SG24-6185-00

IBM Certified Specialist

IBM Certification Study Guide AIX Version 4.3 Problem Determination

Thomas C. Cederlöf
André de Klerk
Thomas Herlin
Tomasz Ostaszewski

Redbooks

ibm.com/redbooks

SG24-6185-00

International Technical Support Organization

IBM Certification Study Guide
AIX Version 4.3
Problem Determination

October 2000

Take Note!

Before using this information and the product it supports, be sure to read the general information in Appendix B, “Special notices” on page 261.

First Edition (October 2000)

This edition applies to AIX Version 4.3 (5765-C34) and subsequent releases running on an RS/6000 server.

This document created or updated on July 19, 2000.

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. JN9B Building 003 Internal Zip 2834
11400 Burnet Road
Austin, Texas 78758-3493

Contents

Figures

Tables

Preface

The AIX and RS/6000 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 and RS/6000. A complete set of professional certifications are available. They include:

- IBM Certified AIX User
- IBM Certified Specialist - AIX System Administration
- IBM Certified Specialist - AIX System Support
- IBM Certified Specialist - AIX HACMP
- IBM Certified Specialist - Business Intelligence for RS/6000
- IBM Certified Specialist - Domino for RS/6000
- IBM Certified Specialist - RS/6000 Solution Sales
- IBM Certified Specialist - RS/6000 SP and PSSP V3
- IBM Certified Specialist - RS/6000 SP
- RS/6000 SP - Sales Qualification
- IBM Certified Specialist - Web Server for RS/6000
- IBM Certified Advanced Technical Expert - RS/6000 AIX

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 that 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 of this certification is the value of appropriate measurements of the skills required to perform the job role.

This Redbook is designed as a study guide for professionals wishing to prepare for the Installation and System Recovery certification exam as a selected course of study in order to achieve: IBM Certified Advanced Technical Expert - RS/6000 AIX.

This Redbook is designed to provide a combination of theory and practical experience needed for a general understanding of the subject matter. It also provides sample questions that will help in the evaluation of personal progress and provide familiarity with the types of questions that will be encountered in the exam.

This publication does not replace practical experience or is designed to be a stand alone guide for any subject. Instead, it is an effective tool that, when combined with education activities and experience, can be a very useful preparation guide for the exam.

For additional information about certification and instructions on *How to Register* for an exam call IBM at 1-800-426-8322 or visit the 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.

Thomas C. Cederlöf is a Education Specialist at IBM Learning Services in Sweden. After working various professions, he was hired as a System Support Specialist in April 1997 at the Nordic AIX Competence Center. After earning his Advanced Technical Expert Certification in 1998 he worked with level 2 support in Scandinavia and the Baltic States, and participated also in the itrans program in 1999. Since January 2000 he is the main instructor for the AIX curriculum in Sweden.

André de Klerk is a Senior IT Specialist at IBM Global Services in South Africa. He has been working for IBM since May 1996. He started his career as a field technician in 1991 and has performed various support roles including application support and customer consulting. Currently he is team leader for the Midrange UNIX team at IGS SA.

Thomas Herlin is an Advisory IT Specialist at IBM Global Services in Denmark. He has been working for IBM since May 1998. Before joining IBM he has been working as a Software Engineer designing and developing programs on UNIX platforms. His areas of expertise include system architecture and system integration of AIX based solutions. He is also a certified SAP technical consultant.

Tomasz Ostaszewski is a computer network architect. He works for Prokom Software SA in Poland - IBM Business Partner. Prokom is the largest IT solution provider in Poland. They offer total solutions which include application development or third party vendor support. He has three years of experience in RS/6000 and AIX. Currently he is working on network project for an insurances company.

The project that produced this publication was managed by:

Scott Vetter	IBM Austin

Thanks to:

Bill Hughes	Program Manager, AIX & RS/6000 Certification

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

Shawn Mullen	IBM Austin
Robert Olsson	ILS Sweden
Malin Cederberg	ILS Sweden
Karl Borman	ILS Austin

Comments welcome

Your comments are important to us!

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

- Fax the evaluation form found in “IBM Redbooks review” on page 283 to the fax number shown on the form.
- Use the online evaluation form found at **ibm.com**/redbooks
- Send your comments in an Internet note to redbook@us.ibm.com

Chapter 1. Certification overview

This chapter provides an overview of the skill requirements needed to obtain an IBM AIX Specialist certification. The following chapters are designed to provide a comprehensive review of specific topics that are essential for obtaining the certification.

1.1 IBM Certified Advanced Technical Expert - RS/6000 AIX

This level certifies an advanced level of AIX knowledge and understanding, both in breadth and depth. It verifies the ability to perform in-depth analysis, apply complex AIX concepts and provide resolution to critical problems, all in a variety of areas within RS/6000 AIX.

To attain the IBM Certified Advanced Technical Expert - RS/6000 AIX certification, you must pass four tests.

One test is the prerequisite in either AIX System Administration or AIX System Support. The other three tests are selected from a variety of AIX and RS/6000 topics. These requirements are explained in greater detail in the sections that follow.

1.1.1 Required prerequisite

Prior to attaining the IBM Certified Advanced Technical Expert - RS/6000 AIX certification, you must be certified as either an:

- IBM Certified Specialist - AIX System Administration

  or

- IBM Certified Specialist - AIX System Support

1.1.2 Recommended prerequisite

A minimum of six to twelve months experience in performing in-depth analysis and applying complex AIX concepts in a variety of areas within RS/6000 AIX is a recommended prerequisite.

1.1.3 Registration for the certification exam

For information about *How to Register* for the certification exam, please visit the following Web site:

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

1.1.4 Core requirement (select three of the following tests)

You will receive a Certificate of Proficiency for tests when passed.

1.1.4.1 AIX V4.3 Installation and System Recovery

The following objectives were used as a basis when the certification test 183 was developed. Some of these topics have been regrouped to provide better organization when discussed in this publication.

Preparation for this exam is the topic of this publication.

Section 1 - Installation and software maintenance

- Install or migrate the operating system
- Install a licensed program product
- Remove an OPP or an LPP from the system
- Update a system
- Apply a selective fix
- Identify and resolve network install problems

Section 2 - System backup and restore

- Perform a complete backup of the system
- Implement backup using relative and absolute path
- Create a mksysb
- Understand advanced mksysb concepts
- Restore files

Section 3 - System initialization (boot) failures

- Understand concepts of system initialization
- Diagnose the cause of a system initialization failure
- Resolve a system initialization failure

Section 4 - File systems and LVM recovery

- Perform problem determination on a filesystem
- Determine a suitable procedure for replacing a disk
- Resolve problems caused by incorrect actions taken to change a disk drive
- Create a new volume group
- Create a logical volume

- Understand LVM concepts
- Resolve a complex LVM problem

1.1.4.2 AIX V4.3 Performance and System Tuning

The following objectives were used as a basis when the certification test 184 was developed.

Preparation for this exam is the topic of *IBM Certification Study Guide - AIX Performance and System Tuning*, SG24-6184.

Section 1 - Performance Tools & Techniques

- Use the `iostat` command
- Use the `filemon` command
- Use the `tprof` command
- Use the `netpmon` command
- Interpret `iostat` output
- Interpret `lsps` output
- Interpret `netstat` output
- Interpret `vmstat` output
- Know about perfpmr
- Know about performance diagnostic tool
- Look at run queue
- Look at system calls

Section 2 - Correcting performance problems

- Correct disk bottlenecks
- Correct NFS bottlenecks
- Correct network bottlenecks
- Correct communications adapter bottlenecks
- Understand random write-behind concepts
- Understand async I/O performance concepts
- Understand VMM I/O pacing
- Understand file fragmentation
- Understand logical volume fragmentation

Section 3 - VMM

- Identify and correct VMM performance problems
- Correct paging problems
- Know about Tuning File Memory Usage
- Know about memory load control
- Understand Page Space Allocation issues

Section 4 - Multiprocessor and process scheduling

- Know SMP commands
- Use the `bindprocessor` command
- Enable, disable, and show status of processors
- List CPU utilization per processor
- Know about `ps` command and threads
- Understand locking issues in SMP
- Know about process scheduling
- Understand priority calculations
- Understand the effect of schedtune on priorities

Section 5 - Tuning and customization

- Tune a system for optimum performance
- Use the `no` command
- Customize a LV for optimum performance
- Configure system parameters
- Tune network parameters
- Determine when application tuning is needed
- Understand real-time tuning
- Understand disk striping
- Tune I/O performance with `vmtune`
- Understand RAID performance issues
- Perform capacity planning
- Understand memory usage

1.1.4.3 AIX V4.3 Problem Determination Tools and Techniques

The following objectives were used as a basis when the certification test 185 was developed.

Preparation for this exam is the topic of *IBM Certification Study Guide - AIX Problem Determination Tools and Techniques*, SG24-6185.

Section 1 - System dumps

- Create a system dump
- Understand valid system dump devices
- Determine the location of system dump data
- Identify the status of a system dump by the LED codes
- Identify appropriate action to take after a system dump
- Determine if a system dump is successful
- Use the `snap` command

Section 2 - Crash

- Understand the use and purpose of the crash command
- Verify the state of a system dump
- Show the stack trace using crash
- Use the `stat` subcommand in crash
- Manipulate data in the process table
- Interpret crash stack trace output
- Interpret crash process output
- Interpret crash TTY output

Section 3 - Trace

- Start and stop trace
- Run trace
- Report trace information
- Interpret trace output
- Use trace to debug process problems

Section 4 - File system and performance PD tools

- Use tools to identify and correct corrupted file systems
- Understand file system characteristics

- Resolve file system mounting problems
- Repair corrupted file systems
- Use `vmstat` command
- Use `iostat` command
- Use `filemon` command

Section 5 - Network problem determination

- Use PD tools to identify network problems
- Resolve a network performance problem
- Correct problem with host name resolution
- Diagnose the cause of a problem with NFS mounts
- Diagnose the cause of a routing problem
- Resolve a router problem

Section 6 - Error logs and diagnostics

- Use error logging
- Interpret error reports
- Invoke and use diagnostic programs

Section 7 - Other problem determination tools

- Set breakpoints using `dbx`
- Step through a program using `dbx`
- Run a program with arguments using `dbx`
- Read core files and locate traceback
- Debug problem using core files
- Read shell scripts
- Debug shell script problems

1.1.4.4 AIX V4.3 Communications

The following objectives were used as a basis when the certification test 186 was developed.

Preparation for this exam is the topic of *IBM Certification Study Guide - AIX Communications*, SG24-6186.

Section 1 - TCP/IP implementation

- Know TCP/IP concepts

- Understand TCP/IP broadcast packets
- Use and implement name resolution
- Understand TCP/IP protocols
- Know IP address classes
- Use interfaces available in LAN communications
- Understand the relationship between an IP address and the network interface
- Log into remote hosts using telnet and rologin
- Construct /etc/hosts.equiv and ~/.rhosts for trusted users
- Transfer files between systems using ftp or tftp
- Run commands on remote machines

Section 2 - TCP/IP: DNS implementation

- Setup a primary name server
- Setup a secondary name server
- Setup a client in a domain network

Section 3 - Routing: implementation

- Apply knowledge of the IP routing algorithm
- Setup and use the routing table and routes
- Implement and use subnet masking

Section 4 - NFS: implementation

- Manipulate local and remote mounts using the automounter
- Understand NFS daemons and their roles
- Configure and tune an NFS server
- Configure and tune an NFS client
- Setup a file system for mounting
- Understand the /etc/exports file
- Invoke a predefined mount.

Section 5 - NIS: implementation

- Understand the various NIS daemons
- Implement NIS escapes
- Create NIS map files

- Transfer NIS maps

Section 6 - Network problem determination

- Diagnose and resolve TCP/IP problems
- Diagnose and resolve NFS problems
- Diagnose and resolve NIS problems

Section 7 - Hardware related PD (modems)

- Determine appropriate diagnostic approach to resolve a modem connection problem
- Resolve communication configuration problems

1.1.4.5 HACMP for AIX V4.2

The following objectives were used as a basis when the certification test 167 was developed.

Preparation for this exam is the topic of *IBM Certification Study Guide - AIX HACMP*, SG24-5131.

Section 1 - Pre-installation

- Conduct a planning session
    - Set customer expectations at the beginning of the planning session
    - Gather customer's availability requirements
    - Articulate tradeoffs of different HA configurations
    - Assist customer in identifying HA applications
- Evaluate customer environment and tailorable components
    - Evaluate configuration and identify Single Points of Failure (SPOF)
    - Define and analyze NFS requirements
    - Identify components affecting HACMP
    - Identify HACMP event logic customizations
- Plan for installation
    - Develop disk management modification plan
    - Understand issues regarding single adapter solutions
    - Produce a test plan

Section 2 - HACMP implementation

- Configure HACMP solutions

    - Install HACMP code
    - Configure IP Address Takeover (IPAT)
    - Configure non IP heartbeat paths
    - Configure network adapter
    - Customize/tailor AIX
    - Set up shared disk (SSA)
    - Set up shared disk (SCSI)
    - Verify a cluster configuration
    - Create an application server
- Setup event notification
    - Set up event notification and pre/post event scripts
    - Setup error notification
- Post configuration activities
    - Configure client notification and ARP update
    - Implement test plan
    - Create a snapshot
    - Create a customization document
- Testing and Troubleshooting
    - Troubleshoot failed IPAT failover
    - Troubleshoot failed shared volume groups
    - Troubleshoot failed shared volume groups
    - Troubleshoot failed network configuration
    - Troubleshoot failed shared disk tests
    - Troubleshoot failed application
    - Troubleshoot failed pre/post event scripts
    - Troubleshoot failed error notifications
    - Troubleshoot errors reported by cluster verification

Section 3 - System management

- Communicate with customer
    - Conduct turnover session
    - Provide hands-on customer education

  - Set customer expectations of their HACMP solution's capabilities
- Perform systems maintenance
  - Perform HACMP maintenance tasks (PTFs, adding products, replacing disks, adapters)
  - Perform AIX maintenance tasks
  - Dynamically update cluster configuration
  - Perform testing and troubleshooting as a result of changes

1.1.4.6 RS/6000 SP and PSSP V2.4

The following objectives were used as a basis when the certification test 178 was developed.

Preparation for this exam is the topic of *IBM Certification Study Guide - RS/6000 SP*, SG24-5348.

Section 1 - Implementation and planning

- Validate software/hardware capability and configuration.
  - Determine required software levels (for example., version, release, and modification level).
  - Determine the size, model and location of the control workstation.
  - Define disk, memory, and I/O including disk placement.
  - Determine disk space requirements.
  - Understand multi-frame requirements and switch partitioning.
  - Determine the number and type of nodes needed (including features).
  - Determine the number of types of I/O devices (for example, SCSI, RAID, SSA, etc.) needed.
  - Configure external I/O connections.
  - Determine additional network connections required.
  - Create the logical plan for connecting into networks outside the SP.
  - Identify the purpose and bandwidth of connections.
- Plan implementation of key aspects of TCP/IP networking in the SP environment.
  - Create specific host names (both fully qualified and aliases), TCP/IP address,
  - Netmask value and default routes.

    - Determine the mechanism (for example, /etc/hosts, NIS, DNS) by which they will be made available across the system.
    - Choose the IP name/address resolver.
- Determine the appropriate common, distributed, and local files/file systems.
    - Determine the physical locations of the file system and home directories.
    - Determine the number of types of I/O devices (for example, SCSI, RAID, SSA, etc.) needed.
    - Configure internal I/O.
    - Determine the mechanism (for example, NFS, AFS, DRS, local) by which they will be made available across the system.
- Configure and administer the Kerberos Authentication subsystem and manage user IDs on the SP system.
    - Define administrative functions.
    - Determine the Kerberos administration ID.
    - Define Administrative functions
    - Understand the options of end-user management.
    - Understand how to administer authenticated users and instances.
- Define a backup/recovery strategy for the SP which supports node images, control workstation images, applications, and data.
    - Determine backup strategy and understand the implications of multiple unique mksysb images.

Section 2 - Installation and configuration

- Configure an RS/6000 as an SP control workstation.
    - Verify the control workstation system configuration.
    - Configure TCP/IP network on the control workstation.
    - Install PSSP.
    - Load the SDR with SP configuration information.
    - Configure the SP System Data Repository.
    - Verify control workstation software.
    - Configure TCP/IP name resolution (for example, /etc/passwd, DNS, NIS).

- Perform network installation of images on nodes, using any combination of boot/install servers.
  - Install the images on the nodes.
  - Create boot/install servers
- Exercise the SP system resources to verify the correct operation of all required subsystems.
  - Verify all network connections.
  - Verify internal and external I/O connections.
  - Verify switch operations

Section 3 - Application enablement

- Determine whether LoadLeveler would be beneficial to a given SP system configuration.
  - Understand the function of LoadLeveler.
- Define and implement application specific FS, VG, and VSDs for a parallel application.
  - Define application-specific file systems, logical volumes, volume groups, or VSDs.
  - Implement application-specific file systems, logical volumes, volume groups, or VSDs.
- Install and configure problem management tools (for example, event manager, problem manager, perspectives)
  - Install and Configure user-management tools.

Section 4 - Support

- Utilize Problem Determination methodologies (for example, HOSTRESPONDS, SWITCHRESPONDS, error report, log files, DAEMONS, GUIs).
  - Handle resolution of critical problems.
  - Conduct SP-specific problem diagnosis.
  - Interpret error logs that are unique to SP.
- Isolate cause of degraded SP performance, and tune the system accordingly.
  - Understand performance analysis and tuning requirements

1.1.4.7 RS/6000 SP and PSSP V3

The following objectives were used as a basis when the certification test 188 was developed.

Preparation for this exam is the topic of *IBM Certification Study Guide - RS/6000 SP*, SG24-5348.

Section 1 - Implementation planning

- Validate software/hardware capability and configuration.
  - Determine required software levels (for example, version, release, and modification level)
  - Determine the size, model and location of the control workstation.
  - Define disk, memory, and I/O including disk replacement.
  - Define disk space requirements.
  - Understand multi-frame requirements and switch partitioning.
  - Determine the number and types of nodes needed (including features).
  - Determine the number and types of I/O devices (for example, SCSI, RAID,SSA, etc.) needed (including features).
  - Configure external I/O connections.
  - Determine additional network connections required.
  - Create the logical plan for connecting into networks outside the SP.
  - Identify the purpose and bandwidth of connections.
  - Determine if boot/install servers are needed and, if needed, where they are located.
- Implement key aspects of TCP/IP networking in the SP environment.
  - Create specific host names (both fully qualified and aliases), TCP/IP address, Netmask value and default routes.
  - Determine the mechanism (for example, /etc/hosts, NIS, DNS) by which they will be made available across the system.
  - Determine SP Ethernet topology (segmentation, routing).
  - Determine TCP/IP addressing for switch network.
- Determine the appropriate common, distributed and/or local files/file systems.
  - Determine the physical locations of the file system and home directories.

  - Determine the mechanism (for example, NFS, AFS, DRS, local) by which they will be made available across the system.
- Define a backup/recovery strategy for the SP which supports node image(s), control workstation images, applications, and data.
  - Determine backup strategy including node and CWS images.
  - Determine backup strategy and tools for application data.

Section 2 - Installation and configuration

- Configure an RS/6000 as an SP control workstation.
  - Verify the control workstation system configuration.
  - Configure TCP/IP network on the control workstation.
  - Install PSSP.
  - Configure the SDR with SP configuration information.
  - Verify control workstation software.
- Perform network installation of images on nodes, using any combination of boot/install servers.
  - Install the images on the nodes.
  - Define and configure boot/install servers.
  - Check SDR information.
  - Check RSCT daemons (hats, hags, haem).
- Thoroughly exercise the SP system resources to verify correct information of all required subsystems.
  - Verify all network connections.
  - Verify switch operations.
- Configure and administer the Kerberos Authentication subsystem and manage user IDs.
  - Plan and configure Kerberos functions and procedures.
  - Configure the Kerberos administration ID.
  - Understand and use the options of end-user management.
- Define and configure system partition and perform switch installation.

Section 3 - Application Enablement

- Determine whether additional SP-related products (for example, Loadleveler, PTPE, HACWS, NetTAPE, CLIOS) would be beneficial.
- Understand the function of additional SP-related products.

- Define and implement application-specific file systems, logical volumes, VGs and VSDs.
- Install and configure problem management tools (for example, event manager, problem manager, perspectives).
    - Define and manage monitors.

Section 4 - Ongoing support

- Perform software maintenance.
    - Perform system software recovery.
    - Upgrade and migrate system software (applying PTFs, migration).
- Perform SP reconfiguration.
    - Add frames.
    - Add nodes.
    - Migrate nodes.
    - Add/replace switch.
- Utilize Problem Determination methodologies (for example, HOSTRESPONDS,SWITCHRESPONDS, error report, log files, DAEMONS,GUIS).
    - Interpret error logs that are unique to the SP.
    - Diagnose networking problems.
    - Diagnose host response problems.
    - Diagnose switch-specific problems.
- Isolate cause of degraded SP performance and tune the system accordingly.
    - Understand Performance analysis and tuning requirements.

1.2 Certification education courses

Courses are offered to help you prepare for the certification tests. These courses are recommended, but not required, before taking a certification test. At the publication of this guide, the following courses are available. For a current list, please visit the following Web site:

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

AIX Version 4 System Administration	
Course Number	Q1114 (USA), AU14 (Worldwide)
Course Duration	Five days
Course Abstract	Learn the basic system administration skills to support AIX RS/6000 running the AIX Version 4 operating system. Build your skills in configuring and monitoring a single CPU environment. Administrators who manage systems in a networked environment should attend additional LAN courses.
Course Content	•Install the AIX Version 4 operating system, software bundles, and filesets •Perform a system startup and shutdown •Understand and use AIX system management tools •Configure ASCII terminals and printer devices •Manage physical and logical volumes •Perform file systems management •Create and manage user and group accounts •Use backup and restore commands •Use administrative subsystems, including cron, to schedule system tasks and security to implement customized access of files and directories

AIX Version 4.3 Advanced System Administration	
Course Number	Q1116 (USA), AU16 (Worldwide)
Course Duration	Five days
Course Abstract	Learn how to identify possible sources of problems on stand-alone configurations of the RS/6000 and perform advanced system administration tasks.
Course Content	•Identify the different RS/6000 models and architects •Explain the ODM purpose for device configuration •Interpret system initialization and problems during the boot process •Customize authentication and set up ACLs •Identify the TCB components, commands, and their use •Obtain a system dump and define saved data •Identify the error logging facility components and reports •List ways to invoke diagnostic programs •Customize a logical volume for optimal performance and availability •Manage a disk and the data under any circumstance •Use the standard AIX commands to identify potential I/O, disk, CPU, or other bottlenecks on the system •Customize SMIT menus and define how SMIT interacts with the ODM •Define the virtual printer database and potential problems •List the terminal attributes and create new terminfo entries •Define the NIM installation procedure

AIX Version 4 Configuring TCP/IP and Accessing the Internet	
Course Number	Q1107 (USA), AU07 or AU05 (Worldwide)
Course Duration	Five days
Course Abstract	•Learn how to perform TCP/IP network configuration and administration on AIX Version 4 RS/6000 systems. •Learn the skills necessary to begin implementing and using NFS, NIS, DNS, network printing, static and dynamic routing, SLIP and SLIPLOGIN, Xstations, and the Internet.
Course Contents	•Describe the basic concepts of TCP/IP protocols and addressing •Explain TCP/IP broadcasting and multicasting •Configure, implement, and support TCP/IP on an IBM RS/6000 system •Use networking commands for remote logon, remote execution, and file transfer •Configure SLIP and SLIPLOGIN •Use SMIT to configure network printing •Connect multiple TCP/IP networks using static and dynamic routing •Implement DNS, NFS, and NIS •Perform basic troubleshooting of network problems •Configure an Xstation in the AIX environment •Explain how to access Internet services •Understand and support TCP/IP •Plan implementation of NFS •Support LAN-attached printers •Support AIX networking •Determine network problems •Implement network file systems

1.3 Education on CD: IBM AIX Essentials

The new IBM AIX Essentials series offers a dynamic training experience for those who need convenient and cost-effective AIX education. The series consists of five new, content rich, computer-based multimedia training courses based on highly acclaimed, instructor-led AIX classes that have been successfully taught by IBM Education and Training for years.

To order, and for more information and answers to your questions:

- In the U.S., call 800-IBM-TEACH (426-8322) or use the online form at the following URL: `http://www-3.ibm.com/services/learning/aix/#order`
- Outside the U.S., contact your IBM Sales Representative or
- Contact an IBM Business Partner.

Chapter 2. Customer Relations

The following topics are discussed in this chapter:

- Problem description and definition
- Collecting information from the user
- Collecting information from the system

This chapter is intended for system support people which have to help and assist customers with a certain problem. The intention is to provide some methods for decribing a problem and collecting the necessary information about the problem, in order to make the right diagnostic of the problem.

2.1 Defining the problem

The first step in problem resolution is to define the problem. It is important that the person trying to solve the problem understands exactly what the users of the system perceive the problem to be. A clear definition of the problem is useful in two ways. First of all, it can give you a hint as to the cause of the problem. Secondly, it is much easier to demonstrate to the users that the problem has been solved if you know how the problem is seen from their point of view.

Take, for example, the situation where a user is unable to print a document. The problem may be due to the /var file system running out of space. The person solving the problem may fix this and demonstrate that the problem has been fixed by using the `df` command to show that the /var file system is no longer full.

This example can also be used to illustrate another difficulty with problem determination. Problems can be hidden by other problems. When you fix the most visible problem, another one may come to light. The problems that are unearthed during the problem determination process may be related to the one that was initially reported, in other words, multiple problems with the same symptoms. In some cases, you may discover problems that are completely unrelated to the one that was initially reported.

In the example described above, simply increasing the amount of free space in the /var file system may not solve the problem being experienced by the user. The printing problem may turn out to be a cable problem, a problem with the printer, or perhaps a failure of the lpd daemon. This is why understanding the problem from the users perspective is so important. In this example, a better way of proving that the problem has been resolved is to get the user to print their document.

2.2 Collecting information from the user

The best way of understanding the problem from the users' perspective is to ask them questions. From their perception of the situation, you can deduce if in fact they have a problem, and the timescale in which they expect it to be resolved. Their expectations may be beyond the scope of the machine or the application it is running.

The following questions should be asked when collecting information from the user during performing problem determination:

- What is the problem?

  Try to get the user to explain what the problem is and how it affects them. Depending on the situation and the nature of the problem, this question can be supplemented by either of the following two questions:

  - What is the system doing?
  - What is the system *not* doing?

  Once you have determined what the symptoms of the problem are, you should try to establish the history of the problem.

- How did you first notice the problem? Did you do anything different that made you notice the problem?
- When did it happen? Does it always happen at the same time, for example, when the same job or application is run?
- Does the same problem occur elsewhere? Is only one machine experiencing the problem or are multiple machines experiencing the same problem?
- Have any changes been made recently?

  This refers to any type of change made to the system, ranging from adding new hardware or software, to configuration changes to existing software.

- If a change has been made recently, were all of the prerequisites met before the change was made?

Software problems most often occur when changes have been made to the system, and either the prerequisites have not been met, for example, system firmware not at the minimum required level, or instructions have not been followed exactly in order, for example, the person following the instructions second guesses what the instructions are attempting to do and decides they know a quicker route. The second guess then means that, because the person has taken a perceived better route, prerequisites for subsequent steps may not have been met, and so, the problem develops into the situation you are confronted with.

Other changes, such as the addition of hardware, bring their own problems, such as cables incorrectly assembled, contacts bent, or addressing misconfigured.

The *How did you first notice the problem?* question may not help you directly, but it is very useful in getting the person to talk about the problem. Once they start talking, they invariably tell you things that will enable you to build a picture to help you to decide the starting point for problem resolution.

If the problem occurs on more than one machine, look for similarities and differences between the situations.

2.3 Collecting information about the system

The second step in problem determination is collecting information about the system. Some information will already have been obtained from the user during the process of defining the problem.

It is not only the user of the machine that can provide information on a problem. By using various commands, it is possible to determine how the machine is configured, the errors that are being produced, and the state of the operating system.

The use of commands, such as `lsdev`, `lspv`, `lsvg`, `lslpp`, `lsattr`, and others enable you to gather information on how the system is configured. Other commands, such as `errpt`, can give you an indication of any errors being logged by the system.

If the system administrator uses SMIT or Web-based System Manager to perform administrative tasks, examine the log files for these applications to look for recent configuration changes. The log files are normally contained in the home directory of the root user and by default are named /smit.log for SMIT and /websm.log for the Web-based System Manager.

If you are looking for something specific based on the problem described by the user, then often other files are viewed or extracted for sending to your IBM support function for analysis, such as system dumps or checkstop files.

Chapter 3. Errors booting

The following topics are discussed in this chapter:

- A general boot process
- Differences between MCA and PCI systems
- Boot phase 1
- Boot phase 2
- Boot phase 3
- Common boot problem scenarios and how to fix them

This chapter is familiar from the *Installation and System Recovery Study guide*, but because boot problems are among the most common problems, an overall discussion on the subject is useful. This chapter begins with a general overview of the boot process, then expands on the details and discusses the process with their LED codes in further detail. A summary of the LED codes can been found in , “LED codes” on page 64.

3.1 General overview of the boot process

Both hardware and software problems can cause the system to halt in the boot process. The boot process is also dependent on what hardware platform is used. In the initial startup phase there are some important differences between MCA and PCI systems, and these differences will determine the way to handle a hardware related boot problem. These differences will be covered in section 3.2, “BIST - POST” on page 43.

A general workflow of the boot process is shown in Figure 1 on page 42.

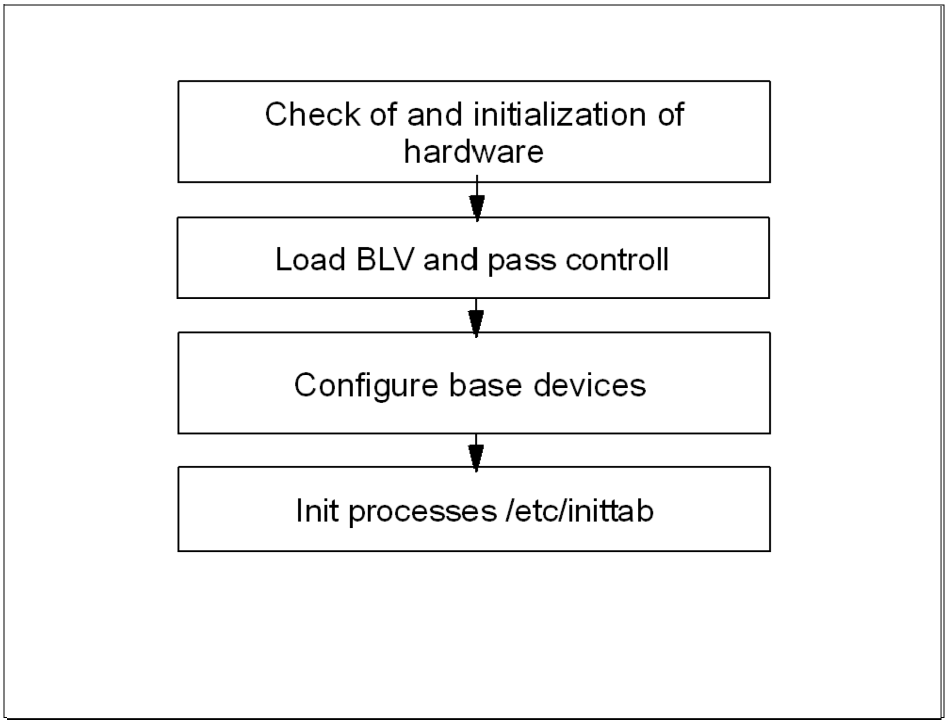

Check of and initialization of hardware
Load BLV and pass controll
Configure base devices
Init processes /etc/inittab

Figure 1. General boot order

The initial hardware check is to verify that the primary hardware is OK. This phase is divided into two separate phases on a MCA system, first built-in self test (BIST) and the second a power-on self test (POST). On PCI systems, it is handled by a single POST. After this, the system loads the boot logical volume (BLV) into a RAM file system (RAMFS) and passes control to the BLV.

Content of the BLV

AIX kernel

- The kernel is always loaded from the BLV. There is a copy of the kernel in /unix (soft link to /usr/lib/boot/ unix_mp or unix_up).

rc.boot

- This is the configuration script that will be called three times by the init process during boot.

Reduced ODM

- Device support is provided only to devices marked as base devices ODM

Boot commands

- For example cfgmgr, bootinfo.

Because the rootvg is not available at this point, all the information needed for boot has to be included in the BLV, used for creation of the RAMFS in the memory. After this, the init process is loaded and starts to configure the base devices. This is called boot phase 1 (init executes the rc.boot script with an argument of 1).

The next step, called phase 2, aims at activating rootvg, and this is probably the phase where the most common boot problems occur - for example, the file systems or the jfslog is corrupt. Next, the control is passed to the rootvg init command and the RAMFS is released.

Finally, the init process, now loaded from disk (not the BLV init) executes the rc.boot script with parameter of 3 to configure the remaining devices. This final stage is done from /etc/inittab. This is called phase 3.

3.2 BIST - POST

As mentioned before, there are differences between the classic RS/6000 system with MCA architecture and the PCI systems that are delivered today. The MCA system will be discussed first.

3.2.1 MCA systems

At a system startup of an MCA system, the first thing that happens is a BIST. These tests resides in EPROM chips, and the hardware tested by BIST are mainly components on the motherboard. After this the POST will be initialized. LED codes shown during this phase of the startup will be in the range of 100 - 195, defining a hardware problem.

The task of the POST is to find a successful hardware path to a BLV. All hardware that is required to load a boot image is tested. The LED codes at this stages are in the range of 200 - 2E7, and both hardware and software problems can cause a halt in the startup process at this stage.

On an MCA system, the load of the BLV starts with checking the bootlist. The bootlist is defined by the key position. When the key is in normal position applications will be started as well as network services. This is done when the init process reads /etc/inittab and executes the configuration scripts referenced in the file. A normal boot is represented by the runlevel 2. The etc/inittab file is discussed in further detail in sections 3.1, “General overview of the boot process” on page 41 and 3.5.1, “/etc/inittab” on page 59. To manipulate the boot list for normal mode, use the following command:

```
#bootlist -m normal hdisk0 hdisk1 rmt0 cd0
```

This will make the system to first search hdisk0 for a usable BLV. If there is no BLV at hdisk0 then hdisk1 will be searched, and so on.

The service boot list is used when booting the system for maintenance tasks. No applications or network services will be started. To check what the service bootlist looks like, use the -o option which was introduced with AIX Version 4.2, as follows.

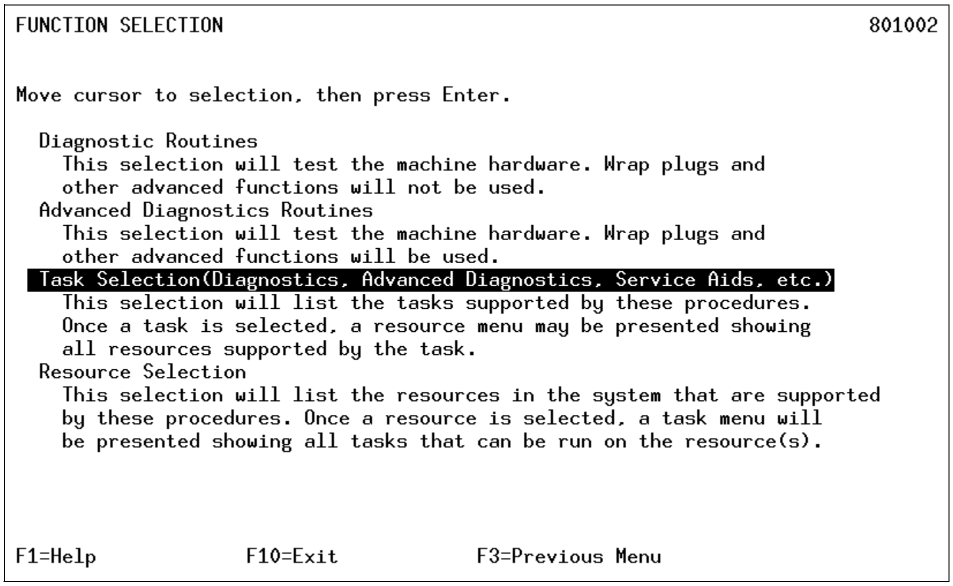
```
# bootlist -m service -o
fd0
cd0
rmt0
hdisk2
ent0
```

Another feature introduced with AIX Version 4.2, is the use of generic device names. Instead of pointing out the specified disk, with hdisk0 or hdisk1, you can use the generic definition of SCSI disks. For example.

```
#bootlist -m service cd rmt scdisk
```

This will cause the system to probe any CD, then probe any tape drive and finally probe any SCSI disk, for a BLV. The actual probing of the disk is a check of sector 0 for a boot record which in turn will point out the boot image.

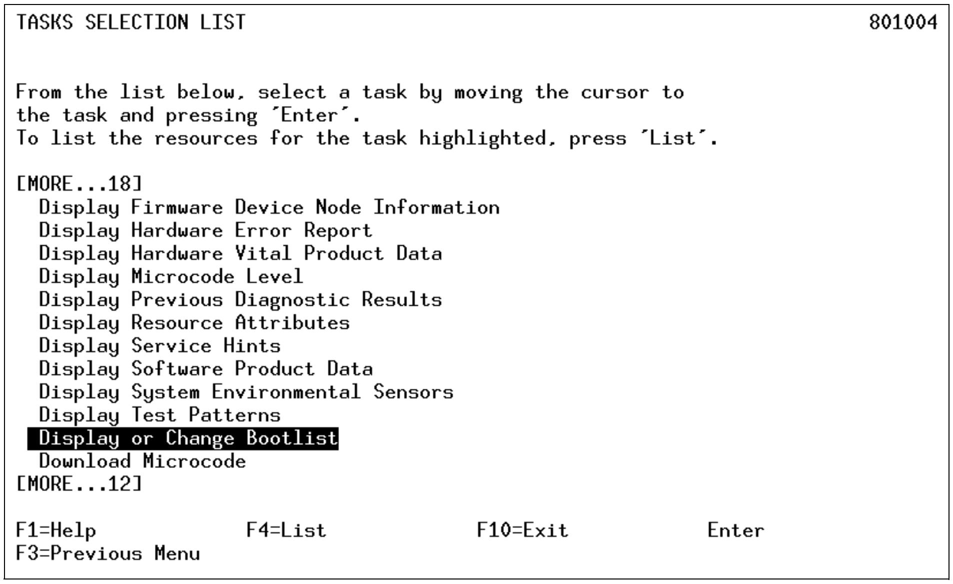
Changes to the boot list can also be done through the diag menus. At the Function Selection menu choose **Task Selections**, as shown in Figure 4.



```
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 2. Function selection menu in diag

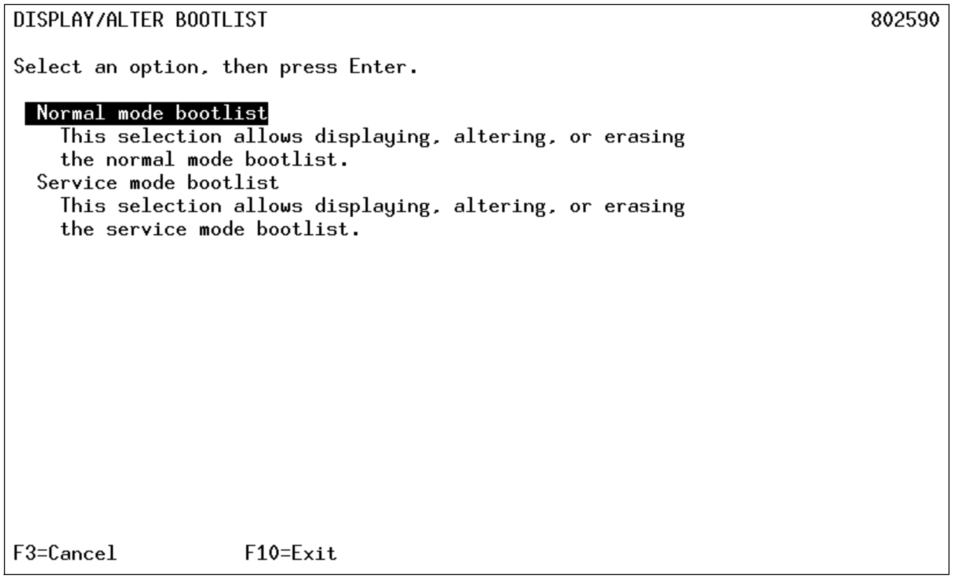
In the list of tasks, choose **Display or Change Bootlist**, as shown in Figure 3 on page 46:



```
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...18]
  Display Firmware Device Node Information
  Display Hardware Error Report
  Display Hardware Vital Product Data
  Display Microcode Level
  Display Previous Diagnostic Results
  Display Resource Attributes
  Display Service Hints
  Display Software Product Data
  Display System Environmental Sensors
  Display Test Patterns
  Display or Change Bootlist
  Download Microcode
[MORE...12]

F1=Help            F4=List            F10=Exit           Enter
F3=Previous Menu
```

Figure 3. Task selection menu in diag

Finally, you have to choose whether to change the **Normal mode bootlist** or the **Service mode bootlist**, as shown in Figure 4:



```
DISPLAY/ALTER BOOTLIST                                                 802590

Select an option, then press Enter.

  Normal mode bootlist
    This selection allows displaying, altering, or erasing
    the normal mode bootlist.
  Service mode bootlist
    This selection allows displaying, altering, or erasing
    the service mode bootlist.














