
Hello world
#
```

2. Display the message `Hello world` enclosed within double quotation marks:

```
# echo "\"Hello world\""
"Hello world"
#
```

When you use double quotation marks, if a backslash is immediately followed by a character that would be interpreted as having a special meaning, the backslash is deleted, and the subsequent character is taken literally. If a backslash does not precede a character that would have a special meaning, it is left in place unchanged, and the character immediately following it is also left unchanged.

For example:

1. Define one variable:

```
# export $H=Hello
#
```

2. Precede the dollar sign with a backslash to preserve its literal value:

```
# echo "Value of \$H variable is: $H"
Value of $H variable is: Hello
#
```

3. Precede characters that are not metacharacters with a backslash:

```
# echo \H\e\l\l\o \w\o\r\l\d
Hello world
#
```

11.10.5 Metacharacter examples

The following examples show you how you can use metacharacters to achieve your daily administration tasks:

Example 1

You want to list the active volume groups and show the hour on which you issued this verification:

```
# echo "Active vgs on: `date +\"%D %H:%M:%S\"`\n$(lsvg -o)"
“Active vgs on: 11/14/05 18:00:44”
testvg
rootvg
#
```

Example 2

You have the following list of files on your system:

```
# ls
Xname.be      Xname.br      Xname.mx      Xname.ro      Xname.us
Xname.be.toc  Xname.br.doc  Xname.mx.doc  Xname.ro.doc  Xname.us.doc
Xname.be.txt  Xname.br.txt  Xname.mx.txt  Xname.ro.txt  Xname.us.txt
#
```

And you want to create a tar file containing all the files except for those with the .doc extension (notice that there is a file with .toc extension, and it must be included on the tar file):

```
# tar -cvf Xnames.tar *[!d]??
a Xname.be 1 blocks.
a Xname.be.toc 1 blocks.
a Xname.be.txt 1 blocks.
a Xname.br 1 blocks.
a Xname.br.txt 1 blocks.
a Xname.mx 1 blocks.
a Xname.mx.txt 1 blocks.
a Xname.ro 1 blocks.
a Xname.ro.txt 1 blocks.
a Xname.us 1 blocks.
a Xname.us.txt 1 blocks.
#
```

Example 3

You want to add a command to the /etc/inittab file that prints the date at inittab's execution into a file. The example includes command substitution syntax, double quotation marks and stdout redirection:

1. Using double quotation marks:

```
# mkitab "echomsg:2:once:echo \`date +\"%m/%d/%y\"\` >> /tmp/echomsgs"
#
```

   The result on the /etc/inittab file will look like:

```
# lsitab echomsg
echomsg:2:once:echo `date +"%m/%d/%y"` >> /tmp/echomsgs
#
```

   If you avoid escaping backquotes, the result looks like:

```
# mkitab "echomsg:2:once:echo `date +\"%m/%d/%y\"` >> /tmp/echomsgs"
# lsitab echomsg
echomsg:2:once:echo 11/16/05 >> /tmp/echomsgs
#
```

   which is not compliant with our objective, because the output is statically set to 11/16/05 in this example.

2. Using single quotation marks:

```
# mkitab 'echomsg:2:once:echo `date +"%m/%d/%y"` >> /tmp/echomsgs'
# lsitab echomsg
echomsg:2:once:echo `date +"%m/%d/%y"` >> /tmp/echomsgs
#
```

11.11 Redirecting stdin, stdout, and stderr

The operating system allows you to manipulate the input and output (I/O) of data to and from your system by using specific I/O commands and symbols. You can control input by specifying the location from which to gather data. For example, you can specify to read input entered on the keyboard (standard input) or to read input from a file. You can control output by specifying where to display or store data. For example, you can specify to write output data to the screen (standard output) or to write it to a file.

11.11.1 File descriptors

When a command begins running, it usually expects that the following files are already open: standard input, standard output, and standard error. A number, called a file descriptor, is associated with each of these files. The default assignments for these descriptors are as follows:

- **0** Represents standard input (stdin)
- **1** Represents standard output (stdout)
- **2** Represents standard error (stderr)

These file descriptors are connected by default to the terminal, so that if a program reads file descriptor 0 and writes file descriptors 1 and 2, the program collects input from the terminal and sends output also to the terminal.

11.11.2 Redirecting input (stdin)

When the notation `< filename` is added to the end of a command, the default file descriptor for input of the command is changed, and input is read from the specified file name. The < (less than) symbol is known as the input redirection operator.

> **Note:** Only commands that normally take their input from the keyboard can have their input redirected.

For example, to send the file letter1 as a message to user denise with the **`mail`** command, type:

```
# mail denise < letter1
#
```

Using inline input (here) documents

If a command is followed by the notation << *`eofstring`* and *`eofstring`* is any string that does not contain pattern-matching characters (for example *), then the shell takes the subsequent lines as the standard input of the command until the shell reads a line consisting of only *`eofstring`* (possibly preceded by one or more tab characters). The lines between the first *`eofstring`* and the second are frequently referred to as an inline input, or here document. If a hyphen (-) immediately follows the << redirection characters, the shell strips leading tab characters from each line of the here document before it passes the line to the command.

The here document is especially useful for a small amount of input data that is more conveniently placed in the shell procedure rather than kept in a separate file (such as editor scripts).

For example, you could type:

```
# cat <<- xyz
>       This message will be shown on the
>       display with leading tabs removed.
>       xyz
This message will be shown on the
display with leading tabs removed.
#
```

11.11.3 Redirecting output (stdout)

The output of a command can be redirected to a file typing the command followed by the notation > filename. For example, to save the value of the PATH variable on a file, type:

```
# echo $PATH > path
#
```

To see the contents of the path file, type:

```
# cat path
/usr/bin:/etc:/usr/sbin:/usr/ucb:/usr/bin/X11:/sbin:/usr/java14/jre/bin:/usr/ja
va14/bin
#
```

Redirecting output and appending to a file

When the notation >> filename is added to the end of a command, the output of the command is appended to the specified file name, rather than writing over any existing data. The >> symbol is known as the append redirection operator.

For example, we want to add the contents of file2 at the end of file1:

```
# cat file1
This is line1
# cat file2
This is line 2
```

To append file2 to file1, type:

```
#cat file2 >> file1
#
```

The result is:

```
# cat file1
This is line 1
This is line 2
#
```

Note: If the file1 file does not exist, it is created, unless the noclobber option of the set built-in ksh (Korn shell) or csh (C shell) command is specified.

Creating a text file with redirection from the keyboard

Used alone, the **cat** command uses whatever you type at the keyboard as input. You can redirect this input to a file. Enter Ctrl-D on a new line to signal the end of the text.

At the system prompt, type:

```
# cat > test
This is a test.
^D
```

The contents of the file test is:

```
# cat test
This is a test.
#
```

Concatenating text files

Combining various files into one single file is known as concatenation.

For example, we have three files, and we want to integrate the information at these three files into a single one. The contents of each file is:

```
# cat file1
This is line 1
# cat file2
This is line 2
# cat file3
This is line3
#
```

Type the following command to concatenate all of them on file4:

```
# cat file1 file2 file3 > file4
#
```

Looking at the contents of file4 gives us the following result:

```
# cat file4
This is line 1
```

```
This is line 2
This is line 3
#
```

The previous example created file4, which consists of file1, file2, and file3, appended in the order given.

The following example shows a common error when concatenating files:

```
# cat file1 file2 file3 > file1
#
```

Note: In this example, you might expect the **cat** command to append the contents of file1, file2, and file3 into file1. The **cat** command creates the output file first, so it actually erases the contents of file1 and then appends file2 and file3 to it.

11.11.4 Redirecting standard error (stderr)

In addition to the standard input and standard output, commands often produce other types of output, such as error or status messages known as diagnostic output. Like standard output, standard error output is written to the screen unless redirected.

To redirect standard error output, type the file descriptor number 2 before the output or append redirection symbols (> or > >) and a file name after the symbol. For example, the following command tries to list a file that does not exist on the system:

```
# ls -l dailyTasks.sh
ls: 0653-341 The file dailyTasks.sh does not exist.
#
```

Redirecting stdout gives us this result:

```
# ls -l dailyTasks.sh > list
ls: 0653-341 The file dailyTasks.sh does not exist.
#
```

If we look at the contents of file list:

```
# cat list
#
```

it has no content at all.

In this case, the command is sending its output to stderr, so in order to save this message on a file, you should type:

```
# ls -l dailyTasks.sh 2> list
#
```

No message is sent to the screen. If we look at the contents of file list, we can see that it now has contents in it:

```
# cat list
ls: 0653-341 The file dailyTasks.sh does not exist
#
```

11.11.5 Examples

You can redirect stdin, stdout, and stderr individually or group them (stdout and stderr only). For example, to list the files on a directory, run:

```
# ls *.dat *.txt > files.out 2> files.err
#
```

If we look at the contents of file files.out:

```
# cat files.out
names.dat
source.dat
people.dat
#
```

and the contents of files.err:

```
# cat files.err
ls: 0653-341 The file *.txt does not exist
#
```

you may find some cases where you just want to keep all the output (stdout and stderr) in one single file. You can do this by using the following notation:

```
command > output 2>&1
```

The listing files example above can be changed to:

```
# ls *.dat *.txt > files.out 2>&1
#
```

And the contents of the file files.out looks like:

```
# cat files.out
ls: 0653-341 The file *.txt does not exist
names.dat
source.dat
people.dat
#
```

> **Note:** The ampersand character (&) after the greater than symbol (>) does not mean that this command should run as a background process; instead, you are instructing the shell that stderr must be redirected where stdout is.

For example, you are given the task to automate an **ftp** command transfer and log all the events related to that transfer:

```
#ftp -n -iv server <<EOT >>results.ftp
user yourUser yourPassword
ascii
get file1
bye
EOT
#
```

Summary

As we have shown, the following symbols redirect I/O, and when used, the shell's default file descriptor assignments are changed. You can combine them to meet your administration tasks and keep control when using I/O operations on commands or shell scripts.

Table 11-8 provides a list of redirection symbols and their meanings.

Table 11-8 Redirection symbols

Symbol	Meaning
< (less than)	Redirects input (stdin).
> (greater than)	Redirects output (stdout).
>>	Appends output.
<<	Inline input (here documents).
2>	Redirects output (stderr).
1>&2	Redirects stdout to stderr.
2>&1	Redirects stderr to stdout.

12

Electronic mail and sendmail

The mail facility provides a method for exchanging electronic mail between the users on the same system or on multiple systems connected by a network. This chapter discusses mail configuration tasks, mail configuration files, mail aliases, and mail logs.

© Copyright IBM Corp. 2006. All rights reserved.

12.1 Overview of mail system

The mail system is an internetwork mail delivery facility that consists of a user interface, a message routing program, and a message delivery program (or mailer).

A mail user agent (MUA) enables users to create, send, and receive messages from other users. The mail system provides two user interfaces: **mail** and **mhmail**. The **mail** command is the standard mail user interface available on all UNIX systems. The **mhmail** command is the Message Handler (MH) user interface, an enhanced mail user interface designed for experienced users.

A message routing program (MTA) routes messages to their destinations. The mail system's message routing program is the **sendmail** command. Depending on the type of route to the destination, the **sendmail** command uses different mailers to deliver messages, as shown in Figure 12-1.

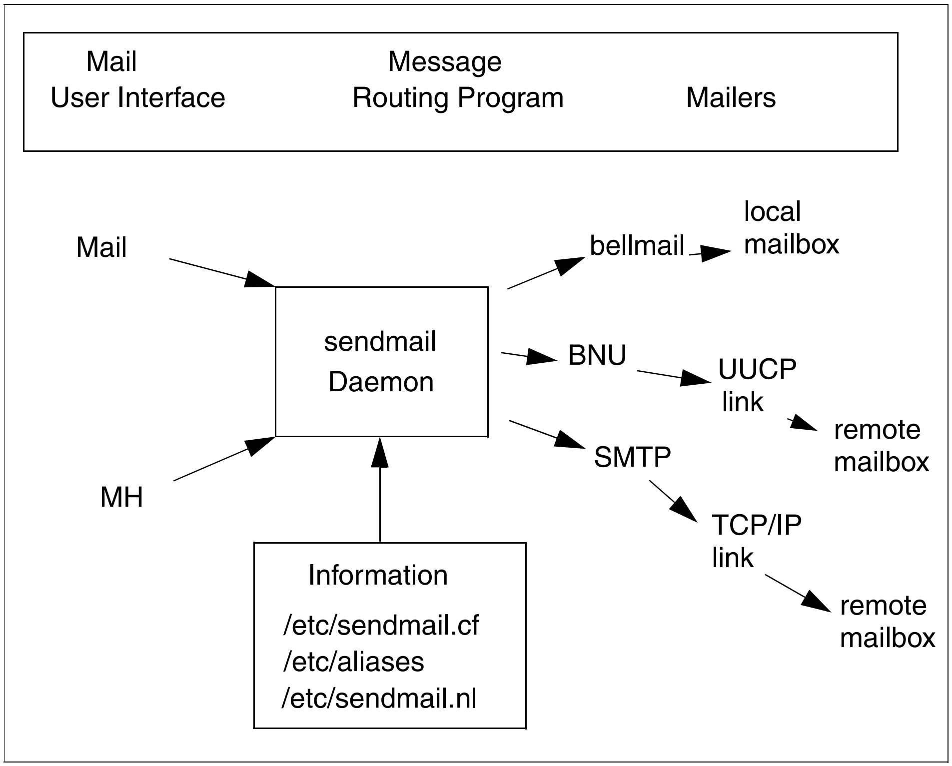


Figure 12-1 Overview of mail system

To deliver local mail, the **sendmail** program routes messages to the **bellmail** program. The **bellmail** program delivers all local mail by appending messages to the user's system mailbox, which is in the /var/spool/mail directory.

To deliver mail over a UNIX-to-UNIX Copy Program (UUCP) link, the **sendmail** program routes messages using Basic Network Utilities (BNU).

To deliver Transmission Control Protocol/Internet Protocol (TCP/IP)-routed mail, the **sendmail** command establishes a TCP/IP connection to the remote system and then uses Simple Mail Transfer Protocol (SMTP) to transfer the message to the remote system.

Figure 12-2 shows the mail management tasks for a system administrator.

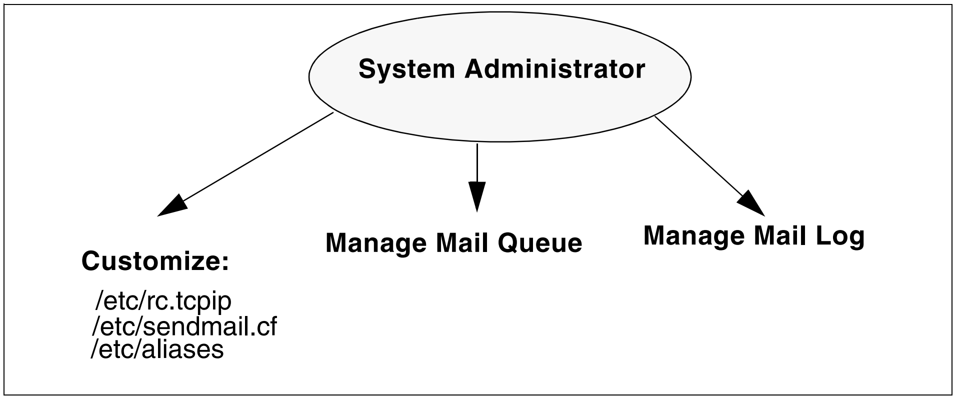


Figure 12-2 Mail management tasks

12.2 Mail daemons

The following is a description of the daemons used by the mail system:

sendmail daemon A message routing program routes messages to their destinations. The mail system's message routing program is the **sendmail** command, which is part of the Base Operating System (BOS) and is installed with the BOS. It uses information in the /etc/sendmail.cf file, the /etc/aliases file, and the /etc/sendmail.nl file to perform the necessary routing. The **sendmail** command must be run in the daemon mode to receive mail that is sent to it over the network.

Syslogd daemon The **sendmail** command logs mail system activity through the **syslogd** daemon. The **syslogd** daemon must be configured and running for logging to occur. Refer to 12.4, “Mail logs” on page 588 for more information about the **syslogd** daemon.

12.2.1 Starting the sendmail daemon

To start the **sendmail** daemon, enter either of the following commands:

```
#startsrc -s sendmail
```

or

```
#sendmail
```

If the **sendmail** daemon is already active when you enter one of these previous commands, you will see the following message on the screen:

```
The sendmail subsystem is already active. Multiple instances are not
supported.
```

If the **sendmail** daemon is not already active, then a message indicating that the **sendmail** daemon has been started will be generated.

12.2.2 Stopping the sendmail daemon

Execute the **stopsrc -s** command to stop the **sendmail** daemon. If the **sendmail** daemon was not started with the **startsrc** command, then kill the process by issuing the following command:

```
#kill -9 ‘cat /etc/mail/sendmail.pid‘
```

12.2.3 Refreshing the sendmail daemon

To refresh the **sendmail** daemon, issue the command:

```
#refresh -s sendmail
```

12.2.4 Getting the status of sendmail daemon

The following example shows how to get the status of the **sendmail** daemon using **lssrc** command with the -s flag. The status can be active or inoperative.

```
# lssrc -s sendmail
Subsystem         Group            PID     Status
 sendmail         mail             5422    active
```

12.2.5 Autostart of the sendmail daemon (/etc/rc.tcpip)

The **sendmail** daemon is started from the /etc/rc.tcpip file. To configure this file so that the **sendmail** daemon will be started at system boot time:

1. Edit the /etc/rc.tcpip file.
2. Find the line that begins with `start /usr/lib/sendmail`. By default, this line should be uncommented; that is, there is no # (pound sign) at the beginning of the line. However, if it is commented, delete the pound sign.

12.2.6 Specifying time values in sendmail (in rc.tcpip)

The interval at which the **sendmail** daemon processes the mail queue is determined by the value of the -q flag when the daemon starts. The **sendmail** daemon is usually started by the /etc/rc.tcpip file at system startup. The /etc/rc.tcpip file contains a variable called the queue processing interval (QPI), which it uses to specify the value of the -q flag when it starts the **sendmail** daemon. By default, the value of QPI is 30 minutes. To specify a different queue processing interval:

1. Edit the /etc/rc.tcpip file.
2. Find the line that assigns a value to the qpi variable, such as qpi=30m.
3. Change the value assigned to the qpi variable to the time value you prefer using the values shown in the next section.

These changes will take effect at the next system restart. For the changes to take effect immediately, stop and restart the **sendmail** daemon while specifying the new -q flag value.

12.2.7 Specifying time values in sendmail (not in rc.tcpip)

To set the message time-out and queue processing interval, you must use a specific format for the time value. The format of a time value is -qNumberUnit, where Number is an integer value and Unit is the unit letter. Unit may have one of the following values:

s Seconds

m Minutes

h Hours

d Days

w Weeks

If Unit is not specified, the **sendmail** daemon uses minutes (m) as the default. Here are three examples:

To process the queue every 15 days, issue the command:

```
#sendmail -q15d
```

To process the queue every 15 hours, issue the command:

```
#sendmail -q15h
```

To process the queue every 15 minutes, issue the command:

```
#sendmail -q15
#sendmail -q15m
```

12.3 Mail queue directory: /var/spool/mqueue

The mail queue is a directory that stores data and control files for mail messages that the **sendmail** command delivers. By default, the mail queue is /var/spool/mqueue. Mail messages may be queued for several reasons. First, the **sendmail** command can be configured to process the queue at certain intervals rather than immediately. If this is so, mail messages must be stored temporarily. Second, if a remote host does not answer a request for a mail connection, the mail system queues the message and tries again later.

12.3.1 Printing the mail queue

The contents of the queue can be printed using the **mailq** command (or by specifying the -bp flag with the **sendmail** command). This produces a listing of the queue IDs, the size of the message, the date the message entered the queue, and the sender and recipients.

12.3.2 Mail queue files

The mail queue directory /var/spool/mqueue contains four types of mail queue files:

- Data file
- Control file
- Temporary file
- Transcript file

Each message in the queue has four files associated with it. For example, if a message has a queue ID of AA00269, the following files are created and deleted in the mail queue directory while the **sendmail** command tries to deliver the message:

```
dfAA00269Data file
qfAA00269Control file
tfAA00269Temporary file
xfAA00269Transcript file
```

12.3.3 Forcing the mail queue to run

In some cases, the mail queue becomes unresponsive. To force a queue to run, use the **sendmail** command with a -q flag (with no value). You can also use the -v flag (verbose) to watch what happens:

```
#sendmail -q -v
```

12.3.4 Moving the mail queue

When a host goes down for an extended period, many messages routed to (or through) that host may be stored in your mail queue. As a result, the **sendmail** command spends a long time sorting the queue, severely degrading your system's performance. If you move the queue to a temporary place and create a new queue, the old queue can be run later when the host returns to service. To move the queue to a temporary place and create a new queue:

1. Stop the **sendmail** daemon.
2. Move the entire queue directory by entering:

   ```
   #cd /var/spool
   #mv mqueue omqueue
   ```

3. Restart the **sendmail** daemon.
4. Process the old mail queue by entering:

   ```
   #sendmail -oQ/var/spool/omqueue -q
   ```

   The -oQ flag specifies an alternate queue directory. The -q flag specifies to run every job in the queue. To get a report about the progress of the operation, use the -v flag. This operation can take a long time.

5. Remove the log files and the temporary directory when the queue is empty by entering:

   ```
   #rm /var/spool/omqueue/*
   #rmdir /var/spool/omqueue
   ```

12.4 Mail logs

The **sendmail** command logs mail system activity through the **syslogd** daemon. The **syslogd** daemon must be configured and running for logging to occur. Specifically, the /etc/syslog.conf file may contain the uncommented line:

```
mail.debug          /var/spool/mqueue/log
```

If it does not, use an editor to make this change; be certain that the path name is correct. If you change the /etc/syslog.conf file while the **syslogd** daemon is running, refresh the **syslogd** daemon by entering the command:

```
#refresh -s syslogd
```

If the /var/spool/mqueue/log file does not exist, you must create it by entering the command:

```
#touch /var/spool/mqueue/log
```

12.4.1 Managing the mail log files

Because information is continually appended to the end of the log file, it can become very large. Also, error conditions can cause unexpected entries to the mail queue. To keep the mail queue and log from growing too large, execute the /usr/lib/smdemon.cleanu shell script. This script forces the **sendmail** command to process the queue and maintains four progressively older copies of log files named log.0, log.1, log.2, and log.3. Each time the script runs, it moves:

1. log.2 to log.3
2. log.1 to log.2
3. log.0 to log.1
4. log to log.0

This allows logging to start over with a new file. Run this script either manually or at a specified interval with the **cron** daemon. A typical log file is shown in Figure 12-3 on page 589. The highlighted field, `stat=Deferred` refers to a message that could not get routed to the destination.

```
# pg /var/spool/mqueue/log
Nov  3 09:49:00 sv1051c sendmail[29038]: JAA29038: from user root: size is 43, c
lass is 0, priority is 30043, and nrcpts=1, message id is <199811031549.JAA29038
@sv1051c.itsc.austin.ibm.com>, relay=root@localhost
Nov  3 09:49:00 sv1051c sendmail[33716]: JAA29038: to=smith, ctladdr=root (0/0),
 delay=00:00:00, xdelay=00:00:00, mailer=local, stat=Sent
Nov  3 09:49:51 sv1051c sendmail[29042]: JAA29042: from user root: size is 57, c
lass is 0, priority is 30057, and nrcpts=1, message id is <199811031549.JAA29042
@sv1051c.itsc.austin.ibm.com>, relay=root@localhost
Nov  3 09:49:51 sv1051c sendmail[29330]: JAA29042: to=npsingh@in.ibm.com, ctladd
r=root (0/0), delay=00:00:00, xdelay=00:00:00, mailer=esmtp, relay=relay2.server
.ibm.com. [::ffff:9.14.2.99], stat=Deferred: Network is unreachable
# 
```

Figure 12-3 /var/spool/mqueue/log file

12.4.2 Logging mailer statistics

The **sendmail** command tracks the volume of mail being handled by each of the mailer programs that interface with it (the mailers are defined in the /etc/sendmail.cf file).

To start the accumulation of mailer statistics, create the /etc/sendmail.st (refer to the sendmail.cf file for the exact file path) by entering:

```
#touch /etc/sendmail.st
```

The **sendmail** command updates the information in the file each time it processes mail. The size of the file does not grow, but the numbers in the file do. They represent the mail volume since the time you created or reset the /etc/sendmail.st file.

12.4.3 Displaying mailer information

The statistics kept in the /var/tmp/sendmail.st file are in a database format that cannot be read as a text file. To display the mailer statistics, enter the command:

```
#mailstats
```

This reads the information in the /var/tmp/sendmail.st file, formats it, and writes it to standard output in the format shown in Figure 12-4.

```
# mailstats
Sendmail statistics from Tue Nov  3 11:43:27 CST 1998
 M msgsfr bytes_from  msgsto   bytes_to  Mailer
 3      3         3K       3         3K  local
 5      1         1K       1         1K  esmtp
========================================
 T      4         4K       4         4K
#
```

Figure 12-4 Displaying mailer information

The fields in the report have the following meanings:

msgs_from Contains the number of messages received by the local machine from the indicated mailer.

bytes_from Contains the number of bytes in the messages received by the local machine from the indicated mailer.

msgs_to Contains the number of messages sent from the local machine using the indicated mailer.

bytes_to Contains the number of bytes in the messages sent from the local machine using the indicated mailer.

12.5 Mail aliasing and forwarding

Aliases map names to address lists. The aliases are defined in the /etc/aliases file by the user administrator. The /etc/aliases file consists of a series of entries in the following format:

```
Alias: Name1, Name2, ... NameX
```

Alias can be any alphanumeric string that you choose (not including special characters, such as @ or !). Name1 through NameX is a series of one or more recipient names. The /etc/aliases file must contain the following three aliases (a sample file is shown in Figure 12-5):

- MAILER-DAEMON
- postmaster
- nobody

```
# Licensed Materials - Property of IBM
#
# US Government Users Restricted Rights - Use, duplication or
# disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
#
##
#  Aliases in this file will NOT be expanded in the header from
#  Mail, but WILL be visible over networks or from /bin/bellmail.
#
#       >>>>>>>>>>      The command "sendmail -bi" must be run after
#       >> NOTE >>      this file is updated for any changes to
#       >>>>>>>>>>      affect sendmail operation.
##

# Alias for mailer daemon
MAILER-DAEMON:root

# Following alias is required by the new mail protocol, RFC 822
postmaster:root

# Aliases to handle mail to msgs and news
nobody: /dev/null

#
```

Figure 12-5 /etc/aliases File

12.5.1 Creating or modifying local system aliases

To add the alias programmer for four users working together in the same department (John, Smith, Mary, and Bob), perform the following functions:

1. Edit the /etc/aliases file.
2. On a blank line, add an alias, followed by a colon (:), followed by a list of comma-separated recipients. For example, the following entry defines an alias named programmer to be the names of the people in that group:

```
programmer: john, smith, mary@sv1051c, bob@sv1051c
```

3. Create an owner for any distribution list aliases. If the **sendmail** command has trouble sending mail to the distribution list, it sends an error message to the owner of that list. For example, the owner of the above list is root of system sv1051a and is defined by the following entry in the /etc/aliases file:

```
owner-programmer: root@sv1051a
```

4. Recompile the /etc/aliases file, as described in 12.5.2, "Building the alias database" on page 592.

12.5.2 Building the alias database

The **sendmail** command does not directly use the alias definitions in the local system /etc/aliases file. Instead, the **sendmail** command reads a processed database manager (dbm) version of the /etc/aliases file. You can compile the alias database using the following method.

Run the **sendmail** command using the -bi flag or run **newaliases**. This command causes the **sendmail** command to read the local system /etc/aliases file and creates two additional files containing the alias database information:

- /etc/aliases.dir
- /etc/aliases.pag

After you have completed building the alias database, you can use the alias (programmer) to send mail to the users (smith and john) on the local system and the users (mary and bob) on system sv1051c by using the **mail** command as shown:

```
#mail programmer
```

12.5.3 Forwarding mail with /etc/aliases

The /etc/aliases file also allows the automatic forwarding of mail to other systems or to other users on the system. This can be helpful, for example, if you are administering large amounts of systems and need a common address to place the mail for root from every system. To do this, place an entry in /etc/aliases in the following format:

```
User: Name1, Name2, ... NameX
```

where User is a user ID that exists on the system and Name1 through NameX is a list of names to which mail is to be forwarded.

For example, if the mail for root was to be forwarded to Smith and Bob, on a blank line in /etc/aliases, you would enter:

```
root: smith, bob@sv1051c
```

By doing this, however, root would no longer receive any mail. If you want to have a user retain a copy of its mail and still forward mail to other systems or users on the same system, place an entry in /etc/aliases in the following format:

```
User: \User, Name1, Name2, ... NameX
```

The backslash (\) instructs **sendmail** to keep a copy of mail for User, and send copies to Name1 through NameX. If the backslash were omitted, this would create an infinite forwarding loop and could lead to mail flooding for all the names in the list.

For example, if the mail for root was to be forwarded to Smith and Bob, and root was to retain a copy, in /etc/aliases you would enter:

```
root: \root, smith, bob@sv1051c
```

After /etc/aliases has been modified, the **sendmail -bi** or **newaliases** command must be run for the changes to take effect, as described in 12.5.2, "Building the alias database" on page 592.

12.5.4 Forwarding mail with $HOME/.forward

By creating a file named .forward in a user's home directory, mail can also be forwarded to other systems or other users on a system. This is a useful option for users on a system that cannot edit the /etc/aliases file. The .forward file uses the following format:

```
Name1, Name2, ... NameX
```

where Name1 through NameX is a list of names to which mail is to be forwarded.

For example, if John wants to forward his mail to Smith and Mary, he would do the following:

```
#vi $HOME/.forward
```

This will allow him to create the .forward file in his home directory through **vi**. On a blank line, he would enter:

```
smith, mary@sv1051c
```

Then he would save and quit the file. If John wants to retain a copy of his mail, he would place a backslash (\) and his own user ID in the .forward file:

```
\john, smith, mary@sv1051c
```

The **sendmail -bi** or **newaliases** command does not have to be run for the changes to take effect.

Any modifications to the .forward file become effective as soon as the file is saved. To stop mail forwarding, delete the .forward file:

```
#rm $HOME/.forward
```

12.5.5 Forwarding mail to /dev/null

To automatically delete mail for a particular user ID, mail can be forwarded to /dev/null in /etc/aliases or $HOME/.forward. This can be helpful, for example, for users who do not need to actively check their mail. To do this in /etc/aliases, you would enter:

```
User: /dev/null
```

where User is the user ID that will have its mail automatically deleted. Then you would run **sendmail -bi** or **newaliases** to have the changes take effect.

To do this in $HOME/.forward, you would enter:

```
/dev/null
```

12.6 Mail addressing

Mail is sent to a user's address. How you address mail to another user depends upon the user's location with respect to your system. The address would depend on whether you are sending the mail:

- To users on your local system.
- To users on your network.
- To users on a different network.
- Over a BNU or UUCP link.

12.6.1 To address mail to users on your local system

To send a message to a user on your local system (to someone whose login name is listed in your /etc/passwd file), use the login name for the address. At your system command line prompt, you can use the **mail** command in the way shown in the following example:

```
# mail LoginName
```

If smith is on your system and has the login name smith, use the command:

```
# mail smith
```

12.6.2 To address mail to users on your network

To send a message through a local network to a user on another system, at the command line, enter:

```
# mail LoginName@SystemName
```

For example, if john is on system sv1051c, use the following command to create and send a message to him:

```
# mail john@sv1051c
```

12.6.3 To address mail to users on a different network

If your network is connected to other networks, you can send mail to users on the other networks. The address parameters differ depending on how your network and the other networks address each other and how they are connected.

Using a central database of names and addresses

Use the **mail** command in the way shown in the following example:

```
# mail LoginName@SystemName
```

Using domain name addressing

Use the **mail** command in the way shown in the following example:

```
# mail LoginName@SystemName.DomainName
```

For example, to send mail to a user john, who resides in a remote network with a domain name in.ibm.com, use the following command:

```
# mail john@in.ibm.com
```

12.6.4 To address mail over a BNU or UUCP link

To send a message to a user on another system connected to your system by the Basic Networking Utilities (BNU) or another version of UNIX-to-UNIX Copy Program (UUCP), you must know the login name, the name of the other system, and the physical route to that other system.

When your computer has a BNU or UUCP link, you can use the command as shown in the following:

```
# mail UUCPRoute!LoginName
```

When the BNU or UUCP link is on another computer, use the **mail** command, as shown below:

```
# mail @InternetSystem:UUCPSystem!username
```

Notice that, in this format, you are not sending mail to a user at any of the intermediate systems; no login name precedes the @ in the domain address.

12.7 Storing mail

Mail is stored in different ways depending on the specific situation, as shown in Figure 12-6 on page 597. The mail program uses the following type of mailboxes or folders:

System mailbox This resides in /var/spool/mail directory and each system mailbox is named by the user ID associated with it. For example, if the user ID is smith, the system mailbox is /var/spool/mail/smith. When the mail arrives for any user ID, it is placed in the respective system mailbox. The shell checks for the new mail and issues the following message when the user logs in:

```
YOU HAVE NEW MAIL
```

Personal mailbox Each user has a personal mailbox. When the mail is read using the **`mail`** command by the user, and if it is not saved in a file or deleted, it is written to user's personal mailbox, $HOME/mbox ($HOME is the default login directory). For user ID smith, the personal mailbox is /home/smith/mbox.

dead.letter file If the user interrupts the message being created to complete some other tasks, the system saves the incomplete message in the dead.letter file in the user's home directory ($HOME). For user ID smith, /home/smith/dead.letter is the dead.letter file.

Folders To save a message in an organized fashion, users can use folders. Messages can be put into a user's personal folder from the system mailbox or the personal mailbox, as shown in Figure 12-6 on page 597.

The **`mail`** command can be used with various flags, as shown below:

`mail` Displays the system mailbox.

`mail -f` Displays your personal mailbox (mbox).

`mail -f +folder` Displays a mail folder.

`mail user@address` Addresses a message to the specified user.

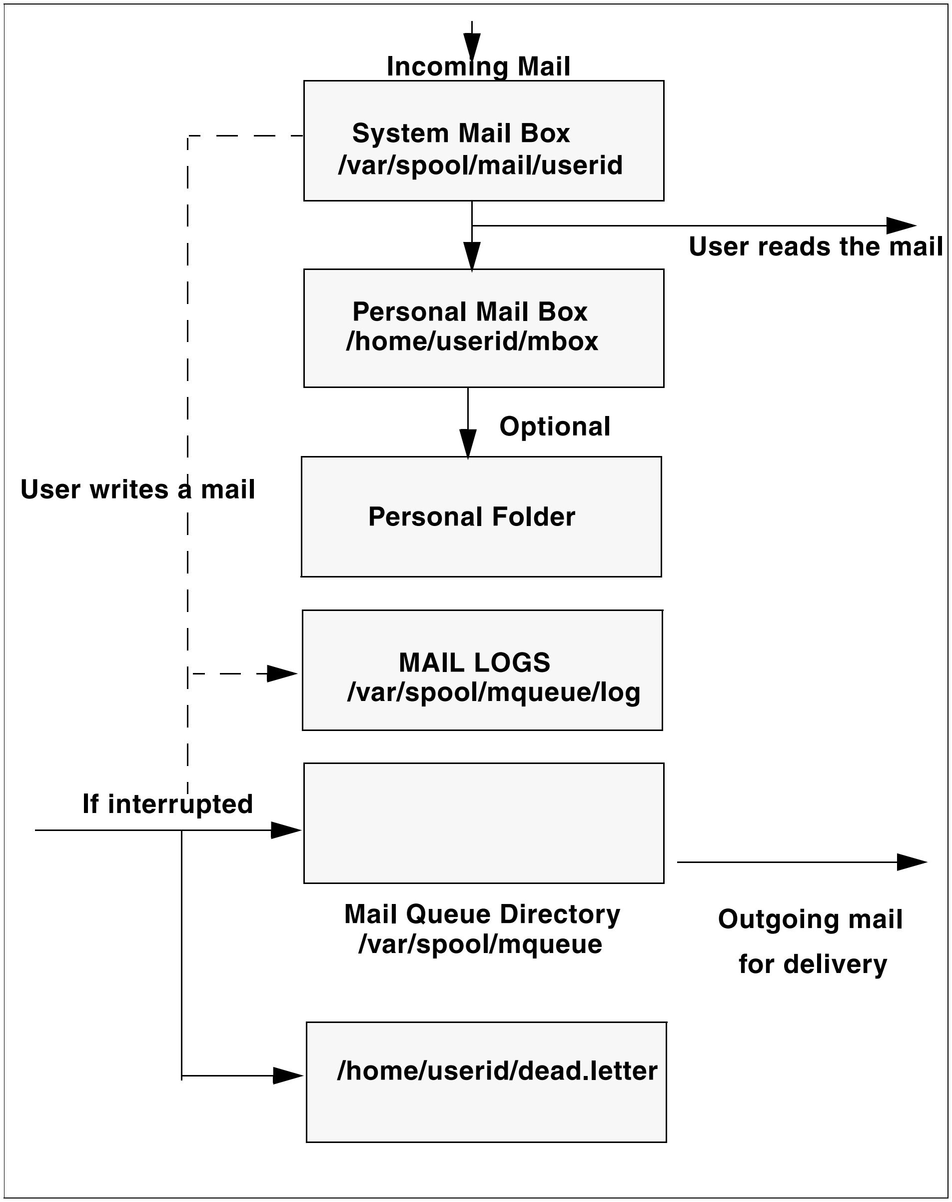


Figure 12-6 Message path for mail

12.8 Mail administrator's reference

This section provides a quick reference to the various mail commands, files, and directories.

12.8.1 List of mail commands

This list includes commands for using and managing the mail program.

`mailq` Prints the contents of the mail queue.

`mailstats` Displays statistics about mail traffic.

`newaliases` Builds a new copy of the alias database from the /etc/aliases file.

`sendmail` Routes mail for local or network delivery.

`smdemon.cleanu` Cleans up the sendmail queue for periodic housekeeping.

12.8.2 List of mail files and directories

This list of files and directories is arranged by function.

The mail program

The following is a list of files associated with the mail program.

$HOME/.mailrc Enables the user to change the local system defaults for the mail program.

$HOME/mbox Stores processed mail for the individual user.

/usr/bin/Mail, **/usr/bin/mail**, and **/usr/bin/mailx**
Specifies three names linked to the same program. The mail program is one of the user interfaces to the mail system.

/var/spool/mail Specifies the default mail drop directory. By default, all mail is delivered to the /var/spool/mail/UserName file.

/var/spool/mqueue Contains the log file and temporary files associated with the messages in the mail queue.

The sendmail command

The following is a list of files associated with **sendmail**:

/usr/sbin/sendmail	The **sendmail** command.
/usr/ucb/mailq	Links to the /usr/sbin/sendmail. Using **mailq** is equivalent to using the **/usr/sbin/sendmail -bp** command.
/usr/ucb/newaliases	Links to the /usr/sbin/sendmail file. Using **newaliases** is equivalent to using the **/usr/sbin/sendmail -bi** command.
/usr/sbin/mailstats	Formats and prints the **sendmail** statistics as found in the /etc/sendmail.st file, if it exists. The /etc/sendmail.st file is the default, but you can specify an alternative file.
/etc/aliases	Describes a text version of the aliases file for the **sendmail** command. You can edit this file to create, modify, or delete aliases for your system.
/etc/sendmail.cf	Contains the sendmail configuration information in text form. Edit the file to change this information.
/etc/sendmail.cfDB	Contains the processed version of the /etc/sendmail.cf configuration file. This file is created from the /etc/sendmail.cf file when you run the **/usr/sbin/sendmail -bz** command.
/etc/sendmail.nl	Contains the **sendmail** National Language Support (NLS) configuration information in text form. Edit the file to change this information.
/usr/lib/smdemon.cleanu	Specifies a shell file that runs the mail queue and maintains the sendmail log files in the /var/spool/mqueue directory.
/var/tmp/sendmail.st	Collects statistics about mail traffic. This file does not grow. Use the **/usr/sbin/mailstats** command to display the contents of this file.
/var/spool/mqueue	Describes a directory containing the temporary files associated with each message in the queue. The directory can contain the log file.

13

Editing files using vi

In this chapter, we provide a basic overview of how to use the **vi** editor to edit ASCII files. You will also learn how to view a file using various commands and the differences between these commands.

© Copyright IBM Corp. 2006. All rights reserved.

13.1 The vi file editor

The **vi** editor is a full-screen file editor that was developed from the **ex** editor. It is also common to all versions of UNIX. The **vedit** command starts a version of the **vi** editor intended for beginners. **vedit** corresponds to **vi -v** (vi in verbose mode).

13.1.1 Editor limitations with vi

The following list provides the maximum limits of the **vi** editor. These counts assume single-byte characters.

- 8,192 characters per line
- 256 characters per global command list
- 128 characters in the previously inserted and deleted text
- 2,048 characters in a shell escape command
- 128 characters in a string-valued option
- 30 characters in a tag name
- 128 map macros with 2048 characters total
- 1,048,560 lines silently enforced

Note: The **vi** editor supports a maximum of 2 GB edit buffer.

13.1.2 Editing modes in vi

The **vi** editor operates in the following modes:

Command mode When you start the **vi** editor, it is in command mode. You can enter any subcommand except those designated for use only in the text input mode. The **vi** editor returns to command mode when subcommands and other modes end. Press the Esc key to cancel a subcommand.

Text-input mode You use the **vi** editor in this mode to add text. Enter text input mode with any of the following subcommands: a, A, i, I, o, O, cx (where the x represents the scope of the subcommand), C, s, S, or R. After entering one of these subcommands, you can enter text into the editing buffer. To return to command mode, press the Esc key for normal exit or press Interrupt (the Ctrl-C key sequence) to end abnormally.

Last-line mode Subcommands with the prefix : (colon), / (slash), ? (question mark), ! (exclamation point), or !! (two exclamation points) read input on a line displayed at the bottom of the screen. When you enter the initial character, the **vi** editor places the cursor at the bottom of the screen, where you enter the remaining characters of the command. Press the Enter key to run the subcommand, or press Interrupt (the Ctrl-C key sequence) to cancel it. When the !! prefix is used, the cursor moves only after both exclamation points are entered. When you use the : prefix to enter the last-line mode, the **vi** editor gives special meaning to the following characters when they are used before commands that specify counts:

% All lines regardless of cursor position

$ Last line

. Current line

13.1.3 The vi command flags

Table 13-1 provides the flags that can be used with the **vi** command.

Table 13-1 Commonly used flags of the vi editor

Flag	Flag description
-c*Subcommand*	Carries out the ex editor subcommand before viewing with **vi** begins. The cursor moves to the line affected by the last subcommand to be carried out.
-l	Enters the **vi** editor in LISP mode. In this mode, the **vi** editor creates indents appropriate for LISP code, and the (,), {, }, [[, and]] subcommands are modified to act appropriately for LISP.
-r*[File]*	Recovers a file after a **vi** editor or system malfunction. If you do not specify the File variable, the **vi** editor displays a list of all saved files.
-R	Sets the read-only option to protect the file against overwriting.
-t*Tag*	Edits the file containing the *Tag* variable and positions the **vi** editor at its definition. To use this flag, you must first create a database of function names and their locations using the **ctags** command.
-v	Enters the **vi** editor in the verbose mode.
-w*Number*	Sets the default window size to the value specified by the Number variable. This flag is useful when you use the **vi** editor over a low-speed line.

Flag	Flag description
-y*Number*	Overrides the maximum line setting of 1,048,560 with any value greater than 1024. You should request twice the number of lines that you require because the `vi` editor uses the extra lines for buffer manipulation.
+*[Subcommand]*	Carries out the ex editor subcommand before editing begins. If you do not specify the Subcommand variable, the cursor is placed on the first line of the file. This + flag is incompatible with the -c flag. Do not specify both flags at the same time.

13.1.4 Editor subcommands in vi

The list of subcommands is very extensive, so you are only going to learn some of the more common subcommands used. You would use the subcommands to perform these kinds of actions:

- Moving the cursor
- Editing text
- Manipulating files
- Other actions

Enter the following subcommands in command mode. You can cancel an incomplete command by pressing the Esc key.

Moving the cursor

Use subcommands to move the cursor within a file in these ways:

- Moving within a line

Left Arrow, **h**, or **Ctrl-H**	Moves the cursor one character to the left.
Down Arrow, **j**, **Ctrl-J**, or **Ctrl-N**	Moves the cursor down one line (it remains in the same column).
Up Arrow, **k**, or **Ctrl-P**	Moves the cursor up one line (it remains in the same column).
Right Arrow or **l**	Moves the cursor one character to the right.

- Moving within a line by character position

  ^ Moves the cursor to the first nonblank character.

  0 Moves the cursor to the beginning of the line.

  $ Moves the cursor to the end of the line.

  fx Moves the cursor to the next x character.

  Fx Moves the cursor to the last x character.

  tx Moves the cursor to one column before the next x character.

  Tx Moves the cursor to one column after the last x character.

  ; Repeats the last f, F, t, or T subcommand.

  , Repeats the last f, F, t, or T subcommand in the opposite direction.

- Moving to words

  w Moves the cursor to the next small word.

  b Moves the cursor to the previous small word.

  e Moves the cursor to the next end of a small word.

  W Moves the cursor to the next big word.

  B Moves the cursor to the previous big word.

  E Moves the cursor to the next end of a big word.

- Moving by line position

  H Moves the cursor to the top line on the screen.

  L Moves the cursor to the last line on the screen.

  M Moves the cursor to the middle line on the screen.

  G Moves the cursor to the last line of the file.

  1G Moves the cursor to the first line of the file.

  nG Moves the cursor to the *n*th line of the file.

  + Moves the cursor to the next line at its first nonblank character.

  - Moves the cursor to the previous line at its first nonblank character.

  Enter Moves the cursor to the next line at its first nonblank character.

- Moving to sentences, paragraphs, or sections

  (Places the cursor at the beginning of the previous sentence.

) Places the cursor at the beginning of the next sentence.

  { Places the cursor at the beginning of the previous paragraph.

} Places the cursor at the beginning of the next paragraph, or at the next section if you are in C mode.

]] Places the cursor at the next section.

[[Places the cursor at the previous section.

Editing text

The subcommands for editing enable you to perform the following tasks:

- Marking a specific location in a file and returning

  " Moves the cursor to the previous location of the current line.

  " Moves the cursor to the beginning of the line containing the previous location of the current line.

  mx Marks the current position with the letter specified by the x parameter.

  `x Moves the cursor to the mark specified by the x parameter.

  'x Moves the cursor to the beginning of the line containing the mark specified by the x parameter.

- Adding text to a file

  a*Text* Inserts text specified by the *Text* parameter after the cursor. End text input mode by pressing the Esc key.

  A*Text* Adds text specified by the *Text* parameter to the end of the line. End text input mode by pressing the Esc key.

  i*Text* Inserts text specified by the *Text* parameter before the cursor. End text input mode by pressing the Esc key.

  I*Text* Inserts text specified by the *Text* parameter before the first non-blank character in the line. End text input mode by pressing the Esc key.

  o Adds an empty line below the current line. End text input mode by pressing the Esc key.

  O Adds an empty line above the current line. End text input mode by pressing the Esc key.

- Changing text while in input mode

  Ctrl-D Goes back to previous autoindent stop.

  ^ Ctrl-D Ends autoindent for this line only.

  0Ctrl-D Moves cursor back to left margin.

  Esc Ends insertion and returns to command state.

  Ctrl-H Erases the last character.

Ctrl-Q	Enters any character if xon is disabled.
Ctrl-V	Enters any character.
Ctrl-W	Erases the last small word.
****	Quotes the erase and kill characters.
Ctrl-?	Interrupts and ends insert or the Ctrl-D key sequence.

- Changing text from command mode

C	Changes the rest of the line (same as c$).
cc	Changes a line.
cw	Changes a word.
cw*Text*	Changes a word to the text specified by the Text parameter.
D	Deletes the rest of the line (same as d$).
dd	Deletes a line.
dw	Deletes a word.
J	Joins lines.
rx	Replaces the current character with the character specified by x.
R*Text*	Overwrites characters with the text specified by the Text parameter.
s	Substitutes characters (same as cl).
S	Substitutes lines (same as cc).
u	Undoes the previous change.
x	Deletes a character at the cursor.
X	Deletes a character before the cursor (same as dh).
<<	Shifts one line to the left.
<L	Shifts all lines from the cursor to the end of the screen to the left.
>>	Shifts one line to the right.
>L	Shifts all lines from the cursor to the end of the screen to the right.
~	Changes letter at the cursor to the opposite case.

- Copying and moving text

p	Puts back text from the undo buffer after the cursor.
P	Puts back text from the undo buffer before the cursor.
"xp	Puts back text from the x buffer.
"xd	Deletes text into the x buffer.

y Places the object that follows (for example, w for word) into the undo buffer.

"xy Places the object that follows into the x buffer, where x is any letter.

Y Places the line in the undo buffer.

- Restoring and repeating changes

u Undoes the last change.

> **Note:** After an undo, the cursor moves to the first non-blank character on the updated current line.

U Restores the current line, if the cursor has not left the line since the last change.

. Repeats the last change or increments the "np subcommand.

> **Note:**
>
> 1. This subcommand will repeat the last change, including an undo. Therefore, after an undo, repeat performs an undo rather than repeat the last change.
> 2. This subcommand is not meant for use with a macro. Enter @@ (two at signs) to repeat a macro.

"np Retrieves the nth last delete of a complete line or block of lines.

Manipulating files

The subcommands for manipulating files allow you to do the tasks outlined in the following sections:

- Saving changes to a file

:w Writes the edit buffer contents to the original file.

:w *File* Writes the edit buffer contents to the file specified by the File parameter.

:w! *File* Overwrites the file specified by the File parameter with the edit buffer contents.

- Editing a second file

:e *File* Edits the specified file.

:e! Re-edits the current file and discards all changes.

:e + *File* Edits the specified file starting at the end.

:e + *Number File* Edits the specified file starting at the specified line number.

:e *#* Edits the alternate file. The alternate file is usually the previous file name before accessing another file with a `:e` command. However, if changes are pending on the current file when a new file is called, the new file becomes the alternate file. This subcommand is the same as the Ctrl-A subcommand.

:r *File* Reads the file into the editing buffer by adding new lines below the current line.

:r !*Command* Runs the specified AIX 5L command and places its output into the file by adding new lines below the current cursor position.

:ta *Tag* Edits a file containing the Tag tag starting at the location of the tag. To use this subcommand, you must first create a database of function names and their locations using the `ctags` command.

Ctrl-A Edits the alternate file. The alternate file is usually the previous current file name. However, if changes are pending on the current file when a new file is called, the new file becomes the alternate file. This subcommand is the same as the :e # subcommand.

- Editing a list of files

  :n Edits the next file in the list entered on the command line.

  :n *Files* Specifies a new list of files to edit.

- Finding file information

  Ctrl-G Shows the current file name, current line number, number of lines in the file, and percentage of the way through the file where the cursor is located.

Other actions

The `vi` editor provides the subcommands described in the following sections:

- Adjusting the screen

  Ctrl-L Clears and redraws the screen.

  Ctrl-R Redraws the screen and eliminates blank lines marked with @ (at sign).

  z*Number* Makes the window the specified number of lines long.

- Entering shell commands

:sh Enters the shell to allow you to run more than one AIX 5L command. You can return to the **vi** editor by pressing the Ctrl-D key sequence.

:!*Command* Runs the specified AIX 5L command and then returns to the **vi** editor.

> **Note:** The # (alternate file), % (current file), and ! (previous command) special characters are expanded when following a :! subcommand. To prevent any of these characters from being expanded, use the **** (backslash).

:!! Repeats the last :!Command subcommand.

Number*!!*Command Runs the specified AIX 5L command and replaces the lines specified by Number with the output of the command. If a number is not specified, the default value is 1. If the command expects standard input, the specified lines are used as input.

!*Object Command* Runs the specified AIX 5L command and replaces the object specified by the Object parameter with the output of the command. If the command expects standard input, the specified object is used as input.

- Interrupting and ending the **vi** editor

Q Enters the ex editor in command mode.

ZZ Exits the **vi** editor, saving changes.

:q Quits the **vi** editor. If you have changed the contents of the editing buffer, the **vi** editor displays a warning message and does not quit.

:q! Quits the **vi** editor, discarding the editing buffer.

Esc Ends text input or ends an incomplete subcommand.

Ctrl-? Interrupts a subcommand.

Other actions

To set the **vi** editor up as your default editor, use the EDITOR command. The EDITOR variable can also be put into the user .profile or the /etc/profile file:

```
# EDITOR=vi
```

> **Note:** The EDITOR command sets a variable and must not be confused with the **set -o vi** command, which sets up command line editing.

13.2 Editing a file using the vi editor

In this section, you will learn how to edit a file. You will learn how to replace multiple characters, replace one character, delete a character, delete a line, copy a line, find a word, insert characters, and insert a line.

In this example you have a file called *yourfile.* In this file, you are running a few simple commands and sending the output for each to your screen.

The yourfile file looks like this:

```
#
# $HOME/yourfile
#
# This file contains some commands that are run by the user.
df
lsfs rootvg
lslv hdisk1
lsdev -Cc disk
```

13.2.1 Inserting text using the vi editor

As you look at the file above, you see that you want the **df** command to display the file system space in 1024 KB blocks, not 512. You also want the script to list what is in the current directory at the end of the script. To edit the file, enter:

```
#vi yourfile
```

Figure 13-1 is the display for the **vi** editor.

```
#
# $HOME/yourfile
#
# This file contains some commands that are run by the user.
df
lsfs rootvg
lspv hdisk1
lsdev -Cc disk
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"yourfile" 8 lines, 124 characters
```

Figure 13-1 Image of yourfile opened by the vi editor

You will note that the cursor is at the top left of the screen under the first character. Move your cursor down five lines to under the `df` using one of the following keys: Down Arrow, j, Ctrl-J, or Ctrl-N. There are now two options for you to add the -k flag to the **df** command.

- Option 1

  Go to the end of the text using Shift-A, which appends to the end of the line. You will notice that your cursor is to the right of the f; press the Space Bar and insert the -k.

- Option 2

  Go to the end of the line using one of the following keys: Right Arrow, l, or $. Your cursor will now be under the f; press the a key, and your cursor will move one space to the right. Press the Space key and then insert the -k.

Your line will look like this:

```
df -k
```

Press the Esc key once. You have now moved out of edit mode back into command mode. Your cursor will be under the k.

Now add the **ls -l** of the current directory at the end of the file. Using any one of the Down Arrow, j, Ctrl-J, or Ctrl-N keys, go to the bottom of the file. The cursor will be under the v in the **lsdev** command; either use the Left Arrow, h, Ctrl-H, or 0 keys to move to the beginning of the line. Your cursor should be under the l of the **lsdev** command. To insert a new line below the current line, press o; you will see the cursor under the l of the **lsdev** command but one line down. Type the following:

```
s -l
```

Press Esc once to go back into command mode. You will notice that this command is incorrect. What you need to do is insert the l for the **ls -l** command and press 0 to take you to the beginning of the line. Your cursor is under the s; press the i key, then type:

```
l
```

Press Esc once to go back to command line. Your line should look like this:

```
ls -l
```

To save the file, there are two options to save and quit.

- Option 1

  Hold the Shift and **; (:)**; the cursor will go to the bottom left of the screen next to a **:**. At the **:**, you can type `wq!`, which overwrites the original file and then quits.

- Option 2

  Hold the Shift and press z twice (ZZ). This will save and quit, taking you back to your AIX 5L command prompt.

To view the file, type:

```
# cat yourfile
```

The output of the **cat** command is as follows:

```
#
# $HOME/yourfile
#
# This file contains some commands that are run by the user.
df -k
lsfs rootvg
lspv hdisk1
lsdev -Cc disk
ls -l
```

13.2.2 Changing text using the vi editor

On closer inspection of the file yourfile, you notice that there are a few mistakes, and you now want to correct them. To correct them, type:

```
#vi yourfile
```

You will see the cursor at the top right side of your monitor.

The first thing you want to do is fix the **lsfs** command. You do not want it to be **lsfs**, because it should be the **lsvg** command. You also want to check the volume group uservg, not rootvg.

Press any of the Down Arrow, j, Ctrl-J, or Ctrl-N keys until you are under the l of the **lsfs** command.

You can also use the find option. Press the / key; it will take you to the bottom right of the screen and you will see a / before the cursor. Here you can type **lsfs** and it will take you to the beginning of the **lsfs** command.

Using the Right Arrow or l key, move the cursor below the f of the **lsfs** command. Press the Shift key and r (R) simultaneously. The cursor will still be under the f of the **lsfs** command. Type `vg`; you will see that it has overwritten the fs with vg. Press Esc once to go back into command mode. Your cursor is now under the g in the **lsvg** command.

Using the Right Arrow, move the cursor to underneath the r of rootvg. Type `cw`, and you will see a $ sign at the end of the word rootvg. Type `uservg`; you will notice that it has overwritten rootvg. Press Esc once and your cursor will be under the g of uservg.

Issue the command **lspv** on both hdisk1 and hdisk0. Press the Down Arrow key once to go down one line. You will notice that your cursor is under the 1 of hdisk1. Press your y key twice (yy); you have just yanked or copied the line. Press the p key to place your yanked line and you will see that the line has been duplicated. Notice that your cursor is at the beginning of the new line. Your file should look like this:

```
lspv hdisk1
lspv hdisk1
```

Using the Right Arrow key, move your cursor until it is under the 1 of the second hdisk1. Use your Up Arrow key to go up to the first hdisk1. Press the r key and then the 0. You will notice that the 1 has changed to a 0. You do not need to press the Esc key, because this is a single character replacement. Your lines should now look like this:

```
lspv hdisk0
```

```
lspv hdisk1
```

Looking at the line with the **lsdev -Cc disk** command, you decide that you want this command to display all the devices, not just the disks. Using the Down Arrow key, go to the line with the **lsdev** command. Using the Right Arrow key, put the cursor under the c, press the x, and you will see the c disappear. Use the Right Arrow key to go under the d of the word disk. Press the d key twice (dd), you will notice the whole line has disappeared. Press the u key once and you will notice that the line has come back. Now press the d key, then the w key. The word disk has now disappeared. Press the Esc key once. Your line will now look like this:

```
lsdev -C
```

Press the Shift zz (ZZ) to save and exit.

Type the following command:

```
#cat yourfile
```

to display your file. It should appear like this:

```
#
# $HOME/yourfile
#
# This file contains some commands that are run by the user.
df -k
lsvg uservg
lspv hdisk1
lspv hdisk0
lsdev -C
ls -l
```

To replace multiple occurrences of text, you can use the s subcommand. In the above example, if you wanted to change the **lspv** commands to have the -p flag, edit the file using **vi** and then change the command using a subcommand as follows:

```
#vi yourfile
```

The cursor will be at the top left hand side. Type Shift and **;** (**:**); the cursor will go into the command line editor at the bottom left side of the screen. To edit, type:

```
%s/lspv/lspv -p
```

This will replace all **lspv** commands to **lspv -p**. Save the file and exit.

Type the command:

```
#cat yourfile
```

to display your file. It should appear like this:

```
#
# $HOME/yourfile
#
# This file contains some commands that are run by the user.
df -k
lsvg uservg
lspv -p hdisk1
lspv -p hdisk0
lsdev -C
ls -l
```

To replace multiple occurrences of text that include the forward slash (/) character, the s subcommand can also be used with a slight variation. In the above example, you decide that you want to have the **df** and **ls** commands display only information for /home. First type:

```
#vi yourfile
```

You can see that the **ls -l** command is at the last line of the file, so press the Shift key and g (G) simultaneously. This will take the cursor down to the last line of the file. Using the Right Arrow, l, or $ key, go to the end of the line and press a to go into insert mode. Type in home/, misplacing the forward slash at the end of the file system name. Your line will look like this:

```
ls -l home/
```

Press Esc twice to go back into command mode. Since you can see that the **df -k** command is at the fifth line of the file, press the 5 key, then press the Shift key and g (G) simultaneously. This will take the cursor to the fifth line of the file. Using the Right Arrow, l, or $ key, go to the end of the line and press a to go into insert mode. Type in home/, misplacing the forward slash again. Your line will look like this:

```
df -k home/
```

Before you save and quit the file, you want to correct the two misplacements you just made, changing all instances of home/ to /home. To do this, press Esc twice to go into command mode. Type Shift and **; (:)**, causing the cursor to go into the command line editor at the bottom left of the screen. To edit, type:

```
%s/home\//\/home
```

This will replace all instances of home/ to /home. The **vi** command interprets the backslash as a marker to read the next character normally, not as part of a command. It is important to put in the backslash (\) before every instance of forward slash (/) that is to be part of the replacement; otherwise, **vi** will misinterpret the forward slash and may make unwanted modifications to your file.

Save and quit the file by typing `:wq!` while in command mode. View the contents of the file by typing:

```
#cat yourfile
```

Your file should look like this:

```
#
# $HOME/yourfile
#
# This file contains some commands that are run by the user.
df -k /home
lsvg uservg
lspv -p hdisk1
lspv -p hdisk0
lsdev -C
ls -l /home
```

14

Printing

This chapter discusses the AIX 5L print subsystem, including the optional SRV4 print subsystem, which is available in AIX 5L Version 5.3.

The following are terms commonly used when discussing UNIX printing:

- Print job

  A print job is a unit of work to be run on a printer. A print job can consist of printing one file or multiple files depending on how the print job is requested. The system assigns a unique job number to each job it runs.

- Queue

  The queue is where a print job is directed. It is a stanza in the /etc/qconfig file whose name is the name of the queue and points to the associated queue device.

- Queue device

  The queue device is the stanza in the /etc/qconfig file that normally follows the local queue stanza. It specifies the /dev file (printer device) that should be used.

Note: There can be more than one queue device associated with a single queue.

© Copyright IBM Corp. 2006. All rights reserved.

- **qdaemon**

  The **qdaemon** is a process that runs in the background and controls the queues. It is generally started during IPL with the **startsrc** command. See Chapter 9 for more information about **startsrc**.

- Print spooler

  A spooler is not specifically a print job spooler. Instead, it provides a generic spooling function that can be used for queuing various types of jobs including print jobs queued to a printer.

  The spooler does not normally know what type of job it is queuing. When the system administrator defines a spooler queue, the purpose of the queue is defined by the spooler back-end program that is specified for the queue. For example, if the spooler back-end program is the **piobe** command (the printer I/O backend), the queue is a print queue. Likewise, if the spooler back-end program is a compiler, the queue is for compile jobs. When the spooler's **qdaemon** command selects a job from a spooler queue, it runs the job by invoking the back-end program specified by the system administrator when the queue was defined.

  The main spooler command is the **enq** command. Although you can invoke this command directly to queue a print job, three front-end commands are defined for submitting a print job: The **lp**, **lpr**, and **qprt** commands. A print request issued by one of these commands is first passed to the **enq** command, which then places the information about the file in the queue for the **qdaemon** to process.

- Real printer

  A real printer is the printer hardware attached to a serial or parallel port at a unique hardware device address. The printer device driver in the kernel communicates with the printer hardware and provides an interface between the printer hardware and a virtual printer, but it is not aware of the concept of virtual printers. Real printers sometimes run out of paper.

- Local and remote printers

  When a printer is attached to a node or host, the printer is referred to as a local printer. A remote print system allows nodes that are not directly linked to a printer to have printer access.

  To use remote printing facilities, the individual nodes must be connected to a network using Transmission Control Protocol/Internet Protocol (TCP/IP) and must support the required TCP/IP applications.

- Printer backend

  The printer backend is a collection of programs called by the spooler's **qdaemon** command to manage a print job that is queued for printing. The printer backend performs the following functions:

  – Receives a list of one or more files to be printed from the **qdaemon** command.

  – Uses printer and formatting attribute values from the database; overridden by flags entered on the command line.

  – Initializes the printer before printing a file.

  – Runs filters as necessary to convert the print data stream to a format supported by the printer.

  – Provides filters for simple formatting of ASCII documents.

  – Provides support for printing national language characters.

  – Passes the filtered print data stream to the printer device driver.

  – Generates header and trailer pages.

  – Generates multiple copies.

  – Reports paper out, intervention required, and printer error conditions.

  – Reports problems detected by the filters.

  – Cleans up after a print job is cancelled.

  – Provides a print environment that a system administrator can customize to address specific printing needs.

Table 14-1 provides a list of commands that can perform the same function.

Table 14-1 Print commands and their equivalents

Submit print jobs	Status print jobs	Cancel print jobs
enq	**enq -A**	**enq -x**
qprt	**qchk**	**qcan**
lp	**lpstat**	**lprm**
lpr	**lpq**	**cancel**

14.1 Creating a new print queue

The best way to create a new print queue is by using the SMIT interface. Here are the steps to follow.

Enter the following command:

```
# smitty
```

The System Management menu will display. Select Print Spooling, as shown in Figure 14-1, and press Enter.

```
                              System Management

Move cursor to desired item and press Enter.

  Software Installation and Maintenance
  Software License Management
  Devices
  System Storage Management (Physical & Logical Storage)
  Security & Users
  Communications Applications and Services
  Print Spooling
  Problem Determination
  Performance & Resource Scheduling
  System Environments
  Processes & Subsystems
  Applications
  Using SMIT (information only)




F1=Help             F2=Refresh          F3=Cancel           F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 14-1 System Management menu screen - Print Spooling option

Select AIX 5L Print Spooling, as shown in Figure 14-2 on page 623, and press Enter. This is the only option available on this screen unless System V printing has been enabled on the system. To enable System V printing, a number of filesets, included in the installation media, need to be installed manually, as discussed in 14.10, “Enabling System V printing” on page 642.

```
                              Print Spooling

Move cursor to desired item and press Enter.

  AIX Print Spooling











F1=Help             F2=Refresh          F3=Cancel           F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 14-2 AIX 5L Print Spooling option

Select Add a Print Queue, as shown in Figure 14-3, and press Enter.

```
                                AIX Print Spooling

Move cursor to desired item and press Enter.

  Start a Print Job
  Manage Print Jobs
  List All Print Queues
  Manage Print Queues
  Add a Print Queue
  Add an Additional Printer to an Existing Print Queue
  Change / Show Print Queue Characteristics
  Change / Show Printer Connection Characteristics
  Remove a Print Queue
  Manage Print Server
  Programming Tools

  Change / Show Current Print Subsystem





F1=Help             F2=Refresh          F3=Cancel           F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 14-3 AIX 5L Print Spooling menu screen - Add a Print Queue option

Figure 14-4 on page 625 shows the AIX 5L Print Spooling menu, in which an Add a Print Queue sub menu will appear. Select what printer the system is connected to, in this case, local, and press Enter.

```
                              AIX Print Spooling

Move cursor to desired item and press Enter.

  ┌──────────────────────────────────────────────────────────────────────┐
  │                          Add a Print Queue                           │
  │                                                                      │
  │ Move cursor to desired item and press Enter. Use arrow keys to scroll.│
  │                                                                      │
  │   # ATTACHMENT TYPE      DESCRIPTION                                 │
  │     local                Printer Attached to Local Host              │
  │     remote               Printer Attached to Remote Host             │
  │     xstation             Printer Attached to Xstation                │
  │     ascii                Printer Attached to ASCII Terminal          │
  │     hpJetDirect          Network Printer (HP JetDirect)              │
  │     file                 File (in /dev directory)                    │
  │     ibmNetPrinter        IBM Network Printer                         │
  │     ibmNetColor          IBM Network Color Printer                   │
  │     other                User Defined Backend                        │
  │                                                                      │
  │ F1=Help                 F2=Refresh              F3=Cancel            │
  │ F8=Image                F10=Exit                Enter=Do             │
F1│ /=Find                  n=Find Next                                  │
F9└──────────────────────────────────────────────────────────────────────┘
```

Figure 14-4 Add a Print Queue menu screen - print queue selection

Once the printer connection location is selected, select what kind of printer it is and press Enter. In Figure 14-5, Other (select this if your printer type is not listed above) has been selected.

```
                          AIX Print Spooling

Move cursor to desired item and press Enter.
  +----------------------------------------------------------------------+
  |                         Printer Type                                 |
  |                                                                      |
  | Move cursor to desired item and press Enter.                         |
  |                                                                      |
  |   Bull                                                               |
  |   Canon                                                              |
  |   Dataproducts                                                       |
  |   Hewlett-Packard                                                    |
  |   IBM                                                                |
  |   Lexmark                                                            |
  |   OKI                                                                |
  |   Printronix                                                         |
  |   QMS                                                                |
  |   Texas Instruments                                                  |
  |   Other (Select this if your printer type is not listed above)       |
  |                                                                      |
  | F1=Help                 F2=Refresh              F3=Cancel            |
  | F8=Image                F10=Exit                Enter=Do             |
F1| /=Find                  n=Find Next                                  |
F9+----------------------------------------------------------------------+
```

Figure 14-5   AIX 5L Print Spooling menu screen - Printer Type selection

Figure 14-6 on page 627 shows the Printer Type selected.

```
                          AIX Print Spooling

Move cursor to desired item and press Enter.

  Start a Print Job
  Manage Print Jobs
  List All Print Queues
  Manage Print Queues
  Add a Print Queue
  Add an Additional Printer to an Existing Print Queue
  Change / Show Print Queue Characteristics
  Change / Show Printer Connection Characteristics
  Remove a Print Queue
  +--------------------------------------------------------------------------+
  |                             Printer Type                                 |
  |                                                                          |
  | Move cursor to desired item and press Enter.                             |
  |                                                                          |
  |   generic  Generic Printer                                               |
  |                                                                          |
  | F1=Help                 F2=Refresh              F3=Cancel                |
  | F8=Image                F10=Exit                Enter=Do                 |
F1| /=Find                  n=Find Next                                      |
F9+--------------------------------------------------------------------------+
```

Figure 14-6 AIX 5L Print Spooling menu - Print Type selection

Now select the Printer Interface. As shown in Figure 14-7, select parallel and press Enter.

```
                              AIX Print Spooling

Move cursor to desired item and press Enter.

  Start a Print Job
  Manage Print Jobs
  List All Print Queues
  Manage Print Queues
  Add a Print Queue
  Add an Additional Printer to an Existing Print Queue
  Change / Show Print Queue Characteristics

                            Printer Interface

    Move cursor to desired item and press Enter.

       parallel
       rs232
       rs422

    F1=Help                 F2=Refresh              F3=Cancel
    F8=Image                F10=Exit                Enter=Do
F1  /=Find                  n=Find Next
F9
```

Figure 14-7 AIX 5L Print Spooling menu - Printer Interface selection

Once the Printer Interface has been selected, select the Parent Adapter and press Enter.