F3=Cancel          F10=Exit
```

Figure 4. Display/alter bootlist menu in diag

At this point a lot of things can cause a boot problem. The boot list could point out a device that does not have a BLV, or the devices pointed out are not accessible because of hardware errors.

The following sections cover some problems that can cause a halt. All problems at this stage of the startup process have an error code defined which is shown in the LED display on the front panel.

3.2.1.1 LED 200

The LED code 200 is connected to the secure key position. When the key is in the secure position - the boot will stop until the key is turned, either to the normal position or the service position, then the boot will continue.

3.2.1.2 LED 299

An LED code of 299 shows that the BLV will be loaded. If this LED code is passed, then the load has been successful. If you, after passing 299, get a stable 201 then you have to recreate the BLV as discussed in section 3.2.1.4, “How-to recreate the BLV” on page 47.

3.2.1.3 MCA LED codes

Table 1 provides a list of the most common LED codes on MCA systems. More of these can be found in the *Message Guide and References* which is part of the AIX version 4 Base Documentation.

Table 1. Common MCA LED codes

LED	Description
100 - 195	Hardware problem during BIST
200	Key mode switch in secure position
201	1. If LED 299 passed recreate BLV 2. If LED 299 has not passed, POST encountered a hardware error

221,
721,
221 - 229,
223 - 229,
225 - 229,
233 - 235
	bootlist in NVRAM is incorrect, or
	`(boot from media and change the bootlist)`
	bootlist device has no bootimage, or
	`(boot from media and recreate the BLV)`
	bootlist device is unavailable
	`(Check for hardware errors)`

3.2.1.4 How-to recreate the BLV

When the LED code indicates that the BLV cannot be loaded, you should start by checking for hardware problems, for example cable connections. The next step is to start the system in maintenance mode from an external media, for example an AIX installation CD. Use the Access this Volume Group and start a shell menu for recreation of BLV (this menu is also used if the boot problem was due to an incorrect bootlist). Execute the following command if you want to recreate the BLV on hdisk0:

```
#bosboot -ad /dev/hdisk0
```

Another scenario when you might want to create a BLV with the `bosboot` command is with mirrored rootvg. Just mirroring this volume group does not make the disks containing the secondary copy bootable. You still have to define the disks in the bootlist and execute the `bosboot` command on the secondary copy.

Accessing rootvg

The following is a short summary on how to access the maintenance menus. For more detailed information see *Installation Guide, Chapter 10 - Accessing a system that will not boot*, SC23-4112-02

1. Boot the system from media
2. At the installation menu - choose **Start Maintenance for System Recovery**
3. On the next menu - choose **Access a Root Volume Group**
4. A list of accessible disks are shown - choose the rootvg disk
5. Finally choose the **Access this Volume Group and start a shell** when you want to recreate the BLV; change the bootlist or forgotten root password.

   Choose the **Access this Volume Group and start a shell before mounting file systems** if file systems or the jfslog in rootvg are corrupt

3.2.2 PCI systems

When booting PCI systems there are some important differences from the MCA systems. It is already mentioned that there is an absence of BIST. Another difference is the absence of the key switch. Modern PCI systems uses a logical keymode switch, which is handled by the use of function key. And third, the diag function is missing on some older PCI systems. The following section discusses how to change the bootlist, and the support of the normal and service boot options on PCI systems.

3.2.2.1 Changing the bootlist on PCI systems

All PCI systems have System Management Services (SMS) menus. On most systems these menus can be accessed by pressing function key **1** (**F1**) or **1**, when the console is initiated (the use of **1** or **F1** depends on the use of graphical display or ASCII terminal). At this time a double beep is heard. Depending on the PCI model, there are three or four choices in the SMS main menu. One of these is named boot. Under this menu you can define the bootlist. The SMS main menu from a 43P-140 is shown in Figure 5. Newer PCI systems also have an additional selection called multiboot.

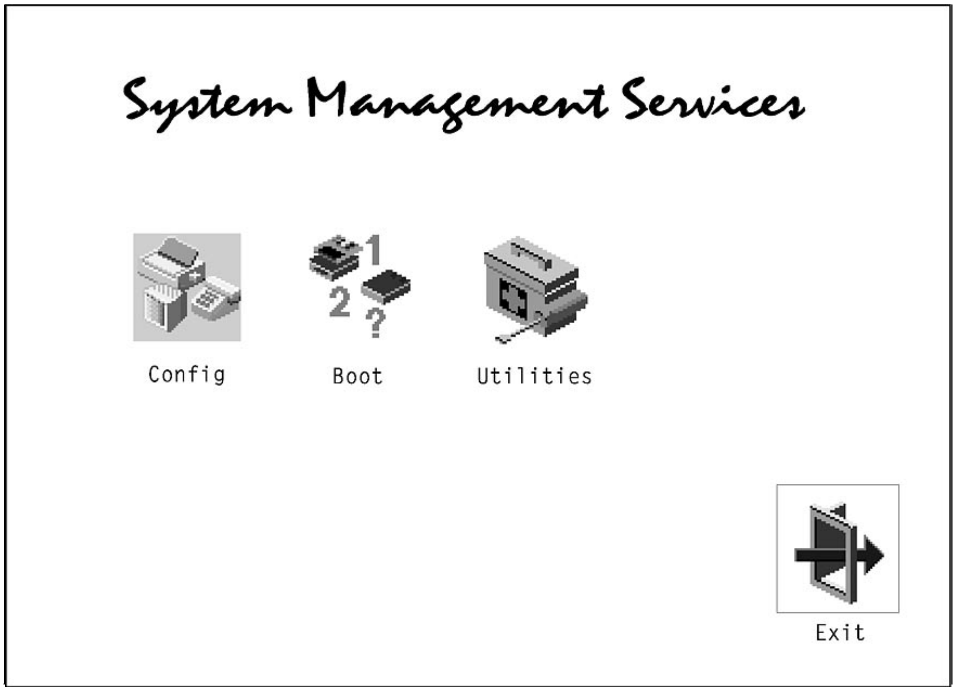


Figure 5. SMS main menu

Changing the boot order can also be done with the bootlist command.

3.2.2.2 Normal boot and Service boot on PCI systems

Some PCI systems do not support service mode, for example the 7048-43P. The only way to boot in another mode, such as maintenance mode, is to change the normal bootlist. This can be done with the `bootlist -m normal` command if the system accessible. If the system is not accessible, this can be done by booting from media and changing the bootlist through the SMS menus.

All PCI systems have a default bootlist. On modern PCI systems this default bootlist can be accessed (and if `diag` is available on the media it is started), by using the **F5** function key. This is a good option to use when booting the system in single user mode for accessing standalone `diag` functions. On older PCI systems this cannot be done. Instead a single bootlist provided, can be reset to the default values by removing the battery for about 30 seconds. This is because the bootlist is stored in NVRAM and the NVRAM is only non-volatile as long as the battery is maintaining the memory.

Newer PCI architecture machines, for example the 43P-150, supports a service bootlist. The simplest way to find out if a particular system supports the service boot option is to execute:

```
# bootlist -m service -o
0514-220 bootlist: Invalid mode (service) for this model
```

When receiving the error message above the system does not support the service boot option.

All new PCI systems supports the following key allocation standard:

PCI key allocation standard

- F1 or 1 on ASCII terminal: Starts System Management Services
- F5 or 5 on ASCII terminal: Boot diag, use default boot list of fd, cd, scdisk, network adapter
- F6 or 6 on ASCII terminal: Boot diag, use of custom service boot list

3.2.2.3 POST LED codes on PCI systems

On old PCI systems like the 40P or the 7248-43P the LED display is missing, so there will be no LED codes helping solving boot problems. Fortunately this has been changed on modern PCI systems, but the error codes generated during this phase of the system startup differs from model to model. The only way to figure out the exact meaning of an error code is to refer to the *Service Guide* delivered with the system. IBM provides a Web page where *Service Guides* for most PCI systems are available in HTML and PDF format. the URL is:

```
http://www.rs6000.ibm.com/resource/hardware_docs/
```

3.3 Boot phase 1

So far the system has tested the hardware, found a BLV, created the RAMFS and started the init process from BLV. The rootvg has not yet been activated. From now on, the boot sequence is the same on both MCA systems and PCI systems.

The workflow for boot phase 1 is shown in Figure 6.

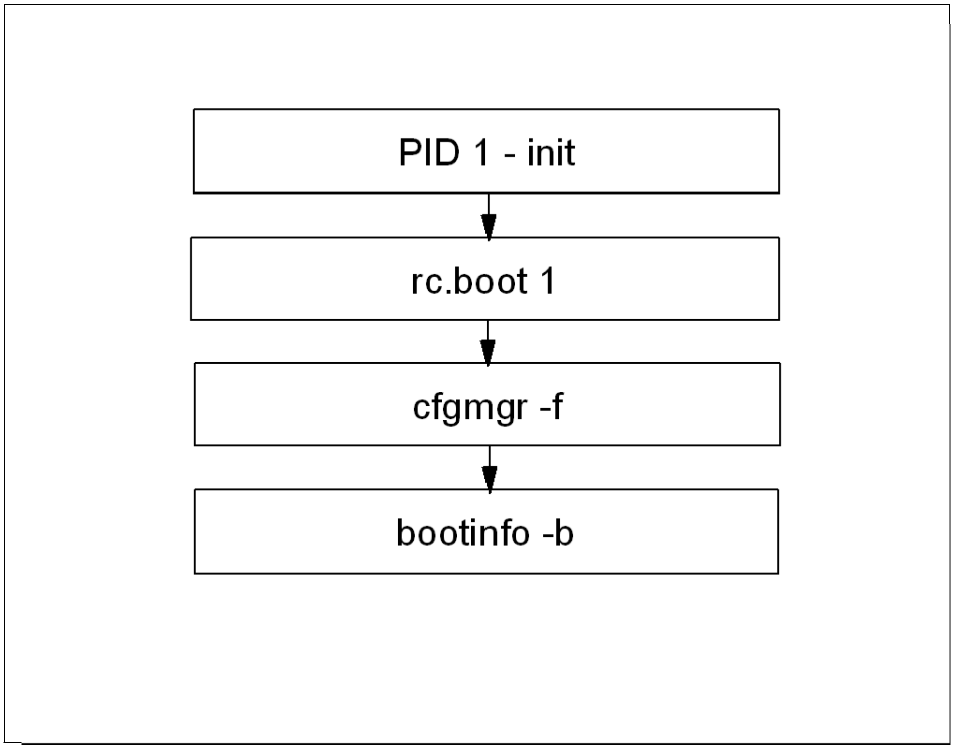


Figure 6. Boot phase 1

The init process started from RAMFS executes the boot script rc.boot 1. At this stage the `restbase` command is called to copy the reduced ODM from the BLV into the RAMFS. If this operation fails you will see a LED code of 548.

After this `cfgmgr -f` reads the Config_Rules class from the reduced ODM. In this class devices with the attribute phase=1 will be considered base devices. Base devices are all devices that are necessary to access rootvg. The process invoked with rc.boot 1 attempts to configure devices so that rootvg can be activated in the next rc.boot phase.

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.

3.4 Boot phase 2

In boot phase 2 the rc.boot script is passed the parameter 2. The first part of this phase is shown in Figure 7.

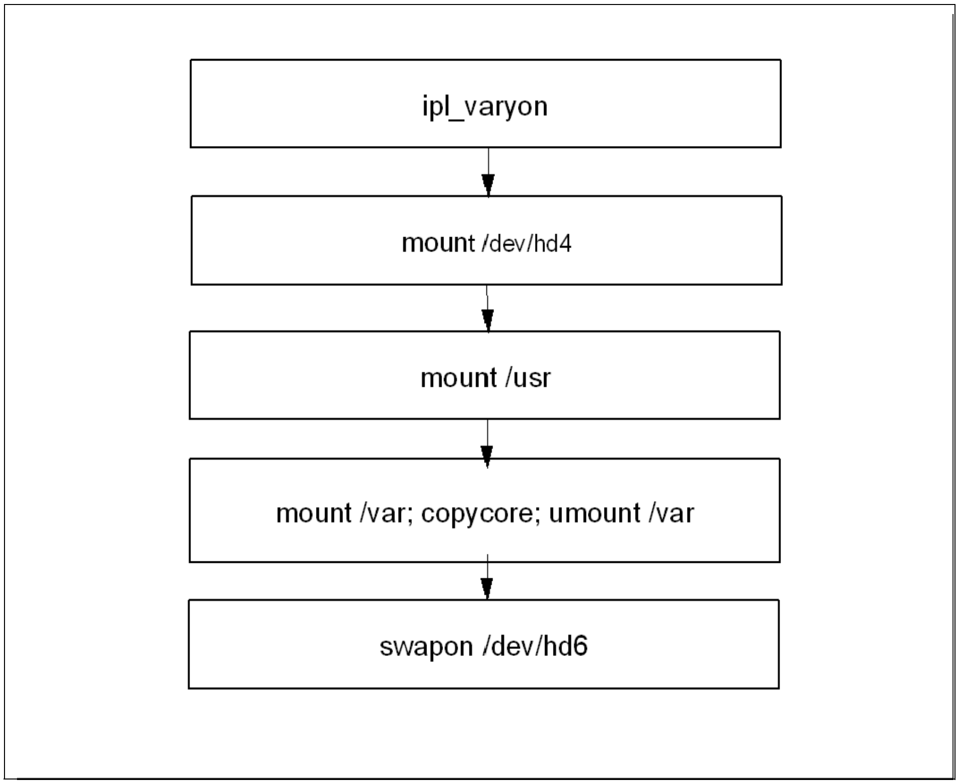


Figure 7. Boot phase 2 figure one

- During this phase rootvg will be varied on with the special ilp_varyon utility. If this command is not successful one of the following LED codes will appear - 552, 554, 556.
- After the successful execution of ipl_varyon, the root file system (/dev/hd4) will be mounted on a temporary mount point (/mnt) in RAMFS. If this fails, 555 or 557 will appear in the LED display.
- Next, the /usr and /var file system be mounted. If this fails the LED 518 appears. The mounting of /var at this point enables the system to copy an

eventual dump from default dump devices, /dev/hd6, to default copy directory, /var/adm/ras.

- After this rootvg's primary paging space, /dev/hd6, will be activated.

The second part of this phase is shown in Figure 8.

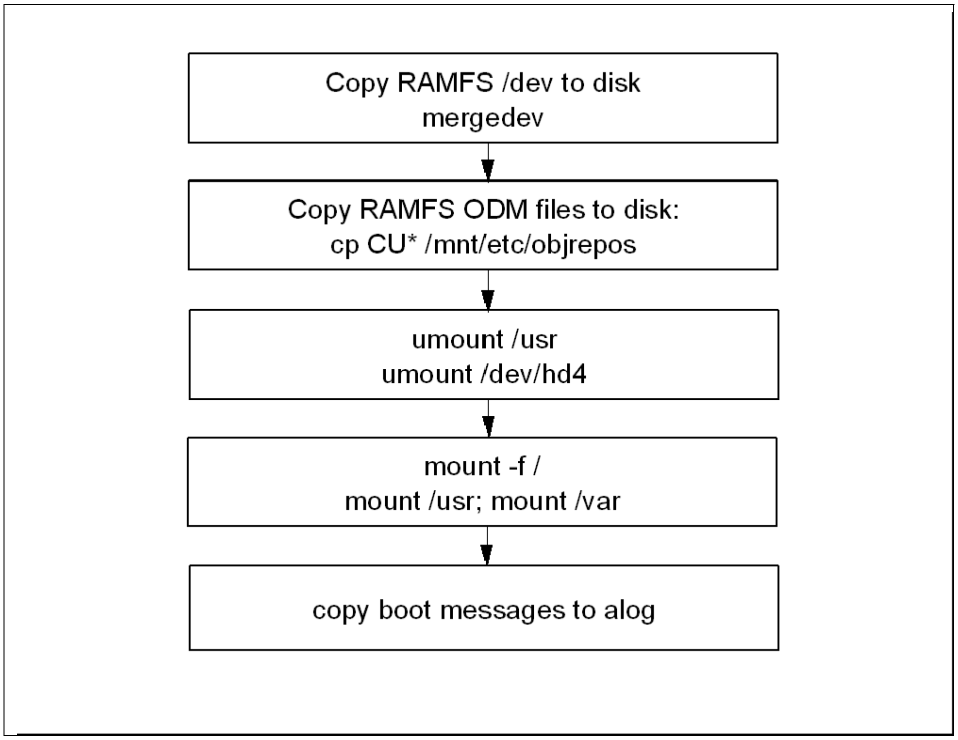


Figure 8. Boot phase 2 figure two

- Next the synchronization of rootvg's and RAMFS' ODM and /dev directories will occur (`mergedev`). This is possible because of the temporary mount point /mnt is used for the mounted root file system.
- Next the /usr and /var from the RAMFS is unmounted
- Finally the root file system from rootvg (disk) is mounted over the root file system from the RAMFS. Now the mount points for the rootvg file systems are available, so now can the /var and /usr file systems from the rootvg be mounted again on their ordinary mount points.
- There is no console available at this stage, so all boot messages will be copied to alog.

As mentioned, there are a lot of different problem possibilities in this phase of the boot. The following section discuss how to correct some of them.

3.4.1 LED 551, 555, or 557

There can be several reason to a system halt with LED codes - 551, 555 or 557. For example:

- A damaged file system
- A damaged Journaled-file-system (JFS) log device
- A bad disk in the machine that is a member of the rootvg

To diagnose and fix these problems you will need to boot from a bootable media, access the maintenance menus, choose **Access a Volume Group and start a shell before mounting file systems**, and then do one or all of these actions.

To ensure file system integrity run `fsck` to fix any file systems that may be corrupted:

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

To ensure the correct function of the log device, run `logform` on /dev/hd8 to recreate the logdevice:

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

or if the BLV is corrupted, recreate the BLV and update the bootlist:

```
bosboot -a -d /dev/hdisk0
bootlist -m normal hdisk0
```

3.4.2 LED 552, 554, 556

An LED code of 552, 554, or 556 during a standard disk based boot indicates a failure occurred during the varyon of the rootvg volume group. This can be caused of:

- A damaged file system
- A damaged Journaled File System (JFS) log device
- A bad IPL-device record or bad IPL-device magic number (The magic number indicates the device type)

- A damaged copy of the Object Data Manager (ODM) database on the boot logical volume
- A hard disk in the inactive state in the root volume group
- A damaged superblock

To diagnose and fix the problem, you will need to boot from media and use the menu access the volume group and start a shell before mounting file systems.

If `fsck` indicates that block 8 could not be read when used as shown in section 3.4.1, “LED 551, 555, or 557” on page 54, the file system is probably unrecoverable. The easiest way to fix an unrecoverable file system is to recreate it. This involves deleting it from the system and restoring it from a backup. Note that /dev/hd4 cannot be recreated. If /dev/hd4 is unrecoverable, you must reinstall AIX.

A corrupted ODM in the BLV is also a possible cause to these LED codes. To create a correct one, run the following commands that remove the system's configuration and save it to a backup directory:

```
mount /dev/hd4 /mnt
mount /dev/hd2 /usr
mkdir /mnt/etc/objrepos/bak
cp /mnt/etc/objrepos/Cu* /mnt/etc/objrepos/bak
cp /etc/objrepos/Cu* /mnt/etc/objrepos
/etc/umount all
exit
```

After this you must copy this new version of the ODM in the RAMFS to the BLV. This is done with the `savebase` command. Before that make sure you place it on the disk used for normal boot, by executing:

```
lslv -m hd5
```

Save the clean ODM database to the boot logical volume. For example:

```
savebase -d /dev/hdisk0
```

Finally recreate the BLV, and reboot the system. For example:

```
bosboot -ad /dev/hdisk0
shutdown -Fr
```

Another possible reason to these error codes is a corrupted superblock. If you boot in maintenance mode and get error messages such as Not an AIX file system or Not a recognized file system type, it is probably due to a corrupted superblock in the file system.

Each file system has two super blocks, one in logical block 1 and a copy in logical block 31. To copy the superblock from block 31 to block 1 for the roof file system, issue the following command:

```
#dd count=1 bs=4k skip=31 seek=1 if=/dev/hd4 of/dev/hd4
```

3.4.3 LED 518

The 518 LED code has an unclear definition in the *Messages Guide and Reference*, which says:

Display Value 518

Remote mount of the / (root) and /usr file systems during network boot did not complete successfully.

This is not the entire problem. If the system runs into problems while mounting the /usr from disk (locally, not network mount) you will get the same error. Fix this problem the same way as any other rootvg file system corruption is fixed.

3.4.4 The alog command

Up until this stage the system has not yet configured the console, so there is no stdout defined for the boot processes. At this stage the alog comes to good use.

The alog can maintain and manages logs. All boot information is sent through the alog. To look at the boot messages, use the following command:

```
#alog -ot boot
****************** no stderr ***********
----------------
Time: 12        LEDS: 0x538
invoking top level program -- "/usr/lib/methods/definet > /dev/null
2>&1;opt=`/u
sr/sbin/lsattr -E -l inet0 -a bootup_option -F value`
        if [ $opt = "no" ];then nf=/etc/rc.net
        else nf=/etc/rc.bsdnet
        fi;$nf -2;x=$?;test $x -ne 0&&echo $nf failed. Check for invalid
command
s >&2;exit $x"
Time: 21        LEDS: 0x539
return code = 0
```

```
****************** no stdout ***********
```

At this point, the bootup_option checked to determine if a BSD style configuration of TCP/IP services is to be used, or if the default of ODM supported configuration should be used. During this stage is the LED codes 538 and 539 shown, as shown in the preceding example.

3.5 Boot phase 3

So far the following boot tasks has been accomplished:

- Hardware configuration performed during BIST and / or POST
- The load of BLV
- Phase 1 where base devices are configured to prepare the system for activating the rootvg
- Phase 2 where rootvg is activated

Finally phase 3, initiated by the init process loaded from rootvg, is shown in Figure 9 on page 58.

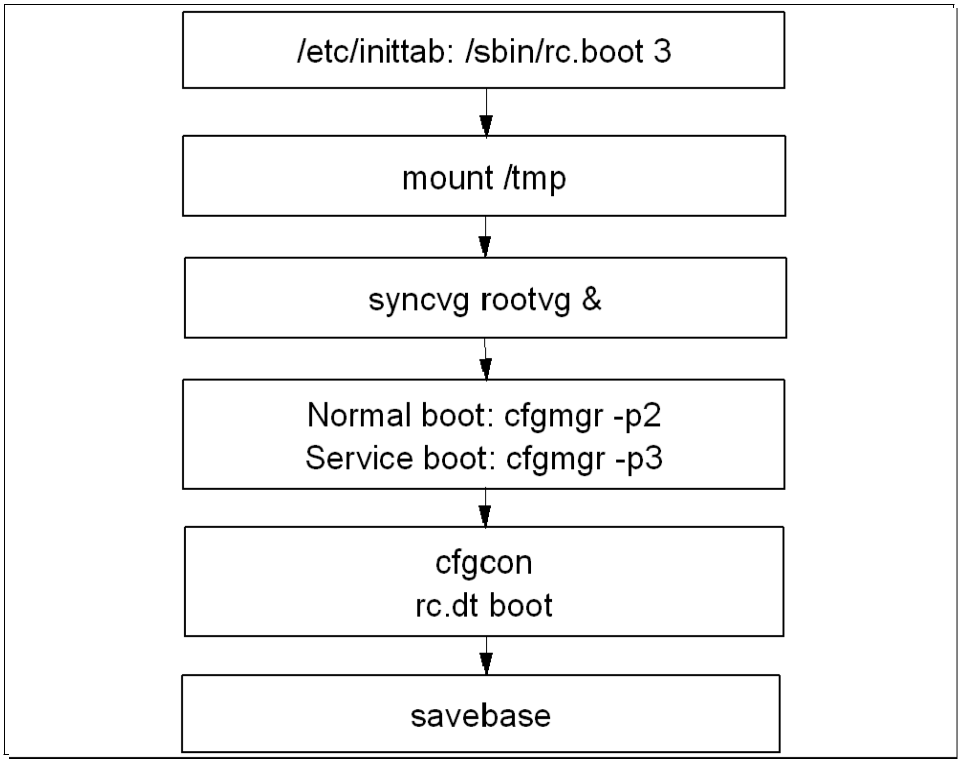


Figure 9. Boot phase 3

The order of boot phase 3 is as follows:

- Phase 3 is started in /etc/inittab
- First it will mount /tmp
- After this the rootvg will be synchronized. This can take some time, that is why the `syncvg rootvg` is executed as a background process. At this stage the LED code 553 is shown
- At this stage is also the `cfgmgr -p2` for normal boot and the `cfgmgr -p3` for service mode executed. `cfgmgr` reads the Config_rules file from ODM and checks for devices with phase=2 or phase=3
- Next the console will be configured. LED codes shown when configuring the console is shown in “cfgcon LED codes” on page 59. After the configuration of the console, boot messages will be sent to the console if no STDOUT redirection is made. Many of these boot messages will scroll

past at a fast phase, so there is not always time to read all messages. Therefore all missed messages can be found in /var/adm/ras/conslog.

- And finally the synchronization of the ODM in the BLV with the ODM from the / (root) file system is down by the `savebase` command

When `cfgcons` is called, different LED codes are shown depending on which device is configured

.

cfgcon LED codes

c31: Console not yet configured. Provides instructions to select console.

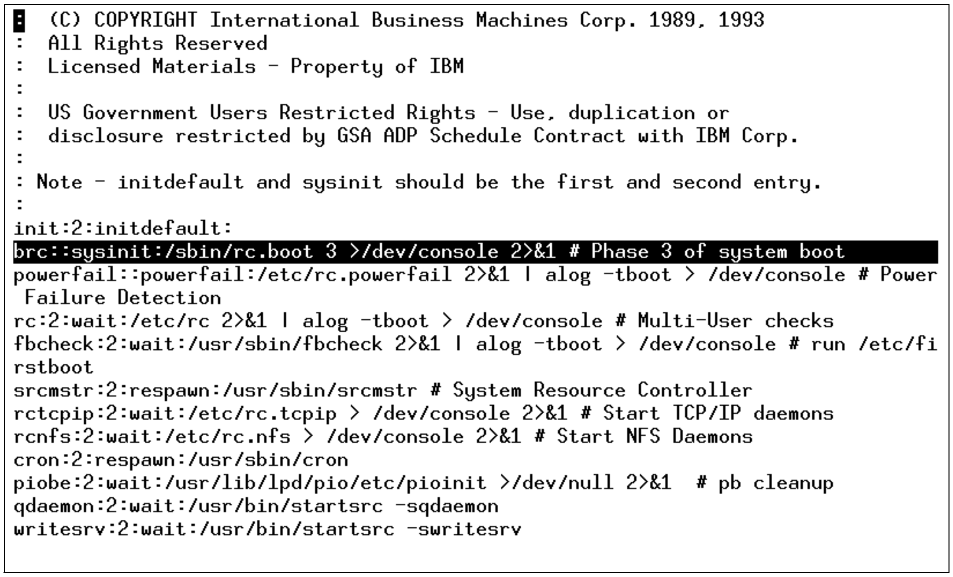
c32: Console is a lft terminal

c33: Console is a tty

c34: Console is a file on the disk

3.5.1 /etc/inittab

The /etc/inittab file supplies configuration scripts to the init process. In Figure 10 the highlighted line is the rc.boot with parameter 3 executed.



```
:  (C) COPYRIGHT International Business Machines Corp. 1989, 1993
:  All Rights Reserved
:  Licensed Materials - Property of IBM
:
:  US Government Users Restricted Rights - Use, duplication or
:  disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
:
: Note - initdefault and sysinit should be the first and second entry.
:
init:2:initdefault:
brc::sysinit:/sbin/rc.boot 3 >/dev/console 2>&1 # Phase 3 of system boot
powerfail::powerfail:/etc/rc.powerfail 2>&1 | alog -tboot > /dev/console # Power
 Failure Detection
rc:2:wait:/etc/rc 2>&1 | alog -tboot > /dev/console # Multi-User checks
fbcheck:2:wait:/usr/sbin/fbcheck 2>&1 | alog -tboot > /dev/console # run /etc/fi
rstboot
srcmstr:2:respawn:/usr/sbin/srcmstr # System Resource Controller
rctcpip:2:wait:/etc/rc.tcpip > /dev/console 2>&1 # Start TCP/IP daemons
rcnfs:2:wait:/etc/rc.nfs > /dev/console 2>&1 # Start NFS Daemons
cron:2:respawn:/usr/sbin/cron
piobe:2:wait:/usr/lib/lpd/pio/etc/pioinit >/dev/null 2>&1  # pb cleanup
qdaemon:2:wait:/usr/bin/startsrc -sqdaemon
writesrv:2:wait:/usr/bin/startsrc -swritesrv
```

Figure 10. Example of rc.boot 3 in /etc/inittab

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

```
Identifier:RunLevel:Action:Commnad
```

The first line in /etc/inittab (initdefault) defines what runlevel is to be considered as a default runlevel. In this example, the runlevel is 2, which means a normal multi-user boot. In the case of a a multi-user boot all lines with the runlevel 2 will be executed from the /etc/inittab. If this line is missing, you are prompted at boot to define the runlevel.

The rc.boot line is to be executed on all run levels: (this equals runlevel 0123456789). The action defined, sysinit, has to finish, before continuing with the next line in /etc/inittab. From rc.boot 3 is, among a lot of other things, the rootvg synchronized, the mirrored, is started and the /tmp directory mounted. A detailed description of /etc/inittab is provided in *IBM Certification Study Guide AIX V4.3 System Support*, SG24-5129.

3.5.2 LED 553

As mentioned previously, an LED code of 553 is caused when the /etc/inittab cannot be read. To recover from an LED 553, check /dev/hd3 and /dev/hd4 for space problems and erase files if necessary. Check the /etc/inittab file for corruption and fix it if necessary. Examples of syntax errors in /etc/inittab seen at the support centers are incorrectly defined entries in the file. When editing /etc/inittab, the inittab commands should be issued. For example:

- `mkitab`
- `chitab`

It is helpful to remember that /etc/inittab is very sensitive to even the most trivial syntax error. A misplaced dot can halt the system boot.

3.5.3 LED c31

LED c31 is not really an error code, but the system is waiting for input from you on the keyboard. This is usually encountered when booting from CD or mksysb tape. This is normally the dialog to select the system console.

3.5.4 LED 581

This LED code is not really an error code either. LED 581 is shown during the time that the configuration manager configures TCP/IP and runs /etc/rc.net to do specific adapter, interface, and host name configuration.

This problem is when this system hangs while executing /etc/rc.net. Then the problem can be either a system or a network problem that happens because TCP/IP waits for replies over some interfaces. If there are no replies, it eventually times out on the attempt and marks the interface as down. This time-out period varies and can range from around three minutes to an indefinite period.

The following problem determination procedure is used to verify that the methods and procedures run by /etc/rc.net are causing the LED 581 hang:

1. Boot the machine in Service mode.
2. Move the /etc/rc.net file:

   ```
   mv /etc/rc.net /etc/rc.net.save
   ```

3. Reboot in Normal mode boot to see if the system continues past the LED 581 and allows you to log in.

Note

The above steps assume that neither DNS or NIS is configured.

If you determine that the procedures in /etc/rc.net are causing the hang, that is, the system continued past LED 581 when you performed the steps above, the problem may be one of the following:

- Ethernet or token-ring hardware problems

  Run diagnostics and check the error log.

- Missing or incorrect default route
- Networks not accessible

  Check that gateways, name servers, and NIS masters are up and available.

- Bad IP addresses or masks

  Use the `iptrace` and `ipreport` commands for problem determination.

- Corrupt ODM

  Remove and recreate network devices.

- Premature name or IP address resolution

  Either named, ypbind/ypserv, or /etc/hosts may need correction.

- Extra spaces at the ends of lines in configuration files

Use the vi editor with the `set list` subcommand to check files, such as the /etc/filesystems file, for this.

- Bad LPPs

  Reinstall the LPP.

A specific LED 581 hang case occurs when ATMLE is being used with a DNS. If you are experiencing this problem, you can either work around the problem by adding a `host=local,bind` entry to /etc/netsvc.conf file or by adding the following lines to the /etc/rc.net file as follows:

```
###################################################################
# Part III - Miscellaneous Commands.
###################################################################
# Set the hostid and uname to `hostname`, where hostname has been
# set via ODM in Part I, or directly in Part II.
# (Note it is not required that hostname, hostid and uname all be
# the same).
export NSORDER="local"            <<============NEW LINE ADDED HERE
/usr/sbin/hostid `hostname`             >>$LOGFILE 2>&1
/bin/uname -S`hostname|sed 's/\..*$//'` >>$LOGFILE 2>&1
unset NSORDER                     <<============NEW LINE ADDED HERE

###################################################################
```

3.6 Boot related information in the error log

Because the functionality of the error log should be familiar to you, this section will only cover boot related messages.

The error log facility provides good historical information on when the system has been rebooted, and often also out of what reason. One way to find the reboot timestamp is simply to check for when error logging has been turned on, as shown in the following example:

```
# errpt
IDENTIFIER TIMESTAMP  T C RESOURCE_NAME  DESCRIPTION
499B30CC   0711125600 T H ent1           ETHERNET DOWN
1104AA28   0711125200 T S SYSPROC        SYSTEM RESET INTERRUPT RECEIVED
9DBCFDEE   0711125500 T O errdemon       ERROR LOGGING TURNED ON
499B30CC   0707114100 T H ent1           ETHERNET DOWN
499B30CC   0707113700 T H ent1           ETHERNET DOWN
C60BB505   0705101400 P S SYSPROC        SW PROGRAM ABNORMALLY TERMINATED
35BFC499   0705101100 P H cd0            DISK OPERATION ERROR
0BA49C99   0705101100 T H scsi0          SCSI BUS ERROR
9DBCFDEE   0704153700 T O errdemon       ERROR LOGGING TURNED ON
192AC071   0704153700 T O errdemon       ERROR LOGGING TURNED OFF
9DBCFDEE   0704152600 T O errdemon       ERROR LOGGING TURNED ON
```

Every time the system is booted, the error log facility is by default started. In the previous example the system has been gracefully shutdown two times on the 4th of July. When the system is gracefully shutdown, as it happened on the 4th of July, the error logging facility is also shutdown, as the error log entry `192AC071` shows. In the case of the reboot at 11th of July, there is no stopping of the error log facility reported, in other words that shutdown cannot be considered graceful. Three minutes before the reboot (12:55) a system reset is reported (the line above with the 12:52 timestamp). The reason to the non-graceful reboot is often reported sequentially later than the reboot, but by checking the timestamp the right relationship is revealed. When looking at the detailed report of the reason to the reboot, the reason, use of the reset button, is shown:

```
# errpt -aj 1104AA28
--------------------------------------------------------------------------
-
LABEL:          SYS_RESET
IDENTIFIER:     1104AA28

Date/Time:       Tue Jul 11 12:52:54
Sequence Number: 12
Machine Id:      000BC6DD4C00
Node Id:         server3
Class:           S
Type:            TEMP
Resource Name:   SYSPROC

Description
SYSTEM RESET INTERRUPT RECEIVED

Probable Causes
SYSTEM RESET INTERRUPT

Detail Data
KEY MODE SWITCH POSITION AT BOOT TIME
normal
KEY MODE SWITCH POSITION CURRENTLY
normal
```

3.7 Summary

In the next sections are short summaries of the boot phases and some common LED codes.

Boot phases

BIST and POST is used to test hardware and to find a successful hardware path to a BLV.

Boot phase 1 (init rc.boot 1) is used to configure base devices.

Boot phase 2 (init rc.boot 2) is used to activate the rootvg.

Boot phase 3(init /sbin/rc.boot 3) is used to configure the rest of devices.

LED codes

LED codes during POST on a MCA system are listed in Table 2 on page 64

Table 2. MCA POST LED

LED	Reason / Action
100 - 195	Hardware problem during BIST
200	Key mode switch in secure position
201	1. If LED 299 passed recreate BLV 2. If LED 299 has not passed, POST encountered a hardware error

221
721
221 - 229
223 - 229
225 - 229
233 - 235
	Bootlist in NVRAM is incorrect, or
	`(boot from media and change the bootlist)`
	Bootlist device has no bootimage, or
	`(boot from media and recreate the BLV)`
	Bootlist device is unavailable
	`(Check for hardware errors)`

LED codes shown during boot phase 2 is shown in Table 3

Table 3. Boot phase 2 LED codes

LED	Reason / Action

551
555
557
	1. Corrupted file system
	`(fsck -y <device>)`
	2. Corrupted jfslog
	`(/usr/sbin/logform /dev/hd8)`
	3. Corrupted BLV -
	`(bosboot -ad <device>)`

552
554
556
	The ipl_varyon failed. Except for the reason mentioned above (551, 555, or 557):
	1. Corrupted ODM
	`(backup ODM, recreate with savebase)`
	2. Superblock dirty
	`(Copy in superblock from block 31)`

LED	Reason / Action

518	/usr cannot be mounted
	1. If /usr should be mounted over the network
	`(check for network problem)`
	2. If /usr is to be mounted locally
	`(fix the file system)`

LED codes shown during boot phase 3 is shown in Table 4

Table 4. Boot phase 3 LED codes

LED	Reason / Action
553	Syntax error in /etc/inittab
c31	Define the console

errpt

The errpt command is used to check for errors reported by the error log facility

The syntax of the `errpt` command is:

```
To Process a Report from the Error Log

errpt [ -a ] [ -A ] [ -c ] [ -d ErrorClassList ] [ -D ] [ -e EndDate ] [ -g
] [ -i File ] [ -I
File ] [ -j ErrorID [ ,ErrorID ] ] | [ -k ErrorID [ ,ErrorID ] ] [ -J
ErrorLabel
[ ,ErrorLabel ] ] | [ -K ErrorLabel [ ,ErrorLabel ] ] [ -l SequenceNumber ]
[ -m
Machine ] [ -n Node ] [ -s StartDate ] [ -F FlagList ] [ -N ResourceNameList
] [ -P ]
[ -R ResourceTypeList ] [ -S ResourceClassList ] [ -T ErrorTypeList ] [ -y
File ] [ -z
File ]

To Process a Report from the Error Record Template Repository

errpt [ -a ] [ -A ] [ -I File ] [ -t ] [ -d ErrorClassList ] [ -j ErrorID [
,ErrorID ] ] | [ -k
ErrorID [ ,ErrorID ] ] [ -J ErrorLabel [ ,ErrorLabel ] ] |
[ -K ErrorLabel [ ,ErrorLabel ] ] [ -F FlagList ] [ -P ] [ -T ErrorTypeList
] [ -y File ]
[ -z File ]
```

Some useful `errpt` flags:

Table 5. Some useful errpt flags

Flags	Description
-a	Detailed output
-j error identifier	Includes only the error-log entries specified by the ErrorID (error identifier) variable
-s StartDate	Specifies all records posted on and after the StartDate variable
-T ErrorTypeList	Limits the error report to error types specified by the valid ErrorTypeList variables: INFO, PEND, PERF, PERM, TEMP, and UNKN

w

Prints a summary of current system activity.

The syntax of the `w` command is:

```
w [ -h ] [ -u ] [ -w ] [ -l | -s ] [ User ]
```

Some useful `w` falgs

Table 6. Some useful w flags

Flags	Description
-u	Prints the time of day, amount of time since last system startup, number of users logged on, and number of processes running. Same output as the `uptime` command

3.8 Quiz

3.8.1 Answers

3.9 Exercises

Do not perform these exercises on an existing file system or on a production system.

1. Create a file system for this exercise, and copy in some files to the file system. Then destroy the first super block. This can be done by copying 4 KB from /dev/zero to block one on your logical volume, for example:

```
dd count=1 bs=4k seek=1 if=/dev/zero of=/dev/thomasclv
```

Try to mount the file system and run `fsck` on the file system to determine the problem. Finally fix the problem as described in this chapter.

2. Still on your test system, with verified mksysb at hand: make a backup of /etc/inittab. remove the first line (first uncommented line, that is) and try to reboot. You are at reboot prompted for what?

   After the boot has finished, edit the /etc/inittab and change a dot to a comma or a colon to semicolon on a line with action=wait. What happens? What is the LED code displayed? What do you have to do to fix this?

Chapter 4. System access problems

The following topics are discussed in this chapter:

- User Licenses problems
- Telnet problems
- Adjusting AIX kernel parameters
- Tracing of hung processes

It can be very frustrating not to able to access a system. There can be many reasons for having problems to access an AIX system, despite a valid user account and corresponding password. This chapter diatribes some of the reasons why a system can have these problems and solutions to these access problems are discussed as well.

4.1 User License

If it is not possible to login into an AIX system and as soon as you try to login at AIX prompt, the session get disconnected there is an indication of AIX license problem.

Following are ways that a user can access the system that require an AIX Version 4 user license:

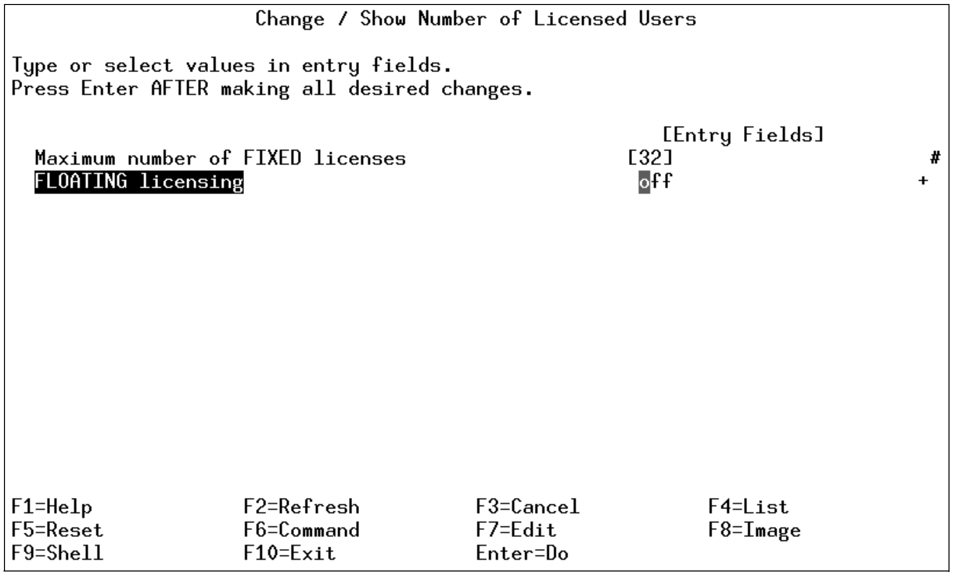
- Logins provided via a getty (from an active, local terminal)
- Logins provided using the `rlogin` or `rsh` -l command
- Logins provided using the `telnet` or tn command
- Logins provided through the Common Desktop Environment (visual login CDE)
- Any other way of accessing the AIX Version 4 BOS system does not require AIX user licenses (for example: `ftp`, `rexec`, `rsh` without the -l flag).

The `lslicense` command displays the number of fixed licenses and the status of the floating licensing.

Example:

```
# lslicense
Maximum number of fixed licenses is 32.
Floating licensing is disabled.
```

To change the number and licenses use the SMIT menu: `smit chlicense`. The Figure 11 shows the corresponding SMIT screen:



```
                     Change / Show Number of Licensed Users

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

                                                        [Entry Fields]
  Maximum number of FIXED licenses                   [32]                  #
  FLOATING licensing                                  off                 +









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

Figure 11. SMIT menu to change the number of licensed users.

In order for the changes to take effect a reboot is required.

4.2 Telnet

If a telnet to an AIX system is not possible there can be a number of reasons:

- No network connection
- inetd server not running
- telnet subserver not configured
- slow login times because of name server problems

In the following these problem areas are discussed in detail.

4.2.1 Network problem

If telnet from a client shows the following error message:

```
# telnet server1
Trying...
telnet: connect: A remote host did not respond within the timeout period.
```

it is likely to be related to network problems.

Try to use the ping command to see if the system can be reached at all. If you cannot ping the system, your problem is related to the network and the problem can either be the system itself or a access error to network due to an erroneous router or gateway.

4.2.2 telnetd subserver

The telnet service is a subserver controlled by the inetd super daemon. If a telnet from a client shows the following error message:

```
# telnet server1
Trying...
telnet: connect: A remote host refused an attempted connect operation.
```

Use the following steps to analyze and recover the problem.

1. Check to see if the inetd subsystem is running, by using the system resources controller (SRC) command `lssrc`.

   ```
   # lssrc -s inetd
   Subsystem         Group            PID     Status
    inetd            tcpip            7482    active
   ```

2. Check to see if the telnet subserver is running.

   ```
   # grep telnet /etc/inetd.conf
   #telnet  stream  tcp6    nowait  root    /usr/sbin/telnetd      telnetd
   -a
   # lssrc -t telnet
   Service       Command                  Description              Status
   ```

3. Start the telnet subserver using the SRC command `startsrc` with the -t option.

   ```
   # startsrc -t telnet
   0513-124 The telnet subserver has been started.
   ```

   Verify that the telnet subserver is running with the `lssrc` command.

   ```
   # lssrc -t telnet
   Service       Command                  Description              Status


    telnet       /usr/sbin/telnetd        telnetd -a               active
   ```

Now the telnet subserver is running and a login screen similar to the one in below should be presented.

```
# telnet server1
Trying...
Connected to server1.
Escape character is '^]'.

telnet (server1)

AIX Version 4
(C) Copyrights by IBM and by others 1982, 1996.
login:
```

If the `telnet` command displays the following error:

```
# telnet server1
telnet: tcp/telnet: unknown service
```

the telnet problem is likely to be related to the /etc/services file. The file might either be corrupt or the telnet entry is missing. Following stanza should be present in the /etc/services file, mapping the telnet service to port 23.

```
# grep telnet /etc/services
telnet          23/tcp
```

4.2.3 Slow telnet login

If the login with telnet takes a long time for example above 2 minutes, it is likely that the problem is related to domain name system (DNS) name server resolution. On the server on where the telnet daemon is running check the file /etc/resolv.conf.

The /etc/resolv.conf file defines the DNS name server information for local resolver routines. If the /etc/resolv.conf file does not exist, the DNS is not available and the system will attempt name resolution using the default paths, the /etc/netsvc.conf file (if it exists), or the NSORDER environment variable (if it exists).

When a DNS server is specified during TCP/IP configuration a /etc/resolv.conf file is generated. Further configuration of the resolv.conf file can be done using the SMIT menu: `smit resolv.conf`.

Determine the IP address of your name server from the /etc/resolv.conf file and test if name resolution is working correctly using the command `nslookup` to determine the IP address of your telnet client machine, given the hostname as input. If the DNS name server does not respond, contact the network administration to fix the problem or alternatively provide you with another name server. Additionally change the name resolution order, by either editing or creating the file /etc/netsvc.conf. Change the search order to be:

```
hosts=local, bind
```

This will force the system to use the /etc/host file for name resolution first. Enter a stanza for your telnet client machine and your login time would improve significantly.