When this process is complete, Add a Print Queue, as shown in Figure 14-8 on page 629. This is where a printer selection is made. In this case, the printer is called *lpforu*. Any of the characteristics of the printer can be changed if needed; however, normally this is not required.

```
                              Add a Print Queue

ype or select values in entry fields.
ress Enter AFTER making all desired changes.

                                                      [Entry Fields]
 Description                                         Generic Printer

 Names of NEW print queues to add
    ASCII                                           [lpforu]
    GL Emulation                                    []
    PCL Emulation                                   []
    PostScript                                      []

 Printer connection characteristics
    PORT number                                     [p]                     +
    Type of PARALLEL INTERFACE                      [standard]              +
    Printer TIME OUT period (seconds)               [60]                    +
    STATE to be configured at boot time              available              +

1=Help              F2=Refresh          F3=Cancel           F4=List
5=Reset             F6=Command          F7=Edit             F8=Image
9=Shell             F10=Exit            Enter=Do
```

Figure 14-8 Add a Print Queue menu - printer characteristics

Once the characteristics of the printer have been entered, the COMMAND STATUS menu will appear showing the success of the action, as shown in Figure 14-9.

```
                                COMMAND STATUS

Command: OK            stdout: yes           stderr: no

Before command completion, additional instructions may appear below.

Added printer 'lp0'.

Added print queue 'lpforu'.









F1=Help             F2=Refresh          F3=Cancel           F6=Command
F8=Image            F9=Shell            F10=Exit            /=Find
n=Find Next
```

Figure 14-9 New print queue Command Status

Press F10 to exit.

The print queue lpforu is now installed on the printer lp0.

14.2 The print configuration file

The /etc/qconfig file holds the printer configurations. It is the most important file in the spooler domain for the following reasons:

- It contains the definition of every queue known to the spooler.
- A system administrator can read this file and discern the function of each queue.
- Although it is not recommended, this file can be edited to modify spooler queues without halting the spooler.

The /etc/qconfig file describes all of the queues defined in the AIX 5L operating system. A queue is a named, ordered list of requests for a specific device. A device is something (either hardware or software) that can handle those requests one at a time. The queue provides serial access to the device. Each queue must be serviced by at least one device; often it can be handled by more than one device.

The following is an example of the contents of the /etc/qconfig file:

```
* @(#)33        1.6  src/bos/usr/bin/que/qconfig.sh, cmdque, bos430, 9737A_430
2/4/94 10:45:05
* IBM_PROLOG_BEGIN_TAG
... lines omitted ...
* IBM_PROLOG_END_TAG
*
* COMPONENT_NAME: cmdque configuration file for spooling
... lines omitted ...
* PRINTER QUEUEING SYSTEM CONFIGURATION
*
* This configuration file contains valid configurations for remote
* print queue rp0, local print queue lp0 and batch queue bsh.
* They may be deleted or changed as necessary.
*
* EXAMPLE of remote print queue configuration
* rp0:
*       host = hostname
*       s_statfilter = /usr/lib/lpd/aixshort
*       l_statfilter = /usr/lib/lpd/aixlong
*       rq = queuename
*       device = drp0
*
```

```
* drp0:
*       backend = /usr/lib/lpd/rembak
*
* EXAMPLE of local print queue configuration
*lp0:
*       discipline = fcfs
*       up = TRUE
*       device = dlp0
*
*dlp0:
*       backend = /usr/lib/lpd/piobe
*       file = FALSE
*       access = write
*       feed = never
*       header = never
*       trailer = never
*
* BATCH queue for running shell scripts
*
*bsh:
*       device = bshdev
*       discipline = fcfs
*bshdev:
*       backend = /usr/bin/bsh
lpforu:
device = lp0
lp0:
        file = /dev/lp0
        header = never
        trailer = never
        access = both
        backend = /usr/lib/lpd/piobe
```

The file /etc/qconfig as shown above is composed of text blocks referred to as stanzas. Each queue is represented by a pair of stanzas. The first stanza in a pair is referred to as the queue stanza; the second stanza in a pair is referred to as the device stanza. Stanzas are composed of parameters and parameter values that describe the queue's properties and functions.

14.3 Controlling the print queue

This section examines some of the commands that are used with the print queue.

- The **lpstat** command displays information about the current status of the line printer.

  The **lpstat** command syntax is as follows:

  lpstat [-a*List*] [-c*List*] [-d] [-o*List*] [-p*List*] [-r] [-s]
  [-t] [-u*List*] [-v*List*] [-W]

  The following is an example of the **lpstat** command used without any flags:

```
# lpstat
Queue  Dev  Status  Job Files            User         PP% Blks Cp Rnk
------ ---- ------- --- ---------------- ------------ --- ---- -- ---
lpforu lp0  READY
```

- The **qchk** command displays the current status of the specified print jobs, print queues, or users.

  The **qchk** command syntax is as follows:

  qchk [-A] [-L | -W] [-P *Printer*] [-# *JobNumber*] [-q]
  [-u *UserName*] [-w *Delay*]

  The following is an example of the **qchk** command used without any flags:

```
# qchk
Queue  Dev  Status  Job Files            User         PP% Blks Cp Rnk
------ ---- ------- --- ---------------- ------------ --- ---- -- ---
lpforu lp0  READY
```

- The **lpq** command reports the status of the specified job or all jobs associated with the specified *UserName* and *JobNumber* variables.

  The **lpq** command syntax is as follows:

  lpq [+ [*Number*]] [-l | -W] [-P *Printer*] [*JobNumber*] [*UserName*]

  The following is an example of the **lpq** command used without any flags:

```
# lpq
Queue  Dev  Status  Job Files            User         PP% Blks Cp Rnk
------ ---- ------- --- ---------------- ------------ --- ---- -- ---
lpforu lp0  READY
```

- The **lpr** command uses a spooling daemon to print the named File parameter when facilities become available.

  The **lpr** command syntax is as follows:

  lpr [-f] [-g] [-h] [-j] [-l] [-m] [-n] [-p] [-r] [-s]
  [-P *Printer*] [-# *NumberCopies*] [-C *Class*] [-J *Job*] [-T *Title*]
  [-i [*NumberColumns*]] [-w *Width*] [*File* ...]

The following is an example of using the **lpr** command to print the file /etc/passwd.

```
# lpr /etc/passwd
# lpstat
Queue  Dev  Status   Job  Files            User       PP %  Blks Cp Rnk
------ ---- -------- ---  ---------------- -------- ---- -- ---- -- ---
lpforu lp0  RUNNING    3  /etc/passwd      root        1 100   1  1   1
```

14.3.1 Editing /etc/qconfig

The /etc/qconfig file can be modified with your favorite text editor. This may be required if a printer definition cannot be removed by any other means. There are unenforced rules concerning whether or not edit /etc/qconfig can be edited without halting or corrupting the operation of the spooler. This is discussed in the next section.

14.3.2 Modifying /etc/qconfig while jobs are processing

The /etc/qconfig file should never be edited when jobs are processing. This is especially true when the system has a large number (greater than 25) of printers that are generally busy. When the **qdaemon** receives notification from **enq** that a new Job Description File (JDF) exists, the **qdaemon** examines the dates on both /etc/qconfig and /etc/qconfig.bin, the binary version of /etc/qconfig. If /etc/qconfig is younger than /etc/qconfig.bin, the **qdaemon** does not accept any new jobs, including the one that caused it to examine the aforementioned files, until all currently running jobs have finished processing. When the jobs have finished processing, the **qdaemon** creates a new version of /etc/qconfig.bin.

If the **qdaemon** goes into this state while jobs are processing, it is possible for the spooler to hang.

14.4 Stopping the print queue

In the following scenario, a job is printing on a print queue, but the queue must be stopped in order to load paper into the printer.

Check the print queue using the **lpstat** command, as shown in the following example. The reason for the -v flag is so that a listing of all the printers is not generated. See Table 14-4 on page 636 for a list of commonly used flags for the **lpstat** command.

```
# lpstat -vlpforu
Queue  Dev  Status   Job  Files            User      PP %  Blks Cp Rnk
------ ---- -------- ---  ---------------- -------- ---- -- ---- -- ---
lpforu lp0  RUNNING    3  /etc/passwd      root        1 100   1  1   1
```

Disable the print queue using the **enq** command, as shown in the following example. See Table 14-2 for a list of commonly used flags for the **enq** command.

```
# enq -D -P 'lpforu:lp0'
```

Check the printer queue using the **qchk** command, as shown in the following example. See Table 14-3 for a list of commonly used flags for the **qchk** command.

```
# qchk -P lpforu
Queue  Dev  Status   Job  Files            User      PP %  Blks Cp Rnk
------ ---- -------- ---  ---------------- -------- ---- -- ---- -- ---
lpforu lp0  DOWN       3  /etc/passwd      root        1 100   1  1   1
```

Table 14-2 Commonly used flags for the enq command

Flag	Description
-D	Device DOWN. Turns off the device associated with the queue. The qdaemon process no longer send jobs to the device.
-U	Brings UP the device associated with a queue. The qdaemon process sends jobs to it again.
-P *Queue*	Specifies the queue to which the job is sent. A particular device on a queue can be specified by typing -P *Queue:Device*.

Table 14-3 Commonly used flags for the qchk command

Flag	Description
-# *JobNumber*	Requests the status of the job number specified by the *JobNumber* variable. The **qchk** command looks for *JobNumber* on the default queue when the *-#JobNumber* flag is used alone. To search for *JobNumber* on all queues, the *-#* flag must be used with the -A flag. The *-#* flag may also be used in conjunction with the -P *Queue* flag.
-A	Requests the status of all queues.

Flag	Description
-P Printer	Requests the status of the printer specified by the *Printer* variable.
-u *UserName*	Requests the status of all print jobs sent by the user specified by the *UserName* variable.
-w *Delay*	Updates requested status information at intervals, in seconds, as specified by the *Delay* variable until all print jobs are finished.

14.5 Starting the print queue

To restart the print queue and finish the print job after loading the paper, use the following commands:

```
# lpstat -vlpforu
Queue  Dev  Status   Job  Files            User       PP %  Blks Cp Rnk
------ ---- -------- ---  ---------------- -------- ---- -- ---- -- ---
lpforu lp0  DOWN       3  /etc/passwd      root        1 100   1  1   1
# enq -U -P 'lpforu:lp0'
# qchk -P lpforu
Queue  Dev  Status   Job  Files            User       PP %  Blks Cp Rnk
------ ---- -------- ---  ---------------- -------- ---- -- ---- -- ---
lpforu lp0  RUNNING    3  /etc/passwd      root        1 100   1  1   1
```

The **enable** command can also be used to start the print queue. It takes the printer queue device as its only argument, as in the following example:

```
# enable lpforu:lp0
```

14.6 Flushing a print job

To delete an incorrect print job after starting the correct one, perform the following:

1. Check the status of the print queue:

```
# lpstat -vlpforu
Queue  Dev  Status   Job  Files            User       PP %  Blks Cp Rnk
------ ---- -------- ---  ---------------- -------- ---- -- ---- -- ---
lpforu lp0  RUNNING    3  /etc/passwd      root        1 100   1  1   1
```

2. Print the /etc/hosts file to the default printer:

```
# lpr -dlpforu /etc/hosts
```

3. Check the status of the print queue:

```
# lpstat -vlpforu
Queue  Dev  Status   Job  Files            User      PP %  Blks Cp Rnk
------ ---- -------- ---  ---------------- -------- ---- -- ---- -- ---
lpforu lp0  RUNNING    3  /etc/passwd      root        1 100    1  1   1
            QUEUED     4  /etc/hosts       root        1 100    2  1   2
```

4. Cancel the print job for /etc/passwd using one of the following commands:

```
# qcan -P lpforu -x 3
# cancel 3
# lprm -P lpforu 3
# enq -P lpforu -x 3
```

5. Check the print queue using the **qchk** command.

```
# qchk -P lpforu
Queue  Dev  Status   Job  Files            User      PP %  Blks Cp Rnk
------ ---- -------- ---  ---------------- -------- ---- -- ---- -- ---
lpforu lp0  RUNNING    4  /etc/hosts       root        1 100    2  1   2
```

14.7 How to check the print spooler

There are various commands to check a print spooler. This section covers the **lpstat** command and some of the flags that can be used. The **enq** command has a similar function.

In Table 14-4 are some of the commonly used flags for the **lpstat** command and, where available, an equivalent **enq** command.

Table 14-4 Commonly used flags for the lpstat and enq command equivalents

Flag	enq equivalent	Description
-a*List*	**enq -q -P*Queue1***	Provides status and job information about queues.
-d	**enq -q**	Displays the status information for the system default destination for the **lp** command.
-o*List*		Displays the status of print requests or print queues.
-p*List*		Displays the status of printers.
-r	**enq -A**	Provides status and job information about queues.

Flag	enq equivalent	Description
-s	**enq -A**	Displays a status summary, including a list of printers and their associated devices.
-t	**enq -AL**	Displays all status information, including a list of printers and their associated devices.
-u*List*	**enq -u**	Prints the status of all print requests for users specified in List. The list variable is a list of login names.
-v*List*		Prints the status of printers. The List variable is a list of printer names.

The following is an example of using the **lpstat** command with different flag settings to get the status of the print queue lpforu:

```
# lpstat -plpforu
Queue   Dev Status    Job Files              User         PP %   Blks  Cp Rnk
------ --- --------- --- ------------------ ---------- ---- -- ----- --- ---
lpforu  lp0 RUNNING    2 /etc/hosts         root                 2   1   1
# lpstat -u"root"
Queue   Dev Status    Job Files              User         PP %   Blks  Cp Rnk
------- --- --------- --- ------------------ ---------- ---- -- ----- --- ---
lpforu  lp0   RUNNING   2 /etc/hosts         root                 2   1   1
```

The output for the **lpstat -t** command is the same as the output for the **lpstat -u** and **lpstat -p** commands, except that it gives the queue file, as well as the time stamp, for the file in the queue.