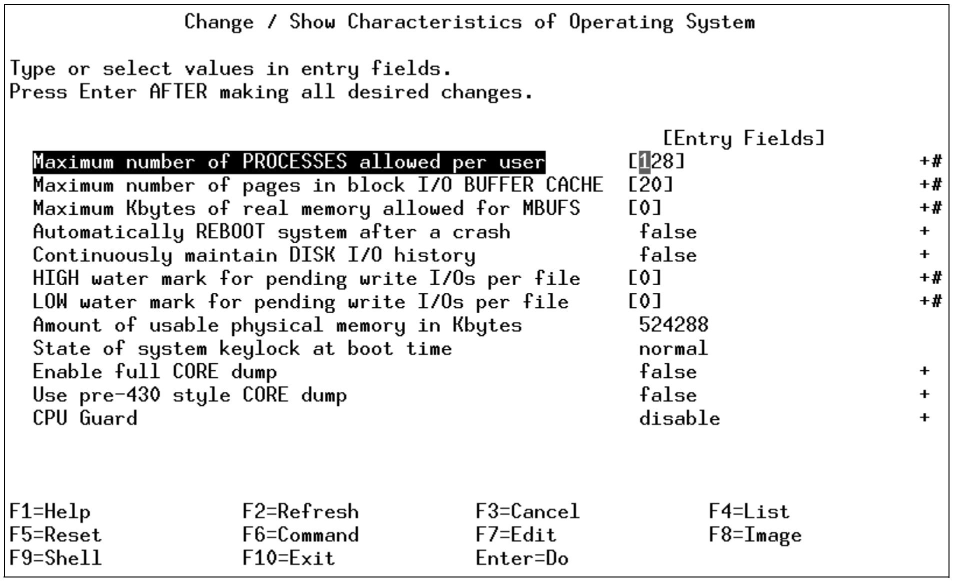
4.3 Adjusting AIX kernel parameters

Some applications need to run as a certain type of user for example database applications. Depending on the implementation some of these applications might require a large set of running processes. However the number of processes per user is limited and define as an AIX kernel parameter. If you see the error message:

```
0403-030 fork function failed too many processes exist
```

it is likely that you have reached the maximum possible number of processes per user.

This can be changed via the SMIT menu: `smit chgsys`, The Figure 12 shows the corresponding SMIT screen.



```
                 Change / Show Characteristics of Operating System

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

                                                        [Entry Fields]
  Maximum number of PROCESSES allowed per user       [128]                +#
  Maximum number of pages in block I/O BUFFER CACHE  [20]                 +#
  Maximum Kbytes of real memory allowed for MBUFS    [0]                  +#
  Automatically REBOOT system after a crash           false               +
  Continuously maintain DISK I/O history              false               +
  HIGH water mark for pending write I/Os per file    [0]                  +#
  LOW water mark for pending write I/Os per file     [0]                  +#
  Amount of usable physical memory in Kbytes          524288
  State of system keylock at boot time                normal
  Enable full CORE dump                               false               +
  Use pre-430 style CORE dump                         false               +
  CPU Guard                                           disable             +



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

Figure 12. SMIT screen for changing AIX operating system characteristics.

The same value can be changed using the command `chdev` on the device sys0 setting the attribute maxuproc.

4.3.1 Full file system

When the file system on your system runs full, it can lead to that it is not possible to login to the system either using telnet or a directly connected TTY or system console. The following message is typically displayed:

```
telnet problem 004 - 004 you must exect "login from the lowest login shell"
```

or on system console:

```
3004-004 you must 'exec' login from the lowest login shell
```

Check your file systems are not full especially the / (root) file system. Use the `df` command to verify the status of free disk space on your file systems. When a file system is full then enlarge the file system using the `chfs` command.

If your file systems are not full check if the following files are okay:

- /etc/utmp
- /etc/security/limits

Check the files both for existence, permissions, ownerships are fine. If the problem persist check the if there is an APAR that addresses this or a similar problem.

4.4 Tracing a hung process

The trace system is a tool allowing you to capture the sequential flow of system activity or system events. Unlike a stand-alone kernel dump that provides a static snapshot of a system, the trace facility provides a more dynamic way to gather problem data.

Trace can be used to:

- Isolate, understand, and fix system or application problems
- Monitor system performance

The events that are traced are timestamped as they are written to a binary trace file named /var/adm/ras/trcfile.

There are events pre-defined in AIX and included in selected commands, libraries, kernel extensions, devices drivers, and interrupt handlers. A user can define their own trace events in application code.

The trace facility generates a large amount of data. For example, a trace session capturing one second of events from an idle system gathered four thousand events in the trace. This value depends on what events you trace and the CPU performance of the system.

The trace facility and commands are provided as part of the Software Trace Service Aids fileset named bos.sysmgt.trace.

Note

Before tracing events it is important to have a strategy for what to trace, and to time the tracing.

Follow these steps to gather a useful trace:

1. Select the trace hook IDs for tracing.
2. Start the trace.
3. Recreate the problem.
4. Stop the trace.

5. Generate the trace report.

4.4.1 Trace hook IDs

The events traced are referenced by hook identifiers. Each hook ID uniquely refers to a particular activity that can be traced.

HookIDs are defined in the /usr/include/sys/trchkid.h file. When tracing you can select the hook IDs of interest, by using the trace flag -j and exclude others that are not relevant to your problem, by using the trace flag -k.

Following is extracted from the trchkid.h file:

```
...
#define HKWD_SYSC_MKDIR           0x15600000
#define HKWD_SYSC_MKNOD           0x15700000
#define HKWD_SYSC_MNTCTL          0x15800000
#define HKWD_SYSC_MOUNT           0x15900000
#define HKWD_SYSC_NICE            0x15a00000
#define HKWD_SYSC_OPEN            0x15b00000
#define HKWD_SYSC_OPENX           0x15c00000
#define HKWD_SYSC_OUNAME          0x15d00000
#define HKWD_SYSC_PAUSE           0x15e00000
#define HKWD_SYSC_PIPE            0x15f00000
#define HKWD_SYSC_PLOCK           0x16000000
#define HKWD_SYSC_PROFIL          0x16100000
#define HKWD_SYSC_PTRACE          0x16200000
#define HKWD_SYSC_READ            0x16300000
#define HKWD_SYSC_READLINK        0x16400000
#define HKWD_SYSC_READX           0x16500000
#define HKWD_SYSC_REBOOT          0x16600000
#define HKWD_SYSC_RENAME          0x16700000
#define HKWD_SYSC_RMDIR           0x16800000
...
```

When specifying the hook ID to the trace command then only the three leftmost digits need to be specified. For example when the *open* system call is traced the value *15b* needs to be specified.

Specifying relevant (or irrelevant hook IDs) can be rather difficult as you don’t know the actual cause of the problem. If source code access to the application is available or the developer is known, then this can be helpful for specifying the hook IDs of interest.

Specifying good hook IDs can reduce the mount of data significantly and make the analysis part of the problem easier.

4.4.2 Starting trace

Trace can be started in background mode or interactive mode.

To perform a trace in interactive mode, invoke the `trace` command with a list of events you want to monitor and the name of the trace log output file. The events have been assigned numbers that are called trace hooks.

The usual way to perform a `trace` is in the background. To do so use the `-a` parameter. An ampersand (&) is not necessary at the end of the command, as the `trace` command will spawn the trace daemon, and return to the shell prompt immediately. The trace is stopped using the `trcstop` command.

Typical command sequence could be:

```
# trace -a -j 15b
# myprogram
# trcstop
```

This example traces only the open operating system, made on the system.

Trace uses in-memory buffers to save the trace data. There are three methods of using the trace buffers:

Alternate mode This is the default mode. All trace events will be recorded in the trace log file.

Circular mode The trace events wrap within the in-memory buffers and are not captured in the trace log file until the trace data collection is stopped.

Single mode The collection of trace events stops when the in-memory trace buffer fills up and the contents of the buffer are captured in the trace log file.

4.4.3 Trace reports

The binary /var/adm/ras/trcfile trace file containing all system events collected during the trace period. To get a readable format this file needs to be translated using the command `trcrpt`, which generated a output report.

To output a formatted trace report to the /tmp/trace.out file, run the following command:

```
# trcrpt -o /tmp/trace.out
```

The output of the trace report file is usually very large, depending on the length of trace as well as the system activity. Despite selecting a narrow time period to do your trace, the system may be tracing a large set of unrelated events from the execution of other threads or interrupt handlers.

To generate a more reduced report a set of filters can be specified, to get a trace report that isolates the problem better.

The `trcrpt` command allows a large set of filters. Following is a types of filters are possible:

- limit report on event hook IDs
- limit report on process IDs
- limit the report to specific time

The report format can be customized using the `trcrpt` flag -O and defining an option value. For example adding the process id of a calling process into the report can be done by:

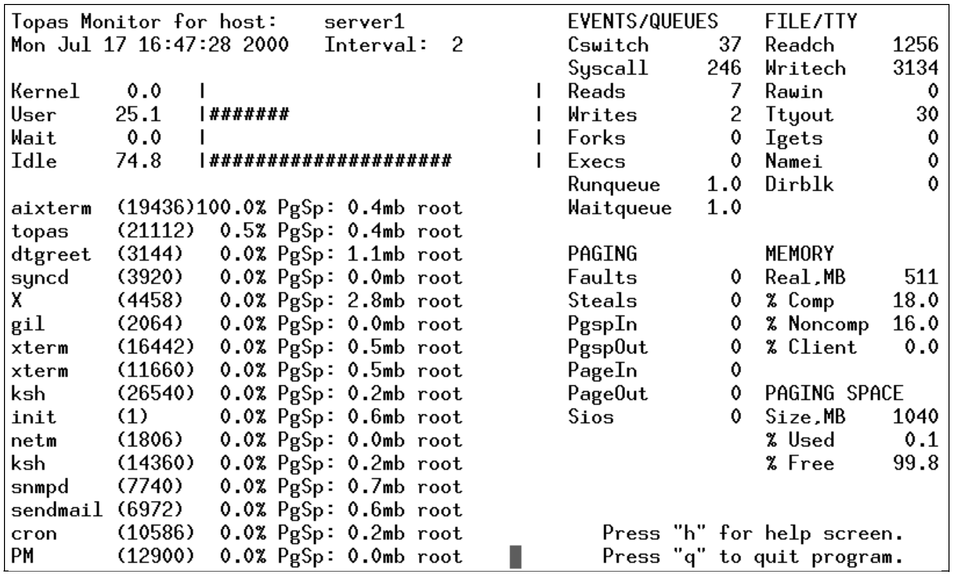
```
# trcprt -O pid=on -o /tmp/trace.out
```

For a complete set of option refer to the manual page of the `trcrpt` command.

4.4.4 Tracing example

The following example scenario will describe how to use the trace facility to analyze a hung process.

In this example system a `aixterm` process is using 100% of one CPU. The Figure 13 shows the output of an `topas` command display.



```
Topas Monitor for host:    server1              EVENTS/QUEUES    FILE/TTY
Mon Jul 17 16:47:28 2000   Interval:  2         Cswitch      37  Readch     1256
                                                Syscall     246  Writech    3134
Kernel    0.0   |                          |    Reads         7  Rawin         0
User     25.1   |#######                   |    Writes        2  Ttyout       30
Wait      0.0   |                          |    Forks         0  Igets         0
Idle     74.8   |#####################     |    Execs         0  Namei         0
                                                Runqueue    1.0  Dirblk        0
aixterm  (19436)100.0% PgSp: 0.4mb root         Waitqueue   1.0
topas    (21112)  0.5% PgSp: 0.4mb root
dtgreet  (3144)   0.0% PgSp: 1.1mb root         PAGING           MEMORY
syncd    (3920)   0.0% PgSp: 0.0mb root         Faults        0  Real,MB     511
X        (4458)   0.0% PgSp: 2.8mb root         Steals        0  % Comp     18.0
gil      (2064)   0.0% PgSp: 0.0mb root         PgspIn        0  % Noncomp  16.0
xterm    (16442)  0.0% PgSp: 0.5mb root         PgspOut       0  % Client    0.0
xterm    (11660)  0.0% PgSp: 0.5mb root         PageIn        0
ksh      (26540)  0.0% PgSp: 0.2mb root         PageOut       0  PAGING SPACE
init     (1)      0.0% PgSp: 0.6mb root         Sios          0  Size,MB    1040
netm     (1806)   0.0% PgSp: 0.0mb root                          % Used      0.1
ksh      (14360)  0.0% PgSp: 0.2mb root                          % Free     99.8
snmpd    (7740)   0.0% PgSp: 0.7mb root
sendmail (6972)   0.0% PgSp: 0.6mb root
cron     (10586)  0.0% PgSp: 0.2mb root            Press "h" for help screen.
PM       (12900)  0.0% PgSp: 0.0mb root            Press "q" to quit program.
```

Figure 13. Output display of the topas command

As this system is a 4-way SMP the overall CPU usage is only 25%. To analyze what is actually happening on this system we want to use the trace facility. Notice that the process id of the `aixterm` is 19436.

As this `aixterm` process seems to loop continuously, the tracing time is limited to 1 second.

Using the command sequence:

```
# trace -a; sleep 1; trcstop
```

will trace all system events for one second. As we do not know what the `aixterm` process is doing no event hook IDs can be specified at this point. The trace generates a raw trace file of the size:

```
# ls -l  /var/adm/ras/trcfile
-rw-rw-rw-   1 root     system    557152 Jul 17 14:27 /var/adm/ras/trcfile
```

Based on this file a trace report can be generated with `trcrpt`. As we know the process id, we can use this information as filter and limit the output of the report:

```
# trcrpt -p 19436 > /tmp/trace.out
# ls -l  /tmp/trace.out
-rw-r--r--   1 root     system    201014 Jul 17 14:31 /tmp/trace.out
```

The contents of the trace report is the following as an extracted subpart of the complete report:

```
Mon Jul 17 14:27:27 2000
System: AIX server1 Node: 4
Machine: 000BC6FD4C00
Internet Address: 0903F038 9.3.240.56
The system contains 4 cpus, of which 4 were traced.
Buffering: Kernel Heap
This is from a 32-bit kernel.



trace -a


ID     ELAPSED_SEC     DELTA_MSEC   APPL    SYSCALL KERNEL  INTERRUPT

100    0.004256674      4.256674                        DECREMENTER INTERRUPT iar=D031EB
60 cpuid=FFFFFFFF
234    0.004258505      0.001831                 clock:   iar=D031EB60 lr=D036CA24 [2503
usec]
112    0.004260143      0.001638                  lock:         lock lock addr=352118 loc
k status=10000001 requested_mode=LOCK_READ return addr=2D80C name=0000.0000
113    0.004261661       0.001518                   unlock: lock addr=352118 lock
status=000
0 return addr=2D8D8 name=0000.0000
112    0.004270432      0.008771                  lock:         lock lock addr=352118 loc
k status=10000001 requested_mode=LOCK_READ return addr=2D3C4 name=0000.0000
113    0.004271781       0.001349                   unlock: lock addr=352118 lock
status=000
0 return addr=2D5A4 name=0000.0000
112    0.004272503      0.000722                  lock:         lock lock addr=352118 loc
k status=10000001 requested_mode=LOCK_READ return addr=2DEA8 name=0000.0000
113    0.004274213       0.001710                   unlock: lock addr=352118 lock
status=000
0 return addr=2E0EC name=0000.0000
112    0.004274863      0.000650                  lock:         lock lock addr=352118 loc
k status=10000001 requested_mode=LOCK_READ return addr=2D90C name=0000.0000
113    0.004275706       0.000843                   unlock: lock addr=352118 lock
status=000
0 return addr=2D980 name=0000.0000
10E    0.004278910      0.003204                 relock: lock addr=34DEA0  oldtid=12679
newtid=1033
10E    0.004279946      0.001036                 relock: lock addr=34DEA0  oldtid=1033  n
ewtid=12679
106    0.004280644      0.000698                 dispatch:   cmd=aixterm pid=19436 tid=12
679 priority=93 old_tid=12679 old_priority=93 CPUID=2 [3551 usec]
200    0.004283438       0.002794                   resume  aixterm iar=D031EB60 cpuid=02
100    0.014254631      9.971193                        DECREMENTER INTERRUPT iar=D031EB
60 cpuid=02
234    0.014256414      0.001783                 clock:   iar=D031EB60 lr=D036CA24 [2497
usec]
112    0.014258004      0.001590                  lock:         lock lock addr=352118 loc
k status=10000001 requested_mode=LOCK_READ return addr=2D80C name=0000.0000

...
```

The heading shows some system information like 32 bit kernel, 4 CPUs and so on. The next part show the parameters used to activate the trace command. In last part is the actual report, where each line is the actual event recorded. The first column shows the event hook IDs the system has performed.

From the output of this example it seems like the `aixterm` process is hung up in some kernel resources, as the only events the process is performing are lock and unlock operations. To go into deeper analysis of this problem would typically require to look into the program sources of the application.

4.5 Command summary

The following are commands discussed in this Chapter and the flags most often used. For a complete reference of the following command use the *AIX Version 4.3 Command Reference* or the online man pages.

4.5.1 lslicense

The lslicense displays the number of fixed licenses and the status of the floating licensing. The command has the following syntax:

```
lslicense [ -c ]
```

4.5.2 lssrc

The lssrc command gets the status of a subsystem, a group of subsystems, or a subserver. The command has the following syntax:

Subsystem Status:

```
lssrc [ -h Host ] { -a | -g GroupName | [ -l ] -s Subsystem | [ -l ] -p
SubsystemPID }
```

Subserver Status

```
lssrc [ -h Host ] [ -l ] -t Type [ -p SubsystemPID ] [ -o Object ] [ -P
SubserverPID ]
```

Note the SMIT format command flags are omitted.

Table 7. Commonly used flags of the lssrc command

Flag	Description
-a	Lists the current status of all defined subsystem.
-g Group	Specifies a group of subsystems to get status for. The command is unsuccessful if the GroupName variable is not contained in the subsystem object class.

Flag	Description
-s Subsystem	Specifies a subsystem to get status for. The Subsystem variable can be the actual subsystem name or the synonym name for the subsystem. The command is unsuccessful if the Subsystem variable is not contained in the subsystem object class.
-t Type	Requests that a subsystem send the current status of a subserver. The command is unsuccessful if the subserver Type variable is not contained in the subserver object class.

4.5.3 startsrc

The startsrc starts a subsystem, a group of subsystems, or a subserver. The command has the following syntax:

For Subsystem

```
startsrc [-a Argument] [-e Environment] [-h Host] {-s Subsystem |-g Group}
```

For Subserver

```
startsrc [-h Host] -t Type [-o Object] [-p SubsystemPID]
```

Table 8. Commonly used flags of the startsrc command

Flag	Description
-s Subsystem	Specifies a subsystem to be started. The Subsystem can be the actual subsystem name or the synonym name for the subsystem. The command is unsuccessful if the Subsystem is not contained in the subsystem object class.
-t Type	Specifies that a subserver is to be started. The command is unsuccessful if Type is not contained in the subserver object class.

4.5.4 trace

The trace records selected system events. The command has the following syntax:

```
trace [ -a [ -g ] ] [ -f | -l ] [-b | -B] [-c] [ -d ] [ -h ] [-j Event [
,Event] ] [-k Event [ ,Event ] ] [ -m Message ] [ -n ] [ -o Name ] [ -o- ]
```

```
[ -s ] [ -L Size ] [ -T Size ]startsrc [-a Argument] [-e Environment] [-h
Host] {-s Subsystem |-g Group}
```

Table 9. Commonly used flags of the trace command

Flag	Description
-a	-a Runs the trace daemon asynchronously (i.e. as a background task). Once trace has been started this way, you can use the trcon, trcoff, and trcstop commands to respectively start tracing, stop tracing, or exit the trace session. These commands are implemented as links to trace.
-j Event[,Event] or -k Event[,Event]	Specifies the user-defined events for which you want to collect (-j) or exclude (-k) trace data. The Event list items can be separated by commas, or enclosed in double quotation marks and separated by commas or blanks. Note: The following events are used to determine the pid, the cpuid and the exec path name in the trcrpt report: 001 TRACE ON 002 TRACE OFF 106 DISPATCH 10C DISPATCH IDLE PROCESS 134 EXEC SYSTEM CALL 139 FORK SYSTEM CALL 465 KTHREAD CREATE If any of these events is missing, the information reported by the trcrpt command will be incomplete. Consequently: when using the -j flag, you should include all these events in the Event list; conversely, when using the -k flag, you should not include these events in the Event list.

4.5.5 trcrpt

The trcrpt formats a report from the trace log. The command has the following syntax:

```
trcrpt [ -c ] [ -d List ] [ -e Date ] [ -h ] [ -j ] [ -n Name ] [ -o File ]
[ -p List ] [ -r ] [ -s Date ] [ -t File ] [ -T List ] [ -v ] [ -O Options
] [ -x ] [ File ]
```

Table 10. Commonly used flags of the trcrpt command

Flag	Description
-o File	Writes the report to a file instead of to standard output.

Flag	Description
-O Options	Specifies options that change the content and presentation of the trcrpt command. Arguments to the options must be separated by commas. Examples of options are: cpuid=[on|off] Displays the physical processor number in the trace report. The default value is off. endtime=Seconds Displays trace report data for events recorded before theseconds specified. Seconds can be given in either an integral or rational representation. If this option is used with the starttime option, a specific range can be displayed. exec=[on|off] Displays exec path names in the trace report. The default value isoff. pid=[on|off] Displays the process IDs in the trace report. The default value is off. svc=[on|off] Displays the value of the system call in the trace report. The default value is off. For a complete list of pptions please refer to the manual page of trcrpt

4.6 Quiz

4.6.1 Answers

4.7 Exercises

The following exercises provide the setting for additional learning.

1. Verify the number of licenses available on your system.
2. List the AIX kernel parameters on your system by using the `lsattr` command on device sys0.

3. Perform a trace on your system to see what your system is actually doing right now. Limit the trace to only a few seconds.
4. Generate a report of the trace performed in the previous step, adding the option for showing the process ID in the report.

Chapter 5. Hardware problem determination

The following topics are discussed in this chapter:

- Hardware inventory
- Diagnostic program
- SSA problem determination
- Three-Digit display codes

This chapter describes common hardware-related problem determination. It provides problem resolving procedures depend on the system architecture.

5.1 Finding Out about Your System

RS/6000 servers are available in a variety of models. An RS/6000 system can be single processor or multiprocessor. Currently, models comply to a number of architecture specifications such as Micro Channel, PowerPC Reference Platform (PREP), Common Hardware Reference Platform (CHRP), and RS/6000 Platform Architecture (RPA).

The hardware platform type is an abstraction that allows machines to be grouped according to fundamental configuration characteristics such as the number of processors and/or I/O bus structure. Machines with different hardware platform types will have basic differences in the way their devices are dynamically configured at boot time. Currently available hardware platforms, which are able to be differentiated by software, in RS/6000 family are:

rs6k	Micro Channel-based uni-processor models
rs6ksmp	Micro Channel-based symmetric multiprocessor models
rspc	ISA-bus models
chrp	PCI-bus models

In order to determine the hardware platform type on your machine, enter the following command:

```
# bootinfo -p
chrp
```

5.1.1 Hardware inventory

For system hardware inventory use either the `lsdev` command or the `lscfg` command. These commands show different aspects of installed hardware.

The `lsdev` command displays information about devices in the Device Configuration database.

```
# lsdev -C
sys0       Available 00-00         System Object
sysplanar0 Available 00-00         System Planar
pci0       Available 00-fef00000   PCI Bus
pci1       Available 00-fee00000   PCI Bus
pci2       Available 00-fed00000   PCI Bus
isa0       Available 10-58         ISA Bus
sa0        Available 01-S1         Standard I/O Serial Port
sa1        Available 01-S2         Standard I/O Serial Port
scsi1      Available 30-58         Wide SCSI I/O Controller
cd0        Available 10-60-00-4,0  SCSI Multimedia CD-ROM Drive
mem0       Available 00-00         Memory
proc0      Available 00-00         Processor
proc1      Available 00-01         Processor
proc2      Available 00-02         Processor
proc3      Available 00-03         Processor
L2cache0   Available 00-00         L2 Cache
sioka0     Available 01-K1-00      Keyboard Adapter
fd0        Available 01-D1-00-00   Diskette Drive
rootvg     Defined                 Volume group
hd5        Defined                 Logical volume
tok0       Available 10-68         IBM PCI Tokenring Adapter (14103e00)
ent0       Available 10-80         IBM PCI Ethernet Adapter (22100020)
ent1       Available
```

The output shows whether the device is in the Available or Defined state.

Use the `lscfg` command to display vital product data (VPD) such as part numbers, serial numbers, microcode level and engineering change levels from either the Customized VPD object class or platform specific areas. To display all of this features, for the `hdisk1`, enter:

```
# lscfg -vp -l hdisk1
  DEVICE            LOCATION          DESCRIPTION

  hdisk1            10-60-00-9,0      16 Bit SCSI Disk Drive (9100 MB)

        Manufacturer................IBM
        Machine Type and Model......DNES-309170W
        FRU Number..................25L3101
        ROS Level and ID............53414730
        Serial Number...............AJ286572
        EC Level....................F42017
        Part Number.................25L1861
        Device Specific.(Z0)........000003029F00013A
        Device Specific.(Z1)........25L2871
        Device Specific.(Z2)........0933
        Device Specific.(Z3)........00038
        Device Specific.(Z4)........0001
        Device Specific.(Z5)........22
        Device Specific.(Z6)........F42036


  PLATFORM SPECIFIC

  Name:  sd
    Node:  sd
    Device Type:  block
```

The most important for you are:

ROS Level and ID This is the microcode level and it is used to determine firmware version in your device.

FRU Number You will use this number to order the same device in cause of damage the original one.

To displays attribute characteristics and possible values of attributes for devices in the system use the lsattr command:

```
# lsattr -El hdisk1
pvid        000bc6ddc63c40380000000000000000 Physical volume identifier
False
queue_depth 3                               Queue DEPTH              False
size_in_mb  9100                            Size in Megabytes        False
```

Note

This is a good practice to have print outs from the `lscfg`, `lsdev` and `lsattr` commands.

5.2 Running diagnostics

Diagnostics on hardware can be run in three different ways. The first way of running the diagnostics is in concurrent mode, that is to say the system is up and running with users on and all processes running and all volume groups being used. The second way is Service mode, this is when you have the machine with AIX running but with the minimum of processes started and only rootvg varied on. Finally, the third way is stand-alone diagnostics from CD.

The CD-based diagnostics are a completely isolated version of AIX and so any diagnostics run are totally independent of the AIX setup on the machine being tested.

Which of these methods you use depends upon the circumstances such as:

- Are you able to test the device? Is the device in use?
- Do you need to decide if the problem is related to hardware or AIX? Stand-alone diagnostics from CD or diskette are independent of the machine operating system. Advanced diagnostics run using the diagnostic CD or diskettes and completing successfully should be taken as proof of no hardware problem.

Note

If you are going to boot from CD or a mksysb tape on a machine that is in any configuration that has two or more SCSI adapters sharing the same SCSI bus, check that no SCSI adapters on the shared bus are set at address 7. If you boot from bootable media, the bootable media will automatically assign address 7 to all SCSI adapters in the machine being booted and will cause severe problems on any other machines sharing the same SCSI bus that have address 7 IDs set on their adapters.

The method by which you run diagnostics varies between machine type. The next sections describe in detail how to run all the diagnostic modes on all machine types.

There are two RS/6000 models that do not have the capability to run AIX-based diagnostics. These machines are 7020-40P and 7248-43P. To run diagnostics on these models requires you to have the SMS diskette for the machine.

5.2.1 Concurrent mode

Concurrent mode diagnostics are run while AIX is running on the machine and potentially with users on. To run diagnostics concurrently, you must have root authority and use one of the methods listed below:

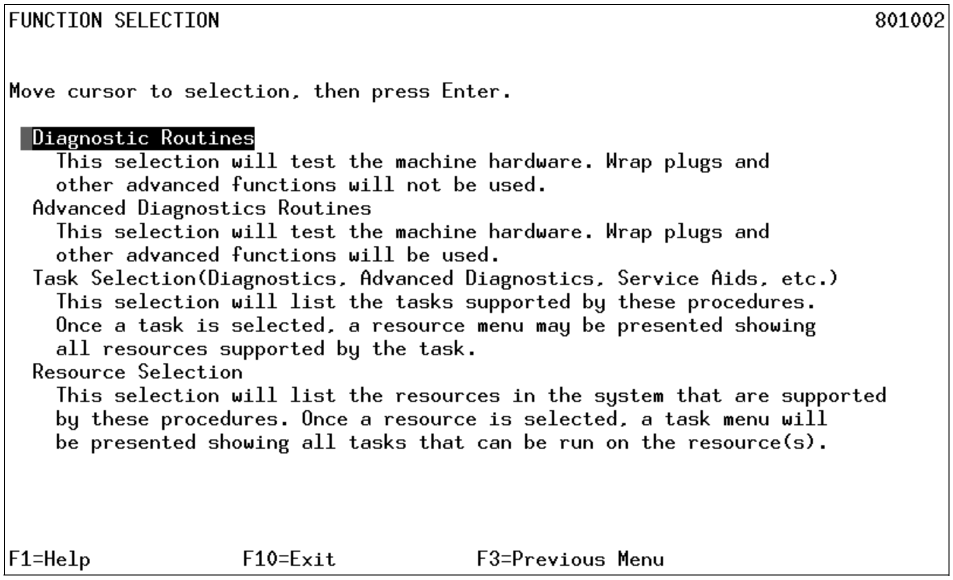
- To run diagnostics on a specific device, use the following command:

  ```
  diag -d [resource name]
  ```

  This command will enable you to test a specific device directly without the need to pass through a number of menus. The diagnostic process run is the Advanced Diagnostic process.

- To go directly to the main diagnostics menu, use the `diag` command.
- Using SMIT take the following menu route:
  - Problem Determination
  - Hardware Diagnostics
  - Current shell

Methods 2 and 3 will get you to the entry screen of the diagnostics menu. If you press **Enter** to continue from the entry screen, you will be presented with a menu as shown in Figure 14.



```
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 14. Main Diagnostics menu

The menu options shown in Figure 14 are explained in the following paragraphs:

Diagnostic Routines

This set of routines is primarily aimed at the operator of the machine. When the diagnostics are run using this option, there will be no prompts to unplug devices or cables, and no wrap plugs are used. Therefore, the testing done by this method is not as comprehensive as the testing performed under Advanced Diagnostics. In some cases, it can produce a `No Trouble Found` result when there is an actual problem.

Advanced Diagnostics
This set of routines will run diagnostic tests that will ask you to remove cables, plug and unplug wrap plugs, and use various other items. As a result, the tests run are as detailed as possible. Generally, if you get a `No Trouble Found` result using Advanced Diagnostics, you can be reasonably certain the devices tested have no hardware defects.

Task Selection
This section is sometimes referred to as Service Aids. There are many useful tools within this section. The use of this option is discussed in Section 5.2.4, “Task selection or service aids” on page 96.

After you have selected the level of diagnostics you wish to run, you will then be presented with a menu for you to decide to use either the Problem Determination method or the System Verification method.

Problem Determination
This selection will make the diagnostic routine search the AIX error log for any errors posted in the previous 24 hours against the device you are testing. It will then use the sense data from any error log entry for the device being tested in conjunction with the results of the diagnostic testing of the device to produce a Service Request Number (SRN). This method must be used to determine the cause of machine checks and checkstops on 7025 and 7026 machine types. If you are performing diagnostics more than seven days since the machine check occurred then you will need to set the system date and time to within seven days of the machine check timestamp. The seven day period is required when using AIX 4.3.1 and later. If you are using AIX 4.3.0 or below, then the system date and time must be within 24 hours of the checkstop entry.

System Verification
Use this selection if you have just replaced a part or performed a repair action. System verification runs a diagnostic to the device but does not refer to the AIX error log, so it reflects the machines condition at the time of running the test. You can also use system verification when you just want to run a straight test to a device or whole machine.

Concurrent mode provides a way to run Online Diagnostics on the system resources while AIX is up and running and users are logged on.

Since the system is running in normal operation, some resources cannot be tested in concurrent mode. The following list shows which resources cannot be tested:

- SCSI adapters used by disks connected to paging devices
- The disk drives used for paging
- Memory
- Processor

Depending on the status of the device being tested, here are four possible test scenarios in concurrent mode:

- Minimal testing is used when the device being tested is under the control of another process.
- Partial testing occurs when testing is performed on an adapter or device that has some processes controlling part of it. For example, testing unconfigured ports on an 8-port RS232 adapter.
- Full testing requires the device be unassigned and unused by any other process. Achieving this condition may require commands to be run prior to the commencement of the diagnostic testing.
- When tests are run to the CPU or memory, the diagnostics refer to an entry in the NVRAM that records any CPU or memory errors generated during initial testing at system power on time. By analyzing these entries, the diagnostics produce any relevant SRNs.

5.2.2 Stand-alone diagnostics from disk

This mode enables you to run tests to the devices that would ordinarily be busy if you ran diagnostics with the machine up in Normal mode boot, for example, the network adapter ent0. However, you still will not be able to test any SCSI device that is attached to disks containing paging space or rootvg. Stand-alone diagnostics from disk is started when you boot up the machine in Service mode boot. The method that you employ to get a Service mode boot depends upon the type of machine.

5.2.2.1 MCA machines

To start a Service mode boot, power off the machine, then:

1. Set the key mode switch of the machine to the Service position.
2. Power on the machine without a CD, tape, or diskette in the machine.

After a period of time, you will see the Diagnostics Entry screen appear on the console. Press **Enter** and you will then get to the screen giving you the choice of diagnostics to run.

5.2.2.2 PCI machines

This section applies to machines of model type 7017, 7024, 7025, 7026, 7043 and 7046. It does not apply to PCI machine types 7020 or 7248.

To start a Service mode boot, power off the machine, then:

1. Turn on the machine power.
2. After a short period of time you will see the Icons screen. At this point, press **F6** if using a graphics console, or **6** if using an ASCII terminal. If you are using the graphics console, sometimes the display device will have power saving enabled, and so will take time to warm up and display images. This can lead you to miss the Icon screen being displayed. In this situation, observe the power LED on the display device, and when it changes from orange to green, simply press the **F6** key.

Once the keyboard input has been processed, the machine will display a Software Starting screen. This can then be followed by more information indicating the SCSI ID of the boot device being used. Once diagnostics have been loaded, you will have the Diagnostic Entry screen displayed.

5.2.3 Stand-alone diagnostics from CD

Stand-alone diagnostics run from CD or diskettes is a good way of proving if the problem is a hardware fault or an AIX problem. The CD or diskettes load a totally independent version of AIX onto the machine as a RAM image. If you get a `No Trouble Found` result using advanced diagnostics using all of the test equipment asked for during the diagnostic, the probability of there being a hardware problem is extremely small. In such cases, the underlying cause of the problem is most often software related.

Note

If you are going to boot from CD or a mksysb tape on a machine that is in any configuration that has two or more SCSI adapters sharing the same SCSI bus, check that no SCSI adapters on the shared bus are set at address 7. If you boot from bootable media, the bootable media will automatically assign address 7 to all SCSI adapters in the machine being booted, and so will cause severe problems on any other machines sharing the same SCSI bus that have address 7 IDs set on their adapters

5.2.3.1 MCA machines

This section describes how to boot from CD on MCA machines and from diskette for the early level of MCA machines.

Boot from CD

To boot from CD, complete the following steps:

1. Power off the machine.
2. Turn the key mode switch to the Service position.
3. Power on the machine, then place the Diagnostic CD in the drive.

   For the machine to boot from the Diagnostic CD, there must be an entry in the boot list including the CD. Using the code on the CD, the machine will boot, eventually pausing when displaying c31 in the LED panel. The code c31 is an indication to you that you need to select a system console. After selecting a console at the prompt, you will get the Diagnostic Entry screen followed by subsequent screens. One of these subsequent screens will prompt you to enter the terminal type. Make sure you know the type before you proceed, since a wrong entry could result in you having to restart the whole process again.

5.2.3.2 PCI Bus machines

This section applies to machines of model type 7017, 7024, 7025, 7026,7043 and 7046. It does not apply to PCI machine types 7020 or 7248.

To start a CD boot:

1. Power off the machine.
2. Turn on machine power.
3. Place the CD into the drive.
4. After a short period of time, you will see the Icons screen. At this point, press **F5** if you are using a graphics console, or **5** if you are using an ASCII terminal. If you are using the graphics console, sometimes the display screen will have power saving enabled, and so take time to warm up before anything can be seen on the screen. This can lead you to miss the Icon screen being displayed. In this situation, observe the power LED on the display device, and when it changes from orange to green then press the **F5** key.

After performing the previous steps, you will get various screens displayed, one of which will indicate to you the SCSI address of the device that the machine is booting from. Following on from this screen, you will then have the Diagnostic Entry screen displayed.

5.2.4 Task selection or service aids

This section is known by two names *service aids* or *task selection* dependent upon the level of diagnostics you are using. Task selection is the name used by AIX 4.3.2; however, in AIX 4.1.4, the same menu is known as service aids. This portion of the diagnostic package is equally as useful in the diagnosis of faults as the diagnostic routines themselves. The next few sections will cover a selection of the service aids available.

5.2.4.1 Local area network service aid

This service aid is useful in the diagnosis of network problems. It enables you to type in IP addresses of both a source machine and a target machine. When activated, it will tell you if it managed to connect to the target machine. If it failed, it will try and give you a reason why it could not reach the destination host. The result of this can sometimes help in furthering fault diagnosis.

5.2.4.2 Microcode download

Using this service aid makes manipulation of microcode much easier than doing it from the command line. As a result, you are less liable to make a mistake.

The microcode download facility is also available when using the Diagnostic CD. This enables the down loading of microcode to devices that are not capable of being updated when AIX is running.

5.2.4.3 SCSI bus analyzer

This is probably one of the most useful service aids. It enables you to issue a SCSI inquiry command to any device on any SCSI bus connected to the machine. The results that are returned give you a good idea of the problem. The results returned are:

- The exerciser transmitted a SCSI Inquiry command and did not receive any response back. Ensure that the address is valid, then try this option again.
- The exerciser transmitted a SCSI Inquiry command and received a valid response back without any errors being detected.
- A check condition was returned from the device.

To run this service aid:

1. From the Task Selection menu, select **SCSI Bus Analyzer**.
2. At the next screen, select the adapter that has the device that you wish to test attached to it.

3. Use the **Tab** key to increment the SCSI ID field to the number you want to test.
4. Press **F7** to confirm your selection.
5. Press **Enter** to commence the test.

If the device is working correctly, the response saying so should be returned almost instantly. If there is a problem, it should return an answer after a few seconds. Sometimes, a device that has a severe check condition will hang the service aid. If this is the case, you need to **Control-C** out of the service aid.

5.2.4.4 Disk maintenance

The disk to disk copy will only work with SCSI disks that pass diagnostics and ideally have minimal errors when the certify process is run. If the error rate is too high when disk-to-disk copy is being run, the program will fail. You will find it useful if the customer situation is such that they have no backup and the disk is unstable but running. Disk-to-disk copy differs from an AIX-based migrate operation because it does not alter the source disk when finished as the `migratepv` command does. Disk-to-disk copy is best run from CD diagnostics which requires you to have the exclusive use of the machine while the disk copying takes place. Also, the disk to be copied to *must not* be smaller than the source disk or more than 10 percent larger in size than the source disk. The copied disk will have the same PVID as the original, so the defective disk must be removed from the machine before starting AIX.

5.2.4.5 SSA service aids

This service aid can be used to help diagnose SSA subsystem problems. It is also used to physically identify and control SSA disks in the tower or drawer. This function greatly speeds the locating of specific disks, especially in very large installations.

Note

This service aid is only present when SSA devices are configured on the machine.

5.3 Serial Storage Architecture (SSA) disks

This disk subsystem is capable of being externally connected to one or more RS/6000 systems. Certain models of RS/6000 can also be configured with internal SSA disks. SSA devices are connected through two or more SSA links to an SSA adapter that is located in the system used. The devices, SSA links, and SSA adapters are configured in loops. Each loop provides a data path that starts at one connector of the SSA adapter and passes through a link (SSA cable) to the devices. The loop continues through the devices, then returns through another link to a second connector on the SSA adapter. Each adapter is capable of supporting two loops. Each loop can have between one and 48 devices. A loop can have as many as eight SSA adapters connected in up to eight systems, but this is dependent on the type of SSA adapter being used and how they are configured. Again dependent on adapters, disk subsystem, and cables in use, the aggregate loop speed per adapter can either be 80 MB/sec or 160 MB/sec. As you can see, the number of possible combinations is almost endless and changes at each product announcement. Therefore, the SSA configuration rules detailed below cover basic considerations.

5.3.1 General SSA setup rules

The following rules must be followed when connecting a 7133 subsystem:

- Each SSA loop must be connected to a valid pair of connectors on the SSA adapter card.

  A1 and A2 form one loop, and B1 and B2 form another loop.
- Only one pair of connectors of a SSA adapter can be connected in a particular SSA loop.

  A1 or A2, with B1 or B2, cannot be in the same SSA loop.
- A maximum of 48 disks can be connected in a SSA loop.
- A maximum of three dummy disk drive modules can be connected next to each other.
- A maximum of two adapters can be in the same host per SSA loop.
- Cabling joining SSA nodes should not exceed 25 meters.
- There is no addressing setup for any SSA device.
- There is no termination since all connections should form a loop.

The maximum number of adapters per SSA loop at the time of this writing is shown in Table 11.

Table 11. SSA adapter information

Feature Code	Description	Identifier	Maximum Number per Loop
6214	MCA Adapter	4-D	2
6216	MCA Enhanced SSA 4 port adapter	4-G	8

Feature Code	Description	Identifier	Maximum Number per Loop
6217	MCA SSA RAID adapter	4-I	1
6218	PCI SSA RAID adapter	4-J	1
6219	MCA Enhanced RAID adapter	4-M	Between 1 and 8 per loop depending on microcode level, and whether RAID and Fast Write Cache are used
6215	PCI Enhanced RAID Adapter	4-N	
6225	PCI Advanced Serial RAID adapter	4-P	

For the most comprehensive and up to date information on SSA adapters, refer to the following URL:

`http://www.hursley.ibm.com/~ssa/`

The user guides for each SSA adapter are available on this Web site. They contain information on the valid adapter combinations allowed on the same loop.

5.3.2 SSA devices

SSA subsystem components use microcode to control their function. When working on SSA problem, you should ensure that the microcode level and any drivers on all devices in the loop are at the latest published level.

5.3.3 SSA disk does not configure as hdisk

If you configure an SSA disk into a system, and it only shows as a pdisk with no corresponding hdisk, the most probable cause is that the disk was originally part of a RAID array set up on another machine. If disks are removed from a RAID array for any reason to be incorporated into any other system as a normal disk, the following procedure must be used:

1. Type `smitty ssaraid` (the fast path to SSA RAID SMIT panels).
2. Select `Change Show use of an SSA Physical disk`. The disk must be returned to general use as an AIX system disk.
3. If the disk is to be removed from the system, use the relevant AIX commands. Do not remove the pdisk until you have removed the disk from the system using the SSA service aids.

Obviously, if you are presented with this situation, and the disk with the problem was not a member of a RAID set on this machine, your only option to return this disk to normal use is to do a low-level format using the SSA service aid. This can take some time if the disk is 9 GB or larger.

5.3.3.1 SSA RAID

The SSA subsystem is capable of being operated by some adapters as either single system disks or as RAID LUNs. Provided that all has been set up correctly, then the RAID implementation works well. If you have any doubts as to how the RAID is set up, refer to the *SSA Adapters: User's Guide and Maintenance Information*, SA33-3272.

If you propose to do any actions involving an SSA RAID array then use the relevant procedure listed. This will ensure that the integrity of the RAID set is maintained at all times.

5.3.3.2 Changing SSA disks

SSA disk changing activity is hot swapable. When preparing AIX for the removal of an SSA disk, do not `rmdev` the pdisk prior to physically removing the disk from the enclosure. You will need the pdisk to do the following steps. Remove the pdisk only when all steps are completed. Use the SSA Service aid to power the disk off prior to removal. This is done by using the set Service mode and identify facility. This will put the disks on either side of the one you want to remove into string mode and power off the disk to be removed. When the replacement disk or blanking module is inserted, you use the same service aid to reset Service mode. This will start up the new disk and take the other disks out of string mode. At this point, you can now `rmdev` the pdisk allocated to the disk you removed. The disk change procedures will tell you to run the `cfgmgr` command.

Note

The `cfgmgr` command should *not* be executed on any system that is in an HACMP cluster. To do so will seriously damage the configuration of the machine possibly resulting in the crashing of the complete cluster.

If the disk to be changed is a defective RAID disk and was in use by the system, then you need to follow the procedures in *SSA Adapters: Users Guide and Maintenance Information,* SA33-3272. Read these procedures carefully because some of the earlier editions of this book indicate you have finished the procedure when, in fact, you need to perform other steps to return the array to a protected state. Below is a list of the important steps that need to be completed before you can be sure that the array will function correctly.

Steps involved in the replacement of a RAID SSA disks are:

1. Addition of the replacement disk to the system using `cfgmgr` command or the `mkdev` command on HACMP systems.
2. Make the disk an array candidate or hot spare using SMIT.

If the disk was removed from a RAID array leaving it in an exposed or degraded state, you now need to add the disk to the array using SMIT. While the array is being rebuilt, error messages will be seen each hour in the error log. These will cease when the array is completely rebuilt.

5.4 Three-digit display values

Three-digit display messages are system error indicators that display on the system operator panel. Most of the three-digit display values are progress indicators that only display briefly. This section enables you to interpret the codes displayed on the system operator panel.

5.4.1 Common boot time LEDs

The following sections cover some hardware related problems that can cause a halt. All problems at this stage of the startup process have an error code defined which is shown in the LED display on the front panel.

5.4.1.1 LED 200

The LED code 200 is connected to the secure key position. When the key is in the secure position - the boot will stop until the key is turned, either to the normal position or the service position, then the boot will continue.

5.4.1.2 LED 299

An LED code of 299 shows that the BLV will be loaded. If this LED code is passed, then the load has been successful. If you, after passing 299, get a stable 201 then you have to recreate the BLV.

5.4.1.3 MCA LED codes

Table 12 provides a list of the most common LED codes on MCA systems. More of these can be found in the *Message Guide and References* which is part of the AIX version 4 Base Documentation.

Table 12. Common MCA LED codes

LED	Description
100 - 195	Hardware problem during BIST

LED	Description
200	Key mode switch in secure position
201	1. If LED 299 passed recreate BLV 2. If LED 299 has not passed, POST encountered a hardware error

221,
721,
221 - 229,
223 - 229,
225 - 229,
233 - 235
	bootlist in NVRAM is incorrect, or
	`(boot from media and change the bootlist)`
	bootlist device has no bootimage, or
	`(boot from media and recreate the BLV)`
	bootlist device is unavailable
	`(Check for hardware errors)`

5.4.2 888 in the Three-Digit Display

A flashing 888 indicates that a problem was detected, but could not be displayed on the console. A message is encoded as a string of three-digit display values. The 888 will be followed by either a 102, 103, or 105. The reset button is used to scroll the message.

5.4.2.1 The 102 code

A 102 indicates that dump has occurred- your AIX kernel crashed due to bad circumstances. LED code description:

- 888 - This value flashes to indicate a system crash.
- 102 - This value indicates an unexpected system halt.
- nnn - This value is the cause of the system halt (reason code).
- 0cx - The value 0cx indicates dump status.

The reason code is the second value after 888 appears. Also, this code can be found using the `stat` subcommand in `crash`.

- 000 - Unexpected system interrupt (hardware related)
- 2xx - Machine check A machine check can occur due to hardware problems, for example, bad memory, or because of a software reference to a non-existent address.
- 3xx - Data storage interrupt. A page fault always begins as a DSI, which is handled in the exception processing of the VMM. However, if a page fault can not be resolved, or if a page fault occurs when interrupts are disabled, the DSI will cause a system crash. The page fault may not be resolved if, for example, an attempt is made to read or write a pointer that has been

freed, in other words, the segment register value is no longer valid, and the address is no longer mapped.

- 400 - Instruction access exception. Instruction Access Interrupt. This is similar to a DSI, but occurs when fetching instructions, not data.
- 5xx - External interrupt. Interrupt arriving from an external device.
- 700 - Program interrupt. Usually caused by a trap instruction that can be a result of failing an *assert*, or hitting a *panic* within kernel or kernel extension code.
- 800 - Floating point unavailable An attempt is made to execute a floating point instruction but the floating point available bit in the Machine Status Register (MSR) is disabled.

For more information about system dump and the dump status code check //// dump chapter made by Andre////

5.4.2.2 The 103 and 105 code

A 103 message indicates that a Service Request Number (SRN) follows the 103. The SRN consists of the two sets of digits following the 103 message. This number together with other system related data is used to analyze the problem. Record and report the SRN to your service representative. A 105 message indicates that an encoded SRN follows the 105. Record and report SRN 111-108 to your service representative. These format is shown on the

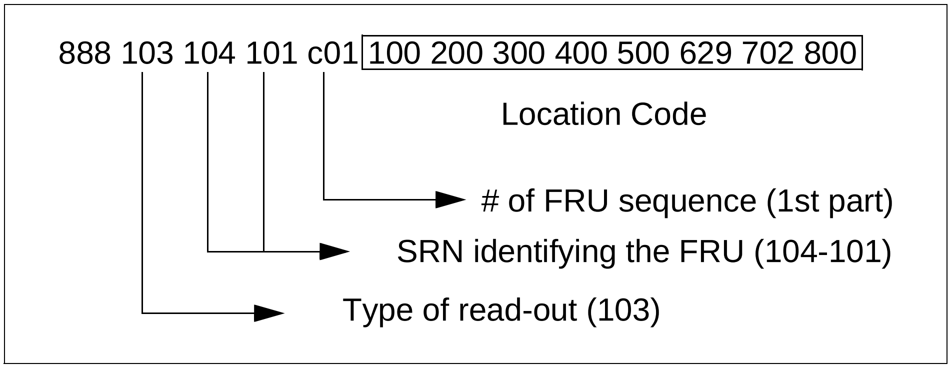

888 103 104 101 c01 100 200 300 400 500 629 702 800
Location Code
# of FRU sequence (1st part)
SRN identifying the FRU (104-101)
Type of read-out (103)

Figure 15. Format of the 103 code message

The 5th value identifies the FRU number (number of defect part) because more than one part could be described in the `888` message. The next eight identifiers describe the location code of the defect part. These shoud be mapped with the Table 13., “Location code mapping table” on page 104 to identify the location code.

Table 13. Location code mapping table

00 = 0	09 = 9	19 = I	28 = S
01 = 1	11 = A	20 = J	30 = T
02 = 2	12 = B	21 = K	31 = U
03 = 3	13 = C	22 = L	32 = V
04 = 4	14 = D	23 = M	33 = W
05 = 5	15 = E	24 = N	34 = X
06 = 6	16 = F	25 = O	35 = Y
07 = 7	17 = G	26 = P	36 = Z
08 = 8	18 = H	27 = R	

5.5 Commands

For a complete reference of the following command use the *AIX Version 4.3 Command Reference* or the online man pages.

5.5.1 chdev

Changes the characteristics of a device. The command has the following syntax:

```
chdev -l Name [ -a Attribute=Value ... ]
```

Table 14. Commonly used flags of the chdev command

Flag	Description
-l *Name*	Specifies the device logical name, specified by the Name parameter, in the Customized Devices object class whose characteristics are to be changed.
-a *Attribute*=*Value*	Specifies the device attribute value pairs used for changing specific attribute values.

5.5.2 lsattr

Displays attribute characteristics and possible values of attributes for devices in the system. The command has the following syntax:

```
lsattr -E -l Name [ -a Attribute ] ...
```

Table 15. Commonly used flags of the lsattr command

Flag	Description
-E	Displays the attribute names, current values, descriptions, and user-settable flag values for a specific device.
-l *Name*	Specifies the device logical name in the Customized Devices object class whose attribute names or values are to be displayed.
-a *Attribute*	Displays information for the specified attributes of a specific device or kind of device.

5.6 References

The following publications contain more information about network tuning procedures.

- *Problem Solving and Troubleshooting in AIX Version 4.3*, SG24-5496
- *AIX Version 4.3 System Management Concepts: Operating System and Devices*, SC23-4126
- *AIX Versions 3.2 and 4 Performance Tuning Guide*, SC23-2365
- *AIX Version 4.3 Commands Reference, Volume 3*, SC23-4117
- *AIX Version 4.3 Commands Reference, Volume 4*, SC23-4118
- *AIX Version 4.3 Messages Guide and Reference*, SC23-4129

5.7 Quiz

5.7.1 Answers

5.8 Exercises

1. Make hardware inventory of your system.
2. Check all possible menu in concurrent mode diagnostics.

Chapter 6. System dumps

In this chapter the system dump will be discussed and how that dump is managed and read. The way to set up the dump device will also be discussed.

A system dump is created when the system has an unexpected system halt or a system failure. The dump will only be a snapshot of the system at the time of the dump, it does not collect data about what lead to the system dump. This dump is written to the primary dump device and if this is not available it will write the dump info to the secondary device. A system dump can also be initiated by a user and then a different device can be initiated.

6.1 Configuring the dump device

Versions prior to AIX 4.1 set up the default dump device as /dev/hd7, in AIX versions after 4.1 the default dump device is /dev/hd6 which is the default paging space logical volume. The secondary dump device is /dev/sysdumpnull.Once the system is booted this image is copied from /dev/hd6 to the directory /var/adm/ras.

The current dump configuration can be determined by running the `sysdumpdev` command as follows:

```
# sysdumpdev

primary             /dev/hd6
secondary           /dev/sysdumpnull
copy directory      /var/adm/ras
forced copy flag    TRUE
always allow dump   FALSE
dump compression    OFF
```

You can use a logical volume outside the root volume group, if it is not a permanent dump device. The primary dump devices must always be in the root volume group for permanent dump devices. The secondary device may be outside the root volume group unless it is a paging space.

Note

Do not use a mirrored, or copied, logical volume as the active dump device. System dump error messages will not be displayed, and any subsequent dumps to a mirrored logical volume will fail.

Do not use a diskette drive as your dump device.

AIX Version 4.2.1 or later supports using any paging device in the root volume group (rootvg) as the secondary dump device

The `sysdumpdev` command can also be used to configure remote dump devices.

The following conditions must be met before a remote dump device can be configured:

- The local and the remote host must have Transmission Control Protocol/Internet Protocol (TCP/IP) installed and configured.
- The local host must have Network File System (NFS) installed.
- The remote host must support NFS.
- The remote host must be operational and on the network. This condition can be tested by issuing the ping command.
- The remote host must have an NFS exported directory defined such that the local host has read and write permissions as well as root access to the dump file on the remote host.
- The remote host cannot be the same as the local host.

To change a primary dump device permanently use the `sysdumpdev` command as follows:

```
# sysdumpdev -P -p /dev/hd3
primary              /dev/hd3
secondary            /dev/sysdumpnull
copy directory       /var/adm/ras
forced copy flag     TRUE
always allow dump    FALSE
dump compression     OFF
```

This will be the permanent dump device until it is changed with the `sysdumpdev` command.

To change the secondary device permanently the sysdumpdev command is used as follows:

```
# sysdumpdev -P -s /dev/rmt0
primary              /dev/hd3
secondary            /dev/rmt0
copy directory       /var/adm/ras
forced copy flag     TRUE
always allow dump    FALSE
dump compression     OFF
```

To change the primary device temporarily to another device the sysdumpdev command is used as follows:

```
# sysdumpdev -p /dev/rmt0
primary              /dev/rmt0
secondary            /dev/sysdumpnull
copy directory       /var/adm/ras
forced copy flag     TRUE
always allow dump    FALSE
dump compression     OFF
```

This will change the primary dump device to /dev/rmt0 until the next system reboot.

6.2 Starting a system dump

A user-initiated dump is different from a dump initiated by an unexpected system halt because the user can designate which dump device to use. When the system halts unexpectedly, a system dump is initiated automatically to the primary dump device. Do not start a system dump if the flashing **888** number shows in your operator panel display. This number indicates your system has already created a system dump and written the information to your primary dump device. If you start your own dump before copying the information in your dump device, your new dump will overwrite the existing information.

You can start a system dump by using one of the methods listed below.

If you have the Software Service Aids Package installed you have access to the sysdumpstart command and can start a dump using one of these methods:

- Using the Command Line
- Using SMIT

If you do not have the Software Services Aids Package installed, you must use one of these methods to start a dump:

- Using the Reset Button
- Using Special Key Sequences

6.2.1 Using the command line

To create a system dump use the following steps to choose a dump device, initiate the system dump, and determine the status of the system dump:

Check which dump device is appropriate for your system (the primary or secondary device) by using the following `sysdumpdev` command:

```
# sysdumpdev -l
```

```
primary             /dev/hd6
secondary           /dev/sysdumpnull
copy directory      /var/adm/ras
forced copy flag    TRUE
always allow dump   FALSE
dump compression    OFF
```

This command lists the current dump devices. You can use the `sysdumpdev` command to change device assignments.

Start the system dump by entering the following `sysdumpstart` command:

```
# sysdumpstart -p
```

This command starts a system dump on the default primary dump device. You can use the -s flag to specify the secondary dump device. If a code shows in the operator panel display, refer to “System dump status check” on page 115.

If the dump was successful reboot the system and during the boot process, if the forced copy flag is set to TRUE, a menu will be displayed on the primary console requesting the removable media to copy the dump to. The options are /dev/rmtx and /dev/fd0. The size of the dump in /dev/hd6 is also displayed. It is advisable not to use /dev/fd0 for the copy of the dump. Once the copy has been completed exit the copy screen and the system will continue the boot process.

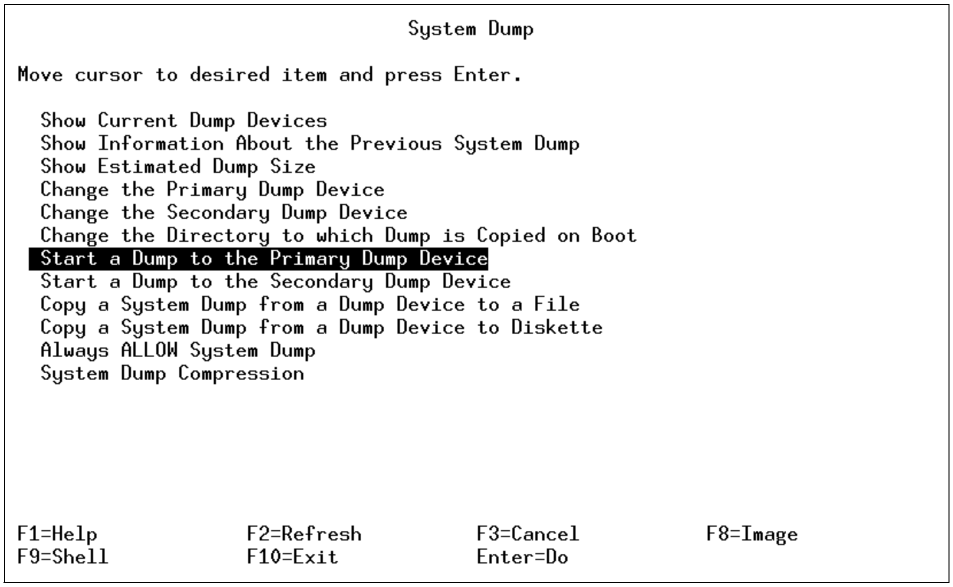
6.2.2 Using the SMIT interface

Use the following SMIT commands to choose a dump device and start the system dump:

```
# smit dump
```

The Choose the Show Current Dump Devices option can be used to note the available dump devices.

Select either the primary or secondary dump device to hold your dump information as shown in Figure 16



```
                                  System Dump

Move cursor to desired item and press Enter.

  Show Current Dump Devices
  Show Information About the Previous System Dump
  Show Estimated Dump Size
  Change the Primary Dump Device
  Change the Secondary Dump Device
  Change the Directory to which Dump is Copied on Boot
  Start a Dump to the Primary Dump Device
  Start a Dump to the Secondary Dump Device
  Copy a System Dump from a Dump Device to a File
  Copy a System Dump from a Dump Device to Diskette
  Always ALLOW System Dump
  System Dump Compression







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

Figure 16. SMIT dump screen

A command status screen will be displayed and once the dump has completed the system will need to be reset.

If the dump was successful reboot the system and during the boot process, if the forced copy flag is set to TRUE, a menu will be displayed on the primary console requesting the removable media to copy the dump too. The options are /dev/rmtx and /dev/fd0. The size of the dump in /dev/hd6 is also displayed. It is advisable not to use /dev/fd0 for the copy of the dump. Once the copy has been completed exit the copy screen and the system will continue the boot process.

6.2.3 Using the reset button

To start a dump with the reset button the key switch must be in the Service position, if the system does not have a key switch set the Always Allow

System Dump value to true. To set this use the `sysdumpdev` command as follows:

```
# sysdumpdev -K
```

The value can be checked using the `sysdumpdev` command without flags as follows:

```
# sysdumpdev
primary              /dev/hd6
secondary            /dev/sysdumpnull
copy directory       /var/adm/ras
forced copy flag     TRUE
always allow dump    TRUE
dump compression     OFF
```

To get the system dump press the reset button. This will initiate the system dump and may take some time.

If the dump was successful reboot the system and during the boot process, if the forced copy flag is set to TRUE, a menu will be displayed on the primary console requesting the removable media to copy the dump to. The options are /dev/rmtx and /dev/fd0. The size of the dump in /dev/hd6 is also displayed. It is advisable not to use /dev/fd0 for the copy of the dump. Once the copy has been completed exit the copy screen and the system will continue the boot process.

If the system does not have a key switch to set the Always Allow System Dump value to back to false use the `sysdumpdev` command as follows:

```
# sysdumpdev -k
```

Ensure the system always allow dump option has been set back to FALSE use the `sysdumpdev` command as follows:

```
# sysdumpdev
primary              /dev/hd6
secondary            /dev/sysdumpnull
copy directory       /var/adm/ras
forced copy flag     TRUE
always allow dump    FALSE
dump compression     OFF
```

6.2.4 Using special key sequences

To start a dump with a key sequence you must have the key switch in the Service position, or have set the Always Allow System Dump value to true. To set this use the `sysdumpdev` command as follows:

```
# sysdumpdev -K
```

The value can be checked using the sysdumpdev command without flags as follows:

```
# sysdumpdev

primary              /dev/hd6
secondary            /dev/sysdumpnull
copy directory       /var/adm/ras
forced copy flag     TRUE
always allow dump    TRUE
dump compression     OFF
```

Press the **Ctrl-Alt 1** key sequence to write the dump information to the primary dump device.

Press the **Ctrl-Alt 2** key sequence to write the dump information to the secondary dump device.

Both these key sequences will initiate the system dump and this process may take some time.

If the dump was successful reboot the system and during the boot process if the forced copy flag is set to TRUE a menu will be displayed on the primary console requesting the removable media to copy the dump to. The options are /dev/rmtx and /dev/fd0. The size of the dump in /dev/hd6 is also displayed. It is advisable not to use /dev/fd0 for the copy of the dump. Once the copy has been completed exit the copy screen and the system will continue the boot process.

If the system does not have a key switch to set the Always Allow System Dump value to back to false use the sysdumpdev command as follows:

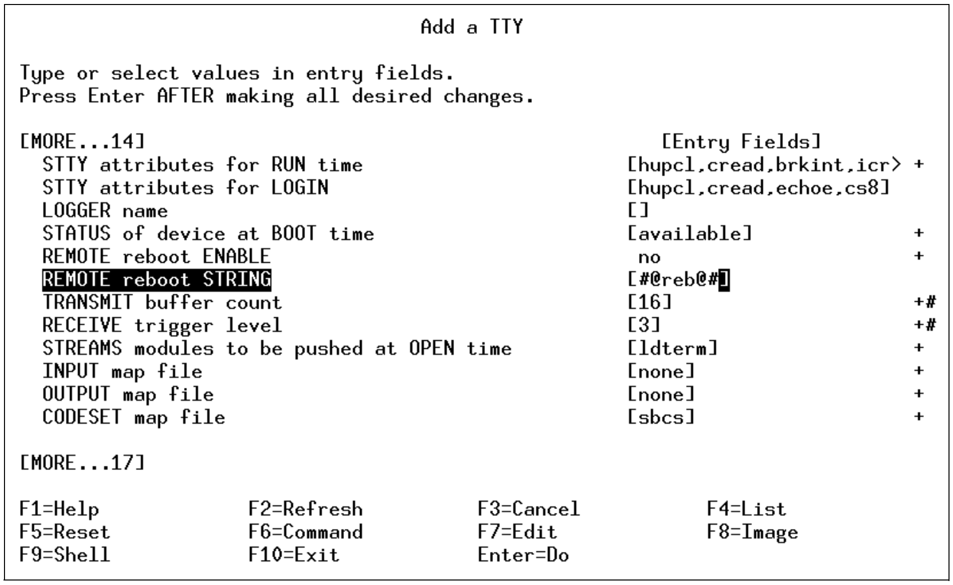
```
# sysdumpdev -k
```

Ensure the system always allow dump option has been set back to FALSE use the sysdumpdev command as follows:

```
# sysdumpdev
primary              /dev/hd6
secondary            /dev/sysdumpnull
copy directory       /var/adm/ras
forced copy flag     TRUE
always allow dump    FALSE
dump compression     OFF
```

6.2.4.1 The TTY remote reboot

From AIX 4.3.2 has added the ability do do a remote reboot of a system across native serial ports only by using a user defined string. This feature is configured by setting up two ODM attributes that have been added to the native serial ports. Figure 17 shows the options as they are set up in the SMIT screen.



```
                              Add a TTY

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

[MORE...14]                                        [Entry Fields]
  STTY attributes for RUN time                  [hupcl,cread,brkint,icr> +
  STTY attributes for LOGIN                     [hupcl,cread,echoe,cs8]
  LOGGER name                                   []
  STATUS of device at BOOT time                 [available]              +
  REMOTE reboot ENABLE                           no                      +
  REMOTE reboot STRING                          [#@reb@#]
  TRANSMIT buffer count                         [16]                    +#
  RECEIVE trigger level                         [3]                     +#
  STREAMS modules to be pushed at OPEN time     [ldterm]                 +
  INPUT map file                                [none]                   +
  OUTPUT map file                               [none]                   +
  CODESET map file                              [sbcs]                   +

[MORE...17]

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

Figure 17. SMIT Add a TTY screen - Remote reboot options

The settings for the **REMOTE reboot ENABLE** attribute are described in Table 16.

Table 16. Remote reboot enable settings

REMOTE reboot Enable settings	Description
no	Remote reboot is disabled and no action will be taken if the reboot string is entered.
reboot	If the reboot string is entred the system will reboot.
dump	If the rebbot string is entered the system will execute a system dump.

The **REMOTE reboot STRING** option is a user defined string that can be used used to perform the function as set up in the **REMOTE reboot ENABLE** option. For more information see *AIX Version 4.3 Differences Guide SG24-2014-02*.

6.3 System dump status check

When a system dump is taking place, status and completion codes are displayed in the operator panel display on the operator panel. When the dump is complete, a 0cx status code displays if the dump was user initiated, a flashing 888 displays if the dump was system initiated.

You can check whether the dump was successful, and if not, what caused the dump to fail. If a 0cx is displayed, see “Status codes” on page 115. If a flashing 888 is displayed, refer to the chapter on 888 in the operator panel display in the *AIX Version 4.3 Messages Guide and Reference*.

Note

If the dump fails and upon reboot there is an error log entry with the label DSI_PROC or ISI_PROC , and the Detailed Data area shows an EXVAL of 000 0005, this is probably a paging space I/O error. If the paging space is the dump device or on the same hard drive as the dump device, the dump may have failed due to a problem with the hard drive. Diagnostics should be run against that disk.

6.3.1 Status codes

Below are the list of status codes for the system dump.

000 The kernel debugger is started. If there is an ASCII terminal attached to one of the native serial ports, enter q dump at the debugger prompt (>) on that terminal and then wait for flashing 888 s to appear in the operator panel display. After the flashing 888 appears, go to “Copy a system dump” on page 117 which describes how to check the dump status.

0c0 The dump completed successfully. Go to “Copy a system dump” on page 117.

0c1 An I/O error occurred during the dump.

0c2 A user-requested dump is not finished. Wait at least 1 minute for the dump to complete and for the operator panel display value to change. If the operator panel display value changes, find the new value on this list. If the value does not change, then the dump did not complete due to an unexpected error. Complete the Problem Summary Form, and report the problem to your software service department.

0c4 The dump ran out of space . A partial dump was written to the dump device, but there is not enough space on the dump device to contain the entire dump. To prevent this problem from occurring again, you must increase the size of your dump media. Go to “Increasing the size of the dump device” on page 119.

0c5 The dump failed due to an internal error. Wait at least 1 minute for the dump to complete and for the operator panel display value to change. If the operator panel display value changes, find the new value on the list. If the value does not change, then the dump did not complete due to an unexpected error. Complete the Problem Summary Form and report the problem to your software service department.

0c7 A network dump is in progress, and the host is waiting for the server to respond. The value in the operator panel display should alternate between 0c7 and 0c2 or 0c9. If the value does not change, then the dump did not complete due to an unexpected error. Complete the Problem Summary Form, and report the problem to your software service department.

0c8 The dump device has been disabled. The current system configuration does not designate a device for the requested dump. Enter the sysdumpdev command to configure the dump device.

0c9 A dump started by the system did not complete. Wait at least 1 minute for the dump to complete and for the operator panel display value to change. If the operator panel display value changes, find the new value on the list. If the value does not change, then the dump did not complete due to an unexpected error. Complete the Problem Summary Form and report the problem to your software service department.

0cc (For AIX Version 4.2.1 and later only) An error occurred dumping to the primary device; the dump has switched over to the secondary device. Wait at least 1 minute for the dump to complete and for the three-digit display value to change. If the three-digit display value changes, find the new value on this list. If the value does not change, then the dump did not complete due to an unexpected error. Complete the Problem Summary Form and report the problem to your software service department.

c20 The kernel debugger exited without a request for a system dump. Enter the quit dump subcommand. Read the new three-digit value from the LED display.

6.4 Copy a system dump

If the dump is not copied to an external device during boot it can be copied to the external device using the snap command. The snap command will check for and existing dump on the system and copy it to the tape or if no dump is available on the system it will prompt for the dump to be copied from the external device.

The last system dump can be checked using the sysdumpdev command as follows:

```
# sysdumpdev -L

0453-039

Device name:         /dev/hd6
Major device number: 10
Minor device number: 2
Size:                42568192 bytes
Date/Time:           Wed Jul 12 14:53:55 CDT 2000
Dump status:         0
dump completed successfully
Dump copy filename: /usr/dumpdir/vmcore.0
```

In this case, the dump was successfully completed and it can be copied to an external media device such as tape.

Use the snap command as follows to create copy the dump to tape:

```
# snap -gfkD -o /dev/rmt0

Setting output device to /dev/rmt0... done.
Checking space requirement for general
information..............................
................................................... done.
Checking space requirement for kernel information.......... done.
Checking space requirement for dump information...... done.
Checking space requirement for filesys information........................
done.
Checking for enough free space in filesystem... done.

********Checking and initializing directory structure
Directory /tmp/ibmsupt/filesys already exists... skipping
Directory /tmp/ibmsupt/dump already exists... skipping
Directory /tmp/ibmsupt/kernel already exists... skipping
Directory /tmp/ibmsupt/general already exists... skipping
Directory /tmp/ibmsupt/testcase already exists... skipping
Directory /tmp/ibmsupt/other already exists... skipping
********Finished setting up directory /tmp/ibmsupt

Gathering general system
information............................................
..................................... done.
Gathering kernel system information........... done.
Gathering dump system information.... done.

WARNING: The dump and /unix file were not copied.
The /unix file does not match the latest dump on your
system.  The /unix must be the same unix, or linked to
the same unix, that was running when the dump occurred.
Possible Causes:
1) The /unix file does not exist.
2) The dump that was taken is a partial dump.
3) The /unix file on your system was replaced,
   and the bosboot command was not run.  If this is the
   case, then run the bosboot command and reboot your
   system.

Gathering filesys system information........................ done.

Copying information to /dev/rmt0... Please wait... done.

****************************************************************
******
****** Please Write-Protect the output device now...
******
****************************************************************

****************************************************************
******
****** Please label your tape(s) as follows:
****** snap                    blocksize=512
****** problem: xxxxx          Wed Jul 12 15:41:42 CDT 2000
****** 'your name or company's name here'
******
****************************************************************
```

The dumpfile can be copied from the external device using the `tar -x` command. To view the contents of the tape device use the following command:

```
# tar -tvf /dev/rmt0

drwx------   0 0        0 Jul 12 13:48:44 2000 ./dump/
-rw-------   0 0     2555 Jul 12 15:40:21 2000 ./dump/dump.snap
-rw-------   0 0  1770955 Jul 12 13:48:29 2000 ./dump/unix.Z
-rwx------   0 0  41761792 Jul 12 11:03:29 2000 ./dump/dump_file
...
drwx------   0 0        0 Jul 12 11:23:06 2000 ./kernel/
-rw-------   0 0    75122 Jul 12 15:40:21 2000 ./kernel/kernel.snap
drwx------   0 0        0 Jul 12 11:22:58 2000 ./testcase/
drwx------   0 0        0 Jul 12 11:22:58 2000 ./other/
```

The files `dump.snap`, `unix.Z` and `dump_file` should exist on the tape device and must be greater than 0 bytes in size.

6.5 Increasing the size of the 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.

For example, enter:

```
# sysdumpdev -e

0453-041 Estimated dump size in bytes: 38797312
```

If the dump device is the default dump device of /dev/hd6 run the `lsps -a` command to check paging space available as follows:

```
# lsps -a

Page Space  Physical Volume   Volume Group    Size   %Used  Active  Auto  Type
hd6         hdisk0            rootvg         512MB       1     yes   yes    lv
```

If the size of the dump device needs to be increased use the `smit chps` command and change the paging space size. If the dump device is a file ensure that the filesystem has enough space, if not use the `smit chfs` command to increase the size of the file system.

6.6 Reading dumps

To check that the dump is readable, start the `crash` command on the dump files, using the command syntax: `crash <dump> <unix>`. The `crash` command needs a kernel file (unix) to match the dump file. If you do not specify a kernel file, crash uses the file /unix by default:

```
# crash dump unix
>
```

If you do not see any message from `crash` about dump routines failing, you probably have a valid dump file. Then, run the stat subcommand at the > prompt. For example:

```
# crash dump unix
> stat
        sysname: AIX
        nodename: sp5i
        release: 3
        version: 4
        machine: 000126774C00
        time of crash: Tue May  4 04:56:10 CDT 1999
        age of system: 4 min.
        xmalloc debug: disabled
        abend code: 300
        csa: 0x2ff3b400
        exception struct:
                dar:   0x00000003
                dsisr: 0x00000000:
                srv:   0x04000000
                dar2:  0x3c160040
                dsirr: 0x06001000: "(unknown reason code)"
```

Look at the time of the dump and the abend code. If these are reasonable for the dump, then perform some initial analysis. Refer to Section 6.8, “The crash command” on page 123 for more information.

A message stating `dumpfile does not appear to match namelist` means the dump is not valid. For example:

```
# crash dump unix
Cannot locate offset 0x02052b8 in segment 0x000000.
endcomm 0x00000000/0x011c5e70
WARNING: dumpfile does not appear to match namelist
```

```
Cannot locate offset 0x00ccf10 in segment 0x000000.
0452-179: Cannot read v structure from address 0x   ccf10.
Symbol proc has null value.
Symbol thread has null value.
Cannot locate offset 0x00ccf10 in segment 0x000000.
0452-179: Cannot read v structure from address 0x   ccf10.
Cannot locate offset 0x00034c4 in segment 0x000000.
0452-1002: Cannot read extension segment value from address 0x    34c4
```

Any other messages displayed when starting crash may indicate that certain components of the dump are invalid, but these are generally handled by crash. If a required component of the dump image is missing, additional messages will indicate this, and the dump should be considered invalid.

6.7 Core dumps

When a system encounters a core dump a core file is created in the current directory when various errors occur. Errors such as memory-address violations, illegal instructions, bus errors, and user-generated quit signals commonly cause this core dump. The core file that is created contains a memory image of the terminated process. A process with a saved user ID that differs from the real user ID does not produce a memory image.

6.7.1 Checking for core dump

When a core dump is created an error will be reported and this entry can be seen in the error report as follows:

```
# errpt
IDENTIFIER TIMESTAMP  T C RESOURCE_NAME  DESCRIPTION
...
C60BB505   0705101400 P S SYSPROC        SOFTWARE PROGRAM ABNORMALLY
TERMINATED
...
```

From the above report it can be seen that the error has an identifier of `C60BB505`, a detailed report of the error can be displayed as follows:

```
# errpt -a -j C60BB505
---------------------------------------------------------------------------
LABEL:          CORE_DUMP
IDENTIFIER:     C60BB505

Date/Time:       Wed Jul  5 10:14:59
Sequence Number: 8
Machine Id:      000BC6DD4C00
```

```
Node Id:         client1
Class:           S
Type:            PERM
Resource Name:   SYSPROC

Description
SOFTWARE PROGRAM ABNORMALLY TERMINATED

Probable Causes
SOFTWARE PROGRAM

User Causes
USER GENERATED SIGNAL

        Recommended Actions
        CORRECT THEN RETRY

Failure Causes
SOFTWARE PROGRAM

        Recommended Actions
        RERUN THE APPLICATION PROGRAM
        IF PROBLEM PERSISTS THEN DO THE FOLLOWING
        CONTACT APPROPRIATE SERVICE REPRESENTATIVE

Detail Data
SIGNAL NUMBER
           4
USER'S PROCESS ID:
       15394
FILE SYSTEM SERIAL NUMBER
           5
INODE NUMBER
           2
PROGRAM NAME
netscape_aix4
ADDITIONAL INFORMATION
Unable to generate symptom string.
Too many stack elements.
```

In the above output it can be seen that the program that created the core dump was netscape_aix4. See the section “Locating a core dump” on page 123 to determine where the core file is located.

6.7.2 Locating a core dump

When a system does a core dump it writes a file named `core`. This file may be written anywhere in the system and it will need to be found using the `find` command as follows:

```
# find / -name core -ls
  737 10188 -rw-r--r--  1 root     system   10430807 Jul  5 10:14 /core
```

From the above it can be noted that the file is located in the root directory.

6.7.3 Determining the program that caused the core dump

There are two ways to determine which program caused the core dump, one is using the `strings` command and the other is using the `lquerypv` command. Although this information should be in the error report there may be occasion when the error report is not available or has been cleared out.

The strings command will give the full path name of the program and is used as follows:

```
# strings core | grep _=
_=/usr/netscape/communicator/us/netscape_aix4
```

The `lquerypv` command is run as follows:

```
# lquerypv -h core 6b0 64
000006B0   7FFFFFFF FFFFFFFF 7FFFFFFF FFFFFFFF  |................|
000006C0   00000000 000007D0 7FFFFFFF FFFFFFFF  |................|
000006D0   00120000 137084E0 00000000 00000016  |.....p..........|
000006E0   6E657473 63617065 5F616978 34000000  |netscape_aix4...|
000006F0   00000000 00000000 00000000 00000000  |................|
00000700   00000000 00000000 00000000 0000085E  |...............^|
00000710   00000000 00000F5A 00000000 00000776  |.......Z.......v|
```

It can be seen that the file was dumped by the `netscape_aix4` program as displayed in the error report.

6.8 The crash command

This section allows you to recognize some common problems using the `crash` command, and to make a basic determination as to what caused the problem.

6.8.1 Uses of crash

The `crash` command can be used on a running system. Invoking `crash` with no parameters essentially allows you to view the memory and state of the currently running system by examining /dev/mem. The alter subcommand in `crash` allows you to modify the running kernel. This should only be used under the direction of IBM support, since incorrect use can cause the system to fail. The user must be in the system group to run `crash` on the live system.

The `crash` can also be used on a system dump. It is the primary tool used to analyze a dump resulting from a system failure. Invoking `crash` with a parameter specifying a dumpfile allows you to examine a dumpfile for problem analysis.

Using crash, you can examine:

- Addresses and symbols
- Kernel stack traceback
- Kernel extensions
- The process table
- The thread table
- The file table
- The inode table

In addition to the items listed above, you can use `crash` to look at anything else contained in the kernel memory.

6.8.2 What is the kernel?

The kernel is the program that controls and protects system resources. It runs in privileged mode. It operates directly with the hardware. The major functions of the kernel are:

- Creation and deletion of processes/threads
- CPU scheduling
- Memory management
- Device management
- Provides synchronization and communication tools for processes

If the kernel has an error, the machine will crash. A user program will only create a core dump and halt.

The `crash` command is used to debug these kernel problems.

6.8.3 Examining a system dump

The `crash` command needs a kernel /unix file to match the dump file under analysis. For example:

```
itsosrv1:/dumptest> crash dumpfile unix
>
```

If no kernel file is specified, the default is /unix.

```
itsosrv1:/dumptest> crash dumpfile
Using /unix as the default namelist file.
>
```

The `crash` command uses the kernel file to interpret symbols and allows for symbolic translation and presentation. If the kernel file does not match the dump, you will get an error message when you start `crash`.

6.8.4 Basic crash subcommands

Once you initiate the `crash` command, the prompt character is the greater than sign (**>**). For a list of the available subcommands, type the **?** character. To exit, type **q**. You can run any shell command from within the crash command by preceding it with an exclamation mark (**!**).

Please refer to the online documentation of *AIX Version 4.3 Kernel Extensions and Device Support Programming Concepts* for more information of the crash utility and all crash subcommands.

- stat

  Shows dump statistics.
- proc [-] [-r] [processTableEntry]

  Displays the process table (proc.h). Alias p and ps.
- user [ProcessTableEntry]

  Displays user structure of named process (user.h). Alias u.
- thread [-] [-r] [-p] [threadTableEntry]

  Displays the thread table (thread.h).
- mst [addr]

  Displays the mstsave portion of uthread structure (uthread.h, mstsave.h).
- ds [addr]

  Finds the data symbol closest to the given address.
- knlist [symbol]

Displays address of symbol name given. Opposite of ds.

- trace [-k][-m][-r][ThreadTableEntry]

  Displays kernel stack trace. Alias t.

- le

  Displays loader entries.

- nm [symbol]

  Displays symbol value and type as found in the /unix file.

- od [symbol name or addr] [count] [format]

  Dumps count number of data words starting at symbol name or addr in the format specified by format.

- ? or help[]

  Lists all subcommands.

  Provides information about crash subcommands.

- cm [thread slot][seg_no]

  Changes the map of the `crash` command internal pointers for any process thread segment not paged out. Resets the map of internal pointers if no parameters are used.

- fs [thread slotNumber]

  Dumps the kernel stack frames for the specified thread.

- dlock [tid] **|** -p [processor_num]

  Displays deadlock information about all types of locks: simple, complex, and lockl.

- errpt [count]

  Displays error log messages. The errpt subcommand always prints all messages that have not yet been read by the errdemon. Count specifies the number of messages to print.

- du

  Dump user area of process.

- ppd

  Display per processor data area, useful for multiprocessor systems. Shows all data that varies for each processor, such as Current Save Area (CSA).

6.8.4.1 stat subcommand

The stat subcommand gives plenty of useful information about a dump, such as the dump code, the panic string, time of the crash, version and release of the operating system, name of the machine that crashed, and how long the machine had been running since the last crash or power off of the system. For example:

```
> stat
        sysname: AIX
        nodename: kmdvs
        release: 3
        version: 4
        machine: 000939434C00
        time of crash: Mon May  3 17:49:46 KORST 1999
        age of system: 2 day, 4 hr., 28 min.
        xmalloc debug: disabled
        dump code: 700
        csa: 0x384eb0
        exception struct:
                0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
        panic: HACMP for AIX dms timeout - ha
```

The stat subcommand should always be the first command run when examining a system crash.

6.8.4.2 trace -m subcommand

The trace -m subcommand gives you a kernel stack traceback.

This is typically the second command you will run when examining a system dump.

This subcommand gives you information on what was happening in the kernel when the crash occurred. The trace -m subcommand gives you a history of function calls and what interrupt processing was going on in the system. If the crash occurred while interrupt processing was going on, this is the command to use. This command traces the linked list of mstsave areas. The mstsave areas basically contain a history of what interrupt processing was going on in the system.

The machine state save area, or MST, contains a saved image of the machine’s process context. The process context includes the general purpose and floating point registers, the special purpose registers, and other information necessary to restart a thread when it is dispatched. For example:

```
> trace -m
Skipping first MST
```

```
MST STACK TRACE:
0x002baeb0 (excpt=00000000:00000000:00000000:00000000:00000000) (intpri=3)
        IAR:      .[atmle_dd:atmle_ready_ind]+d8 (01b05cb0): tweqi   r5,0x0
        LR:       .[atmle_dd:atmle_ready_ind]+34 (01b05c0c)
        002ba940: .[atmle_dd:atmle_receive_ether_data]+1ec (01b0c35c)
        002ba9a0: .[atm_demux:atm_dmx_receive]+204 (01adc0e8)
        002baa00: .[atmdd:atm_deqhandler]+1254 (01ac7e6c)
        002babc0: .[atmdd:atm_HandleCardRsp]+1a4 (01aba084)
        002baca0: .[atmdd:atm_handler]+48 (01aba350)
        002bad40: .[atmdd:atm_intr]+ac (01ac4a04)
        002bad90: .i_poll_soft+9c (0001ef84)
        002badf0: .i_softmod+c8 (0001e964)
        002bae70: flih_603_patch+c0 (0000bb9c)

0x2ff3b400 (excpt=00000000:00000000:00000000:00000000:00000000)(intpri=11)
        IAR:      .waitproc+c0 (0000edb0):     lwz   r3,0x6c(r28)
        LR:       .waitproc+d4 (0000edc4)
        2ff3b388: .procentry+14 (00045414)
        2ff3b3c8: .low+0 (00000000)
```

In this example, there are two levels of stack traceback. The first level shows the Instruction Address Register (IAR), pointing to a trap instruction, `tweqi r5, 0x0`.

IAR - Instruction Address Register. It has a address which caused the crash.
LR - Link Register who called the fatal function or where last call returns to.

This trap instruction is what you will see when you get a crash of type Program Interrupt, or Dump Status = 700. This was probably the result of assert or panic. It can be seen that the interrupt priority is 3 (`intpri=3`). In this case, it can be seen that interrupt processing was occurring when the crash happened because the interrupt priority was less than 11 or 0xB. The base interrupt priority is indicated by 0xB or 11. This is the level at which a normal process runs.

When looking at a stack traceback, realize that the first thing on the stack was the most recently running function, which was called by the function below it, which was called by the function below it, and so on. So, in the case of the middle stack traceback in our example, it can be seen that `i_softmod` called `i_poll_soft`, which called some functions in `atmdd` and `atm_demux` module, which called `atmle_receive_ether_data`, which called `atmle_ready_ind`, and an assert was hit in `atmle_ready_ind`. Look at the code for this to try to find out the cause of the assert action. The `atmle_dd module` did something wrong.

Make sure the failing module is at the latest version. Problems are frequently resolved in later versions of software. You can use the le subcommand in crash and the `lslpp -w` command to find the fileset that contains the specific module.

Use the le subcommand with an argument of the address listed in the IAR of the topmost MST area. The address is displayed in brackets after the name of the module. For example:

```
> le 01b05cb0
LoadList entry at 0x04db7780
  Module start:0x00000000_01b016e0  Module filesize:0x00000000_00030fbc
  Module *end:0x00000000_01b3269c
  *data:0x00000000_0125ef40  data length:0x00000000_0000375c
  Use-count:0x000c  load_count:0x0001  *file:0x00000000
  flags:0x00000272 TEXT KERNELEX DATAINTEXT DATA DATAEXISTS
  *exp:0x04e0e000  *lex:0x00000000  *deferred:0x00000000
*expsize:0x69626f64
  Name: /usr/lib/drivers/atmle_dd
  ndepend:0x0001  maxdepend:0x0001
  *depend[00]:0x04db7580
  le_next:  04db7380
```

One of the fields listed by the le subcommand is the `Name` of the module. You can then use the `lslpp -w` command to determine the fileset that contains the module. For example:

```
itsosrv1:/> lslpp -w /usr/lib/drivers/atmle_dd
  File                                          Fileset               Type
--------------------------------------------------------------------------
  /usr/lib/drivers/atmle_dd                     bos.atm.atmle         File
```

This command is available in AIX Version 4.2 or later.

Looking at the line:

```
002ba940: .[atmle_dd:atmle_receive_ether_data]+1ec (01b0c35c)
```

It can be seen in the first column the address of the entry on the stack. The last column contains the return address of the code (`01b0c35c`). This address corresponds to the function shown, `atmle_receive_ether_data`, which is contained in the module `atmle_dd`. The square brackets around the [module:function] pair indicate that this is a kernel extension. In addition, the instruction at this return address is at offset `1ec` from the beginning of the module `atmle_dd`.

The last of the stack trace backs indicates the user level process (intpri=b) and the running process is `wait`. If the user subcommand is run, it will be seen that the running process is `wait`. However, `wait` did not cause the problem here, the problem was caused by a program running at interrupt level, and looking at the MST stack traceback is the only way to see the real problem.