14.8 Setting the timeout on a printer

Setting the time out on a printer specifies the amount of time, in seconds, the system waits for an operation to complete on a printer. The value must be greater than zero (0). The default value is calculated based on the device you select.

This option would be used in the following scenarios:

- A large network with many users utilizing the printers.
- A network with printers a long distance from the server or at another location.

The following example shows output from the **lpstat** command that indicates the printer time out should be increased:

```
# lpstat -plpforu
Queue   Dev   Status    Job Files              User         PP %   Blks  Cp  Rnk
------- ----- --------- --- ------------------ ---------- ---- -- ----- --- ---
lpforu  lp0   DEV_WAIT   17 smit.log           root                  50   1   1
```

To increase the printer time out, enter the following SMIT fast path:

```
# smitty spooler
```

In the Print Spooling menu, select Change / Show Printer Connection Characteristics, as in Figure 14-10.

```
                              AIX Print Spooling

Move cursor to desired item and press Enter.

  Start a Print Job
  Manage Print Jobs
  List All Print Queues
  Manage Print Queues
  Add a Print Queue
  Add an Additional Printer to an Existing Print Queue
  Change / Show Print Queue Characteristics
  Change / Show Printer Connection Characteristics
  Remove a Print Queue
  Manage Print Server
  Programming Tools

  Change / Show Current Print Subsystem





F1=Help             F2=Refresh          F3=Cancel           F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 14-10 AIX 5L Print Spooling menu

In the Change/Show Printer Connection Characteristics sub window, select where the printer is connected. In Figure 14-11 on page 639, local is used.

```
                              AIX Print Spooling

Move cursor to desired item and press Enter.

  Start a Print Job
  Manage Print Jobs
  List All Print Queues
  Manage Print Queues
  Add a Print Queue
  Add an Additional Printer to an Existing Print Queue
  Change / Show Print Queue Characteristics
  +-------------------------------------------------------------------------+
  |            Change / Show Printer Connection Characteristics             |
  |                                                                         |
  | Move cursor to desired item and press Enter. Use arrow keys to scroll.  |
  |                                                                         |
  |   # ATTACHMENT TYPE      DESCRIPTION                                    |
  |     local                Printer Attached to Local Host                 |
  |     xstation             Printer Attached to Xstation                   |
  |                                                                         |
  | F1=Help                  F2=Refresh                F3=Cancel            |
  | F8=Image                 F10=Exit                  Enter=Do             |
F1| /=Find                   n=Find Next                                    |
F9+-------------------------------------------------------------------------+
```

Figure 14-11 Change/Show Printer Connection Characteristics pop-up screen

In the Local Printers selection sub-menu, select lp0 Available 01-d0-00-00 Other parallel printer, and press Enter.

Once everything is selected, go to the Change/Show Printer Connection Characteristics menu, as displayed in Figure 14-12. Here, select the Printer TIME OUT period (seconds) and change it, in this case, to 60 seconds.

```
                Change / Show Printer Connection Characteristics

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                          [Entry Fields]
  Printer name                                          lp0
  Description                                           Other parallel printer
  Interface                                             parallel
  Status                                                Available
  Location                                              01-D0-00-00
  Parent adapter                                        ppa0

* PORT number                                          [p]                      +
  Type of PARALLEL INTERFACE                           [standard]               +
  Printer TIME OUT period (seconds)                    [60]                     +#
  STATE to be configured at boot time                   available               +
  Microseconds to delay between characters             [0]                      +#



F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 14-12   Change/Show Printer Connection Characteristics screen

Once the command has completed, the COMMAND STAUS menu appears, as displayed in Figure 14-13 on page 641, which shows the status of command completion.

```
                                COMMAND STATUS

Command: OK            stdout: yes           stderr: no

Before command completion, additional instructions may appear below.

lp0 changed










F1=Help             F2=Refresh          F3=Cancel           F6=Command
F8=Image            F9=Shell            F10=Exit            /=Find
n=Find Next
```

Figure 14-13 Changing printer connection characteristics results

14.9 Basic printer diagnostics checklist

This section has some troubleshooting tips. This is not a comprehensive list, but it will assist with resolving some of the more common problems that may be encountered.

- Verify that the **qdaemon** is running. Make sure there are no forked processes running from the **qdaemon**.
- Make sure the system date is correct. The **qdaemon** automatically rebuilds the /etc/qconfig.bin file when the qconfig file changes. If the date on the qconfig file is earlier than the date on the /etc/qconfig.bin file, the qconfig file is not digested, even if it was just modified. Use the **enq -Y** command to redigest the qconfig file.
- If the dates on the /etc/qconfig.bin file and the /etc/qconfig file are correct, and changes to the qconfig file are correct, the /etc/qconfig file may no longer be linked to the /usr/lpd/qconfig file.
- Check that the /tmp directory is not full. The /tmp directory may be full if a message is received, such as *No Virtual Printers Defined*, or if your are unable to print from InfoExplorer.

- If only the root user can print, check the permissions of the /tmp directory. Also, check the permissions of the print commands being used (including **enq**).
- Check for obsolete queue names in the /var/spool/lpd/qdir file. A problem with the installation of a new /etc/qconfig file occurs when a queue is removed from the new /etc/qconfig file and a print request is made using the obsolete queue name. In this case, the **qdaemon** logs an error message. Determine if the message refers to an old queue. If so, the problem will persist until the obsolete queue entries from the /var/spool/lpd/qdir file are removed.
- If operator-attention messages requested by remote print commands are not being received, make sure the socket is connected and the host name can be pinged.

14.10 Enabling System V printing

Print spooling only for AIX 5L is enabled by default when AIX 5L Version 5.3 is installed on a system. In order to enable System V printing, the System V printing filesets need to be manually installed. These filesets are included in the installation media.

The best way to enable System V printing is by using the SMIT interface. Here are the steps to follow:

Enter the following command:

```
# smitty
```

The System Management menu will appear. Select Print Spooling, as shown in Figure 14-1 on page 622, and press Enter.

Select System V Print Spooling, as shown in Figure 14-14 on page 643, and press Enter.

```
                              Print Spooling

Move cursor to desired item and press Enter.

  AIX Print Spooling
  System V Print Spooling
















F1=Help             F2=Refresh          F3=Cancel           F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 14-14 System V Print Spooling

The next few screens show how to add a local printer, how to add a remote printer, how to remove a printer, and how to manage printers. Each of these screens has a number of fields for which inputs are mandatory. See 14.10.5, “The lpadmin command” on page 651 for an explanation of these fields and input the appropriate values.

14.10.1 Add a local printer

Select Add Local Printer Configuration from the System V Print Spooling menu, as shown in Figure 14-15.

```
                          System V Print Spooling

Move cursor to desired item and press Enter.

  Manage Print Requests
  List Print Destinations
  Remove Destinations
  Manage Destinations
  Add Local Printer Configuration
  Add Remote Printer Configuration
  Add JetDirect Printer Configuration
  Manage Print Services
  Manage Remote Systems
  Change / Show Current Print Subsystem









F1=Help             F2=Refresh          F3=Cancel           F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 14-15 Add Local Printer Configuration - System V Print Spooling

The PRINTER name and DEVICE name fields require input, as shown in Figure 14-16 on page 645 and Figure 14-17 on page 645.

```
                        Add Local Printer Configuration

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[TOP]                                                  [Entry Fields]
* PRINTER name                                       []
* DEVICE name                                        []
    Device is hardwired or a terminal?                 hardwired              +
  COMMENT                                            []
  CLASS name                                         []                       +

  ---------- Interface Program Options ----------

  MODEL                                                standard               +
  Printer TYPES                                      []                       +
  CONTENT TYPE                                       [simple]                 +

  --------- Header/Trailer Page Options ---------
[MORE...8]

F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 14-16   Add a Local Printer - System V Print Spooling

```
                        Add Local Printer Configuration

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[MORE...8]                                             [Entry Fields]
  MODEL                                                standard               +
  Printer TYPES                                      []                       +
  CONTENT TYPE                                       [simple]                 +

  --------- Header/Trailer Page Options ---------

  Force BANNER?                                        yes                    +

  ----------- Error Processing Options ----------

  ALERT type on error                                [mail]                   +
    WAIT before alert in minutes                     []                        #
  RECOVERY on fault                                    beginning              +
[BOTTOM]

F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 14-17   Add a Local Printer - System V Print Spooling (more)

14.10.2 Add a remote printer

Select Add Remote Printer Configuration from the System V Print Spooling menu, as shown in Figure 14-18.

```
                              System V Print Spooling

Move cursor to desired item and press Enter.

  Manage Print Requests
  List Print Destinations
  Remove Destinations
  Manage Destinations
  Add Local Printer Configuration
  Add Remote Printer Configuration
  Add JetDirect Printer Configuration
  Manage Print Services
  Manage Remote Systems
  Change / Show Current Print Subsystem








F1=Help             F2=Refresh          F3=Cancel           F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 14-18 Add Remote Printer Configuration - System V Print Spooling

The PRINTER name, SERVER name, and PRINTER NAME on server are mandatory fields, as shown in Figure 14-19 on page 647.

```
                         Add Remote Printer Configuration

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                        [Entry Fields]
* PRINTER name                                       []
* SERVER name                                        []                      +
* PRINTER NAME on server                             []
  COMMENT                                            []
  CLASS name                                         []                      +

  ---------- Interface Program Options ----------

  Printer TYPES                                      []                      +
  CONTENT TYPE                                       [simple]                +

  --------- Header/Trailer Page Options ---------

  Force BANNER?                                       yes                    +

F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```
Figure 14-19 Add a Remote Printer - System V Print Spooling

14.10.3 Remove a printer

Select Remove Destinations from the System V Print Spooling menu, as shown in Figure 14-20.

```
                          System V Print Spooling

Move cursor to desired item and press Enter.

  Manage Print Requests
  List Print Destinations
  Remove Destinations
  Manage Destinations
  Add Local Printer Configuration
  Add Remote Printer Configuration
  Add JetDirect Printer Configuration
  Manage Print Services
  Manage Remote Systems
  Change / Show Current Print Subsystem








F1=Help              F2=Refresh           F3=Cancel            F8=Image
F9=Shell             F10=Exit             Enter=Do
```

Figure 14-20 Remove Destinations - System V Spooling

In this case, only the DESTINATIONS field is mandatory, as shown in Figure 14-21 on page 649.

```
                          Remove Destination

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                   [Entry Fields]
* DESTINATION name                              []                        +













F1=Help              F2=Refresh           F3=Cancel            F4=List
F5=Reset             F6=Command           F7=Edit              F8=Image
F9=Shell             F10=Exit             Enter=Do
```

Figure 14-21 Remove a printer - System V Spooling

14.10.4 Manage destinations

Select Manage Destinations from the System V Print Spooling menu, as shown in Figure 14-22.

```
                          System V Print Spooling

Move cursor to desired item and press Enter.

  Manage Print Requests
  List Print Destinations
  Remove Destinations
  Manage Destinations
  Add Local Printer Configuration
  Add Remote Printer Configuration
  Add JetDirect Printer Configuration
  Manage Print Services
  Manage Remote Systems
  Change / Show Current Print Subsystem










F1=Help             F2=Refresh          F3=Cancel           F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 14-22 Manage Destinations - System V Spooling

There are several choices available to manage printers, as shown in Figure 14-23 on page 651.

```
                           Manage Destinations

Move cursor to desired item and press Enter.

  List Print Destinations
  Change / Show Default Destination
  Accept Requests for Destination
  Reject Requests for Destination
  Enable Printer
  Disable Printer
  Change / Show Default Print Priority for All Destinations
  Remove Destination










F1=Help              F2=Refresh           F3=Cancel            F8=Image
F9=Shell             F10=Exit             Enter=Do
```

Figure 14-23 Options in Manage Destinations - System V Spooling

14.10.5 The lpadmin command

The **lpadmin** command is a System V Print Subsystem command that configures the LP print services by defining printers and devices. It is used to:

- Add and change printers
- Remove printers from the service
- Set or change the system default destination
- Define alerts for printer faults
- Mount print wheels
- Define printers for remote printing services

For additional information about network printers, see the **lpsystem** command.

The **lpadmin** command follows different syntaxes for different print services. The following is a description of some of these services and the syntaxes followed.

- Adding or changing the configuration of a local printer:

  lpadmin -p Printer -v Device [-D Comment] [-A AlertType] [-W Minutes] [-c Class] [-e Printer1] [-F FaultRecovery] [-f allow:FormList | -f denyFormList] [-h] [-I Content-Type-List] [-i Interface] [-l] [-M -f Form-Name [-o File-break]] [-M -S Print-Wheel] [-m Model] [-O Copy-Options] [-o Print-Options] [-o nobanner | -o banner] [-r Class] [-S List] [-s Server-Name [!ServerPrinterName]] [-T Printer-Type-List] [-u allow:Login-Id-List | -u deny:Login-Id-List]]

- Adding or changing the configuration of a remote printer:

  lpadmin -p Printer -s ServerName [!ServerPrinterName] -v Device [-D Comment] [-A AlertType] [-W Minutes] [-c Class] [-e Printer1] [-F FaultRecovery] [-f allow:FormList | -f denyFormList] [-h] [-I Content-Type-List] [-i Interface] [-l] [-M -f Form-Name [-o Filebreak]] [-M -S Print-Wheel] [-m Model] [-O CopyOptions] [-o PrintOptions] [-o nobanner | -o banner] [-r Class] [-S List] [-T PrinterTypeList] [-u allow:LoginIdList | -u deny:LoginIdList] [-v Device]

The -p Printer flag is used to configure a new printer or to change the configuration of an existing printer. When you use this form of the **lpadmin** command, one of the following must be selected:

- -v Device, required to configure a local printer
- -s ServerNname [!ServerPrinterName], required to configure a remote printer

Removing a printer destination

The **lpadmin** command has the following syntax to remove a printer:

```
lpadmin -x [ Destination ]
```

The -x dest flag removes the destination dest (a printer or a class) from the LP print service. If dest is a printer and is the only member of a class, then the class is deleted. If dest is all, all printers and classes are removed. No other parameters are allowed with -x.

Setting or changing the system default destination

The **lpadmin** command has the following syntax to set or change the system default destination:

```
lpadmin -d [ Destination ]
```

The -d [dest] flag makes dest, an existing printer or class, the new system default destination. If dest is not supplied, then there is no system default destination. No other parameters are allowed with -d. To unset the system default printer, the user can enter the keyword none.

Setting an alert for a print wheel

The **lpadmin** command has the following syntax to set an alert:

```
lpadmin -S Print-Wheel -A AlertType [ -W Minutes ] [ -Q Requests ]
```

The -S Print-Wheel flag is used with the -A Alert-Type flag to define an alert to mount the print wheel when there are jobs queued for it. If this command is not used to arrange alerting for a print wheel, no alert will be sent for the print wheel.

The Alert-Types are the same as those available with the -A flag: mail, write, quiet, none, shell-command, and list. See the description of -A for details about each.

If the -W flag is not given, the default procedure is that only one message will be sent per need to mount the print wheel. Not specifying the -W flag is equivalent to specifying -W once or -W 0. If Minutes is a number greater than zero, an alert is sent at intervals specified by minutes.

If the -Q flag is also given, the alert is sent when a certain number (specified by the argument requests) of print requests that need the print wheel are waiting. If the -Q flag is not given, or requests is 1 or the word any (which are both the default), a message is sent as soon as anyone submits a print request for the print wheel when it is not mounted.

14.11 The switch.prt command

The printing subsystem can be switched between AIX 5L Print Spooling and System V Print Spooling with the **switch.prt** command. For this command to work, System V print subsystem has to be enabled on the system.

Abbreviations and acronyms

ABI	Application Binary Interface
AC	Alternating Current
ACL	Access Control List
AFPA	Adaptive Fast Path Architecture
AIO	Asynchronous I/O
AIX	Advanced Interactive Executive
APAR	Authorized Program Analysis Report
API	Application Programming Interface
ARP	Address Resolution Protocol
ASMI	Advanced System Management Interface
BFF	Backup File Format
BIND	Berkeley Internet Name Domain
BIST	Built-In Self-Test
BLV	Boot Logical Volume
BOOTP	Boot Protocol
BOS	Base Operating System
BSD	Berkeley Software Distribution
CA	Certificate Authority
CATE	Certified Advanced Technical Expert
CD	Compact Disk
CDE	Common Desktop Environment
CD-R	CD Recordable
CD-ROM	Compact Disk-Read Only Memory
CEC	Central Electronics Complex
CHRP	Common Hardware Reference Platform
CLI	Command-Line Interface
CLVM	Concurrent LVM
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
CSM	Cluster Systems Management
CoD	Capacity Upgrade on Demand
DCM	Dual Chip Module
DES	Data Encryption Standard
DGD	Dead Gateway Detection
DHCP	Dynamic Host Configuration Protocol
DLPAR	Dynamic LPAR
DMA	Direct Memory Access
DNS	Domain Naming System
DRM	Dynamic Reconfiguration Manager
DR	Dynamic Reconfiguration
DVD	Digital Versatile Disk
EC	EtherChannel
ECC	Error Correction Code
EOF	End of File
EPOW	Environmental and Power Warning
ERRM	Event Response resource manager
ESS	Enterprise Storage Server®
F/C	Feature Code
FC	Fibre Channel
FCAL	Fibre Channel Arbitrated Loop
FDX	Full Duplex

© Copyright IBM Corp. 2006. All rights reserved.

FLOP	Floating Point Operation
FRU	Field Replaceable Unit
FTP	File Transfer Protocol
GDPS®	Geographically Dispersed Parallel Sysplex™
GID	Group ID
GPFS	General Parallel File System
GUI	Graphical User Interface
HACMP	High Availability Cluster Multi Processing
HBA	Host Bus Adapters
HMC	Hardware Management Console
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
Hz	Hertz
I/O	Input/Output
IBM	International Business Machines
ID	Identification
IDE	Integrated Device Electronics
IEEE	Institute of Electrical and Electronics Engineers
IP	Internetwork Protocol
IPAT	IP Address Takeover
IPL	Initial Program Load
IPMP	IP Multipathing
ISV	Independent Software Vendor
ITSO	International Technical Support Organization
IVM	Integrated Virtualization Manager
JFS	Journaled File System
L1	Level 1
L2	Level 2
L3	Level 3
LA	Link Aggregation
LACP	Link Aggregation Control Protocol
LAN	Local Area Network
LDAP	Lightweight Directory Access Protocol
LED	Light Emitting Diode
LMB	Logical Memory Block
LPAR	Logical Partition
LPP	Licensed Program Product
LUN	Logical Unit Number
LV	Logical Volume
LVCB	Logical Volume Control Block
LVM	Logical Volume Manager
MAC	Media Access Control
Mbps	Megabits Per Second
MBps	Megabytes Per Second
MCM	Multichip Module
ML	Maintenance Level
MP	Multiprocessor
MPIO	Multipath I/O
MTU	Maximum Transmission Unit
NFS	Network File System
NIB	Network Interface Backup
NIM	Network Installation Management
NIMOL	NIM on Linux
NVRAM	Non-Volatile Random Access Memory
ODM	Object Data Manager
OSPF	Open Shortest Path First
PCI	Peripheral Component Interconnect
PIC	Pool Idle Count
PID	Process ID
PKI	Public Key Infrastructure
PLM	Partition Load Manager
POST	Power-On Self-test

POWER	Performance Optimization with Enhanced Risc (Architecture)
PPC	Physical Processor Consumption
PPFC	Physical Processor Fraction Consumed
PTF	Program Temporary Fix
PTX®	Performance Toolbox
PURR	Processor Utilization Resource Register
PV	Physical Volume
PVID	Physical Volume Identifier
PVID	Port Virtual LAN Identifier
QoS	Quality of Service
RAID	Redundant Array of Independent Disks
RAM	Random Access Memory
RAS	Reliability, Availability, and Serviceability
RCP	Remote Copy
RDAC	Redundant Disk Array Controller
RIO	Remote I/O
RIP	Routing Information Protocol
RISC	Reduced Instruction-Set Computer
RMC	Resource Monitoring and Control
RPC	Remote Procedure Call
RPL	Remote Program Loader
RPM	Red Hat Package Manager
RSA	Rivet, Shamir, Adelman
RSCT	Reliable Scalable Cluster Technology
RSH	Remote shell
SAN	Storage Area Network
SCSI	Small Computer System Interface
SDD	Subsystem Device Driver
SMIT	System Management Interface Tool
SMP	Symmetric Multiprocessor
SMS	System Management Services
SMT	Simultaneous multithreading
SP	Service Processor
SPOT	Shared Product Object Tree
SRC	System Resource Controller
SRN	Service Request Number
SSA	Serial Storage Architecture
SSH	Secure Shell
SSL	Secure Socket Layer
SUID	Set User ID
SVC	SAN Virtualization Controller
TCP/IP	Transmission Control Protocol/Internet Protocol
TSA	Tivoli System Automation
UDF	Universal Disk Format
UDID	Universal Disk Identification
VIPA	Virtual IP Address
VG	Volume Group
VGDA	Volume Group Descriptor Area
VGSA	Volume Group Status Area
VLAN	Virtual Local Area Network
VP	Virtual Processor
VPD	Vital Product Data
VPN	Virtual Private Network
VRRP	Virtual Router Redundancy Protocol
VSD	Virtual Shared Disk
WLM	Workload Manager

Related publications

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

IBM Redbooks

For information about ordering these publications, see “How to get IBM Redbooks” on page 662. Note that some of the documents referenced here may be available in softcopy only.

- *Advanced POWER Virtualization on IBM System p5*, SG24-7940
- *Effective System Management Using the IBM Hardware Management Console for pSeries*, SG24-7038
- *i5/OS on IBM eServer p5 Models - A Guide to Planning, Implementation, and Operation*, SG24-8001
- *IBM eServer Certification Study Guide eServer p5 and pSeries Enterprise Technical Support AIX 5L V5.3*, SG24-7197
- *IBM eServer p5 590 and 595 System Handbook*, SG24-9119
- *IBM eServer p5 590 and 595 Technical Overview and Introduction*, REDP-6024
- *Introduction to pSeries Provisioning*, SG24-6389.
- *Linux Applications on pSeries*, SG24-6033
- *Managing AIX Server Farms*, SG24-6606
- *NIM: From A to Z in AIX 4.3,* SG24-5524
- *Partitioning Implementations for IBM eServer p5 Servers*, SG24-7039
- *A Practical Guide for Resource Monitoring and Control (RMC)*, SG24-6615
- *Practical Guide for SAN with pSeries*, SG24-6050
- *Problem Solving and Troubleshooting in AIX 5L*, SG24-5496
- *Understanding IBM eServer pSeries Performance and Sizing*, SG24-4810

© Copyright IBM Corp. 2006. All rights reserved.

Other publications

These publications are also relevant as further information sources:

- *AIX 5L Version 5.3 Installation Guide and Reference*, SC23-4887
- *RS/6000 & eServer pSeries Adapter Placement Reference for AIX,* SA38-0538
- The following types of documentation are located through the Internet at the following URL:

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

  – User guides
  – System management guides
  – Application programmer guides
  – All commands reference volumes
  – Files reference
  – Technical reference volumes used by application programmers

Online resources

These Web sites and URLs are also relevant as further information sources:

- AIX 5L operating system maintenance packages downloads.

  `http://www.ibm.com/servers/eserver/support/pseries/aixfixes.html`

- Autonomic computing on IBM eserver pSeries servers.

  `http://www.ibm.com/autonomic/index.shtml`

- Ceramic Column Grid Array (CCGA), see IBM Chip Packaging.

  `http://www.ibm.com/chips/micronews`

- Copper circuitry.

  `http://www.ibm.com/chips/technology/technologies/copper/`

- Frequently asked SSA-related questions.

  `http://www.storage.ibm.com/hardsoft/products/ssa/faq.html`

- Hardware documentation.

  `http://publib16.boulder.ibm.com/pseries/en_US/infocenter/base/`

- IBM AIX Toolbox Download Page - RPM Group Classifications.

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

- IBM Certification Web site.

  http://www.ibm.com/certify

- IBM Directory of worldwide contacts.

  http://www.ibm.com/planetwide/

- IBM Electronic Service Call sign-in page.

  http://www.ibm.com/support/esc/signin.jsp

- IBM eserver Information Center.

  http://publib.boulder.ibm.com/eserver/

- IBM eserver iSeries: Getting Started with Dynamic LPAR - IBM LPAR Validation Tool.

  http://www.ibm.com/servers/eserver/iseries/lpar/systemdesign.html

- IBM eserver pSeries and RS/6000 microcode update.

  http://techsupport.services.ibm.com/server/mdownload2/download.html

- IBM eserver pSeries support.

  http://www.ibm.com/servers/eserver/support/pseries/index.html

- IBM eserver support: Tips for AIX 5L administrators.

  http://techsupport.services.ibm.com/server/aix.srchBroker

- IBM Linux news: Subscribe to the Linux Line.

  https://www6.software.ibm.com/reg/linux/linuxline-i

- IBM online sales manual.

  http://www.ibmlink.ibm.com

- IBM Support Contracts login.

  http://www.ibm.com/software/support/esr/support_contracts

- IBM System p5 servers - Facts and features reports.

  http://www.ibm.com/systems/p/hardware/factsfeatures.html

- Index of ftp://ftp.software.ibm.com/aix/tools/perftools/perfpmr.

  ftp://ftp.software.ibm.com/aix/tools/perftools/perfpmr

- Information about UnitedLinux for pSeries from Turbolinux.

  http://www.turbolinux.co.jp

- Linux for IBM eserver pSeries.

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

- The LVT is a PC based tool intended assist you in logical partitioning.

  http://www-1.ibm.com/servers/eserver/iseries/lpar/systemdesign.htm

- Microcode Discovery Service.

  http://techsupport.services.ibm.com/server/aix.invscoutMDS

- The perfpmr tool ZIP file.

  ftp://ftp.software.ibm.com/aix/tools/perftools/perfpmr/perf53/perf53.tar.Z

- POWER4 system micro architecture, comprehensively described in the *IBM Journal of Research and Development*, Vol 46 No.1 January 2002.

  http://www.research.ibm.com/journal/rd46-1.html

- pSeries and AIX Information Center.

- http://publib.boulder.ibm.com/infocenter/pseries/index.jspSCSI T10 Technical Committee.

  http://www.t10.org

- Silicon-on-insulator (SOI) technology.

  http://www.ibm.com/chips/technology/technologies/soi/

- SSA boot FAQ.

  http://www.storage.ibm.com/hardsoft/products/ssa/faq.html#microcode

- SUSE Linux Enterprise Server 8 for pSeries information.

  http://www.suse.de/us/business/products/server/sles/i_pseries.html

How to get IBM Redbooks

You can search for, view, or download Redbooks, Redpapers, Hints and Tips, draft publications and Additional materials, as well as order hardcopy Redbooks or CD-ROMs, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

© Copyright IBM Corp. 2006. All rights reserved.

B

C

D

F

G

H

J

K

M

O

S

T

U

V

W

X

Z

IBM Certification Study Guide
@server p5 and pSeries Administration and
Support for AIX 5L Version 5.3

(1.0” spine)
0.875”<->1.498”
460 <-> 788 pages

IBM Certification Study Guide eServer p5 and pSeries Administration and Support for AIX 5L Version 5.3

Redbooks

Developed specifically for the purpose of preparing for AIX 5L certification Test 222

Makes an excellent companion to classroom education

For AIX 5L Version 5.3 professionals

This IBM Redbook is designed as a study guide for professionals wishing to prepare for the certification exam (Test 222) to achieve: IBM Certified Specialist - eServer p5 and pSeries Administration and Support for AIX 5L Version 5.3. The certification validates a broad scope of AIX 5L administration, support, problem determination skills, and the ability to perform general AIX 5L software system maintenance.

This redbook helps AIX 5L professionals seeking a comprehensive and task-oriented guide for developing the knowledge and skills required for the certification. It is designed to provide a combination of theory and practical experience needed for a general understanding of the subject matter.

This redbook does not replace practical experience you should have, but is an effective tool that, when combined with education activities and experience, should prove to be a very useful preparation guide for the exam. Due to the practical nature of the certification content, this publication can also be used as a desk-side reference. So, whether you are planning to take the eServer p5 and pSeries Administration and Support for AIX 5L Version 5.3 certification exam, or if you just want to validate your AIX 5L skills, this redbook is for you.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-7199-00 ISBN 0738496499