When a Data Storage Interrupt (DSI) with dump code 300 occurs, the exception structure is filled in as follows:

```
0x2ff3b400 (excpt=DAR:DSISR:SRV:DAR2:DSIRR) (intpri=?)
```

The exception structure shows various machine registers and the interrupt level. The registers shown in the exception structure are defined as follows:

DAR	Data Address Register
DSISR	Data Storage Interrupt Status Register
SRV	Segment Register Value
DAR2	Secondary Data Address Register
DSIRR	Data Storage Interrupt Reason Register

The interrupt priority of the running context is shown in the (intpri=?) field at the end of the line. The intpri value ranges from 0xb (INTBASE) to 0x0 (INTMAX).

The exception structure is not used for code 700 dumps.

6.8.4.3 The proc subcommand

The proc subcommand displays entries in the process table. The process table is made up of entries of type struct proc, one per active process. Entries in the process table are pinned so that they are always resident in physical memory. The process table contains information needed when the process has been swapped out in order to get it running again at some point in the future. For example:

```
> proc - 0
SLT ST    PID   PPID   PGRP   UID  EUID  TCNT  NAME
  0 a       0      0      0     0     0     1  swapper
        FLAGS: swapped_in no_swap fixed_pri kproc

Links:  *child:0xe3000170  *siblings:0x00000000  *uidl:0xe3001fa0
    *ganchor:0x00000000  *pgrpl:0x00000000  *ttyl:0x00000000
Dispatch Fields:  pevent:0x00000000  *synch:0xffffffff
    lock:0x00000000  lock_d:0x01390000
Thread Fields:  *threadlist:0xe6000000  threadcount:1
    active:1  suspended:0  local:0   terminating:0
```

```
Scheduler Fields:   fixed pri: 16  repage:0x00000000  scount:0  sched_pri:0
    *sched_next:0x00000000  *sched_back:0x00000000 cpticks:0
    msgcnt:0    majfltsec:0
Misc:  adspace:0x0001e00f  kstackseg:0x00000000  xstat:0x0000
    *p_ipc:0x00000000  *p_dblist:0x00000000  *p_dbnext:0x00000000
Signal Information:
    pending:hi 0x00000000,lo 0x00000000
    sigcatch:hi 0x00000000,lo 0x00000000 sigignore:hi 0xffffffff,lo
0xfff7ffff
Statistics:  size:0x00000000(pages)  audit:0x00000000
    accounting page frames:0   page space blocks:0

    pctcpu:0    minflt:1802    majflt:7
```

The fields in the first few lines of the output are as follows:

SLT This is the process slot number, and simply indicates the process's position in the process table. Use this number to tell the `crash` command which specific process block or u-block to display. Note that the slot numbers are in decimal.

ST This is a 1-character field indicating the status of the process, and may be a=active, i=idle, t=stopped, or z=zombie.

PID This is the actual process `ID` by which the process is known to the system. The process slot number is used to generate the process ID.

PPID Parent process ID.

PGRP Process group ID.

UID User ID.

EUID Effective user ID.

TCNT Thread count.

NAME Program name.

FLAGS Status flags.

6.8.4.4 The thread subcommand

The thread table contains per-thread information that can be used by other threads in a process. There is one structure allocated per active thread. Entries that are in use are pinned to avoid page faults in kernel critical sections. For example:

```
> thread - 0
SLT ST    TID      PID    CPUID  POLICY PRI CPU    EVENT  PROCNAME
  0 s       3        0  unbound    FIFO  10  78            swapper
```

```
        t_flags:  wakeonsig kthread

Links:  *procp:0xe3000000  *uthreadp:0x2ff3b400  *userp:0x2ff3b6e0
    *prevthread:0xe6000000  *nextthread:0xe6000000,  *stackp:0x00000000
    *wchan1(real):0x00000000  *wchan2(VMM):0x00000000 *swchan:0x00000000
    wchan1sid:0x00000000  wchan1offset:0x00000000
    pevent:0x00000000  wevent:0x00000001  *slist:0x00000000
Dispatch Fields:  *prior:0xe6000000  *next:0xe6000000
    polevel:0x0000000a  ticks:0x0139  *synch:0xffffffff  result:0x00000000
    *eventlst:0x00000000  *wchan(hashed):0x00000000  suspend:0x0001
    thread waiting for:  event(s)
Scheduler Fields:  cpuid:0xffffffff  scpuid:0xffffffff  pri: 16
policy:FIFO
    affinity:0x0003  cpu:0x0078    lpri:  0  wpri:127    time:0x00
sav_pri:0x10
Misc:  lockcount:0x00000000  ulock:0x00000000  *graphics:0x00000000
    dispct:0x000000e4  fpuct:0x00000001  boosted:0x0000
    userdata:0x00000000
Signal Information:  cursig:0x00  *scp:0x00000000
    pending:hi 0x00000000,lo 0x00000000  sigmask:hi 0x00000000,lo
0x00000000
```

The fields in the output of the thread subcommand are as follows:

SLT Slot number.

ST Status. This may be i=idle, r=running, s=sleeping, w=swapped out, t=stopped, or z=zombie.

TID Thread ID.

PID Process id of the associated process. There may be multiple threads per process, but only one process per thread.

CPUID CPU ID of the CPU running the thread. On a uniprocessor system, this will always be 0.

POLICY This is the scheduling policy used for the thread and may have the values FIFIO, RR, or other.

PRI Dispatch priority. This is not the *nice* value.

CPU CPU utilization. This value is used for scheduling.

PROCNAME The name of the process for this thread.

EVENTS This is the wait channel if not zero.

FLAGS Status flags.

6.8.4.5 Display memory with od

To display and examine memory areas from the dump use the od subcommand. The syntax of the subcommand is as follows:

```
od [symbol name] [count] [format]
```

Formats are ASCII, octal, decimal, hex, byte, character, instruction, long octal, and long decimal. For example:

```
> od vmker 15
000bde48: 00002001 00006003 00000000 00008004
000bde58: 00200000 00000012 0000000d 00000200
000bde68: 00080000 00000017 00078c93 00066320
000bde78: 00000ab2 00020000 00002870

> od 0xbde48 15 a
000bde48: 00002001 00006003 00000000 00008004  |.. ...`.........|
000bde58: 00200000 00000012 0000000d 00000200  |. ..............|
000bde68: 00080000 00000017 00078c93 00066320  |..............c |
000bde78: 00000ab2 00020000 00002870           |..........(p|
```

6.8.4.6 Looking for the error log

To examine the last few error log entries from the dump use the errpt subcommand. For example:

```
> errpt
ERRORS NOT READ BY ERRDEMON (MOST RECENT LAST):
Sun Apr  6 01:01:11 1997 : DSI_PROC data storage interrupt : processor
Resource Name: SYSVMM
42000000 007fffff 80000000 fffffffa
>
```

6.8.4.7 Finding addresses in kernel extensions

The le subcommand can indicate what kernel extension an address belongs to. Take, for example, the address 0x0123cc5c. This is a kernel address, since it starts 0x01, which indicates it is in segment 0, the kernel segment. To find the kernel module that contains the code at this address, use the le subcommand. For example:

```
> le 0123cc5c
LoadList entry at 0x04db7780
  Module start:0x00000000_012316e0  Module filesize:0x00000000_00030fbc
  Module *end:0x00000000_0126269c
  *data:0x00000000_0125ef40  data length:0x00000000_0000375c
  Use-count:0x000c  load_count:0x0001  *file:0x00000000
  flags:0x00000272 TEXT KERNELEX DATAINTEXT DATA DATAEXISTS
  *exp:0x04e0e000  *lex:0x00000000  *deferred:0x00000000
*expsize:0x69626f64
  Name: /usr/lib/drivers/pse/pse
  ndepend:0x0001  maxdepend:0x0001
  *depend[00]:0x04db7580
  le_next:  04db7380
```

In this case, it can be seen that the code at address `0x0123cc5c` is in module `/usr/lib/drivers/pse/pse`.

The le subcommand is only helpful for modules that are already loaded into the kernel.

6.8.4.8 VMM error log

When the Dump Status code indicates a DSI or an ISI, look at the VMM error log. This is done using the od subcommand and looking at the `vmmerrlog` structure. See Table 17. For example:

```
> od vmmerrlog 9 a
000c95b0: 9d035e4d 53595356 4d4d2000 00000000  |..^MSYSVMM .....|
000c95c0: 00000000 0a000000 00000000 0000000b  |................|
000c95d0: 00000086                             |....|
```

Table 17. vmmerrlog structure components

Offset	Meaning
0x14	The Data Storage Interrupt Status Register (DSISR)
0x1C	Faulting address
0x20	VMM return code

In this example, the VMM return code 0x86 means PROTECTION EXCEPTION. The various VMM return codes, symbolic names, and meanings are shown in below:

0000000E This return code indicates an EFAULT. It comes from errno.h (14) and is returned if you attempt to access an invalid address.

FFFFFFFA This return code indicates you tried to access an invalid page that is not in memory. This is usually the result of a page fault. This will be returned if you try to access something that is paged out while interrupts are disabled.

00000005 This is a hardware problem. An I/O error occurred when you tried to either page in or page out, or you tried to access a memory mapped file and could not do it. Check the error log for disk or SCSI errors.

00000086 This return code indicates a protection exception. This means that you tried to store to a location that is protected. This is usually caused by low kernel memory.

0000001C This return code indicates no paging space. This means that the system has exhausted its paging space.

6.8.4.9 The symptom subcommand

The symptom[-e] subcommand displays the symptom string for a dump. It is not valid on a running system. The optional -e option will create an error log entry containing the symptom string, and is normally only used by the system and not entered manually. The symptom string can be used to identify duplicate problems.

6.8.5 Handling crash output

Some crash subcommands generate many more lines than can fit on one screen. Also, crash does not pause its output after each screen full. You will want to have some way of seeing scrolled-off data.

In the past, the `script` or `tee` commands were used for this. For example:

```
tee -a outf | crash /tmp/dump /unix | tee -a outf
```

There is now a new way to obtain a log file by using the `set logfile` subcommand. For example:

```
>set logfile crash.log
```

Once this has been entered, crash starts logging all input and output to the specified file. The `set variable` subcommand is available in AIX Versions 4.1.5, 4.2.1, 4.3, and above.

In addition to the logfile support, command pipeline support was added to crash, allowing you to pipe long output to other commands, such as `more`, `pg`, and `grep`. For example:

```
> le 0123cc5c | grep Name
  Name: /usr/lib/drivers/pse/pse
```

6.8.6 Types of crashes

Common problems requiring crash dump analysis include the following:

6.8.6.1 Kernel panic or trap

This is usually the cause of a system crash with the LED sequence 888-102-700-0cx.

In AIX, kernel panics manifest themselves as traps. The panic() routine in the kernel puts its message into a buffer, writes it to the debug tty using the kernel debug program, and calls brkpoint(). If the kernel debugger is loaded, and an ASCII terminal is connected on a serial port, this will start the debugger; otherwise, it will cause a dump. If a panic or assert occurs, you must examine the source code to understand the condition that caused the panic or assert.

6.8.6.2 Addressing exception or data storage interrupt

This type of crash is accompanied by the LED sequence 888-102-300-0cx.

The 300 in the LED sequence indicates an addressing exception (a Data Storage Interrupt or DSI). This is usually caused by a bad address being accessed, or page fault occurring when interrupts are disabled. When you get this type of crash, check the VMM return code.

6.8.6.3 System hang

A dump can be forced when the system locks up to determine the cause of the hang.

A system hang is a total system lockup. A dump forced by turning the key to the Service position and pressing the Reset button can be examined to see what locks are being held by whom. Refer to “Starting a system dump” on page 109for more information.

6.9 The snap command

The `snap` command gathers system configuration information and compresses the information into a tar file. The file can then be downloaded to disk or tape, or transmitted to a remote system. The information gathered with the `snap` command may be required to identify and resolve system problems.

The `snap` command syntax is as follows:

```
snap [ -a ] [ -A ] [ -b ] [ -c ] [ -D ] [ -f ] [ -g ] [ -G ] [  -i  ] [ -k
] [ -l ] [ -L ][ -n ] [ -N ] [ -p ] [ -r ] [ -s ] [ -S ] [ -t ] [ -o
OutputDevice ] [ -d Dir ] [ -v Component ]
```

The `snap` commands flags are listed in Table 18.

Table 18. The snap command flags

Flag	Description
-a	Gathers all system configuration information. This option requires approximately 8MB of temporary disk space.

Flag	Description
-A	Gathers asynchronous (TTY) information.
-b	Gathers SSA information.
-c	Creates a compressed tar image (snap.tar.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. If bosboot -k was used to specify the running kernel to be other than /unix, the incorrect kernel will be gathered. Make sure that /unix is , or is linked to, the kernel in use when the dump was taken.
-dDir	Identifies the optional snap command output directory (/tmp/ibmsupt is the default).
-f	Gathers file system information.
-g	Gathers the output of the `lslpp -hBc` command, which is required to recreate exact operating system environments. Writes output to the /tmp/ibmsupt/general/lslpp.hBc file. Also collects general system information and writes the output to the /tmp/ibmsupt/general/general.snap file.
-G	Includes predefined Object Data Manager (ODM) files in general information collected with the -g flag.
-i	Gathers installation debug vital product data (VPD) information.
-k	Gathers kernel information
-l	Gathers programming language information.
-L	Gathers LVM information.
-n	Gathers Network File System (NFS) information.
-N	Suppresses the check for free space.
-oOutputDevice	Copies the compressed image onto diskette or tape.
-p	Gathers printer information.
-r	Removes snap command output from the /tmp/ibmsupt directory.
-s	Gathers Systems Network Architecture (SNA) information.
-S	Includes security files in general information collected with the -g flag.

Flag	Description
-t	Gathers Transmission Control Protocol/Internet Protocol (TCP/IP) information.
-vComponent	Displays the output of the commands executed by the snap command. Use this flag to view the specified name or group of files.

6.10 The strings command

The `strings` command looks for printable strings in an object or binary file. A string is any sequence of four or more printable characters that end with a new-line or a null character. The `strings` command is useful for identifying random object files.

The `strings` command syntax is as follows:

```
strings [ -a ] [ - ] [ -o ] [ -t Format ] [ -n Number ] [ -Number ] [ File ]
```

The `strings` commands flags are listed in Table 19.

Table 19. The strings command flags

Flag	Description
-a or -	Searches the entire file, not just the data section, for printable strings.
-n Number	Specifies a minimum string length other than the default of 4 characters. The maximum value of a string length is 4096. This flag is identical to the -Number flag.
-o	Lists each string preceded by its octal offset in the file. This flag is identical to the -t o flag.
-t Format	Lists each string preceded by its offset from the start of the file. The format is dependent on the character used as the Format variable. d Writes the offset in decimal. o Writes the offset in octal. x Writes the offset in hexadecimal. When the -o and the -t Format flags are defined more than once on a command line, the last flag specified controls the behavior of the strings command.
-Number	Specifies a minimum string length other than the default of 4 characters. The maximum value of a string length is 4096. This flag is identical to the -n Number flag.
File	Binary or object file to be searched.

6.11 The sysdumpdev command

The sysdumpdev command changes the primary or secondary dump device designation in a system that is running. The primary and secondary dump devices are designated in a system configuration object. The new device designations are in effect until the sysdumpdev command is run again, or the system is restarted.

If no flags are used with the sysdumpdev command, the dump devices defined in the SWservAt ODM object class are used. The default primary dump device is /dev/hd6. The default secondary dump device is /dev/sysdumpnull.

Note

Do not use a mirrored, or copied, logical volume as the active dump device. System dump error messages will not be displayed, and any subsequent dumps to a mirrored logical volume will fail.

Do not use a diskette drive as your dump device.

If you use a paging device, only use hd6, the primary paging device. AIX Version 4.2.1 or later supports using any paging device in the root volume group (rootvg) as the secondary dump device.

The sysdumpdev command syntax is as follows:

```
sysdumpdev [-c | -C] -P { -p Device | -s Device } [ -q ]

sysdumpdev [-c | -C] [ -p Device | -s Device] [ -q ]

sysdumpdev [-c | -C] [ -d Directory | -D Directory | -e | [ -k | -K ] | -l
| -L | -p Device | -q | -r Host: Path | -s Device | -z ]
```

The sysdumpdev command flags are listed in Table 20

Table 20. The sysdumpdev command flags

Flag	Description
-c	Specifies that dumps will not be compressed. The -c flag applies to only AIX Version 4.3.2 and later versions.
-C	Specifies that all future dumps will be compressed before they are written to the dump device. The -C flag applies to only AIX Version 4.3.2 and later versions.
-d Directory	Specifies the Directory the dump is copied to at system boot. If the copy fails at boot time, the -d flag ignores the system dump.

Flag	Description
-D Directory	Specifies the Directory the dump is copied to at system boot. If the copy fails at boot time, using the -D flag allows you to copy the dump to an external media. When using the -d Directory or -D Directory flags, the following error conditions are detected: Directory does not exist. Directory is not in the local journaled file system. Directory is not in the rootvg volume group.
-e	Estimates the size of the dump (in bytes) for the current running system.
-k	Requires the key mode switch to be in the service position before a dump can be forced with the reset button or the dump key sequences. This is the default setting.
-K	The reset button or the dump key sequences will force a dump with the key in the normal position, or on a machine without a key mode switch. On a machine without a key mode switch, a dump can not be forced with the reset button nor the key switch without this value set.
-l	Lists the current value of the primary and secondary dump devices, copy directory, and forcecopy attribute.
-L	Displays statistical information about the most recent system dump. This includes date and time of last dump, number of bytes written, and completion status.
-P	Makes permanent the dump device specified by -p or -s flags. The -P flag can only be used with the -p or -s flags.
-p Device	Temporarily changes the primary dump device to the specified device. The device can be a logical volume or a tape device. For a network dump, the device can be a host name and a path name.
-q	Suppresses all messages to standard output. If this flag is used with the -l, -r, -z or -L flag, the -q command will be ignored.
-r Host:Path	Frees space used by the remote dump file on server Host. The location of the dump file is specified by the Path.
-s Device	Temporarily changes the secondary dump device to the specified device. The device can be a logical volume or a tape device. For a network dump, the device can be a host name and a path name.

Flag	Description
-z	Determines if a new system dump is present. If one is present, a string containing the size of the dump in bytes and the name of the dump device will be written to standard output. If a new system dump does not exist, nothing is returned. After the sysdumpdev -z command is run on an existing system dump, the dump will no longer be considered recent.

6.12 The sysdumpstart command

The `sysdumpstart` command provides a command line interface to start a kernel dump to the primary or secondary dump device. When the dump completes, the system halts. Use the `crash` command to examine a kernel dump. Use the sysdumpdev command to reassign the dump device.

The `sysdumpstart` command syntax is as follows:

```
sysdumpstart { -p | -s [ -f ] }
```

During a kernel dump, the following values can be displayed on the three-digit terminal display as follows:

0c0 Indicates that the dump completed successfully.

0c1 Indicates that an I/O occurred during the dump. This value only applies to AIX Version 4.2.1 or later.

0c2 Indicates that the dump is in progress.

0c4 Indicates that the dump is too small.

0c5 Indicates a dump internal error .

0c6 Prompts you to make the secondary dump device ready. This value does not apply for AIX Version 4.2.1 or later.

0c7 Indicates that the dump process is waiting for a response from the remote host.

0c8 Indicates that the dump was disabled. In this case, no dump device was designated in the system configuration object for dump devices. The sysdumpstart command halts, and the system continues running.

0c9 Indicates that a dump is in progress.

0cc Indicates that the system switched to the secondary dump device after attempting a dump to the primary device. This value only applies to AIX Version 4.2.1 or later.

The `sysdumpstart` command flags are listed in Table 21

Table 21. The sysdumpstart command flags

Flag	Description
-f	Suppresses the prompt to make the secondary dump device ready. This flag does not apply to AIX Version 4.2.1 or later.
-p	Initiates a system dump and writes the results to the primary dump device.
-s	Initiates a system dump and writes the results to the secondary dump device.

6.13 Quiz

6.13.1 Answers

6.14 Exercises

1. Describe the difference ways to start a system dump.
2. On a core dump, name the two ways that can be used to find the program that caused the core dump.
3. Briefly describe how the crash command can be used to analyze system dumps.

Chapter 7. Error report

In AIX when an error is reported it is written to the error report. These may not always be errors as system shutdowns and other system functions that are also recorded in the error log. This chapter will cover the use of the error report and how it can be used to get information about problems and also how the report can be maintained.

7.1 The error daemon

The error logging daemon is started with the `errdemon` command and writes entries to the error log.

The error logging daemon reads error records from the /dev/error file and creates error log entries in the system error log. Besides writing an entry to the system error log each time an error is logged, the error logging daemon performs error notification as specified in the error notification database, /etc/objrepos/errnotify. 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 error logging daemon does not create an error log entry for the logged error if the error record template specifies Log=FALSE .

If you use the error logging daemon without flags, the system restarts the error logging daemon using the values stored in the error log configuration database for the error log file name, the error log file size, and the internal buffer size.

Use the `errclear` command to remove entries from the system error log.

Note

The error logging daemon is normally started during system initialization. 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 `errdemon` command syntax is as follows:

```
errdemon [ [ -B BufferSize ] [ -i File ] [ -s LogSize ] | -l ]
```

The `errdemon` flags are shown in Table 22

Table 22. The errdemon command flags

Flag	Description
-i File	Uses the error log file specified by the File variable. The specified file name is saved in the error log configuration database and is immediately put into use.
-l	Displays the values for the error log file name, file size, and buffer size from the error log configuration database.
-s LogSize	Uses the size specified by the LogSize variable for the maximum size of the error log file. The specified log file size limit is saved in the error log configuration database, and it is immediately put into use. If the log file size limit is smaller than the size of the log file currently in use, the error logging daemon renames the current log file by appending .old to the file name. The error logging daemon creates a new log file with the specified size limit. Generate a report form the old log file using the -i flag of the `errpt` command. If this parameter is not specified, the error logging daemon uses the log file size from the error log configuration database.

Flag	Description
-B BufferSize	Uses the number of bytes specified by the BufferSize parameter for the error log device driver's in-memory buffer. The specified buffer size is saved in the error log configuration database. If the BufferSize parameter is larger than the buffer size currently in use, the in-memory buffer is immediately increased. If the BufferSize parameter is smaller than the buffer size currently in use, the new size is put into effect the next time the error logging daemon is started after the system is rebooted. The buffer cannot be made smaller than the hard-coded default of 8KB. If this parameter is not specified, the error logging daemon uses the buffer size from the error log configuration database. The size you specify is rounded up to the next integral multiple of the memory page size (4KB). The memory used for the error log device driver's in-memory buffer is not available for use by other processes. (The buffer is pinned). Be careful not to impact your system's performance by making the buffer excessively large. On the other hand, if you make the buffer too small, the buffer can become full if error entries arrive faster than they can be read from the buffer and put into the log file. When the buffer is full, new entries are discarded until space becomes available in the buffer. When this situation occurs, the error logging daemon creates an error log entry to inform you of the problem. You can correct the problem by enlarging the buffer.

Example of the `errdemon` commands follow:

To check the attributes of the error log file use the `errdemon` command as follows:

```
# /usr/lib/errdemon -l

Error Log Attributes
--------------------------------------------
Log File                /var/adm/ras/errlog
Log Size                23899 bytes
Memory Buffer Size      8192 bytes
```

To change the current log file the `errdemon` command is used as follows:

```
# /usr/lib/errdemon -i /var/adm/ras/myerrlog
```

To change the error log file size the `errdemon` command is used as follows:

```
# /usr/lib/errdemon -s 47798
```

To change the error log buffer size the `errdemon` command is used as follows:

```
# /usr/lib/errdemon -B 16384
0315-175 The error log memory buffer size you supplied will be rounded up
to a multiple of 4096 bytes.
```

The new status can be checked using the errdemon command as follows:

```
# /usr/lib/errdemon -l
Error Log Attributes
--------------------------------------------
Log File                /var/adm/ras/myerrlog
Log Size                47798 bytes
Memory Buffer Size      16384 bytes
```

The errdemon command without any flags will start the error daemon if it is not running as follows:

```
# /usr/lib/errdemon
```

If the error daemon is running an error will be reported as follows:

```
# /usr/lib/errdemon
0315-100 The error log device driver, /dev/error, is already open.
The error demon may already be active.
```

7.2 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 they occurred and were recorded. By using the - c (concurrent) flag, you can display errors as they occur. If the -i flag is not used with the `errpt` command, the error log file processed by errpt is the one specified in the error log configuration database, by default /var/adm/ras/errlog.

The default summary report contains one line of data for each error. You can use flags to generate reports with different formats.

Note

The `errpt` command does not perform error log analysis; for analysis, use the `diag` command.

The `errpt` command syntax is as follows:

To Process a Report from the Error Log

```
errpt [ -a ] [ -c ] [ -d ErrorClassList ] [ -e EndDate ] [ -g ] [ -i File ]
[ -j ErrorID [ ,ErrorID ] ] | [ -k ErrorID [ ,ErrorID ] ] [ -J ErrorLabel [
,ErrorLabel ] ] | [ -K ErrorLabel [ ,ErrorLabel ] ] [ -l SequenceNumber ] [
-m Machine ] [ -n Node ] [ -s StartDate ] [ -F FlagList ] [ -N
ResourceNameList ] [ -R ResourceTypeList ] [ -S ResourceClassList ] [ -T
ErrorTypeList ] [ -y File ] [ -z File ]
```

Figure 18 shows how the `errpt` command processes a report from the error log.

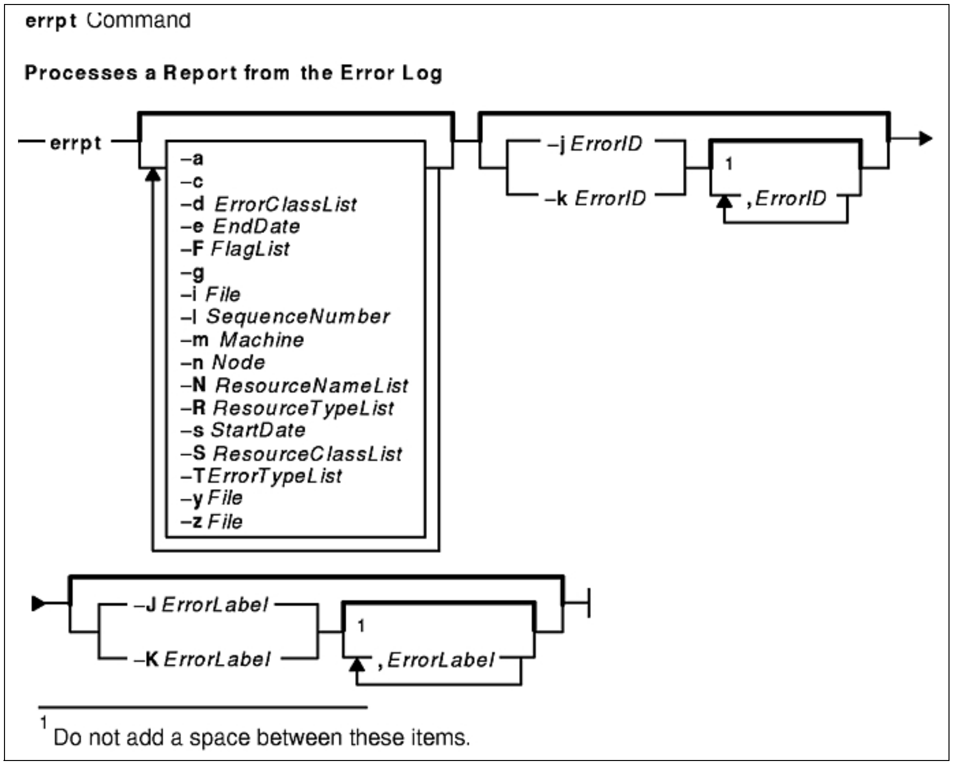


Figure 18. The errpt command error log report process

To Process a Report from the Error Record Template Repository

```
errpt [-a ] [ -t ] [ -d ErrorClassList ] [ -j ErrorID [ ,ErrorID ] ] | [ -k
ErrorID [ ,ErrorID ] ] [ -J ErrorLabel [ ,ErrorLabel ] ] | [ -K ErrorLabel
[ ,ErrorLabel ] ] [ -F FlagList ] [ -T ErrorTypeList ] [ -y File ] [ -z File
]
```

Figure 19 shows how the `errpt` command processes a report from the error record template.

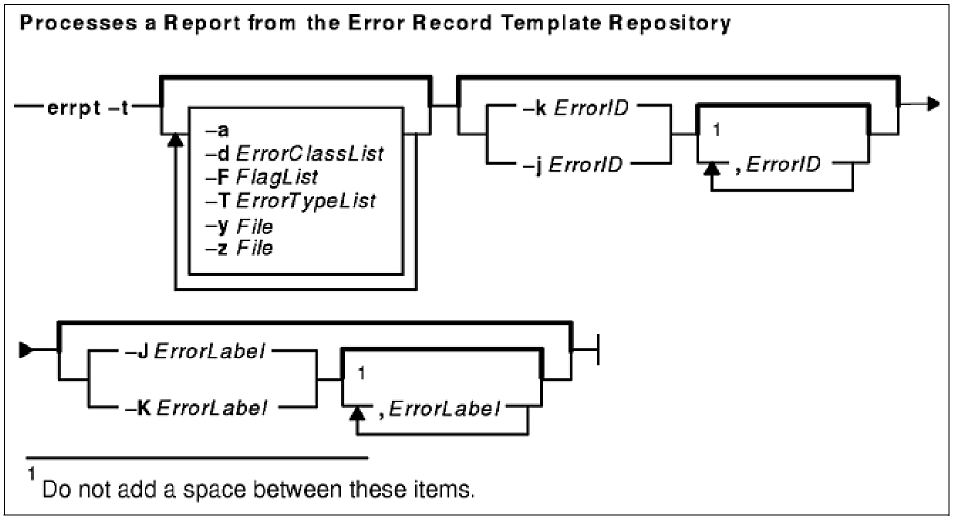


Figure 19. The errpt command error record template repository process

Table 23 is a listing of the errpt command flags

Table 23. The errpt command flags

Flag	Description
-a	Displays information about errors in the error log file in detailed format. If used in conjunction with the - t flag, all the information from the template file is displayed.
-c	Formats and displays each of the error entries concurrently, that is, at the time they are logged. The existing entries in the log file are displayed in the order in which they were logged.
-d ErrorClassList	Limits the error report to certain types of error records specified by the valid ErrorClassList variables: H (hardware), S (software), 0 (`errlogger` command messages), and U (undetermined). The ErrorClassList variable can be separated by , (commas), or enclosed in "" (double quotation marks) and separated by , (commas) or space characters.

Flag	Description
-e EndDate	Specifies all records posted before the EndDate variable, where the EndDate variable has the form mmddhhmmyy (month, day, hour, minute, and year).
-g	Displays the ASCII representation of unformatted error-log entries. The output of this flag is in the following format: *el_sequence* Error-log stamp number *el_label* Error label *el_timestamp* Error-log entry time stamp *el_crcid* Unique cyclic-redundancy-check (CRC) error identifier *el_machineid* Machine ID variable *el_nodeid* Node ID variable *el_class* Error class *el_type* Error type *el_resource* Resource name *el_rclass* Resource class *el_rtype* Resource type *el_vpd_ibm* IBM vital product data (VPD) *el_vpd_user* User VPD *el_in* Location code of a device *el_connwhere* Hardware-connection ID (location on a specific device, such as slot number) *et_label* Error label *et_class* Error class *et_type* Error type *et_desc* Error description *et_probcauses* Probable causes *et_usercauses* User causes *et_useraction* User actions *et_instcauses* Installation causes *et_instaction* Installation actions *et_failcauses* Failure causes *et_failaction* Failure actions *et_detail_length* Detail-data field length *et_detail_descid* Detail-data identifiers *et_detail_encode* Description of detail-data input format *et_logflg* Log flag *et_alertflg* Alertable error flag *et_reportflg* Error report flag *el_detail_length* Detail-data input length *el_detail_data* Detail-data input
-i File	Uses the error log file specified by the File variable. If this flag is not specified, the value from the error log configuration database is used.

Flag	Description
-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. (Otherwise entries are processed from the error-log repository.)
-k ErrorID[,ErrorID]	Excludes the error-log entries specified by the ErrorID 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. (Otherwise entries are processed from the error-log repository.)
-l SequenceNumber	Selects a unique error-log entry specified by the SequenceNumber variable. This flag is used by methods in the error-notification object class. The SequenceNumber variable can be separated by , (commas), or enclosed in "" (double quotation marks) and separated by , (commas) or space characters.
-m Machine	Includes error-log entries for the specified Machine variable. The `uname -m` command returns the Machine variable value.
-n Node	Includes error-log entries for the specified Node variable. The uname -n command returns the Node variable value.
-s StartDate	Specifies all records posted after the StartDate variable, where the StartDate variable has the form mmddhhmmyy (month, day, hour, minute, and year).
-t	Processes the error-record template repository instead of the error log. The -t flag can be used to view error-record templates in report form.
-y File	Uses the error record template file specified by the File variable. When combined with the -t flag, entries are processed from the specified error template repository. (Otherwise, entries are processed from the error log repository, using the specified error template repository.)
-z File	Uses the error logging message catalog specified by the File variable. When combined with the -t flag, entries are processed from the error template repository. (Otherwise, entries are processed from the error log repository.)

Flag	Description
-F FlagList	Selects error-record templates according to the value of the Alert , Log , or Report field of the template. The FlagList variable can be separated by , (commas), or enclosed in "" (double quotation marks) and separated by , (commas) or space characters. The -F flag is used with the -t flag only. Valid values of the FlagList variable include: *alert=0* Selects error-record templates with the Alert field set to False. *alert=1* Selects error-record templates with the Alert field set to True. *log=0* Selects error-record templates with the Log field set to False. *log=1* Selects error-record templates with the Log field set to True. *report=0* Selects error-record templates with the Report field set to False. *report=1* Selects error-record templates with the Report field set to True.
-J ErrorLabel	Includes the error log entries specified by the ErrorLabel variable. The ErrorLabel variable values can be separated by commas or enclosed in double-quotation marks and separated by commas or blanks. When combined with the -t flag, entries are processed from the error template repository. (Otherwise, entries are processed from the error log repository.)
-K ErrorLabel	Excludes the error log entries specified by the ErrorLabel variable. The ErrorLabel variable values can be separated by commas or enclosed in double-quotation marks and separated by commas or blanks. When combined with the -t flag, entries are processed from the error template repository. (Otherwise, entries are processed from the error log repository).
-N ResourceNameList	Generates a report of resource names specified by the ResourceNameList variable. For hardware errors, the ResourceNameList variable is a device name; for software errors it is the name of the failing executable. The ResourceNameList variable can be separated by , (commas), or enclosed in "" (double quotation marks) and separated by , (commas) or space characters.

Flag	Description
-R ResourceTypeList	Generates a report of resource types specified by the ResourceTypeList variable; for hardware errors the ResourceTypeList variable is a device type; for software errors it is the LPP value. The ResourceTypeList variable can be separated by , (commas), or enclosed in "" (double quotation marks) and separated by , (commas) or space characters.
-S ResourceClassList	Generates a report of resource classes specified by the ResourceClassList variable. For hardware errors, the ResourceClassList variable is a device class. The ResourceClassList variable can be separated by , (commas), or enclosed in "" (double quotation marks) and separated by , (commas) or space characters.
-T ErrorTypeList	Limits the error report to error types specified by the valid ErrorTypeList variables: INFO, PEND, PERF, PERM, TEMP, and UNKN. The ErrorTypeList variable can be separated by , (commas), or enclosed in "" (double quotation marks) and separated by , (commas) or space characters.

Examples of the errpt command follows:

To display a complete summary report, enter:

```
# errpt

IDENTIFIER TIMESTAMP  T C RESOURCE_NAME  DESCRIPTION
9DBCFDEE   0713172600 T O errdemon       ERROR LOGGING TURNED ON
9DBCFDEE   0713172400 T O errdemon       ERROR LOGGING TURNED ON
192AC071   0713172400 T O errdemon       ERROR LOGGING TURNED OFF
9DBCFDEE   0713172300 T O errdemon       ERROR LOGGING TURNED ON
192AC071   0713171700 T O errdemon       ERROR LOGGING TURNED OFF
...
35BFC499   0707112300 P H cd0            DISK OPERATION ERROR
0BA49C99   0707112300 T H scsi0          SCSI BUS ERROR
35BFC499   0707104000 P H cd0            DISK OPERATION ERROR
0BA49C99   0707104000 T H scsi0          SCSI BUS ERROR
369D049B   0706151600 I O SYSPFS         UNABLE TO ALLOCATE SPACE IN FILE
SYSTEM
```

To display a complete detailed report, enter:

```
# errpt -a
-------------------------------------------------------------------------
-
```

```
LABEL:          ERRLOG_ON
IDENTIFIER:     9DBCFDEE

Date/Time:       Thu Jul 13 17:26:11
Sequence Number: 143
Machine Id:      000FA17D4C00
Node Id:         server2
Class:           O
Type:            TEMP
Resource Name:   errdemon

Description
ERROR LOGGING TURNED ON

Probable Causes
ERRDEMON STARTED AUTOMATICALLY

User Causes
/USR/LIB/ERRDEMON COMMAND

        Recommended Actions
        NONE
...
Date/Time:       Thu Jul  6 15:16:09
Sequence Number: 8
Machine Id:      000FA17D4C00
Node Id:         server2
Class:           O
Type:            INFO
Resource Name:   SYSPFS

Description
UNABLE TO ALLOCATE SPACE IN FILE SYSTEM

Probable Causes
FILE SYSTEM FULL

        Recommended Actions
        USE FUSER UTILITY TO LOCATE UNLINKED FILES STILL REFERENCED
        INCREASE THE SIZE OF THE ASSOCIATED FILE SYSTEM
        REMOVE UNNECESSARY DATA FROM FILE SYSTEM

Detail Data
MAJOR/MINOR DEVICE NUMBER
002B 0003
FILE SYSTEM DEVICE AND MOUNT POINT
/dev/lv00, /u
```

To display a detailed report of all errors logged for the error identifier `369D049B`, enter:

```
# errpt -a -j 369D049B
---------------------------------------------------------------------------
-
LABEL:          JFS_FS_FULL
IDENTIFIER:     369D049B

Date/Time:       Thu Jul  6 15:16:09
Sequence Number: 8
Machine Id:      000FA17D4C00
Node Id:         server2
Class:           O
Type:            INFO
Resource Name:   SYSPFS

Description
UNABLE TO ALLOCATE SPACE IN FILE SYSTEM

Probable Causes
FILE SYSTEM FULL

        Recommended Actions
        USE FUSER UTILITY TO LOCATE UNLINKED FILES STILL REFERENCED
        INCREASE THE SIZE OF THE ASSOCIATED FILE SYSTEM
        REMOVE UNNECESSARY DATA FROM FILE SYSTEM

Detail Data
MAJOR/MINOR DEVICE NUMBER
002B 0003
FILE SYSTEM DEVICE AND MOUNT POINT
/dev/lv00, /u
```

To display a detailed report of all errors logged in the past 24 hours, enter:

```
# date
Fri Jul 14 14:08:35 CDT 2000

# errpt -a -s 0714140800
```

To list error-record templates for which logging is turned off for any error-log entries, enter:

```
# errpt -t -F log=0
Id       Label                Type CL Description
AF6582A7 LVM_MISSPVRET        UNKN S  PHYSICAL VOLUME IS NOW ACTIVE
```

To view all entries from the alternate error-log file var/adm/ras/*errlog.alternate* enter:

```
# errpt -i /var/adm/ras/myerrlog
```

```
IDENTIFIER TIMESTAMP  T C RESOURCE_NAME  DESCRIPTION
192AC071   0713172300 T O errdemon       ERROR LOGGING TURNED OFF
9DBCFDEE   0713172100 T O errdemon       ERROR LOGGING TURNED ON
192AC071   0713172100 T O errdemon       ERROR LOGGING TURNED OFF
9DBCFDEE   0713171900 T O errdemon       ERROR LOGGING TURNED ON
192AC071   0713171900 T O errdemon       ERROR LOGGING TURNED OFF
9DBCFDEE   0713171700 T O errdemon       ERROR LOGGING TURNED ON
```

where *errlog.alternate* is an alternative error log as specified with the `errdemon -i` command.

To view all hardware entries from the alternate error-log file /var/adm/ras/*errlog.alternate*, enter:

```
# errpt -i /var/adm/ras/errlog.alternate -d H
```

To display a detailed report of all errors logged for the error label `ERRLOG_ON`, enter:

```
# errpt -a -J ERRLOG_ON
```

```
--------------------------------------------------------------------------
-
LABEL:           ERRLOG_ON
IDENTIFIER:      9DBCFDEE

Date/Time:       Thu Jul 13 17:26:11
Sequence Number: 143
Machine Id:      000FA17D4C00
Node Id:         server2
Class:           O
Type:            TEMP
Resource Name:   errdemon

Description
ERROR LOGGING TURNED ON

Probable Causes
ERRDEMON STARTED AUTOMATICALLY

User Causes
/USR/LIB/ERRDEMON COMMAND
```

```
        Recommended Actions
        NONE
...
LABEL:          ERRLOG_ON
IDENTIFIER:     9DBCFDEE

Date/Time:       Fri Jul  7 17:00:46
Sequence Number: 14
Machine Id:      000FA17D4C00
Node Id:         server2
Class:           O
Type:            TEMP
Resource Name:   errdemon

Description
ERROR LOGGING TURNED ON

Probable Causes
ERRDEMON STARTED AUTOMATICALLY

User Causes
/USR/LIB/ERRDEMON COMMAND

        Recommended Actions
        NONE
```

7.3 The errclear command

The `errclear` command deletes error-log entries older than the number of days specified by the Days parameter. To delete all error-log entries, specify a value of `0` for the Days parameter.

If the -i flag is not used with the `errclear` command, the error log file cleared by `errclear` is the one specified in the error log configuration database. (To view the information in the error log configuration database, use the `errdemon` command.)

The errclear command syntax is as follows:

```
errclear [ -d ErrorClassList ] [ -i File ] [ -J ErrorLabel [ ,Errorlabel ]
] | [ -K ErrorLabel [ ,Errorlabel ] ] [ -l SequenceNumber ] [ -m Machine ]
[ -n Node ] [ -N ResourceNameList ] [ -R ResourceTypeList ] [ -S
ResourceClassList ] [ -T ErrorTypeList ] [ -y FileName ] [ -j ErrorID [
,ErrorID ] ] | [ -k ErrorID [ ,ErrorID ] ] Days
```

Table 24 shows the flags for the `errclear` command

Table 24. The errclear command flags

Flag	Description
-d List	Deletes error-log entries in the error classes specified by the List variable. The List variable values can be separated by , (commas), or enclosed in " " (double quotation marks) and separated by , (commas) or space characters. The valid List variable values are H (hardware), S (software), O (errlogger messages), and U (undetermined).
-i File	Uses the error-log file specified by the File variable. If this flag is not specified, the errclear command uses the value from the error-log configuration database.
-j ErrorID[,ErrorID]	Deletes the error-log entries specified by the ErrorID (error identifier) variable. The ErrorID variable values can be separated by , (commas), or enclosed in " " (double quotation marks) and separated by , (commas) or space characters.
-J ErrorLabel	Deletes the error-log entries specified by the ErrorLabel variable. The ErrorLabel variable values can be separated by , (commas), or enclosed in " " (double quotation marks) and separated by , (commas) or space characters.
-k ErrorID[,ErrorID]	Deletes all error-log entries except those specified by the ErrorID (error identifier) variable. The ErrorID variable values can be separated by , (commas), or enclosed in " " (double quotation marks) and separated by , (commas) or space characters.
-K ErrorLabel	Deletes all error-log entries except those specified by the ErrorLabel variable. The ErrorLabel variable values can be separated by , (commas), or enclosed in " " (double quotation marks) and separated by , (commas) or space characters.
-l SequenceNumber	Deletes error-log entries with the specified sequence numbers. The SequenceNumber variable values can be separated by , (commas), or enclosed in " " (double quotation marks) and separated by , (commas) or space characters.
-m Machine	Deletes error-log entries for the machine specified by the Machine variable. The uname -m command returns the value of the Machine variable.

Flag	Description
-n Node	Deletes error-log entries for the node specified by the Node variable. The uname -n command returns the value of the Node variable.
-N List	Deletes error-log entries for the resource names specified by the List variable. For hardware errors, the List variable is a device name. For software errors, the List variable is the name of the unsuccessful executable. The List variable values can be separated by , (commas), or enclosed in " " (double quotation marks) and separated by , (commas) or space characters.
-R List	Deletes error-log entries for the resource types specified by the List variable. For hardware errors, the List variable is a device type. For software errors, the value of the List variable is LPP. The List variable values can be separated by , (commas), or enclosed in " " (double quotation marks) and separated by , (commas) or space characters.
-S List	Deletes error-log entries for the resource classes specified by the List variable. For hardware errors, the List variable is a device class. The List variable values can be separated by , (commas), or enclosed in " " (double quotation marks) and separated by , (commas) or space characters.
-T List	Deletes error-log entries for error types specified by the List variable. Valid List variable values are: PERM, TEMP, PERF, PEND, INFO, and UNKN. The List variable values can be separated by , (commas), or enclosed in " " (double quotation marks) and separated by , (commas) or space characters.
-y FileName	Uses the error-record template file specified by the FileName variable.

Example of the `errclear` command follows:

To delete all entries from the error log, enter:

```
# errclear 0
```

To delete all entries in the error log classified as software errors, enter:

```
# errclear -d S 0
```

To clear all entries from the alternate error-log file /var/adm/ras/*errlog.alternate*, enter:

```
# errclear -i /var/adm/ras/myerrlog 0
```

To clear all hardware entries from the alternate error-log file /var/adm/ras/*errlog.alternate*, enter:

```
# errclear -i /var/adm/ras/myerrlog -d H 0
```

Note

Once the `errclear` command has been run it clears out the error log and this data is no longer available. To get this error information the error log would have to be restored from a backup prior to running the errclear command.

7.4 Quiz

7.4.1 Answers

7.5 Exercises

1. Describe the two methods errpt uses to process a report.
2. Once the errclear command has been run to clear all entries in the error report, what is the only way to restore the error log?
3. What is the file name for the error log and the directory where it is kept.

Chapter 8. LVM, File system and disk problems

The following topics are discussed in this chapter:

- Logical Volume Manager (LVM) problems
- Replacement of physical volumes
- JFS file system problems and their solutions
- Paging space creation and removal as well as recommendations about paging space.

To understand the problems that can happen on an AIX system with volume groups, logical volumes and file system it is important to have a detailed knowledge about the storage is managed by the logical volume manager (LVM). This chapter does not cover the fundamentals of the LVM, they are considered to be prerequisite knowledge required to understand the issues addressed in this chapter. To get a good understanding of the AIX logical volume manager, please refer to the following documentation:

- *AIX Version 4.3 System Management Guide: Operating System and Devices*, SC23-2525.
- *IBM Certification Study Guide AIX V4.3 System Administration*, SG24-5129
- *AIX Logical Volume Manager, from A to Z: Introduction and Concepts*, SG24-5432

8.1 LVM data

The logical volume manager (LVM) data structures required for the LVM to operate are stored in a number of places.

8.1.1 Physical volumes

Each disk is assigned a Physical Volume Identifier (PVID) when it is first assigned to a volume group. The PVID is a combination of the serial number of the machine creating the volume group and the time and date of the operation. The PVID is stored on the disk itself and is also stored in the ODM of a machine when a volume group is created or imported.

You should not use the `dd` command to copy the contents of one physical volume to another, since the PVID will also be copied; this will result in two disks having the same PVID which can confuse the system.

8.1.2 Volume groups

Each volume group has a Volume Group Descriptor Area (VGDA). There are multiple copes of the VGDA in a volume group. A copy of the VGDA is stored on each disk in the volume group. The VGDA stores information about the volume group, such as the logical volumes in the volume group and the disks in the volume group.

The VGDA is parsed by the `importvg` command when importing a volume group into a system. It is also used by the `varyonvg` command in the quorum voting process to decide if a volume group should be varied on.

For a single disk volume group, there are two VGDAs on the disk. When a second disk is added to make a two disk volume group, the original disk retains two VGDAs and the new disk gets one VGDA.

Adding a third disk results in the extra VGDA from the first disk moving to the third disk for a quorum of three with each disk having one vote. Adding each additional disk adds one new VGDA per disk.

A volume group with quorum checking enabled (the default) must have at least 51 percent of the VGDAs in the volume group available before it can be varied on. Once varied on, if the number of VGDAs falls below 51 percent, the volume group will automatically be varied off.

In contrast, a volume group with quorum checking disabled must have 100 percent of the VGDAs available before it can be varied on. Once varied on, only one VGDA needs to remain available to keep the volume group online.

A volume group also has a Volume Group Identifier (VGID), a soft serial number for the volume group similar to the PVID for disks.

Each disk in a volume group also has a Volume Group Status Area (VGSA), a 127 byte structure used to track mirroring information for up to the maximum 1016 physical partitions on the disk.

8.1.3 Logical volumes

Each logical volume has a Logical Volume Control Block (LVCB), that is stored in the first 512 bytes of the logical volume. The LVCB holds important details about the logical volume, including its creation time, mirroring information, and mount point if it contains a Journaled File System (JFS).

Each logical volume has a Logical Volume Identifier (LVID) that is used to represent the logical volume to the LVM libraries and low-level commands.

The LVID is made up of VGID.<num>, where num is the order in which it was created in the volume group.

8.1.4 Object Data Manager (ODM)

The Object Data Manger (ODM) is used by the LVM to store information about the volume groups, physical volumes, and logical volumes on the system. The information held in the ODM is placed there when the volume group is imported or when each object in the volume group is created.

There exists an ODM object known as the vg-lock. Whenever an LVM modification command is started, the LVM command will lock the vg-lock for the volume group being modified. If for some reason the lock is inadvertently left behind, the volume group can be unlocked by running the `varyonvg -b` command, which can be run on a volume group that is already varied on.

8.2 LVM problem determination

The most common LVM problems are related to with disk failures. Depending on the extent of the failure, you may be able to recover the situation with little or no data loss. However a failed recovery attempt may leave the system in a worse condition than before. Leaving you the only way to recover is to restore from a backup. Therefore always keep in mind to make frequent backups of your system.

8.2.1 Data relocation

When a problem occurs with a disk drive, sometimes data relocation takes place. There are three types of data relocation:

- Internal to the disk
- Hardware relocate ordered by LVM
- Software relocation

Relocation typically occurs when the system fails to perform a read or write due to physical problems with the disk platter. In some cases, the data I/O request completes but with warnings. Depending on the type of recovered error, the LVM may be wary of the success of the next request to that physical location, so it orders a relocation to be on the safe side.

The lowest logical layer of relocation is the one that is internal to the disk. These types of relocations are typically private to the disk and there is no notification to the user that a relocation occurred.

The next level up in terms of relocation complexity is a hardware relocation called for by the LVM device driver. This type of relocation will instruct the disk to relocate the data on one physical partition to another portion (reserved) of the disk. The disk takes the data in physical location A and copies it to a reserved portion of the disk, location B. However, after this is complete, the LVM device driver will continue to reference physical location A, with the understanding that the disk itself will handle the true I/O to the real location B.

The top layer of data relocation is the *soft* relocation handled by the LVM device driver. In this case, the LVM device driver maintains a bad block directory, and whenever it receives a request to access a logical location A, the LVM device driver will look up the bad block table and translate it to actually send the request to the disk drive at physical location B.

8.2.2 Backup data

The first step you should perform if you suspect a problem with LVM is to make a backup of the affected volume group and save as much data as possible. This may be required for data recovery. The integrity of the backup should be compared with the last regular backup taken before the problem was detected.

8.2.3 ODM re-synchronization

Problems with the LVM tend to occur when a physical disk problem causes the ODM data to become out of sync with the VGDA, VGSA, and LVCB information stored on disk.

ODM corruption can also occur if an LVM operation terminates abnormally and leaves the ODM in an inconsistent state. This may happen, for example, if the file system on which the ODM resides (normally /) becomes full during the process of importing a volume group.

If you suspect the ODM entries for a particular volume group have been corrupted, a simple way to re-synchronize the entries is to vary off and export the volume group from the system, then import and vary on to refresh the ODM. This process can only be performed for non-rootvg volume groups.

For the rootvg volume group, you can try using the `redefinevg` command that examines every disk in the system to determine which volume group it belongs to, and then updates the ODM. For example:

```
# redefinevg rootvg
```

If you suspect the LVM information stored on disk has become corrupted, use the `synclvodm` command to synchronizes and rebuild the LVCB, the device configuration database, and the VGDAs on the physical volumes. For example:

```
# synclvodm -v myvg
```

If you have a volume group in which one or more logical volumes is mirrored, use the `syncvg` command if you suspect that one or more mirror copies has become stale. The command can be used to re-synchronize an individual logical volume, a physical disk, or an entire volume group. For example:

```
# syncvg -l lv02
```

Will synchronize the mirror copies of the logical volume `lv02`.

```
# syncvg -v myvg
```

Will synchronize all of the logical volumes in the volume group `myvg`.

8.2.4 importvg problems

If importing a volume group into a system is not possible using the `importvg` command, a number of problems can be the reason. The following are the typical problem areas:

- AIX Version Level
- Invalid PVID
- Disk change while volume group was exported
- Shared disk environment

In general if an importvg is unsuccessful always check the error log for information that can point to the problem.

8.2.4.1 AIX Version Level

Verify that the volume group you are importing is supported by the level of AIX running on the system. Various new features have been added to the LVM system at different levels of AIX, such as support for large volume groups. A number of these features require a change to the format of the VGDA stored on the disk, and thus will not be understood by previous levels of AIX.

8.2.4.2 Invalid PVID

Check that all of the disks in the volume group you are trying to import are marked as available to AIX and have valid PVIDs stored in the ODM. This can be checked using the `lspv` command. If any disks do not have a PVID displayed, use the `chdev` command to resolve the problem.

Example:

```
# lspv
hdisk0         000bc6fdc3dc07a7    rootvg
hdisk1         000bc6fdbff75ee2    testvg
hdisk2         000bc6fdbff92812    testvg
hdisk3         000bc6fdbff972f4    None
hdisk4         None                None
# chdev  -l hdisk4 -a pv=yes
hdisk4 changed
# lspv
hdisk0         000bc6fdc3dc07a7    rootvg
hdisk1         000bc6fdbff75ee2    testvg
hdisk2         000bc6fdbff92812    testvg
hdisk3         000bc6fdbff972f4    None
hdisk4         000bc6fd672864b9    None
```

In this example the PVID for `hdisk4` is not shown by the `lspv` command. This is resolved by running the `chdev` command. The PVID is read from the disk and place it in the ODM if the disk is accessible. It will only write a new PVID if there truly is no PVID on the disk. Alternatively the disk can be removed using the `rmdev` command and by running the configuration manager command `cfgmgr` the device is re-created with the correct PVID. After this an import of the volume group with `the importvg` command should be possible.

8.2.4.3 Disk change while volume group was exported

If the `importvg` command fails with an error message similar to this one:

```
0516-056 varyon testvg: The volume group is not varied on because a
        physical volume is marked missing. Run diagnostics.
```

The physical volume is marked missing and it is possible that some disk change to the disks defined in the volume group was done, while the volume group was exported. Check the error log with errpt in order to see what actually did happen to the respective disk.

In order to forces the volume group to be varied online unset the flag `-f` og the `importvg` command. This makes it possible to operate on the volume group and depending of the situation reconfigure the volume group by excluding the disk that is marked missing, with the `reducevg` command.

8.2.4.4 Shared disk environment

In a shared disk environment such as SSA disk system used by two or more systems it is possible that the physical volumes defined not are accessible, because they are already imported and *varied-on* on another machine. Check the volume groups on both machines an compare the PVIDs by using the lspv command.

8.2.5 Extending number of max PPs

In AIX systems where there is a need for more space in a volume group the following situation can occur. If the new disk is much larger disk then the disks in the volume group, there might not be enough PP descriptors per physical volume, because the maximum number of PPs is reached. This situation is typical on older installations, due to the rapid growth of storage technology. To overcome this a change of the volume group modifying the LVM meta data is required.

The `chvg` command is used for this operation using the flag -t and apply a factor value.

Example:

```
# lsvg testvg
VOLUME GROUP:   testvg                   VG IDENTIFIER:  000bc6fd5a177ed0
VG STATE:       active                   PP SIZE:        16 megabyte(s)
VG PERMISSION:  read/write               TOTAL PPs:      542 (8672 megabytes)
MAX LVs:        256                      FREE PPs:       42 (672 megabytes)
LVs:            1                        USED PPs:       500 (8000 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 PV: 1016                     MAX PVs:        32
# chvg -t 2 testvg
0516-1193 chvg: WARNING, once this operation is completed, volume group testvg
        cannot be imported into AIX 430 or lower versions. Continue (y/n) ?
y
0516-1164 chvg: Volume group testvg changed.  With given characteristics testvg
        can include upto 16 physical volumes with 2032 physical partitions each.
```

This example shows that the volume group testvg with a current 9.1 GB disk has a maximum number of 1016 PPs per physical volume. Adding a larger 18.3 GB disk would not be possible. The maximum size of the disk is limited to 17GB unless the maximum number of PPs is increased. Using the `chvg` command to increase the maximum number of PPs by a factor 2 to 2032 allows the volume group to be extended with physical volumes of up to app. 34GB.

8.3 Disk replacement

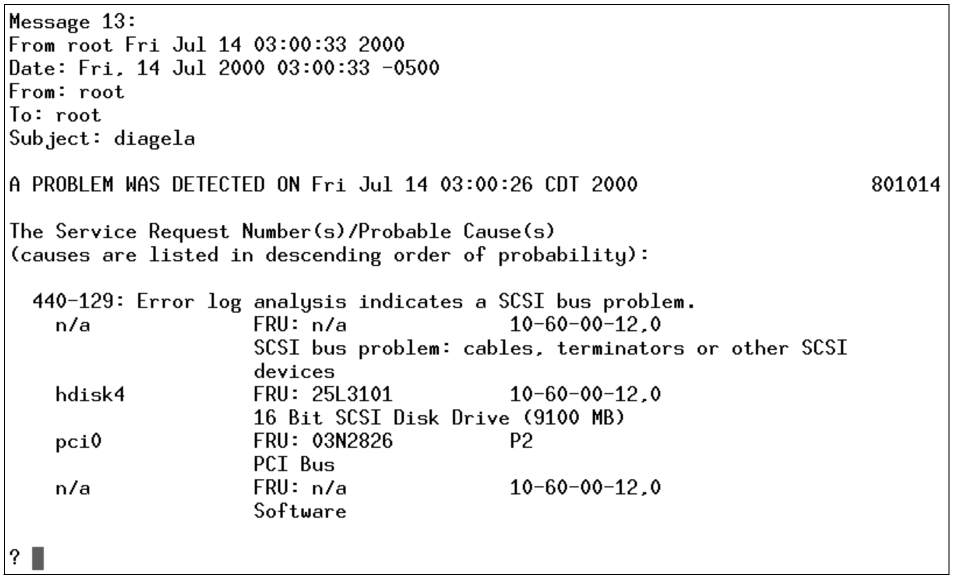
AIX, like all operating systems, can be problematic when you have to change a disk. AIX provides the ability to prepare the system for the change using the LVM. You can then perform the disk replacement and then use the LVM to restore the system back to how it was before the disk was changed. This process manipulates not only the data on the disk itself but is also a way of keeping the Object Data Manager (ODM) intact.

The ODM within AIX is a database that holds device configuration details and AIX configuration details. The function of the ODM is to store the information between reboots and also provide rapid access to system data eliminating the need for AIX commands to interrogate components for configuration information. Since this database holds so much vital information regarding the configuration of a machine, any changes made to the machine, such as the changing of a defective disk, need to be done in such a way as to preserve the integrity of the database.

8.3.1 Replacing a disk scenario

The following example scenario shows a system which has a hardware error on a physical volume. However since the system uses a mirrored environment which has multiple copies of the logical volume is possible to replace the disk while the system is active. The disk hardware in this example scenario are hot-plugable SCSI disks, which permits the replacement of the disk in a running environment.

One important factor is detecting the disk error. Normally a mail it sent to the system administrator (root account) from the Automatic Error Log Analysis (diagela). The Figure 20 on page 169 shows the information of such an diagnostics mail:



```
Message 13:
From root Fri Jul 14 03:00:33 2000
Date: Fri, 14 Jul 2000 03:00:33 -0500
From: root
To: root
Subject: diagela

A PROBLEM WAS DETECTED ON Fri Jul 14 03:00:26 CDT 2000                  801014

The Service Request Number(s)/Probable Cause(s)
(causes are listed in descending order of probability):

  440-129: Error log analysis indicates a SCSI bus problem.
    n/a              FRU: n/a              10-60-00-12,0
                     SCSI bus problem: cables, terminators or other SCSI
                     devices
    hdisk4           FRU: 25L3101          10-60-00-12,0
                     16 Bit SCSI Disk Drive (9100 MB)
    pci0             FRU: 03N2826          P2
                     PCI Bus
    n/a              FRU: n/a              10-60-00-12,0
                     Software

?
```

Figure 20. Disk problem mail from Automatic Error Log Analysis (diagela)

Automatic Error Log Analysis (diagela) provides the capability to do error log analysis whenever a permanent hardware error is logged. Whenever a permanent hardware resource error is logged and the diagela program is enabled, the diagela program is invoked. Automatic Error Log Analysis is enabled by default on all platforms.

The diagela message shows that the hdisk4 has a problem. Another good way of locating a problem is to check the state of the logical volume using the `lsvg` command.

Example:

```
# lsvg -l mirrorvg
mirrorvg:
LV NAME          TYPE      LPs   PPs   PVs  LV STATE      MOUNT POINT
lvdb01           jfs       500   1000  2    open/syncd    /u/db01
lvdb02           jfs       500   1000  2    open/stale    /u/db02
loglv00          jfslog    1     1     1    open/syncd    N/A
```

The logical volume lvdb02 in the volume group mirrorvg is marked with a status stale, indicating the copies in this LV are not synchronized. Any error indication like that should result in a look at the error log using the error reporting command `errpt`.

Example:

```
# errpt
EAA3D429   0713121400 U S LVDD          PHYSICAL PARTITION MARKED STALE
F7DDA124   0713121400 U H LVDD          PHYSICAL VOLUME DECLARED MISSING
41BF2110   0713121400 U H LVDD          MIRROR WRITE CACHE WRITE FAILED
35BFC499   0713121400 P H hdisk4        DISK OPERATION ERROR
```

This error information displays the reason why the LV lvdb02 is marked *stale*. The hdisk4 had an DISK OPERATION ERROR and the LVDD could not write the mirror cache.

It looks like the hdisk4 needs to be replaced. Before doing any action on the physical disk of the mirror LV, it is recommended to do a file system backup in case anything should go wrong. As the other disk of the mirrored LV still is functional all the data should be present. If the LV contains a database then the respective database tools for backup of the data should be used.

8.3.1.1 Removing bad disk

If the system is a high availability (24x7) system you might decide to let the system running while performing the disk replacement, provided that the HW supports a online disk exchange with hot-swappable disks. However the procedure should be agreed upon by the system administrator or customer before continuing.

1. To remove the physical partition copy of the mirrored logical volume from the erroneous disk used the command `rmlvcopy`:

   ```
   # rmlvcopy lvdb02 1 hdisk4
   ```

   The logical volume lvdb02 is now left with only one copy

   ```
   # lslv -l lvdb02
   lvdb02:/u/db02
   PV                 COPIES        IN BAND       DISTRIBUTION
   hdisk3             500:000:000   21%           109:108:108:108:067
   ```

2. Reduce the volume group by removing the disk you want to replace from its volume group.

   ```
   # reducevg  -f mirrorvg  hdisk4

   # lsvg -l mirrorvg
   mirrorvg:
   LV NAME             TYPE       LPs   PPs   PVs  LV STATE      MOUNT
   POINT
   ```

```
      lvdb01              jfs        500   1000  2    open/syncd    /u/db01
      lvdb02              jfs        500   500   1    open/syncd    /u/db02
      loglv00             jfslog     1     1     1    open/syncd    N/A
```

3. Remove the disk as a device from the system and from the ODM database whith the `rmdev` command.

   ```
   # rmdev -d -l hdisk4
   hdisk4  deleted
   ```

   The above command is valid for any SCSI disk. If your system is using SSA disk then an additional step is required. As SSA disks also define the device pdisk the corresponding pdisk device must be deleted as well. Use the SSA menus in SMIT to display the mapping between hdisk and pdisk. These menus can also be used to delete the pdisk device.

4. Now the disk can be safely removed from your system.

8.3.1.2 Adding a new disk

Continuing the example scenario from above, this section describes how to add a new disk into a running environment. After hdisk4 has been removed the he system is now left with the following disks:

```
# lsdev -Cc disk
hdisk0 Available 30-58-00-8,0 16 Bit SCSI Disk Drive
hdisk1 Available 30-58-00-9,0 16 Bit SCSI Disk Drive
hdisk2 Available 10-60-00-8,0 16 Bit SCSI Disk Drive
hdisk3 Available 10-60-00-9,0 16 Bit SCSI Disk Drive
```

1. Plugin the new disk an run the configuration manager command `cfgmgr`. The `cfgmgr` command configures devices controlled by the Configuration Rules object class, which is part of the Device Configuration database. The cfgmgr will see the newly inserted SCSI disk and create the corresponding device. :

   ```
   # cfgmgr
   ```

   The result is a new hdisk4 added to the system:

   ```
   #  lsdev -Cc disk
   hdisk0 Available 30-58-00-8,0  16 Bit SCSI Disk Drive
   hdisk1 Available 30-58-00-9,0  16 Bit SCSI Disk Drive
   hdisk2 Available 10-60-00-8,0  16 Bit SCSI Disk Drive
   hdisk3 Available 10-60-00-9,0  16 Bit SCSI Disk Drive
   hdisk4 Available 10-60-00-12,0 16 Bit SCSI Disk Drive
   ```

2. The new hdisk must now be assigned to the volume group mirrorvg by using the LVM command `extendvg`:

   ```
   # extendvg mirrorvg hdisk4
   ```

3. To re-establish the mirror copy of the LV used the command `mklvcopy`.

   ```
   # mklvcopy lvdb02 2  hdisk4
   ```

   Now the number of copies of LV is two, but the LV stat is still marked as *stale*, because the LV copies are not synchronized with each other.

   ```
   # lsvg -l mirrorvg
   mirrorvg:
   LV NAME              TYPE       LPs   PPs   PVs  LV STATE      MOUNT POINT
   lvdb01                jfs        500   1000  2    open/syncd    /u/db01
   lvdb02                jfs        500   1000  2    open/stale    /u/db02
   loglv00               jfslog     1     1     1    open/syncd    N/A
   ```

4. To get a fully synchronized set of copies of the LV lvdb02 use the command `syncvg`.

   ```
   # syncvg -p hdisk4
   ```

   The `syncvg` command can be used with logical volumes, physical volumes, or volume groups. The synchronization process can be quite time consuming, depending on the hardware characteristics and the amount of data.

   After the synchronization is finished verify the logical volume state using either `lsvg` or `lslv` command:

   ```
   # lsvg -l mirrorvg
   mirrorvg:
   LV NAME              TYPE       LPs   PPs   PVs  LV STATE      MOUNT POINT
   lvdb01                jfs        500   1000  2    open/syncd    /u/db01
   lvdb02                jfs        500   1000  2    open/syncd    /u/db02
   loglv00               jfslog     1     1     1    open/syncd    N/A
   ```

The system is now back to a *normal* situation.

8.3.2 Recovering an incorrectly removed disk

If a disk needs to be recovered that was incorrectly removed from the system and the system was rebooted the `synclvodm` command will need to be run.

Example:

In the example a disk has been incorrectly removed from the system and the logical volume control block needs to be rebuilt.

The disks in the system before the physical volume was removed.

```
# lsdev -Ccdisk

hdisk0 Available 30-58-00-8,0 16 Bit SCSI Disk Drive
hdisk1 Available 30-58-00-9,0 16 Bit SCSI Disk Drive
hdisk2 Available 10-60-00-8,0 16 Bit SCSI Disk Drive
hdisk3 Available 10-60-00-9,0 16 Bit SCSI Disk Drive
```

The allocation of the physical volumes before the disk was removed.

```
# lspv

hdisk0         000bc6fdc3dc07a7    rootvg
hdisk1         000bc6fdbff75ee2    volg01
hdisk2         000bc6fdbff92812    volg01
hdisk3         000bc6fdbff972f4    volg01
```

The logical volumes on the volume group.

```
# lsvg -l volg01

volg01:
LV NAME             TYPE       LPs   PPs   PVs  LV STATE      MOUNT POINT
logvol01            jfs        1000  1000  2    open/syncd    /userfs01
loglv00             jfslog     1     1     1    open/syncd    N/A
```

The logical volume distribution on the physical volumes.

```
# lslv -l logvol01

logvol01:/userfs01
PV                COPIES        IN BAND       DISTRIBUTION
hdisk1            542:000:000   19%           109:108:108:108:109
hdisk3            458:000:000   23%           109:108:108:108:025
```

After a reboot the system looks as follows:

```
# lspv

hdisk0         000bc6fdc3dc07a7    rootvg
hdisk1         000bc6fdbff75ee2    volg01
hdisk3         000bc6fdbff972f4    volg01
```

When trying to mount the file system on the logical volume the error may look like this:

```
# mount /userfs01
mount: 0506-324 Cannot mount /dev/logvol01 on /userfs01: There is an input
or output error.
```

To get the logical volume synchronized correctly the following needs to be done:

```
# synclvodm -v volg01
synclvodm: Physical volume data updated.
synclvodm: Logical volume logvol01 updated.
synclvodm: Warning, lv control block of loglv00 has been over written.
0516-622 synclvodm: Warning, cannot write lv control block data.
synclvodm: Logical volume loglv00 updated.
```

The system can now be repaired, if the file system data was spread across all the disks including the failed disk it may need to be restored from the last backup.

8.4 JFS file system

Similar to the LVM, most JFS problems can be traced to problems with the underlying physical disk.

As with volume groups, various JFS features have been added at different levels of AIX, which preclude those file systems being mounted if the volume group is imported on an earlier version of AIX. Such features include large file enabled file systems and file systems with non-default allocation group size

8.4.1 Creating JFS file systems

In a journaled file system (JFS) files are stored in blocks of contiguous bytes. The default block size also referred to as fragmentation size in AIX is 4096 byte (4 KB). The JFS i-node contains a information structure of the file together with an array of 8 pointers to data blocks. A file which is less then 32 KB is referenced directly from the i-node.

A larger file uses a 4 KB block, referred to as an indirect block, for the addressing of up to 1024 data blocks. Using an indirect block the a file size of 1024x4 KB=4 MB is possible.

For larger files then 4 MB a second block the double indirect block is used. The double indirect block points to 512 indirect blocks providing the possible addressing of 512*1024*4 KB=2 GB files. The Figure 21 on page 175 illustrates the addressing using double indirection.

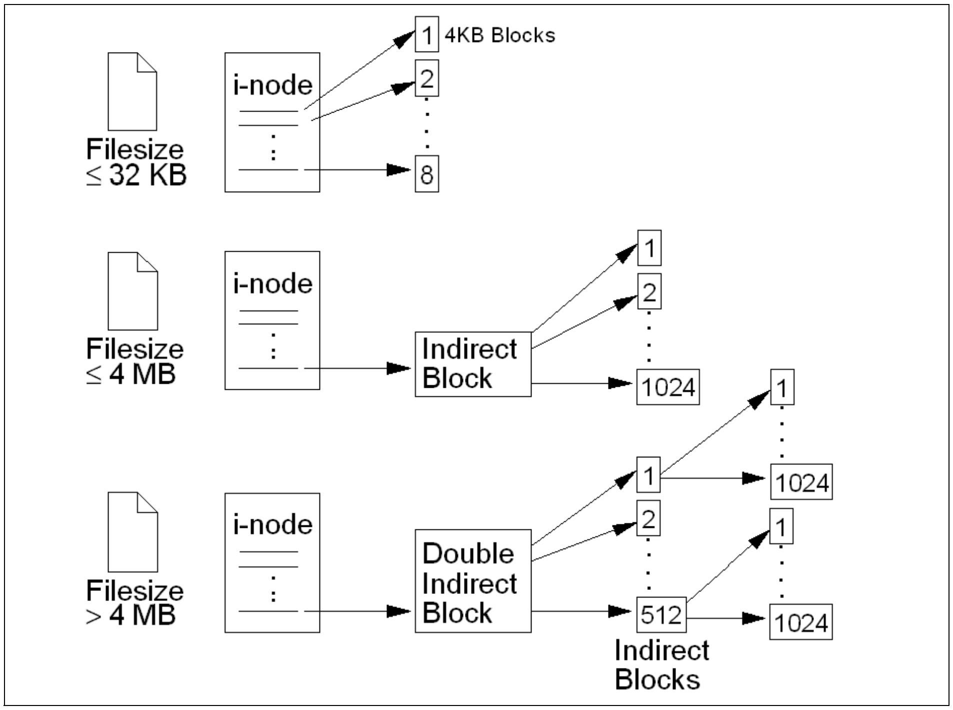


Figure 21. JFS file system organization.

Since AIX V4.2 support for even larger files was added, by defining a new type of JFS file system the *bigfile* file system. In the bigfile file system the double indirect are using references to 128 KB blocks rather then 4 KB blocks. However the first indirect block still points to 4 KB block, so that the large blocks are only used when the file size is above 4 MB. This provides a new maximum file size of just under 64GB.

When creating a JFS file system this structure is defined on either a new logical volume or an already defined logical volume. The parameters of a define JFS can be displayed either via SMIT menus (`smit jfs`) or by using the `lsjfs` command.

```
# lsjfs /u/testfs
#MountPoint:Device:Vfs:Nodename:Type:Size:Options:AutoMount:Acct:OtherOpti
ons:LvSize:FsSize:FragSize:Nbpi:Compress:Bf:AgSize:
/u/testfs:/dev/lv03:jfs:::425984:rw:yes:no::425984:425984:4096:4096:no:fal
se:8:
```

The `lsjfs` command shows the JFS attributes directly using : (colon) and delimiter.

8.4.2 Increasing the file system size

In many occasions the size of a file system needs to be increased because the demand for storage has increased. In AIX this is a very common procedure and it is possible by using the command chfs.

Example:

```
# chfs -a size=+300000 /u/testfs
Filesystem size changed to 458752
```

This example show how the file system testfs is extend with 300000 512-byte blocks. When the filesystem is extended the logical volume holding the JFS file system is extended as well, with the number of logical partitions that is needed to fulfill the space request. If the system does not have enough free space, the volume group can either be extended with an additional physical volume or a the size specified for the chfs command is lowered so it matches the number of free LPs.

8.4.3 File system verification and recovery

The fsck command checks and interactively repairs inconsistent file systems. You should run this command before mounting any file system. You must be able to read the device file on which the file system resides (for example, the /dev/hd0 device).

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. If the file system cannot be repaired, restore it from backup.

Mounting an inconsistent file system may result in a system crash. If you do not specify a file system with the FileSystem parameter, the fsck command will check all the file systems with attribute check=TRUE in /etc/filesystems.

Note

By default, the /, /usr, /var, and /tmp file systems have the check attribute set to False (check=false) in their /etc/filesystem stanzas. The attribute is set to False for the following reasons:

1. The boot process explicitly runs the `fsck` command on the /, /usr, /var, and /tmp file systems.
2. The /, /usr, /var, and /tmp file systems are mounted when the /etc/rc file is run. The `fsck` command will not modify a mounted file system, and `fsck` results on mounted file systems are unpredictable

8.4.3.1 Fixing a bad superblock

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
```

Following is an example where the the superblock is corrupted, and where copying the backup helps solving the problem.

Example:

```
# mount /u/testfs
mount: 0506-324 Cannot mount /dev/lv02 on /u/testfs: A system call received
a parameter that is not valid.
# fsck /dev/lv02

Not a recognized filesystem type. (TERMINATED)

# dd count=1 bs=4k skip=31 seek=1 if=/dev/lv02 of=/dev/lv02
1+0 records in.
1+0 records out.
```

```
# fsck /dev/lv02

** Checking /dev/rlv02 (/u/tes)
** Phase 0 - Check Log
log redo processing for /dev/rlv02
** Phase 1 - Check Blocks and Sizes
** Phase 2 - Check Pathnames
** Phase 3 - Check Connectivity
** Phase 4 - Check Reference Counts
** Phase 5 - Check Inode Map
** Phase 6 - Check Block Map
8 files 2136 blocks 63400 free
```

Once the restoration process is completed, check the integrity of the file system by issuing the `fsck` command.

```
fsck /dev/lv00
```

In many cases, restoration of the backup of the superblock to the primary superblock will recover the file system. If this does not resolve the problem, recreate the file system and restore the data from a backup.

8.4.4 Sparse file allocation

Some applications, particularly databases, maintain data in sparse files. Files that do not have disk blocks allocated for each logical block are called sparse files. Sparse files are created by seeking two different file offsets and writing data. If the file offsets are greater than 4MB, then a large disk block of 128KB is allocated. Applications using sparse files larger than 4MB may require more disk blocks in a file system enabled for large files than in a regular file system.

In case of sparse files the output of `ls` command is not showing the actual files size, but is reporting the number of bytes between the first and last blocks allocated to the file.

Example:

```
# ls -l /tmp/sparsefile
-rw-r--r--   1 root     system  100000000 Jul 16 20:57 /tmp/sparsefile
```

The command `du` can be used to see that actual allocation, as it is reporting the blocks actually allocated and in use by the file. Use `du -rs` to report the number of allocated blocks on disk.

Example:

```
# du -rs /tmp/sparsefile
```

```
256     /tmp/sparsefile
```

Note

When using the tar command for backing up the file system containing sparse file or files the following problem occurs. The `tar` command does not preserve the sparse nature of any file that is sparsely allocated. Any file that was originally sparse before the restoration will have all space allocated within the filesystem for the size of the file.

Only the `dd` command in combination with an own backup script will solve this problem.

8.4.5 Unmount problems

A file system cannot be unmounted if any references are still active within that file system. The following error message will be displayed:

```
Device busy
```

or

```
A device is already mounted or cannot be unmounted
```

The following situations can leave an 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 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:

```
# umount /home
umount: 0506-349 Cannot unmount /dev/hd1: The requested resource is
busy.
# fuser -x -c /home
/home:    11630
# ps -fp 11630
     UID   PID  PPID   C    STIME    TTY  TIME CMD
   guest 11630 14992   0 16:44:51  pts/1  0:00 -sh
# kill -1 11630
# umount /home

#
```

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 still 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. However, the `genkex` command will report on all loaded kernel extensions.
- File systems are still mounted within the file system. Unmount these file systems before the file system can be unmounted. 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. Use the `mount` command to get a list of mounted file systems. Unmount all the file systems that are mounted within the file system to be unmounted.

8.4.6 Removing file systems

When removing a JFS file system the file system has to be unmounted, before it can be removed. The command for removing file systems is `rmfs`.

In the case of an JFS s a journaled file system (JFS), the `rmfs` command removes both the logical volume on which the file system resides and the associated stanza in the /etc/filesystems file. If the file system is not a JFS file system, the command removes only the associated stanza in the /etc/filesystems file.

Example:

```
# lsvg -l testvg
testvg:
LV NAME             TYPE       LPs   PPs   PVs  LV STATE      MOUNT POINT
loglv00             jfslog     1     1     1    open/syncd    N/A
lv02                jfs        2     2     1    open/syncd    /u/testfs
#  rmfs /u/testfs
rmfs: 0506-921 /u/testfs is currently mounted.
# umount /u/testfs
# rmfs /u/testfs
rmlv: Logical volume lv02 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
```

This example shows how the file system testfs is removed. First attempt fails, because the file system is still mounted. The associated logical volume lv02 is also removed. The jfslog remains defined on the volume group.

8.5 Paging space

On AIX systems the following list indicates possible problems associated with paging space.

- All paging spaces defined on one physical volume
- Page space nearly full
- Imbalance in allocation of paging space on physical volumes with paging space
- Fragmentation of a paging space in a volume group

8.5.1 Recommendations for creating or enlarging paging space

Do not put 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 chunks. If you have paging space on multiple physical volumes and put 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 the same physical volume as a heavily active logical volume, such as that 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 roughly 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 best system performance, put paging space logical volumes on physical volumes that are each attached to a different disk controller.

8.5.2 Determining if more paging space is needed

Allocating more paging space than necessary results in unused paging space that is simply wasted disk space. But if you allocate too little paging space, a variety of unpleasant symptoms may occur on your system. To determine how much paging space is needed, use the following guidelines:

- Enlarge paging space if any of the following messages appear on the console or in response to a command on any terminal:

```
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
```

- Enlarge paging space if the %Used column of the `lsps -s` output is greater than 80.

Use the following commands to determine if you need to make changes regarding paging space logical volumes:

```
#iostat
#vmstat
#lsps
```

If you wish to remove a paging space from the system, or reduce the size of a paging space, this should be performed in two steps. The first step in either case is to change the paging space so that it is no longer automatically used when the system starts. This is done with the `chps` command, for example:

```
# chps -a n paging00
```

Once this has been done, you need to reboot the system, since there is no way to dynamically bring a paging space offline. Once the system reboots, the paging space will not be active. At this point, you can remove the paging space logical volume.

If you wanted to reduce the size of the paging space, you should remove the logical volume, and then create the new paging space with the desired size. The new paging space can be created activated without having to reboot the machine using the `mkps` command.

8.5.3 Removing paging space

Removing paging space can be done by using a the following procedure involving the `chps` and the `rmps` command.

> **Note**
>
> Removing default paging spaces incorrectly can prevent the system from restarting.
>
> The paging space must be deactivated before it can be removed. A special procedure is required for removing the default paging spaces (hd6, hd61, and so on). These paging spaces are activated during boot time by shell scripts that configure the system. To remove one of the default paging spaces, these scripts must be altered and a new boot image must be created.

This example scenario describes how to remove an existing additional paging space paging00 from the system.

```
# lsps -a
Page Space  Physical Volume   Volume Group    Size   %Used  Active  Auto Typ
paging00    hdisk2            testvg         3200MB      1     yes   yes   lv
hd6         hdisk0            rootvg         1040MB      1     yes   yes   lv
```

1. As the paging00 paging space is automatically activated use the `chps` command to change its state.

   ```
   # chps -a n paging00
   ```

2. As the paging space is in use a reboot the system is required. Make sure that the system dump device is still pointing to a valid paging space:

   ```
   # sysdumpdev -l
   primary              /dev/hd6
   secondary            /dev/sysdumpnull
   copy directory       /var/adm/ras
   forced copy flag     TRUE
   always allow dump    FALSE
   dump compression     OFF
   ```

3. Remove the paging00 paging space using the `rmps` command.

   ```
   # rmps paging00
   rmlv: Logical volume paging00 is removed.
   ```

If the paging space you are removing is the default dump device, you must change the default dump device to another paging space or logical volume before removing the paging space. To change the default dump device use the following command: `sysdumpdev -P -p /dev/new_dump_device`

8.6 Command summary

The following are commands discussed in this Chapter and the flags most often used. For a complete reference of the following command use the *AIX Version 4.3 Command Reference* or the online man pages.

8.6.1 lsvg

The `lsvg` command sets the characteristics of a volume group. The command has the following syntax:

```
lsvg [ -L ] [ -o ] | [ -n DescriptorPhysicalVolume ] | [ -i ] [ -l | -M | -p
] VolumeGroup ...
```

Table 25. Commonly used flags of the lsvg command

Flag	Description
-l	Lists the following information for each logical volume within the group specified by the VolumeGroup parameter: LV A logical volume within the volume group. Type Logical volume type. LPs Number of logical partitions in the logical volume. PPs Number of physical partitions used by the logical volume. PVs Number of physical volumes used by the logical volume. Logical volume state State of the logical volume. Opened/stale indicates the logical volume is open but contains partitions that are not current. Opened/syncd indicates the logical volume is open and synchronized. Closed indicates the logical volume has not been opened.

8.6.2 chvg

The `chvg` command sets the characteristics of a volume group. The command has the following syntax:

```
chvg [ -a AutoOn { n | y } ] [ -c | -l ] [ -Q { n | y } ] [-u ] [ -x { n |
y } ][ -t [factor ] ] [-B ] VolumeGroup
```

Table 26. Commonly used flags of the chvg command

Flag	Description
-t [factor]	Changes the limit of the number of physical partitions per physical volume, specified by factor. factor should be between 1 and 16 for 32 disk volume groups and 1 and 64 for 128 disk volume groups. If factor is not supplied, it is set to the lowest value such that the number of physical partitions of the largest disk in volume group is less than factor x 1016. If factor is specified, the maximum number of physical partitions per physical volume for this volume group changes to factor x 1016. Notes: 1. If the volume group is created in AIX 3.2/4.1.2 in violation of 1016 physical partitions per physical volume limit, this flag can be used to convert the volume group to a supported state. This will ensure proper stale/fresh marking of partitions. 2. factor cannot be changed if there are any stale physical partitions in the volume group. 3. Once volume group is converted, it cannot be imported into AIX Version 4.3 or lower versions. 4. This flag cannot be used if the volume group is varied on in concurrent mode.

8.6.3 importvg

The `importvg` command imports a new volume group definition from a set of physical volumes. The command has the following syntax:

```
importvg [ -V MajorNumber ] [ -y VolumeGroup ] [ -f ] [ -c ] [ -x ] | [ -L
VolumeGroup ] [ -n ] [ -F ] [ -R ]PhysicalVolume
```

Table 27. Commonly used flags of the importvg command

Flag	Description
-y VolumeGroup	Specifies the name to use for the new volume group. If this flag is not used, the system automatically generates a new name. The volume group name can only contain the following characters: "A" through "Z," "a" through "z," "0" through "9," or "_" (the underscore), "-" (the minussign), or "." (the period). All other characters are considered invalid.
-f	Forces the volume group to be varied online

8.6.4 rmlvcopy

The `rmlvcopy` command sets the characteristics of a volume group. The command has the following syntax:

```
rmlvcopy LogicalVolume Copies [ PhysicalVolume ... ]
```

8.6.5 reducevg

The `reducevg` command removes physical volumes from a volume group. When all physical volumes are removed from the volume group, the volume group is deletedsets the characteristics of a volume group. The command has the following syntax:

```
reducevg [ -d ] [ -f ] VolumeGroup PhysicalVolume ...
```

Table 28. Commonly used flags of the reducevg command

Flag	Description

8.6.6 rmdev

The `rmdev` command removes a device from the system. The command has the following syntax:

```
rmdev -l Name [ -d | -S ] [ -f File ] [ -h ] [ -q ] [ -R ]
```

Table 29. Commonly used flags of the rmdev command

Flag	Description
-l Name	Specifies the logical device, indicated by the Name variable, in the Customized Devices object class.
-d	Removes the device definition from the Customized Devices object class. This flag cannot be used with the -S flag

8.6.7 syncvg

The `syncvg` command synchronizes logical volume copies that are not current. The command has the following syntax:

```
syncvg [ -f ] [ -i ] [ -H ] [ -P NumParallelLps ] { -l | -p | -v } Name ...
```

Table 30. Commonly used flags of the syncvg command

Flag	Description
-p	Specifies that the Name parameter represents a physical volume device name.

8.7 Quiz

8.7.1 Answers

8.8 Exercises

The following exercises provide the setting for additional learning.

1. Verify the maximum number of PPs on your system, using for example rootvg. What is the maximum disk size that can be added to your system ?
2. If you have access to a test system which is equipped with hot-swappable SCSI disk, try the disk replacement example in Chapter 8.3.1, “Replacing a disk scenario” on page 168 as an exercise.

Chapter 9. Network problems

The following topics are discussed in this chapter:

- Network interface problems
- Routing problems
- Name resolution problems
- NFS troubleshooting

This chapter deals with network problem source identification and resolution.

9.1 Network interface problems

If host name resolution does not work and you cannot ping any address in the routing table, the interface itself may be the culprit. The first step should be to check the installed adapter types and states using the `lsdev -Cc adapter` and `lsdev -Cc if` commands.

```
# lsdev -Cc adapter
pmc0   Available 01-A0 Power Management Controller
fda0   Available 01-C0 Standard I/O Diskette Adapter
ide0   Available 01-E0 ATA/IDE Controller Device
ide1   Available 01-F0 ATA/IDE Controller Device
      ....
ppa0   Available 01-D0 Standard I/O Parallel Port Adapter
ent0   Available 04-D0 IBM PCI Ethernet Adapter (22100020)
tok0   Available 04-01 IBM PCI Tokenring Adapter (14101800)
# lsdev -Cc if
en0 Available  Standard Ethernet Network Interface
et0 Defined    IEEE 802.3 Ethernet Network Interface
lo0 Available  Loopback Network Interface
tr0 Available  Token Ring Network Interface
```

As you can see there are two network adapters and four network interfaces. All interfaces can be administrated either by the `chdev` or the `ifconfig` command.

To determine state of the interface use the `ifconfig` command. The following examples show the `en0` interface in the `up`, `down` and `detach` state.

The en0 interface in the `up` state:

```
# ifconfig en0
```

```
en0:
flags=e080863<UP,BROADCAST,NOTRAILERS,RUNNING,SIMPLEX,MULTICAST,GROUPRT,64
BIT>
        inet 10.47.1.1 netmask 0xffff0000 broadcast 10.47.255.255
```

The down state of the interface keeps the system from trying to transmit messages through that interface. Routes that use the interface are not automatically disabled.

```
# ifconfig en0 down
# ifconfig en0
en0:
flags=e080862<BROADCAST,NOTRAILERS,RUNNING,SIMPLEX,MULTICAST,GROUPRT,64BIT
>
        inet 10.47.1.1 netmask 0xffff0000 broadcast 10.47.255.255
```

The interface in the detach state is removed from the network interface list. If the last interface is detached, the network interface driver code is unloaded.

```
# ifconfig en0 detach
# ifconfig en0
en0: flags=e080822<BROADCAST,NOTRAILERS,SIMPLEX,MULTICAST,GROUPRT,64BIT>
```

All changes made to the network interface as shown above can be also done by the chdev command. Changes made by this command are permanent because they are made directly to the ODM database. To list the parameters of the network interface tr0 that you can change by the chdev command, enter:

```
# lsattr -El tr0
mtu           1492          Maximum IP Packet Size for This Device     True
mtu_4         1492          Maximum IP Packet Size for This Device     True
mtu_16        1492          Maximum IP Packet Size for This Device     True
mtu_100       1492          Maximum IP Packet Size for This Device     True
remmtu        576           Maximum IP Packet Size for REMOTE Networks True
netaddr       9.3.240.59    Internet Address                           True
state         up            Current Interface Status                   True
arp           on            Address Resolution Protocol (ARP)          True
allcast       on            Confine Broadcast to Local Token-Ring      True
hwloop        off           Enable Hardware Loopback Mode              True
netmask       255.255.255.0 Subnet Mask                                True
security      none          Security Level                             True
authority                   Authorized Users                           True
broadcast                   Broadcast Address                          True
netaddr6                    N/A                                        True
alias6                      N/A                                        True
prefixlen                   N/A                                        True
alias4                      N/A                                        True
```

```
rfc1323                     N/A                                         True
tcp_nodelay                 N/A                                         True
tcp_sendspace               N/A                                         True
tcp_recvspace               N/A                                         True
tcp_mssdflt                 N/A                                         True
```

For example to setup the broadcast address for the `tr0` interface, enter:

```
# chdev -l tr0 -a broadcast='9.3.240.255'
tr0 changed
```

To check the new value of the `broadcast` parameter, enter:

```
# lsattr -El tr0 -a broadcast
broadcast 9.3.240.255 Broadcast Address True
```

When you have network performance problem and you suspect that network interface could be cause of it, you should check interface statistics. To display statistic for `en0` interface, enter:

```
# netstat -I en0
Name  Mtu   Network     Address            Ipkts Ierrs    Opkts Oerrs  Coll
en0   1500  link#2      8.0.5a.fc.d2.e1     28982     0   579545     0     0
en0   1500  10.47       server4_            28982     0   579545     0     0
```

As you can see, the output shows the number of the input/output errors and the number of input/output packets.

Note

The collision count for ethernet interfaces is not displayed by the `netstat` command. It always shows 0

To see more detailed statistic use the `entstat` command:

```
# entstat -d en0
-------------------------------------------------------------
ETHERNET STATISTICS (en0) :
Device Type: IBM PCI Ethernet Adapter (22100020)
Hardware Address: 08:00:5a:fc:d2:e1
Elapsed Time: 1 days 1 hours 21 minutes 29 seconds

Transmit Statistics:                        Receive Statistics:
--------------------                        -------------------
Packets: 579687                             Packets: 55872
Bytes: 49852606                             Bytes: 4779893
Interrupts: 0                               Interrupts: 55028
Transmit Errors: 0                          Receive Errors: 0
```

```
Packets Dropped: 0                            Packets Dropped: 0
                                              Bad Packets: 0
Max Packets on S/W Transmit Queue: 2
S/W Transmit Queue Overflow: 0
Current S/W+H/W Transmit Queue Length: 0

Broadcast Packets: 2327                       Broadcast Packets: 0
Multicast Packets: 0                          Multicast Packets: 0
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: 34                                  Packet Too Short Errors: 0
SQE Test: 0                                   Packet Too Long Errors: 0
Timeout Errors: 0                           Packets Discarded by Adapter: 0
Single Collision Count: 4                     Receiver Start Count: 0
Multiple Collision Count: 12
Current HW Transmit Queue Length: 0

General Statistics:
-------------------
No mbuf Errors: 0
Adapter Reset Count: 4
Driver Flags: Up Broadcast Running
        Simplex AlternateAddress 64BitSupport

IBM PCI Ethernet Adapter Specific Statistics:
------------------------------------------------
Chip Version: 16
Packets with Transmit collisions:
 1 collisions: 4          6 collisions: 2         11 collisions: 0
 2 collisions: 1          7 collisions: 1         12 collisions: 0
 3 collisions: 4          8 collisions: 1         13 collisions: 0
 4 collisions: 1          9 collisions: 0         14 collisions: 0
 5 collisions: 1         10 collisions: 1         15 collisions: 0
```

To make a test for dropped packet use the `ping` command with `-f` flag. The `-f` flag *floods* or outputs packets as fast as they come back or one hundred times per second, whichever is more. For every ECHO_REQUEST sent, a **.** (period) is printed, while for every ECHO_REPLY received, a backspace is printed. This provides a rapid display of how many packets are being dropped. Only the root user may use this option.

9.2 Routing problems

If you are not able to `ping` by host name or IP address, you may have a routing problem.

First, check the routing tables as follows:

- Use the `netstat -rn` command to show the content of your local routing table using IP addresses.

```
# 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        31   142091  tr0     -   -
10.47.1.2        9.3.240.59        UGH       0        2  tr0     -   -
127/8            127.0.0.1         UR        0        3  lo0     -   -
127.0.0.1        127.0.0.1         UH        3      761  lo0     -   -
195.116.119/24   195.116.119.2     U         2      406  en0     -   -

Route Tree for Protocol Family 24 (Internet v6):
::1              ::1               UH        0        0  lo0 16896   -
```

- Check the netmask displayed and ensure that it is correct (ask the network administrator what it should be if you are unsure).

```
# lsattr -El tr0 -a netmask -F value
255.255.255.0
```

- If there is a default route, attempt to `ping` it.

```
# ping 9.3.240.1
PING 9.3.240.1: (9.3.240.1): 56 data bytes
64 bytes from 9.3.240.1: icmp_seq=0 ttl=64 time=1 ms
64 bytes from 9.3.240.1: icmp_seq=1 ttl=64 time=0 ms
^C
----9.3.240.1 PING Statistics----
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 0/0/1 ms
```

- If you have more than one network interface, attempt to determine if any interfaces are working.

If you cannot `ping` your default route, either it is down, or your local network connection may be down. Attempt to `ping` all of the other gateways listed in the routing table to see if any portion of your network is functioning:

If you cannot ping any host or router interface from among those listed in the routing table, try to ping your loopback interface lo0 with the following command:

```
# ping localhost
PING localhost: (127.0.0.1): 56 data bytes
64 bytes from 127.0.0.1: icmp_seq=0 ttl=255 time=1 ms
^C
----localhost PING Statistics----
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 1/1/1 ms
```

If the ping is successful you have either an adapter or network hardware problem or a routing problem.

If the ping is not successful, you need to:

- Ensure that the inetd process is active using the lssrc -g tcpip command. If inetd is not active issue the startsrc -s inetd or startsrc -g tcpip commands.

  ```
  # lssrc -g tcpip
  Subsystem         Group            PID     Status
   routed           tcpip            5424    active
   inetd            tcpip            6192    active
   snmpd            tcpip            6450    active
   gated            tcpip                    inoperative
   named            tcpip                    inoperative
  ----- the output was edited for brevity -----
  ```

- Check the state of the loopback interface (lo0) with the netstat -i command. If you see lo0* in the output, check the /etc/hosts file for an uncommented local loopback entry as follows:

  ```
  # netstat -I lo0 -n
  Name  Mtu   Network     Address         Ipkts Ierrs    Opkts Oerrs  Coll
  lo0*  16896 link#1                      412934     0   414344     0     0
  lo0*  16896 127         127.0.0.1       412934     0   414344     0     0
  lo0*  16896 ::1                         412934     0   414344     0     0
  # grep localhost /etc/hosts
  127.0.0.1      loopback localhost      # loopback (lo0) name/address
  ```

  An asterix (*) after the interface name in the output from the netstat command indicates that the interface is down.Use the following command to start the lo0 interface:

  ```
  # ifconfig lo0 inet 127.0.0.1 up
  ```

If you can not reach a host which is in the different network you can check connection using traceroute command. The traceroute output shows each gateway that the packet traverses on its way to finding the target host. If possible, examine the routing tables of the last machine shown in the `traceroute` output to check if a route exists to the destination from that host. The last machine shown is where the routing is going astray.

```
# traceroute 9.3.240.56
traceroute to 9.3.240.56 (9.3.240.56), 30 hops max, 40 byte packets
 1  server4e (10.47.1.1)  1 ms  1 ms  0 ms
 2  server1 (9.3.240.56)  1 ms  1 ms  1 ms
```

If you are using the `route` command to change the routing table on your machine and you want this change to be permanent insert appropriate line in the /etc/rc.net file.

9.2.1 Dynamic or static routing

If you suppose that you have a problem with dynamic routing protocol follow the procedure:

If your system is set up to use the `routed` daemon:

- Check if the `routed` is running, if not start it by the `startsrc -s routed` command
- If routed cannot identify the route through queries, check the /etc/gateways file to verify that a route to the target host is defined and that the target host is running the RIP.
- Make sure that gateways responsible for forwarding packets to the host are up and that they are running the RIP (routed or gated active). Otherwise, you will need to define a static route.
- Run the `routed` daemon with the debug option to log such information as bad packets received. Invoke the daemon from the command line using the following command:

  ```
  startsrc -s routed -a "-d"
  ```
- Run the `routed` daemon using the `-t` flag, which causes all packets sent or received to be written to standard output. When routed is run in this mode, it remains under the control of the terminal that started it. Therefore, an interrupt from the controlling terminal kills the daemon.

If your system is set up to use the `gated` daemon:

- Check if the `gated` is running, if not start it by the `startsrc -s gated` command.
- Verify that the /etc/gated.conf file is configured correctly and that you are running the correct routing protocols.

- Make sure the gateway on the source network is using the same protocol as the gateway on the destination network.
- Make sure that the machine with which you are trying to communicate has a return route back to your host machine.

You should set static routes under either of the following conditions:

- The destination host is not running the same protocol as the source host, so it cannot exchange routing information.
- The host must be reached by a distant gateway (a gateway that is on a different autonomous system than the source host). The RIP can be used only among hosts on the same autonomous system.

If you are using dynamic routing, you should not attempt to add static routes to the routing table using the `route` command.

As a very last resort of the problem solving, you may flush the routing table using the `route -f` command, which will cause all the routes to be removed and eventually replaced by the routing demons. This is a last case resort, since any networking that was functioning before will be temporarily cut off once the routes are removed. Be sure no other users will be interrupted by this.

If your system is going to be configured as a router (it has two or more network interfaces), then it needs to be enabled as a router by the `no` command. The network option that controls routing from one network to another is *ipforwarding* and by default is disabled. To enable it, enter:

```
# no -o ipforwarding=1
```

This is not a permanent setting and after the next system reboot will be lost. To make this permanent, add this command to the end of /etc/rc.net file.

Note

When you add the second network interface to your system a new entry appear in the routing table. This is a route associated with new interface

9.3 Name resolution problems

If network connections seems inexplicably slow sometimes but all right at other times, it is good idea to check name resolution configuration for your system. Do a basic diagnostic for name resolving. You can use either the `host` command or the `nslookup` command.

```
# host dhcp240.itsc.austin.ibm.com
dhcp240.itsc.austin.ibm.com is 9.3.240.2
```

The name resolution can be served through either remote DNS server or remote NIS server. So, if one of them is down, you have to wait till TCP time-out occur. The name can be resolved by alternate source, which can be secondary name server or the local /etc/hosts file.

First check the /etc/netsvc.conf file and `NSORDER` environment variable for your particular name resolution ordering. Remember that, the `NSORDER` variable overrides the hosts settings in the /etc/netsvc.conf file. Next, check /etc/resolve.conf file for IP address of name server and try to `ping` it. If you can `ping` it, then it is up and reachable. If not, try to play with different name resolution ordering.

Note

When you can ping the name server it does not mean that the `named` demon is active on this system

By default, resolver routines attempt to resolve names using BIND/DNS. If the /etc/resolv.conf file does not exist or if BIND/DNS could not find the name, NIS is queried if it is running. NIS is authoritative over the local /etc/hosts, so the search will end here if it is running. If NIS is not running, then the local /etc/hosts file is searched. If none of these services could find the name, then the resolver routines return with HOST_NOT_FOUND. If all of the services are unavailable, then the resolver routines return with SERVICE_UNAVAILABLE.

If you want to change name resolution ordering that NIS takes precedence over the BIND/DNS your /etc/netsvc.conf file should looks like:

```
# cat /etc/netsvc.conf
hosts = nis,bind
```

You can ovreride this setting by using the NSORDER environment variable:

```
# export NSORDER=local,bind
```

In this situation the /etc/hosts file will be examined for name resolution first.

When you have protocol level problems with communication between your system and name server you can use the `iptrace` daemon to find the cause of it. The `iptrace` demon records all packets that are exchange between two systems. Command flags provide a filter so that the daemon traces only packets meeting specific criteria. Packets are traced only between the local host on which the iptrace daemon is invoked and the remote host. To format `iptrace` output run the `ipreport` command. The following example shows query from host 9.3.240.59 to DNS server 9.3.240.2. The output from the `nslookup` command is shown below:

```
# nslookup www.prokom.pl
Server:  dhcp240.itsc.austin.ibm.com
Address:  9.3.240.2

Non-authoritative answer:
Name:    mirror.prokom.pl
Address:  153.19.177.201
Aliases:  www.prokom.pl
```

The data was grabbed by the `iptrace` command as shown below:

```
# iptrace -a -P UDP -s 9.3.240.59 -b -d 9.3.240.2 /tmp/dns.query
```

The output form the `iptrace` command was formatted by the `ipreport` command:

```
TOK: ====( 81 bytes transmitted on interface tr0 )==== 17:14:26.406601066
TOK: 802.5 packet
TOK: 802.5 MAC header:
TOK: access control field = 0, frame control field = 40
TOK: [ src = 00:04:ac:61:73:f7, dst = 00:20:35:29:0b:6d]
TOK: 802.2 LLC header:
TOK: dsap aa, ssap aa, ctrl 3, proto 0:0:0, type 800 (IP)
IP:  < SRC =      9.3.240.59 >  (server4f.itsc.austin.ibm.com)
IP:  < DST =       9.3.240.2 >  (dhcp240.itsc.austin.ibm.com)
IP:  ip_v=4, ip_hl=20, ip_tos=0, ip_len=59, ip_id=64417, ip_off=0
IP:  ip_ttl=30, ip_sum=aecc, ip_p = 17 (UDP)
UDP: <source port=49572, <destination port=53(domain) >
UDP: [ udp length = 39 | udp checksum = 688d ]
DNS Packet breakdown:
    QUESTIONS:
   www.prokom.pl, type = A, class = IN

TOK: ====( 246 bytes received on interface tr0 )==== 17:14:26.407798799
TOK: 802.5 packet
TOK: 802.5 MAC header:
TOK: access control field = 18, frame control field = 40
TOK: [ src = 80:20:35:29:0b:6d, dst = 00:04:ac:61:73:f7]
TOK: routing control field = 02c0,  0 routing segments
TOK: 802.2 LLC header:
TOK: dsap aa, ssap aa, ctrl 3, proto 0:0:0, type 800 (IP)
IP:  < SRC =       9.3.240.2 >  (dhcp240.itsc.austin.ibm.com)
IP:  < DST =      9.3.240.59 >  (server4f.itsc.austin.ibm.com)
```

```
IP:  ip_v=4, ip_hl=20, ip_tos=0, ip_len=222, ip_id=2824, ip_off=0
IP:  ip_ttl=64, ip_sum=7cc3, ip_p = 17 (UDP)
UDP: <source port=53(domain), <destination port=49572 >
UDP: [ udp length = 202 | udp checksum = a7bf ]
DNS Packet breakdown:
    QUESTIONS:
   www.prokom.pl, type = A, class = IN
    ANSWERS:
    ->  www.prokom.plcanonical name = mirror.prokom.pl
    ->  mirror.prokom.plinternet address = 153.19.177.201
    AUTHORITY RECORDS:
    ->  prokom.plnameserver = phobos.prokom.pl
    ->  prokom.plnameserver = alfa.nask.gda.pl
    ->  prokom.plnameserver = amber.prokom.pl
    ADDITIONAL RECORDS:
    ->  phobos.prokom.plinternet address = 195.164.165.56
    ->  alfa.nask.gda.plinternet address = 193.59.200.187
    ->  amber.prokom.plinternet address = 153.19.177.200
```

There are two packets shown on the `ipreport` output above. Every packet is divided into a few parts. Each part describes different network protocol level. There is Token Ring (TOK), IP, UDP and application (DNS) part. The first packet is send by host 9.3.240.59 and this is query about IP address of www.prokom.pl host. The second one is the answer.

9.4 NFS troubleshooting

Prior to starting any NFS debugging, it is necessary to ensure the underlying network is up and working correctly. It is also most important to ensure that name resolution is functional and consistent across the network and that end-to-end routing is correct both ways.

9.4.1 General steps for NFS problem solving

The general steps for NFS problem solving are as follows:

1. Check for correct network connectivity and configuration as described in previous sections.
2. Check the following NFS configuration files on the client and server for content and permissions:
   - /etc/exports (servers only)
   - /etc/rc.tcpip
   - /etc/rc.nfs

- /etc/filesystems (clients only)
- /etc/inittab

3. Check that the following NFS daemons are active on the client and server.

   Server NFS daemons required:

   - portmap
   - biod
   - nfsd
   - rpc.mountd
   - rpc.statd
   - rpc.lockd

   Client NFS daemons required:

   - portmap
   - biod (these are dymanically created on AIX Version 4.2.1 and later)
   - rpc.statd
   - rpc.lockd

4. Initiate an `iptrace` (client or server or network), reproduce the problem, then view the ipreport output to determine where the problem is.

9.4.2 NFS mount problems

Mount problems fall into one of the categories below:

- File system not exported, or not exported to a specific client.
  - Correct server export list (/etc/exports)
- Name resolution different from the name in the export list. Normally, it is due to one of the following causes:
  - The export list uses a fully qualified name but the client host name is resolved without network domain. Fully qualified names cannot be resolved – mount permission is denied. Usually, this happens after upgrade activity and can be fixed by exporting to both forms of the name.
  - The client has two adapters and two different names for the two adapters and the export only specifies one. This problem can be fixed by exporting both names.

- Server cannot do a lookuphostbyname or lookuphostbyaddr onto the client. To check, make sure the following commands both resolve to the same thing:
  - `host <name>`
  - `host <ip_addr>`
- The file system mounted on the server after `exportfs` was run. In this case, the `exportfs` command is exporting the mount point and not the mounted file system. To correct this problem run:

  ```
  /usr/etc/exportfs -ua; /usr/etc/exportfs -a
  ```

  Then fix the /etc/filesystems file to mount the file system on boot, so it is already mounted when NFS starts from /etc/rc.nfs at system startup.
- Changes in the exports list, mounts, or somewhere else unexpectedly can sometimes lead to mountd getting confused. This usually happens following mounting, exporting, or because of mount point conflicts and the like. To correct this condition, `mountd` needs to be restarted:

  ```
  # stopsrc -s rpc.mountd
  # startsrc -s rpc.mountd
  ```
- System date being wildly off on one or both machines is another source of mount problems. To fix this, it is necessary to set the correct date and time, then reboot the system.
- Slow mounts from AIX V4.2.1 or later clients running NFS Version 3 to AIX V4.1.5 or earlier and other non-AIX servers running NFS Version 2. NFS Version 3 uses TCP by default while NFS Version 2 uses UDP only. This means the initial client mount request using TCP will fail. To provide backwards compatibility, the mount is retried using UDP, but this only occurs after a timeout of some minutes. To avoid this problem, NFSV3 provided the `proto` and `vers` parameters with the `mount` command. These parameters are used with the `-o` option to hardwire the protocol and version for a specific mount. The following example forces the use of UDP and NFSV2 for the mount request:

  ```
  # mount -o proto=udp,vers=2,soft,retry=1 platypus:/test /mnt
  ```

Note

If the `proto` and the `vers` do not match the server, the mount will fail altogether.

- Older non-AIX clients can also incur mount problems. If your environment has such clients, you need to start mountd with the `-n` option:

  ```
  # stopsrc -s rpc.mountd
  ```

```
# startsrc -s rpc.mountd -n
```

- Another mount problem that can occur with older non-AIX clients is when a user that requests a mount is in more than eight groups. The only workaround for this is to decrease the number of groups the user is in or mount via a different user.

9.5 Commands

For a complete reference of the following command use the *AIX Version 4.3 Command Reference* or the online man pages.

9.5.1 ifconfig

Configures or displays network interface parameters for a network using TCP/IP. The command has the following syntax:

```
ifconfig Interface [ AddressFamily [ Address [ DestinationAddress ] ] [
Parameters... ] ]
```

Table 31. Commonly used flags of the ifconfig command

Flag	Description	
AddressFamily	Specifies which network address family to change	
Parameters	alias	Establishes an additional network address for the interface.
	delete	Removes the specified network address.
	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.
	netmask *Mask*	Specifies how much of the address to reserve for subdividing networks into subnetworks.
	up	Marks an interface as active (up). This parameter is used automatically when setting the first address for an interface.
Address	Specifies the network address for the network interface.	

9.5.2 netstat

Shows network status. The command has the following syntax:

```
/bin/netstat [ -n ] [ { -r  -i  -I  Interface } ] [ -f  AddressFamily ] [ -p
Protocol ] [ Interval ]
```

Table 32. Commonly used flags of the netstat command

Flag	Description
-n	Shows network addresses as numbers.
-r	Shows the routing tables.
-i	Shows the state of all configured interfaces.
-I *Interface*	Shows the state of the configured interface specified by the Interface variable.
-f *AddressFamily*	Limits reports of statistics or address control blocks to those items specified by the AddressFamily variable.
-p *Protocol*	Shows statistics about the value specified for the Protocol variable.

9.5.3 route

Manually manipulates the routing tables. The command has the following syntax:

```
route Command [ Family ] [ [ -net | -host ] Destination [-netmask  [ Address
] ] Gateway ] [ Arguments ]
```

Table 33. Commonly used flags of the route command

Flag	Description	
Command	add	Adds a route.
	flush or -f	Removes all routes.
	delete	Deletes a specific route.
	get	Lookup and display the route for a destination.
-net	Indicates that the Destination parameter should be interpreted as a network.	
-host	Indicates that the Destination parameter should be interpreted as a host.	
Destination	Identifies the host or network to which you are directing the route.	
-netmask	Specifies the network mask to the destination address.	
Gateway	Identifies the gateway to which packets are addressed.	

9.5.4 chdev

Changes the characteristics of a device. The command has the following syntax:

```
chdev -l Name [ -a Attribute=Value ... ]
```

Table 34. Commonly used flags of the chdev command

Flag	Description
-l *Name*	Specifies the device logical name, specified by the Name parameter, in the Customized Devices object class whose characteristics are to be changed.
-a *Attribute*=*Value*	Specifies the device attribute value pairs used for changing specific attribute values.

9.5.5 lsattr

Displays attribute characteristics and possible values of attributes for devices in the system. The command has the following syntax:

```
lsattr -E -l Name [ -a Attribute ] ...
```

Table 35. Commonly used flags of the lsattr command

Flag	Description
-E	Displays the attribute names, current values, descriptions, and user-settable flag values for a specific device.
-l *Name*	Specifies the device logical name in the Customized Devices object class whose attribute names or values are to be displayed.
-a *Attribute*	Displays information for the specified attributes of a specific device or kind of device.

9.5.6 exportfs

Exports and unexports directories to NFS clients. The syntax of the `exportfs` command is:

```
exportfs [ -a ] [ -v ] [ -u ] [ -i ] [ -fFile ] [ -oOption [ ,Option ... ]
] [ Directory ]
```

Some useful `exportfs` flags:

Table 36. Commonly used flags of the exportfs command

Flags	Description
-a	Exports all filesets defined in /etc/exports

Flags	Description
-u	Unexports the directories you specify; can be used with -a
-o <option>	Specifies optional characteristics for the exported directory

9.6 References

The following publications contain more information about network tuning procedures.

- *AIX Version 4.3 System Management Concepts: Operating System and Devices*, SC23-4126
- *AIX Versions 3.2 and 4 Performance Tuning Guide*, SC23-2365
- *AIX Version 4.3 Commands Reference, Volume 3*, SC23-4117
- *AIX Version 4.3 Commands Reference, Volume 4*, SC23-4118
- *IBM Certification Study Guide AIX V 4.3 Communication*, SG24-6186

9.7 Quiz

9.7.1 Answers

9.8 Exercises

- Check setting of your network interface with the `lsattr` command
- Check name resolution ordering of your system
- Try to resolve a few hostnames to IP addresses using either `nlsookup` or `host` command

Chapter 10. Performance Problems

In this chapter will the following topics be covered:

- Performance tuning flowchart
- Tools

Performance tuning issues from a problem determination perspecive is concentrated around the skills of interpreting output from various commands. For a well structured approach to such problems, most problem solvers work accordingly to the following flowchart:

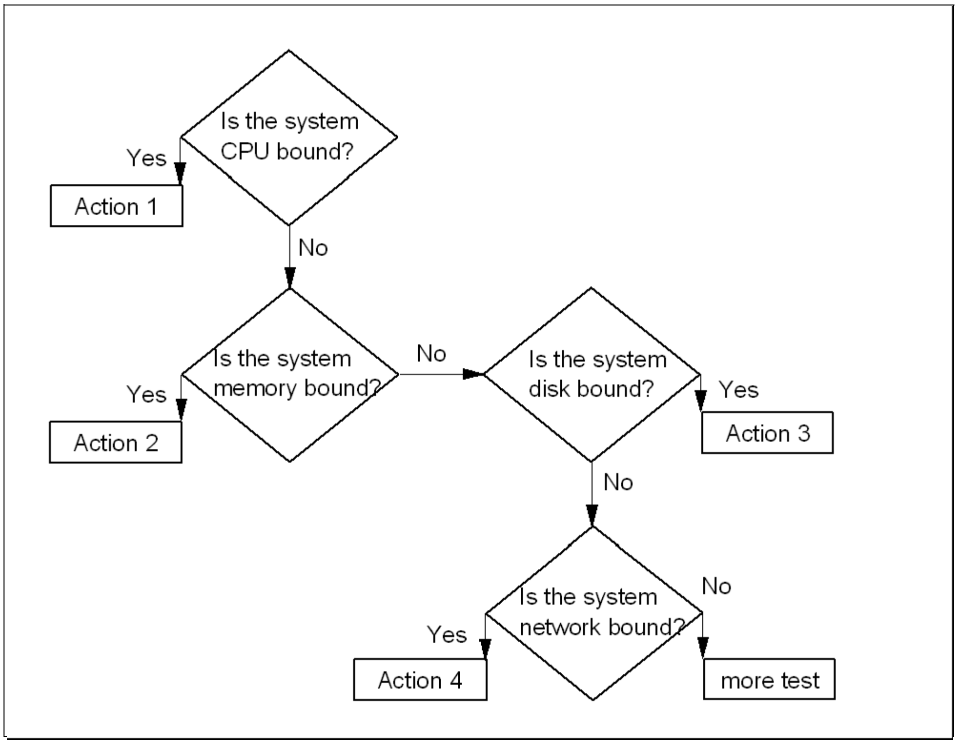


Figure 22. General performance tuning flowchart

When investigating a performance problem, CPU constraint is probably the easiest to find. That is why most performance analysts start with checking for CPU constrains.

10.1 CPU bound system

CPU performance problems can be handled in different way. For example:

- Reschedule tasks to less active time of the day or week
- Change the priority of processes
- Manipulate the scheduler to prioritize foreground processes
- Implement Workload Manager
- By more CPU power

Whatever the solution finally will be, the way to the solution is usually the same; identify the process (or the groups of processes) that constrains the CPU.

When working with CPU performance tuning problems, you have good use of historical performance information for comparison reasons, if such is available. A very useful tool for this task is the `sar` command.

10.1.1 sar

The `sar` command gathers statistical data about the system. Though it can be used to gather some useful data regarding system performance, the `sar` command can increase the system load which will exacerbate a pre-existing performance problem. The system maintains a series of system activity counters which record various activities and provide the data that the `sar` command reports. The `sar` command does not cause these counters to be updated or used; this is done automatically regardless of whether or not the `sar` command runs. It merely extracts the data in the counters and saves it, based on the sampling rate and number of samples specified to the sar command.

There are three situations to use the `sar` command:

Real-time sampling and display

To collect and display system statistic reports immediately, use the following command:

```
# sar -u 2 5
AIX texmex 3 4 000691854C00    01/27/00
17:58:15    %usr    %sys    %wio   %idle
17:58:17      43       9       1      46
17:58:19      35      17       3      45
17:58:21      36      22      20      23
17:58:23      21      17       0      63
```

```
17:58:25      85      12       3       0
Average       44      15       5      35
```

This example is from a single user workstation and shows the CPU utilization.

Display previously captured data

The `-o` and `-f` options (write and read to/from user given data files) allow you to visualize the behavior of your machine in two independent steps. This consumes less resources during the problem-reproduction period. You can use a separate machine to analyze the data by transferring the file because the collected binary file keeps all data the `sar` command needs.

```
# sar -o /tmp/sar.out 2 5 > /dev/null
```

The above command runs the `sar` command in the background, collects system activity data at 2-second intervals for 5 intervals, and stores the (unformatted) `sar` data in the /tmp/sar.out file. The redirection of standard output is used to avoid a screen output.

The following command extracts CPU information from the file and outputs a formatted report to standard output:

```
# sar -f/tmp/sar.out
AIX texmex 3 4 000691854C00    01/27/00
18:10:18    %usr    %sys    %wio   %idle
18:10:20       9       2       0      88
18:10:22      13      10       0      76
18:10:24      37       4       0      59
18:10:26       8       2       0      90
18:10:28      20       3       0      77
Average       18       4       0      78
```

The captured binary data file keeps all information needed for the reports. Every possible `sar` report could therefore be investigated.

System activity accounting via cron daemon

The `sar` command calls a process named sadc to access system data. Two shell scripts (/usr/lib/sa/sa1 and /usr/lib/sa/sa2) are structured to be run by the cron daemon and provide daily statistics and reports. Sample stanzas are included (but commented out) in the /var/spool/cron/crontabs/adm crontab file to specify when the cron daemon should run the shell scripts.

The following lines show a modified crontab for the adm user. Only the comment characters for the data collections were removed:

```
#=========================================================
#      SYSTEM ACTIVITY REPORTS
#  8am-5pm activity reports every 20 mins during weekdays.
#  activity reports every an hour on Saturday and Sunday.
#  6pm-7am activity reports every an hour during weekdays.
#  Daily summary prepared at 18:05.
#=========================================================
0 8-17 * * 1-5 /usr/lib/sa/sa1 1200 3 &
0 * * * 0,6 /usr/lib/sa/sa1 &
0 18-7 * * 1-5 /usr/lib/sa/sa1 &
5 18 * * 1-5 /usr/lib/sa/sa2 -s 8:00 -e 18:01 -i 3600 -ubcwyaqvm &
#=========================================================
```

Collection of data in this manner is useful to characterize system usage over a period of time and to determine peak usage hours.

Another useful feature with `sar` is that the output can be specific about the usage for each processor in a multiprocessor environment, as seen in the following output. The last line is an average output.

```
# sar -P ALL 2 1

AIX client1 3 4 000BC6DD4C00    07/06/00

14:46:52 cpu    %usr    %sys    %wio   %idle
14:46:54  0        0       0       0     100
          1        0       1       0      99
          2        0       0       0     100
          3        0       0       0     100
          -        0       0       0     100
```

If **%usr** plus **%sys** is constantly over 80%, then the system is CPU bound.

10.1.2 vmstat

The `vmstat` command reports statistics about kernel threads, virtual memory, disks, traps and CPU activity. Reports generated by the `vmstat` command can be used to balance system load activity. These system-wide statistics (among all processors) are calculated as averages for values expressed as percentages, and as sums otherwise. most interesting from a CPU point of view are the highlighted two left-hand columns and the highlighted four right-hand columns.

```
# vmstat 2
kthr     memory             page              faults        cpu
----- ----------- ------------------------ ------------ -----------
 r  b   avm   fre  re  pi  po  fr   sr  cy  in   sy  cs us sy id wa
 0  0 16998 14612   0   0   0   0    0   0 101   10   8 55  0 44  0
 0  1 16998 14611   0   0   0   0    0   0 411 2199  54  0  0 99  0
 0  1 16784 14850   0   0   0   0    0   0 412  120  51  0   0 99  0
 0  1 16784 14850   0   0   0   0    0   0 412   88  50  0   0 99  0
```

10.1.2.1 The kthr columns

The **kthr** columns shows how kernel threads are placed on various queues per second over the sampling interval

The r column

The **r** column shows the average number of kernel threads waiting on the run queue per second. This field indicates the number of threads that can be run. This value should be less than five for non-SMP systems. For SMP systems, this value should be less than:

5 x (Ntotal - Nbind)

Where Ntotal stands for total number of processors and Nbind for the number of processors which have been bound to processes, for example, with the `bindprocessor` command.

If this number increases rapidly, examine the applications. But systems may be also running fine with 10 to 15 threads on their run queue, depending on the thread tasks and the amount of time they run.

The b column

The **b** column shows the average number of kernel threads in the wait queue per second. These threads are waiting for resources or I/O. Threads are also located in the wait queue when waiting for one of their thread pages to be paged in. This value is usually near zero. But, if the run-queue value increases, the wait-queue normally also increases. If processes are suspended due to memory load control, the blocked column (b) in the `vmstat` report indicates the increase in the number of threads rather than the run queue.

10.1.2.2 The cpu columns

The four right-hand columns are a breakdown in percentage of CPU time used on user threads, system threads, CPU idle time (running the wait process), and CPU idle time during which the system had outstanding disk/NFS I/O request

The us column

The **us** column shows the percent of CPU time spent in user mode. A UNIX process can execute in either user mode or system (kernel) mode. When in user mode, a process executes within its application code and does not require kernel resources to perform computations, manage memory, or set variables.

The sy column

The **sy** column details the percentage of time the CPU was executing a process in system mode. This includes CPU resource consumed by kernel processes (kprocs) and others that need access to kernel resources. If a process needs kernel resources, it must execute a system call and is thereby switched to system mode to make that resource available. For example, reading or writing of a file requires kernel resources to open the file, seek a specific location, and read or write data, unless memory mapped files are used.

The id column

The **id** column shows the percentage of time which the CPU is idle, or waiting, without pending local disk I/O. If there are no processes available for execution (the run queue is empty), the system dispatches a process called wait. On an SMP system, one wait process per processor can be dispatched. On a uniprocessor system, the process ID (PID) usually is 516. SMP systems will have an idle kproc for each processor. If the `ps` report shows a high aggregate time for this process, it means there were significant periods of time when no other process was ready to run or waiting to be executed on the CPU. The system was therefore mostly idle and waiting for new tasks.

If there are no I/Os pending to a local disk, all time charged to wait is classified as idle time. In operating system version 4.3.2 and earlier, an access to remote disks (NFS-mounted disks) is treated as idle time (with a small amount of sy time to execute the NFS requests) because there is no pending I/O request to a local disk. With operating system version 4.3.3 and later NFS goes through the buffer cache, and waits in those routines are accounted for in the **wa** statistics.

The wa column

The **wa** column details the percentage of time the CPU was idle with pending local disk I/O (in operating system version 4.3.3 and later this is also true for NFS-mounted disks). The method used in operating system version 4.3.2 and earlier versions of the operating system can, under certain circumstances, give an inflated view of **wa** time on SMPs. In AIX Version 4.3.2 and earlier, at each clock interrupt on each processor (100 times a second per processor), a determination is made as to which of the four categories (usr/sys/wio/idle) to place the last 10 ms of time. If any disk I/O is in progress, the **wa** category is incremented. For example, systems with just one thread doing I/O could report over 90 percent **wa** time regardless of the number of CPUs it has.

The change in Version 4.3.3 is to only mark an idle CPU as **wa** if an outstanding I/O was started on that CPU. This method can report much lower **wa** times when just a few threads are doing I/O and the system is otherwise idle. For example, a system with four CPUs and one thread doing I/O will report a maximum of 25 percent **wa** time. A system with 12 CPUs and one thread doing I/O will report a maximum of 8.3 percent **wa** time.

Also, NFS now goes through the buffer cache, and waits in those routines are accounted for in the wa statistics.

A **wa** value over 25 percent could indicate that the disk subsystem might not be balanced properly, or it might be the result of a disk-intensive workload.

10.1.2.3 The fault columns

It may also be worthwhile looking at the **faults** columns, which gives Information about process control, such as trap and interrupt rate.

The in column

In the **in** column is the number of device interrupts per second observed in the interval.

The sy column

In the **sy** column is the number of system calls per second observed in the interval. Resources are available to user processes through well-defined system calls. These calls instruct the kernel to perform operations for the calling process and exchange data between the kernel and the process. Because workloads and applications vary widely, and different calls perform different functions, it is impossible to define how many system calls per-second are too many. But typically, when the **sy** column raises over 10000 calls per second on a uniprocessor, further investigations is called for (on an SMP system the number is 10000 calls per second per processor). One reason could be *polling* subroutines like the select() subroutine. For this column, it is advisable to have a baseline measurement that gives a count for a normal **sy** value.

The cs column

The **cs** column shows the number of context switches per second observed in the interval. The physical CPU resource is subdivided into logical time slices of 10 milliseconds each. Assuming a thread is scheduled for execution, it will run until its time slice expires, until it is preempted, or until it voluntarily gives up control of the CPU. When another thread is given control of the CPU, the context or working environment of the previous thread must be saved and the context of the current thread must be loaded. The operating system has a very efficient context switching procedure, so each switch is inexpensive in terms of resources. Any significant increase in context switches, such as when **cs** is a lot higher than the disk I/O and network packet rate, should be cause for further investigation.

If the system has bad performance because of a lot of threads on the run queue or threads waiting for I/O, then `ps` output will be useful in determine which process has used most CPU resources.

10.1.3 The ps command

The `ps` command is a flexible tool for identifying the programs that are running on the system and the resources they are using. It displays statistics and status information about processes on the system, such as process or thread ID, I/O activity, CPU and memory utilization.

10.1.3.1 ps command output used for CPU usage monitoring

Three of the possible `ps` output columns report CPU usage, each in a different way.

Table 37. CPU related ps output

Column	Value
C	Recent used CPU time for process
TIME	Total CPU time sued by the process since it started
%CPU	Total CPU time used by the process since it started, divided by the elapsed time since the process started. This is a measure of the CPU dependence of the program

The C column

Let’s start with the **C** column. It can be generated by the `-l` and the `-f` flag. In this column is CPU utilization of process or thread reported. The value is incremented each time the system clock ticks and the process or thread is found to be running. Therefore it also can be said to be a process penalty for recent CPU usage. The value is decayed by the scheduler by dividing it by 2 once per second. Large values indicate a CPU intensive process and result in lower process priority whereas small values indicate an I/O intensive process and result in a more favorable priority. In the following example is `tctestprog` running which is a CPU intensive program. The `vmstat` output shows that the CPU is used about 25% by usr processes.

```
# vmstat 2 3
kthr     memory             page              faults        cpu
----- ----------- ------------------------ ------------ -----------
 r  b   avm   fre  re  pi  po  fr   sr  cy  in   sy  cs us sy id wa
 0  0 26468 51691   0   0   0   0    0   0 100   91   6 47  0 53  0
 1  1 26468 51691   0   0   0   0    0   0 415 35918 237 26   2 71  0
 1  1 26468 51691   0   0   0   0    0   0 405    70  26 25  0 75  0
```

Here comes the `ps` command handy. The following formatting sorts the output according to the third column with the biggest value at top, and shows only 15 lines from the total output.

```
# ps -ef | sort +3 -r |head -n 5
    UID   PID  PPID   C    STIME    TTY  TIME CMD
    root 22656 27028 101 15:18:31 pts/11  7:43 ./tctestprog
    root 14718 24618   5 15:26:15 pts/17  0:00 ps -ef
    root  4170     1   3   Jun 15      - 12:00 /usr/sbin/syncd 60
    root 21442 24618   2 15:26:15 pts/17  0:00 sort +3 -r
```

From the example above you can tell that the tctestprog is the process with the most used CPU in recent time.

The TIME column

The second value mentioned is the **TIME** value. This value is generated with all flags, and it shows the total execution time for the process. This calculation does not take into account when the process was started as seen in the following output. The same test program is used again, and event though the C column shows that the process gets a lot of CPU time, it is not yet in top on the **TIME** column:

```
# ps -ef | sort +3 -r |head -n 5
    UID   PID  PPID   C    STIME    TTY  TIME CMD
    root 18802 27028 120 15:40:28 pts/11  1:10 ./tctestprog
    root  9298 24618   3 15:41:38 pts/17  0:00 ps -ef
    root 15782 24618   2 15:41:38 pts/17  0:00 head -n 5
    root 24618 26172   2   Jun 21 pts/17  0:03 ksh

# ps -e |head -n 1 ; ps -e|egrep -v "TIME|0:"|sort +2b -3 -n -r|head -n 10
   PID    TTY  TIME CMD
  4170      - 12:01 syncd
  4460      -  2:07 X
  3398      -  1:48 dtsession
 18802 pts/11  1:14 tctestprog
```

The `syncd`, `X` and `dtsession` is all processes that has been active since IPL, that is why they have accumulated more total TIME than the test program.

The %CPU column

Finally the **%CPU**. This column, generated by the `-u` or the `-v` flags, shows the percentage of time the process has used the CPU since the process started. The value is computed by dividing the time the process uses the CPU, by the elapsed time of the process. In a multi-processor environment, the value is further divided by the number of available CPUs since several threads in the same process can run on different CPUs at the same time. Because the time base over which this data is computed varies, the sum of all **%CPU** fields can exceed 100%. In the example below are two ways to sort the extracted output from a system. The first example includes kprocs, for example PID 516, which is a wait process. The other, more complex command syntax, excludes such kprocs:

```
# ps auxwww |head -n 5
USER       PID %CPU %MEM   SZ  RSS    TTY STAT    STIME  TIME COMMAND
root     18802 25.0  1.0 4140 4160 pts/11 A    15:40:28  5:44 ./tctestprog
root       516 25.0  5.0    8 15136      - A      Jun 15 17246:34 kproc
root       774 20.6  5.0    8 15136      - A      Jun 15 14210:30 kproc
root      1290  5.9  5.0    8 15136      - A      Jun 15 4077:38 kproc

# ps gu|head -n1; ps gu|egrep -v "CPU|kproc"|sort +2b -3 -n -r |head -n 5
USER       PID %CPU %MEM   SZ  RSS    TTY STAT    STIME  TIME COMMAND
root     18802 25.0  1.0 4140 4160 pts/11 A    15:40:28  7:11 ./tctestprog
imnadm   12900  0.0  0.0  264  332      - A Jun 15 0:00 /usr/IMNSearch/ht
root         0  0.0  5.0   12 15140      - A      Jun 15  4:11 swapper
root         1  0.0  0.0  692  764      - A      Jun 15  0:28 /etc/init
root      3398  0.0  1.0 1692 2032      - A Jun 15  1:48 /usr/dt/bin/dtses
```

From the output you can see that the test program, tctestprog, uses about 25% of available CPU resources since process start.

When finding a run-away process the next step in the analysis is to find out what exactly in the process uses CPU. For this is a profiler needed. The AIX profiler of preference is tprof.

10.1.4 The tprof command

The tprof can be used for application tuning and for overall CPU utilization information gathering. The tprof command can be runned over a time period to trace the activity of the CPU.

In the AIX operating system, an interrupt occurs periodically to allow a *housekeeping* kernel routine to run. This occurs 100 times per second. When the tprof command is invoked, it counts every such kernel interrupt as a *tick*. This kernel routine records the process ID and the address of the instruction executing when the interrupt occurred, this information is used by the tprof command. The tprof command also records whether the process counter is in the kernel address space, the user address space, or shared library address space.

10.1.4.1 The tprof summary CPU utilization report

A summary ASCII report with the suffix `.all` is always produced. If no program is specified, the report is named `__prof.all`. If a program is specified, the report is named `__<program>.all`. This report contains an estimate of the amount of CPU time spent in each process that was executing while the `tprof` program was monitoring the system. This report also contains an estimate of the amount of CPU time spent in each of the three address spaces and the amount of time the CPU was idle.

The files containing the reports are left in the working directory. All files created by the tprof command are prefixed by __ (two underscores).

In the following example is a generic report generated:

```
# tprof -x sleep 30
Starting Trace now
Starting  sleep 30
Wed Jun 28 14:58:58 2000
System: AIX server3 Node: 4 Machine: 000BC6DD4C00

Trace is done now
30.907 secs in measured interval
 * Samples from __trc_rpt2
 * Reached second section of __trc_rpt2
```

In this case the `sleep 30` points out to the `tprof` command to run for 30 seconds

The total column

The **Total** collumn in the `__prof.all` is interesting. The first section indicates the use of ticks on a per process basis.

```
Process      PID      TID      Total    Kernel User   Shared    Other
=======      ===      ===      =====    ====== ====   ======    =====
wait          516     517       3237      3237  0        0         0
tctestprg    14746    13783     3207     1     3206      0         0
tctestprg    13730    17293     3195     0     3195      0         0
wait        1032     1033       3105      3105  0        0
wait        1290     1291       138       138   0        0         0
swapper        0        3       10        7     3        0         0
tprof       14156    5443       6         3     3        0         0
trace       16000    14269      3         3     0        0         0
syncd       3158     4735       2         2     0        0         0
tprof       5236      16061     2         2     0        0         0
gil         2064     2839       1         1     0        0         0
gil         2064      3097      1         1     0        0
trace       15536    14847      1         1     0        0         0
```

```
sh          14002    16905    1        1     0        0        0
sleep       14002    16905    1        1     0        0        0
=======      ===      ===    =====   ======    ====   ======    =====
Total                       12910     6503     6407        0        0
```

Each tick is a 1/100 second. By this you can calculate the total amount of available ticks; about 30 seconds, times 100 ticks make a total of 3000 ticks. This according to the theory, but when looking at the output there are over 12000 total ticks. This is because the test system is a 4 way F50, so the available ticks are calculated in the following way:

Time (in seconds) x Number of available CPUs x 100

The user column

If the **user** column shows high values, application tuning might be necessary. In the out put you see that both `tctestprg` used about 3200 ticks. Something around 25% of the total amount of available ticks. This is confirmed with a `ps auxwww` output:

```
#ps auxwww
USER       PID %CPU %MEM   SZ  RSS    TTY STAT    STIME  TIME COMMAND
root     14020 25.0  0.0  300  320  pts/1 A    15:23:55 16:45 ./tctestprg
root     12280 25.0  0.0  300  320  pts/1 A    15:23:57 16:43 ./tctestprg
```

The freq column

In the second section is the total amount of ticks used by a specified type of process defined. Here is the ticks used by all three `wait` processes added together, and the two `tctestprg` are added together. By this the total workload produced by one type of process is shown (as well as the number of instances running of the processes).

```
Process      FREQ    Total   Kernel     User   Shared    Other
=======       ===    =====   ======     ====   ======    =====
wait            3     6480     6480        0        0        0
tctestprg       2     6402        1     6401        0        0
swapper         1       10        7        3        0        0
tprof           2        8        5        3        0        0
trace           2        4        4        0        0        0
gil             2        2        2        0        0        0
syncd           1        2        2        0        0        0
sh              1        1        1        0        0        0
sleep           1        1        1        0        0        0
=======       ===    =====   ======     ====   ======    =====
Total          15    12910     6503     6407        0        0
```

10.2 Memory bound

Memory in AIX is handled by the Virtual Memory Manager (VMM). Virtual memory manager is a method by which real memory appears larger than its true size. The virtual memory system is composed of real memory plus physical disk space where portions of a file that are not currently in use are stored.

VMM maintains a list of free page frames that is used to accommodate pages that must be brought into memory. In memory constrained environments, the VMM must occasionally replenish the free list by moving some of the current data from real memory. This is called page stealing. A page fault is a request to load a 4 KB data page from disk. A number of places are searched in order to find data.

First is the data and instruction caches searched. Next is the *Translation Lookaside Buffer* (TLB) searched. This is an index of recently used virtual addresses with their page frame IDs. If the data is not in the TLB, the *Page Frame Table* (PTF) is consulted. This is an index for all real memory pages, and this index is held in pinned memory. As the table is large, there are indexes to this index. The *Hash Anchor Table* (HAT) links pages of related segments, to get a faster entry point to the main PTF.

From the page stealer perspective the memory is divided into *Computational memory* and *File memory*. The page stealer tries to balance these two types of memory usage when stealing pages. The page replacement algorithm can be manipulated.

Computational memory are pages that belong to the working segment or program text segment.

File memory consists of the remaining pages. These are usually pages from the permanent data file in persistent memory.

When starting a process, a slot has to be assigned and when a process references a virtual memory page that is on the disk, the referenced page must be paged in and probably one or more pages must be paged out, creating I/O traffic and delaying the start up of the process. AIX attempts to steal real memory pages that are unlikely to be referenced in the near future, via the page replacement algorithm. If the system has too little memory, no RAM pages are good candidates to be paged out, as they will be reused in the near future. When this happens, continuos pagein and pageout occurs. This condition is called trashing.

The `vmstat` command can help you in recognizing memory bound systems.

10.2.1 The vmstat command

The vmstat command summarizes the total active virtual memory used by all of the processes in the system, as well as the number of real-memory page frames on the free list. Active virtual memory is defined as the number of virtual-memory working segment pages that have actually been touched. This number can be larger than the number of real page frames in the machine, because some of the active virtual-memory pages may have been written out to paging space.

When determining if a system might be short on memory or if some memory tuning needs to be done, run the vmstat command over a set interval and examine the pi and po columns on the resulting report. These columns indicate the number of paging space page-ins per second and the number of paging space page-outs per second. If the values are constantly non-zero, there might be a memory bottleneck. Having occasional non-zero values is not be a concern because paging is the main principle of virtual memory.

```
# vmstat 2 10
kthr     memory             page              faults        cpu
----- ----------- ------------------------ ------------ -----------
 r  b   avm   fre  re  pi  po  fr   sr  cy  in   sy  cs us sy id wa
 1  3 113726   124   0  14   6 151  600   0 521 5533 816 23 13  7 57
 0  3 113643   346   0   2  14 208  690   0 585 2201 866 16  9  2 73
 0  3 113659   135   0   2   2 108  323   0 516 1563 797 25  7  2 66
 0  2 113661   122   0   3   2 120  375   0 527 1622 871 13  7  2 79
 0  3 113662   128   0  10   3 134  432   0 644 1434 948 22  7  4 67
 1  5 113858   238   0  35   1 146  422   0 599 5103 903 40 16  0 44
 0  3 113969   127   0   5  10 153  529   0 565 2006 823 19  8  3 70
 0  3 113983   125   0  33   5 153  424   0 559 2165 921 25  8  4 63
 0  3 113682   121   0  20   9 154  470   0 608 1569 1007 15  8  0 77
 0  4 113701   124   0   3  29 228  635   0 674 1730 1086 18  9  0 73
```

Notice the high I/O wait in the output and also the number of threads on the blocked queue. Most likely, the I/O wait is due to the paging in/out from paging space.

To see if the system has performance problems with its VMM, examine the columns under memory and page:

10.2.1.1 The memory columns

Provides information about the real and virtual memory.

The avm column

The **avm** (Active Virtual Memory) column gives the average number of 4 K pages that are allocated to paging space. The **avm** value can be used to calculate the amount of paging space assigned to executing processes.

Note

The `vmstat` command (**avm** column), `ps` command (**SIZE, SZ**), and other utilities report the amount of virtual memory actually accessed, but with DPSA, the paging space may not get touched. The `svmon` command (up through operating system version 4.3.2) shows the amount of paging space being used, so this value may be much smaller than the **avm** value of the `vmstat` command.

For more information on DPSA see the *Performance Management Guide* or the *Performance Tuning Study Guide*, SG24-6184

The number in the **avm** field divided by 256 will yield the approximate number of megabytes (MB) allocated to paging space system wide. Prior to operating system version 4.3.2 the same information is reflected in the Percent Used column of the `lsps -s` command output or with the `svmon -G` command under the **pg space inuse** field.

The fre column

The **fre** column shows the average number of free memory pages. A page is a 4 KB area of real memory. The system maintains a buffer of memory pages, called the free list, that will be readily accessible when the VMM needs space. The minimum number of pages that the VMM keeps on the free list is determined by the **minfree** parameter of the `vmtune` command. When an application terminates, all of its working pages are immediately returned to the free list. Its persistent pages (files), however, remain in RAM and are not added back to the free list until they are stolen by the VMM for other programs. Persistent pages are also freed if the corresponding file is deleted.

For this reason, the **fre** value may not indicate all the real memory that can be readily available for use by processes. If a page frame is needed, then persistent pages related to terminated applications are among the first to be handed over to another program.

10.2.1.2 The page columns

The **page** columns shows information about page faults and paging activity. These are averaged over the interval and given in units per second.

The pi column

The **pi** column details the number (rate) of pages paged in from paging space. Paging space is the part of virtual memory that resides on disk. It is used as an overflow when memory is over committed. Paging space consists of logical volumes dedicated to the storage of working set pages that have been stolen from real memory. When a stolen page is referenced by the process, a page fault occurs, and the page must be read into memory from paging space.

Due to the variety of configurations of hardware, software and applications, there is no absolute number to look out for. But five page-ins per second per paging space should be the upper limit. This guideline should not be rigidly adhered to, but used as a reference. This field is important as a key indicator of paging-space activity. If a page-in occurs, there must have been a previous page-out for that page. It is also likely in a memory-constrained environment that each page-in will force a different page to be stolen and, therefore, paged out. But systems could also work fine when they have close to 10 **pi** per second for 1 minute and then work without any page-ins.

The po column

The **po** column shows the number (rate) of pages paged out to paging space. Whenever a page of working storage is stolen, it is written to paging space, if it does not yet reside in paging space or if it was modified. If not referenced again, it will remain on the paging device until the process terminates or disclaims the space. Subsequent references to addresses contained within the faulted-out pages results in page faults, and the pages are paged in individually by the system. When a process terminates normally, any paging space allocated to that process is freed. If the system is reading in a significant number of persistent pages (files), you might see an increase in po without corresponding increases in **pi**. This does not necessarily indicate thrashing, but may warrant investigation into data-access patterns of the applications.

The fr column

The **fr** collumn shows the number of pages that were freed per second by the page-replacement algorithm during the interval. As the VMM page-replacement routine scans the Page Frame Table (PFT), it uses criteria to select which pages are to be stolen to replenish the free list of available memory frames. The criteria include both kinds of pages, working (computational) and file (persistent) pages. Just because a page has been freed, it does not mean that any I/O has taken place. For example, if a persistent storage (file) page has not been modified, it will not be written back to the disk. If I/O is not necessary, minimal system resources are required to free a page. If the ratio of **po/fr** is greater the 1 to 6, this could indicate a trashing system.

The sr column

The **sr** column shows the number of pages that were examined per second by the page-replacement algorithm during the interval. The VMM page-replacement code scans the PFT and steals pages until the number of frames on the free list is at least the maxfree value. The page-replacement code might have to scan many entries in the PFT before it can steal enough to satisfy the free list requirements. With stable, unfragmented memory, the scan rate and free rate might be nearly equal. On systems with multiple processes using many different pages, the pages are more volatile and disjoint. In this scenario, the scan rate might greatly exceed the free rate.

Memory is over committed when the ratio of fr to sr (fr:sr) is high.

An **fr:sr** ratio of 1:4 means that for every page freed, four pages had to be examined. It is difficult to determine a memory constraint based on this ratio alone, and what constitutes a high ratio is workload/application dependent.

The cy column

The **cy** columns shows the number of cycles per second of the clock algorithm. The VMM uses a technique known as the clock algorithm to select pages to be replaced. This technique takes advantage of a referenced bit for each page as an indication of what pages have been recently used (referenced). When the page-stealer routine is called, it cycles through the PFT, examining each page's referenced bit. The **cy** column shows how many times per second the page-replacement code has scanned the PFT. Because the free list can be replenished without a complete scan of the PFT and because all of the `vmstat` fields are reported as integers, this field is usually zero. If not, it indicates a complete scan of the PFT, and the stealer has to scan the PFT again, because **fre** is still under the **maxfree** value.

One way to determine the appropriate amount of RAM for a system is to look at the largest value for **avm** as reported by the `vmstat` command. Multiply that by 4 K to get the number of bytes and then compare that to the number of bytes of RAM on the system. Ideally, **avm** should be smaller than total RAM. If not, some amount of virtual memory paging will occur. How much paging occurs will depend on the difference between the two values. Remember, the idea of virtual memory is that it gives us the capability of addressing more memory than we have (some of the memory is in RAM and the rest is in paging space). But if there is far more virtual memory than real memory, this could cause excessive paging which then results in delays. If **avm** is lower than RAM, then paging-space paging could be caused by RAM being filled up with file pages. In that case, tuning the **minperm/maxperm** values, could reduce the amount of paging-space paging. This can be don with the `vmtune` command. For more information on the `vmtune` command se *Performance Management Guide, Performance Tuning Study Guide*, SG24-6184 and *Commands Reference - Volume 6*, SBOF-1877

Another useful command for memory performance problem determination is the `ps` command.

10.2.2 The ps command

The `ps` command is a flexible tool for identifying the programs that are running on the system and the resources they are using. It displays statistics and status information about processes on the system, such as process or thread ID, I/O activity, CPU and memory utilization.

10.2.2.1 ps command output used for memory usage monitoring

The `ps` command gives useful information on memory usage. The most useful output is presented in the following columns:

Table 38. Memory related ps output

Column	Value
SIZE	The virtual size of the data section of the process in 1KB units
RSS	The real-memory size of the process in 1KB units
%MEM	The percentage of real memory used by this process

The SIZE column

The `v` flag generates the **SIZE** column. This is the virtual size (in paging space) in kilobytes of the data section of the process (displayed as **SZ** by other flags). This number is equal to the number of working segment pages of the process that have been touched times four. If some working segment pages are currently paged out, this number is larger than the amount of real memory being used. **SIZE** includes pages in the private segment and the shared-library data segment of the process.

For example:

```
# ps av |sort +5 -r |head -n 5
   PID    TTY STAT  TIME PGIN  SIZE   RSS   LIM  TSIZ   TRS %CPU %MEM
COMMAND
 25298 pts/10 A     0:00    0  2924    12 32768   159     0  0.0  0.0 smitty
 13160   lft0 A     0:00   17   368    72 32768    40 60  0.0 0.0/usr/sbin
 27028 pts/11 A     0:00   90   292   416 32768   198   232  0.0  1.0 ksh
 24618 pts/17 A     0:04  318   292   408 32768   198   232  0.0  1.0 ksh
```

The RSS column

The `v` flag also produces the **RSS** column as seen in the previous example. This is the real-memory (resident set) size in kilobytes of the process. This number is equal to the sum of the number of working segment and code segment pages in memory times four. Remember that code segment pages are shared among all of the currently running instances of the program. If 26 ksh processes are running, only one copy of any given page of the ksh executable program would be in memory, but the ps command would report that code segment size as part of the **RSS** of each instance of the ksh program.

If you want to sort on the 6th column, you will get the output accordingly to the **RSS** column, as shown in the following example:

```
#ps av |sort +6 -r |head -n 5
PID    TTY STAT  TIME PGIN  SIZE   RSS   LIM  TSIZ   TRS %CPU %MEM COMMAND
21720  pts/1 A     0:00    1   288   568 32768   198   232  0.0  1.0 ksh
27028 pts/11 A     0:00   90   292   416 32768   198   232  0.0  1.0 ksh
24618 pts/17 A     0:04  318   292   408 32768   198   232  0.0  1.0 ksh
15698  pts/1 A     0:00    0   196   292 32768    52    60  0.0  0.0 ps av
```

The %MEM column

Finally the **%MEM** column, generated by the `u` and the `v` flags. This is calculated as the sum of the number of working segment and code segment pages in memory times four (that is, the **RSS** value), divided by the size of the real memory of the machine in KB, times 100, rounded to the nearest full percentage point. This value attempts to convey the percentage of real memory being used by the process. Unfortunately, like **RSS**, it tends the exaggerate the cost of a process that is sharing program text with other processes. Further, the rounding to the nearest percentage point causes all of the processes in the system that have RSS values under .005 times real memory size to have a **%MEM** of 0.0.

For example:

```
# ps au |head -n 1; ps au |egrep -v "RSS"|sort +3 -r |head -n 5
USER       PID %CPU %MEM   SZ  RSS    TTY STAT    STIME  TIME COMMAND
root     22750  0.0 21.0 20752 20812 pts/11 A 17:55:51  0:00./tctestprog2
root     21720  0.0  1.0  484  568  pts/1 A    17:16:14  0:00 ksh
root     25298  0.0  0.0 3080   12 pts/10 A      Jun 16  0:00 smitty
root     27028  0.0  0.0  488  416 pts/11 A    14:53:27  0:00 ksh
root     24618  0.0  0.0  488  408 pts/17 A      Jun 21  0:04 ksh
```

Finally you can combine all these column in one output, by using the gv flags. For example:

```
# ps gv|head -n 1; ps gv|egrep -v "RSS" | sort +6b -7 -n -r |head -n 5
PID    TTY STAT  TIME PGIN  SIZE   RSS   LIM  TSIZ TRS %CPU %MEM COMMAND
15674 pts/11 A  0:01    0 36108 36172  32768  5    24  0.6 24.0 ./tctestp
22742 pts/11 A  0:00    0 20748 20812  32768 5    24  0.0 14.0 ./backups
10256  pts/1 A  0:00    0 15628 15692  32768 5    24  0.0 11.0 ./tctestp
2064      - A   2:13    5    64  6448     xx  0  6392  0.0  4.0 kproc
1806      - A   0:20    0    16  6408     xx  0  6392  0.0  4.0 kproc
```

In the previous output are also these columns interesting:

The PGIN column
Number of page-ins caused by page faults. Since all I/O is classified as page faults, this is basically a measure of I/O volume.

The TSIZ column
Size of text (shared-program) image. This is the size of the text section of the executable file. Pages of the text section of the executable program are only brought into memory when they are touched, that is, branched to or loaded from. This number represents only an upper bound on the amount of text that could be loaded. The **TSIZ** value does not reflect actual memory usage.

The TRS column
Size of the resident set (real memory) of text. This is the number of code segment pages times 4. This number exaggerates memory use for programs of which multiple instances are running.

10.2.3 The svmon command

The svmon command provides a more in-depth analysis of memory usage. It is more informative, but also more intrusive, than the vmstat and ps commands. The svmon command captures a snapshot of the current state of memory. There are some significant changes in the flags and in the output from the svmon command between Version 4.3.2 and Version 4.3.3.

You can use four different reports to analyze the displayed information:

- Global (-G)

  Displays statistics describing the real memory and paging space in use for the whole system.

- Process (-P)

  Displays memory usage statistics for active processes.

- Segment (-S)

Displays memory usage for a specified number of segments or the top ten highest memory-usage processes in descending order.

- Detailed Segment (-D)

  Displays detailed information on specified segments.

Additional reports are available in Version 4.3.3 and later, as follows:

- User (-U)

  Displays memory usage statistics for the specified login names. If no list of login names is supplied, memory usage statistics display all defined login names.

- Command (-C)

  Displays memory usage statistics for the processes specified by command name.

- Workload Management Class (-W)

  Displays memory usage statistics for the specified workload management classes. If no classes are supplied, memory usage statistics display all defined classes.

To support 64-bit applications, the output format of the `svmon` command was modified inVersion 4.3.3 and later. Additional reports are available in operating system versions later than 4.3.3, as follows:

- Frame (-F)

  Displays information about frames. When no frame number is specified, the percentage of used memory is reported. When a frame number is specified, information about that frame is reported.

- Tier (-T)

  Displays information about tiers, such as the tier number, the superclass name when the -a flag is used, and the total number of pages in real memory from segments belonging to the tier.

For more information in the svmon command, see *Performance Management Guide, Performance Tuning Study Guide*, SG24-6184 and *Commands Reference - Volume 5*, SBOF-1877

10.3 Disk bound

The set of operating system commands, library subroutines, and other tools that allow you to establish and control logical volume storage is called the

Logical Volume Manager (LVM). The Logical Volume Manager (LVM) controls disk resources by mapping data between a more simple and flexible logical view of storage space and the actual physical disks. The LVM does this using a layer of device driver code that runs above traditional disk device drivers. If you are not familiar with the concepts of the LVM, see, for example, *System Management Concepts: Operating System*, SC23-4311 and *Devices and Performance Management Guide* and *Performance Tuning Study Guide*, SG24-6184.

While an operating system's file is conceptually a sequential and contiguous string of bytes, the physical reality might be very different. Fragmentation may arise from multiple extensions to logical volumes as well as allocation/release/reallocation activity within a file system. A file system is fragmented when its available space consists of large numbers of small chunks of space, making it impossible to write out a new file in contiguous blocks.

Access to files in a highly fragmented file system may result in a large number of seeks and longer I/O response times (seek latency dominates I/O response time). For example, if the file is accessed sequentially, a file placement that consists of many, widely separated chunks requires more seeks than a placement that consists of one or a few large contiguous chunks.

If the file is accessed randomly, a placement that is widely dispersed requires longer seeks than a placement in which the file's blocks are close together.

The VMM tries to anticipate the future need for pages of a sequential file by observing the pattern in which a program is accessing the file. When the program accesses two successive pages of the file, the VMM assumes that the program will continue to access the file sequentially, and the VMM schedules additional sequential reads of the file. This is called *Sequential-Access Read Ahead*. These reads are overlapped with the program processing, and will make the data available to the program sooner than if the VMM had waited for the program to access the next page before initiating the I/O. The number of pages to be read ahead is determined by two VMM thresholds:

minpgahead - Number of pages read ahead when the VMM first detects the sequential access pattern. If the program continues to access the file sequentially, the next read ahead will be for 2 times **minpgahead**, the next for 4 times **minpgahead**, and so on until the number of pages reaches **maxpgahead**.

maxpgahead - Maximum number of pages the VMM will read ahead in a sequential file.

If the program deviates from the sequential-access pattern and accesses a page of the file out of order, sequential read ahead is terminated. It will be resumed with **minpgahead** pages if the VMM detects a resumption of sequential access by the program. The values of minpgahead and maxpgahead can be set with the `vmtune` command.

To increase write performance, limit the number of dirty file pages in memory, reduce system overhead, and minimize disk fragmentation, the file system divides each file into 16 KB partitions. The pages of a given partition are not written to disk until the program writes the first byte of the next 16 KB partition. At that point, the file system forces the four dirty pages of the first partition to be written to disk. The pages of data remain in memory until their frames are reused, at which point no additional I/O is required. If a program accesses any of the pages before their frames are reused, no I/O is required.

If a large number of dirty file pages remain in memory and do not get reused, the sync daemon writes them to disk, which might result in abnormal disk utilization. To distribute the I/O activity more efficiently across the workload, *write-behind* can be turned on to tell the system how many pages to keep in memory before writing them to disk. The write-behind threshold is on a per-file basis, which causes pages to be written to disk before the sync daemon runs. The I/O is spread more evenly throughout the workload.

There are two types of write-behind: *sequential* and *random*. The size of the write-behind partitions and the write-behind threshold can be changed with the `vmtune` command.

Normal files are automatically mapped to segments to provide mapped files. This means that normal file access bypasses traditional kernel buffers and block I/O routines, allowing files to use more memory when the extra memory is available (file caching is not limited to the declared kernel buffer area).

Because most writes are asynchronous, FIFO I/O queues of several megabytes can build up, which can take several seconds to complete. The performance of an interactive process is severely impacted if every disk read spends several seconds working its way through the queue. In response to this problem, the VMM has an option called *I/O pacing* to control writes.

I/O pacing does not change the interface or processing logic of I/O. It simply limits the number of I/Os that can be outstanding against a file. When a process tries to exceed that limit, it is suspended until enough outstanding requests have been processed to reach a lower threshold.

Disk-I/O pacing is intended to prevent programs that generate very large amounts of output from saturating the system's I/O facilities and causing the response times of less-demanding programs to deteriorate. Disk-I/O pacing enforces per-segment (which effectively means per-file) *high*- and *low-water marks* on the sum of all pending I/Os. When a process tries to write to a file that already has high-water mark pending writes, the process is put to sleep until enough I/Os have completed to make the number of pending writes less than or equal to the low-water mark. The logic of I/O-request handling does not change. The output from high-volume processes is slowed down somewhat.

When gathering information on I/O performance, the first command to use is normally `iostat`.

10.3.1 The iostat command

The `iostat` command is used for monitoring system input/output device loading by observing the time the physical disks are active in relation to their average transfer rates. The `iostat` command generates reports that can be used to change system configuration to better balance the input/output load between physical disks and adapters. the `iostat` command gathers its information on the protocol layer.

In AIX Version 4.3.3 has some significant changes to the output from the iostat command occurred. These changes are similar to the changes described for the vmstat commands in , “The wa column” on page 212.

10.3.1.1 The TTY columns

The two columns of TTY information (**tin** and **tout**) in the `iostat` output show the number of characters read and written by all TTY devices. This includes both real and pseudo TTY devices. Real TTY devices are those connected to an asynchronous port. Some pseudo TTY devices are shells, `telnet` sessions, and `aixterm` windows. Because the processing of input and output characters consumes CPU resources, look for a correlation between increased TTY activity and CPU utilization. If such a relationship exists, evaluate ways to improve the performance of the TTY subsystem. Steps that could be taken include changing the application program, modifying TTY port parameters during file transfer, or perhaps upgrading to a faster or more efficient asynchronous communications adapter.

10.3.1.2 The CPU columns

The CPU statistics columns (**%user**, **%sys**, **%idle**, and **%iowait**) provide a breakdown of CPU usage. This information is also reported in the vmstat command output in the columns labeled us, sy, id, and wa. For a detailed explanation for the values, see , “The us column” on page 211, , “The sy column” on page 212, , “The id column” on page 212 and , “The wa column” on page 212.

On systems running one application, high I/O wait percentage might be related to the workload. On systems with many processes, some will be running while others wait for I/O. In this case, the **% iowait** can be small or zero because running processes *hide* some wait time. Although **% iowait** is low, a bottleneck can still limit application performance.

If the iostat command indicates that a CPU-bound situation does not exist, and **% iowait** time is greater than 20 percent, you might have an I/O or disk-bound situation. This situation could be caused by excessive paging due to a lack of real memory. It could also be due to unbalanced disk load, fragmented data or usage patterns. For resolving such problems an reorganization of logical volumes or a defragmentation of file systems might be necessary. For an unbalanced disk load, the same `iostat` report provides the necessary information. But for information about file systems or logical volumes, which are logical resources, you must use tools such as the `filemon` or `fileplace` commands.

10.3.1.3 The Drive reports

When you suspect a disk I/O performance problem, use the `iostat` command. To avoid the information about the TTY and CPU statistics, use the `-d` option. In addition, the disk statistics can be limited to the important disks by specifying the disk names.Remember that the first set of data represents all activity since system startup. In the following example such information is not available:

```
# iostat 1 2

tty:      tin         tout   avg-cpu:  % user    % sys     % idle    % iowait
          0.0          6.2             16.3      0.0       83.6       0.0
                 " Disk history since boot not available. "


tty:      tin         tout   avg-cpu:  % user    % sys     % idle    % iowait
          0.0        192.7            100.0      0.0        0.0       0.0

Disks:         % tm_act     Kbps      tps    Kb_read   Kb_wrtn
hdisk1            0.0       0.0       0.0          0         0
```

```
hdisk3           0.0       0.0       0.0          0          0
hdisk2           0.0       0.0       0.0          0          0
cd0              0.0       0.0       0.0          0          0
```

In such a case, this can be turned on with the following command;

```
# chdev -l sys0 -a iostat=true
sys0 changed
```

The disks column

Shows the names of the physical volumes. They are either hdisk or cd followed by a number. If physical volume names are specified with the iostat command, only those names specified are displayed.

The %tm_act column

Indicates the percentage of time that the physical disk was active (bandwidth utilization for the drive) or, in other words, the total time disk requests are outstanding. A drive is active during data transfer and command processing, such as seeking to a new location. The *disk active time* percentage is directly proportional to resource contention and inversely proportional to performance. As disk use increases, performance decreases and response time increases. In general, when the utilization exceeds 70 percent, processes are waiting longer than necessary for I/O to complete because most UNIX processes block (or sleep) while waiting for their I/O requests to complete. Look for busy versus idle drives. Moving data from busy to idle drives can help alleviate a disk bottleneck. Paging to and from disk will contribute to the I/O load.

The Kbps column

Indicates the amount of data transferred (read or written) to the drive in KB per second. This is the sum of **Kb_read** plus **Kb_wrtn**, divided by the seconds in the reporting interval.

The tps column

Indicates the number of transfers per second that were issued to the physical disk. A transfer is an I/O request through the device driver level 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.

The Kb_read column

Reports the total data (in KB) read from the physical volume during the measured interval.

The Kb_wrtn column

Shows the amount of data (in KB) written to the physical volume during the measured interval.

Taken alone, there is no unacceptable value for any of the above fields because statistics are too closely related to application characteristics, system configuration, and type of physical disk drives and adapters. Therefore, when you are evaluating data, look for patterns and relationships. The most common relationship is between disk utilization (**%tm_act**) and data transfer rate (**tps**).

To draw any valid conclusions from this data, you have to understand the application’s disk data access patterns such as sequential, random, or combination, as well as the type of physical disk drives and adapters on the system. For example, if an application reads/writes sequentially, you should expect a high disk transfer rate (Kbps) when you have a high disk busy rate (**%tm_act**). Columns **Kb_read** and **Kb_wrtn** can confirm an understanding of an application's read/write behavior. However, these columns provide no information on the data access patterns.

Generally you do not need to be concerned about a high disk busy rate (**%tm_act**) as long as the disk transfer rate (**Kbps**) is also high. However, if you get a high disk busy rate and a low disk transfer rate, you may have a fragmented logical volume, file system, or individual file.

Discussions of disk, logical volume and file system performance sometimes lead to the conclusion that the more drives you have on your system, the better the disk I/O performance. This is not always true because there is a limit to the amount of data that can be handled by a disk adapter. The disk adapter can also become a bottleneck. If all your disk drives are on one disk adapter, and your hot file systems are on separate physical volumes, you might benefit from using multiple disk adapters. Performance improvement will depend on the type of access.

To see if a particular adapter is saturated, use the `iostat` command and add up all the **Kbps** amounts for the disks attached to a particular disk adapter. For maximum aggregate performance, the total of the transfer rates (**Kbps**) must be below the disk adapter throughput rating. In most cases, use 70 percent of the throughput rate. In operating system versions later than 4.3.3 the -`a` or `-A` option will display this information.

When finding performance problems due to disk I/O, the next step is to find the file system causing the problem. This can be done with the `filemon` command.

10.3.2 The filemon command

The `filemon` command uses the trace facility to obtain a detailed picture of I/O activity during a time interval on the various layers of file system utilization, including the logical file system, virtual memory segments, LVM, and physical disk layers. Both summary and detailed reports are generated. Tracing is started by the `filemon` command, optionally suspended with the `trcoff` sub command and resumed with the `trcon` sub command, and terminated with the `trcstop` sub command. As soon as tracing is terminated, the `filemon` command writes its report to stdout. More on the `filemon` command in *Commands Reference, Volume 2*, SBOF-1877, *Performance Management Guide* and *Performance Tuning Study Guide*, SG24-6184

If a file is identified as the problem the `fileplace` command can be used to see how the file is stored.

10.3.3 The fileplace command

The `fileplace` command displays the placement of a specified file within the logical or physical volumes containing the file. By default, the `fileplace` command lists to standard output the ranges of logical volume fragments allocated to the specified file. More on the `fileplace` command in Commands *Reference, Volume 2*, SBOF-1877, *Performance Management Guide* and *Performance Tuning Study Guide*, SG24-6184

10.4 Network bound

When performance problems arise, your system might be totally innocent, while the real culprit is buildings away. An easy way to tell if the network is affecting overall performance is to compare those operations that involve the network with those that do not. If you are running a program that does a considerable amount of remote reads and writes and it is running slowly, but everything else seems to be running normally, then it is probably a network problem. Some of the potential network bottlenecks can be caused by the following:

Client-network interface

Network bandwidth

Network topology

Server network interface

Server CPU load

Server memory usage

Server bandwidth

Inefficient configuration

A large part of network tuning involves tuning TCP/IP to achieve maximum throughput. With the new high bandwidth interfaces like FIDDI and SOCC, this has become even more important.

The first command to use for gathering information on network performance is the `netstat` command.

10.4.1 The netstat command

The `netstat` command is used to show network status. Traditionally, it is used more for problem determination than for performance measurement. However, the `netstat` command can be used to determine the amount of traffic on the network to ascertain whether performance problems are due to network congestion.

10.4.1.1 netstat -i output

Shows the state of all configured interfaces.

The following example shows the statistics for a workstation with an integrated Ethernet and a Token-Ring adapter:

```
# netstat -i
Name  Mtu   Network     Address            Ipkts Ierrs    Opkts Oerrs  Coll
lo0   16896 <Link>                        144834     0   144946     0     0
lo0   16896 127         localhost         144834     0   144946     0     0
tr0   1492  <Link>10.0.5a.4f.3f.61        658339     0   247355     0     0
tr0   1492  9.3.1       ah6000d           658339     0   247355     0     0
en0   1500  <Link>8.0.5a.d.a2.d5               0     0      112     0     0
en0   1500  1.2.3       1.2.3.4                0     0      112     0     0
```

The count values are summarized since system startup.

The Mtu column

Maximum transmission unit. The maximum size of packets in bytes that are transmitted using the interface.

The Ipkts column

Total number of packets received.

The Ierrs column

Total number of input errors. For example, malformed packets, checksum errors, or insufficient buffer space in the device driver.

The Opkts column

Total number of packets transmitted.

The Oerrs column

Total number of output errors. For example, a fault in the local host connection or adapter output queue overrun.

The Coll column

Number of packet collisions detected.

Tuning guidelines based on netstat -i

If the number of errors during input packets is greater than 1 percent of the total number of input packets (from the command `netstat -i`); that is,

Ierrs > 0.01 x **Ipkts**

Then run the `netstat -m` command to check for a lack of memory.

If the number of errors during output packets is greater than 1 percent of the total number of output packets (from the command `netstat -i`); that is,

Oerrs > 0.01 x **Opkts**

Then increase the send queue size (**xmt_que_size**) for that interface. The size of the xmt_que_size could be checked with the following command:

```
# lsattr -El adapter
```

If the collision rate is greater than 10 percent, that is,

Coll / **Opkts** > 0.1

Then there is a high network utilization, and a reorganization or partitioning may be necessary. Use the `netstat -v` or `entstat` command to determine the collision rate.

10.4.1.2 netstat -i -Z

This function of the netstat command clears all the statistic counters for the `netstat -i` command to zero.

10.4.1.3 netstat -m output

Displays the statistics recorded by the mbuf memory-management routines. The most useful statistics in the output of the `netstat -m` command are the counters that show the requests for mbufs denied and non-zero values in the failed column. If the requests for mbufs denied is not displayed, then this must be an SMP system running operating system version 4.3.2 or later; for performance reasons, global statistics are turned off by default. To enable the global statistics, set the `no` parameter **extended_netstats** to 1. This can be done by changing the /etc/rc.net file and rebooting the system.

The following example shows the first part of the `netstat -m` output with extended_netstats set to 1:

```
# netstat -m
29 mbufs in use:
16 mbuf cluster pages in use
71 Kbytes allocated to mbufs
0 requests for mbufs denied
0 calls to protocol drain routines


Kernel malloc statistics:


******* CPU 0 *******
By size       inuse     calls failed   delayed    free   hiwat   freed
32              419    544702      0         0     221     800       0
64              173     22424      0         0      19     400       0
128             121     37130      0         0     135     200       4
256            1201 118326233      0         0     239     480     138
512             330    671524      0         0      14      50      54
1024             74    929806      0         0      82     125       2
2048            384   1820884      0         0       8     125    5605
4096            516   1158445      0         0      46     150      21
8192              9      5634      0         0       1      12      27
16384             1      2953      0         0      24      30      41
32768             1         1      0         0       0    1023       0
```

```
By type       inuse   calls failed delayed  memuse  memmax  mapb

Streams mblk statistic failures:

0 high priority mblk failures

0 medium priority mblk failures

0 low priority mblk failures
```

If global statistics are not on and you want to determine the total number of requests for mbufs denied, add up the values under the failed columns for each CPU. If the `netstat -m` command indicates that requests for mbufs or clusters have failed or been denied, then you may want to increase the value of **thewall** by using the `no -o thewall=NewValue` command.

Beginning with operating system version 4.3.3, a delayed column was added. If the requester of an mbuf specified the M_WAIT flag, then if an mbuf was not available, the thread is put to sleep until an mbuf is freed and can be used by this thread. The failed counter is not incremented in this case; instead, the delayed column will be incremented. Prior to operating system version 4.3.3, the failed counter was also not incremented, but there was no delayed column.

Also, if the currently allocated amount of network memory is within 85 percent of thewall, you may want to increase thewall. If the value of thewall is increased, use the `vmstat` command to monitor total memory use to determine if the increase has had a negative impact on overall memory performance.

10.4.1.4 netstat -v output

The `netstat -v` command displays the statistics for each Common Data Link Interface (CDLI)-based device driver that is in operation. Interface-specific reports can be requested using the `tokstat`, `entstat`, `fddistat`, or `atmstat` commands.

Every interface has its own specific information and some general information. The most important output fields follows:

Transmit and Receive Errors

Number of output/input errors encountered on this device. This field counts unsuccessful transmissions due to hardware/network errors.These unsuccessful transmissions could also slow down the performance of the system.

Max Packets on S/W Transmit Queue

Maximum number of outgoing packets ever queued to the software transmit queue. An indication of an inadequate queue size is if the maximal transmits queued equals the current queue size (**xmt_que_size**). This indicates that the queue was full at some point.

To check the current size of the queue, use the `lsattr -El` adapter command (where adapter is, for example, tok0 or ent0). Because the queue is associated with the device driver and adapter for the interface, use the adapter name, not the interface name. Use the SMIT or the `chdev` command to change the queue size.

S/W Transmit Queue Overflow

Number of outgoing packets that have overflowed the software transmit queue. A value other than zero requires the same actions as would be needed if the Max Packets on S/W Transmit Queue reaches the **xmt_que_size**. The transmit queue size must be increased.

Broadcast Packets

Number of broadcast packets received without any error. If the value for broadcast packets is high, compare it with the total received packets. The received broadcast packets should be less than 20 percent of the total received packets. If it is high, this could be an indication of a high network load; use multicasting. The use of IP multicasting enables a message to be transmitted to a group of hosts, instead of having to address and send the message to each group member individually.

DMA Overrun

The DMA Overrun statistic is incremented when the adapter is using DMA to put a packet into system memory and the transfer is not completed. There are system buffers available for the packet to be placed into, but the DMA operation failed to complete. This occurs when the MCA bus is too busy for the adapter to be able to use DMA for the packets. The location of the adapter on the bus is crucial in a heavily loaded system. Typically an adapter in a lower slot number on the bus, by having the higher bus priority, is using so much of the bus that adapters in higher slot numbers are not being served. This is particularly true if the adapters in a lower slot number are ATM or SSA adapters.

Max Collision Errors

Number of unsuccessful transmissions due to too many collisions. The number of collisions encountered exceeded the number of retries on the adapter.

Late Collision Errors
Number of unsuccessful transmissions due to the late collision error.

Timeout Errors
Number of unsuccessful transmissions due to adapter reported timeout errors.

Single Collision Count
Number of outgoing packets with single (only one) collision encountered during transmission.

Multiple Collision Count
Number of outgoing packets with multiple (2 - 15) collisions encountered during transmission.

Receive Collision Errors
Number of incoming packets with collision errors during reception.

No mbuf Errors
Number of times that mbufs were not available to the device driver. This usually occurs during receive operations when the driver must obtain memory buffers to process inbound packets. If the mbuf pool for the requested size is empty, the packet will be discarded. Use the `netstat -m` command to confirm this, and increase the parameter thewall.

The No mbuf Errors value is interface-specific and not identical to the requests for mbufs denied from the `netstat -m` output. Compare the values of the example for the commands `netstat -m` and n`etstat -v` (Ethernet and Token-Ring part).

Tuning guidelines based on netstat -v
To check for an overloaded Ethernet network, calculate (from the n`etstat -v` command):

(**Max Collision Errors** + **Timeouts Errors**) / **Transmit Packets**

If the result is greater than 5 percent, reorganize the network to balance the load.

Another indication for a high network load is (from the command `netstat -v`):

If the total number of collisions from the n`etstat -v` output (for Ethernet) is greater than 10 percent of the total transmitted packets, as follows:

Number of collisions / **Number of Transmit Packets** > 0.1

For more information in how to use netstat, see *Commands Reference, Volume 4*, SBOF-1877, *Performance Management Guide* and *Performance Tuning Study Guide*, SG24-6184.

If the system suffers from extensive NFS load, the `nfsstat` command give useful information.

10.4.2 The nfsstat command

NFS gathers statistics on types of NFS operations performed, along with error information and performance indicators. You can use the `nfsstat` commands to identify network problems and observe the type of NFS operations taking place on your system. The `nfsstat` command displays statistical information about the NFS and Remote Procedure Call (RPC) interfaces to the kernel. You can also use this command to reinitialize this information. The `nfsstat` command splits its information into server and client parts. For example, use the:

- `nfsstat -r` to see how application uses NFS

  The output is divided into server connection oriented and connectionless, as well as client connection oriented and connectionless.

  `nfsstat -s` to see the server report

  The NFS server displays the number of NFS calls received (calls) and rejected (**badcalls**) due to authentication, as well as the counts and percentages for the various kinds of calls made.

- `nfsstat -c` to see the client part

  The NFS client displays the number of calls sent and rejected, as well as the number of times a client handle was received (**clgets**) and a count of the various kinds of calls and their respective percentages. For performance monitoring, the `nfsstat -c` command provides information on whether the network is dropping UDP packets. A network may drop a packet if it cannot handle it. Dropped packets can be the result of the response time of the network hardware or software or an overloaded CPU on the server. Dropped packets are not actually lost, because a replacement request is issued for them.

  A high **badxid** count implies that requests are reaching the various NFS servers, but the servers are too loaded to send replies before the client's RPC calls time out and are retransmitted. The **badxid** value is incremented each time a duplicate reply is received for a transmitted request (an RPC request retains its XID through all transmission cycles). Excessive retransmissions place an additional strain on the server, further degrading response time.

The **retrans** column displays the number of times requests were retransmitted due to a time-out in waiting for a response. This situation is related to dropped UDP packets. If the retrans number consistently exceeds five percent of the total calls in column one, it indicates a problem with the server keeping up with demand.

For more information on the nfsstat command see *Performance Management Guide* and *Commands Reference - Volume 4*, SBOF-1877.

When going into more detailed output the `netpmon` command, using a trace facility, is useful.

10.4.3 The netpmon command

The `netpmon` command monitors a trace of system events, and reports on network activity and performance during the monitored interval. By default, the `netpmon` command runs in the background while one or more application programs or system commands are being executed and monitored. The `netpmon` command automatically starts and monitors a trace of network-related system events in real time. More on the `netpmon` command in *Performance Management Guide* and *Commands Reference - Volume 4*, SBOF-1877

10.5 Summary

The flowchart starting this chapter is used in the summery, now with some suggestions included:

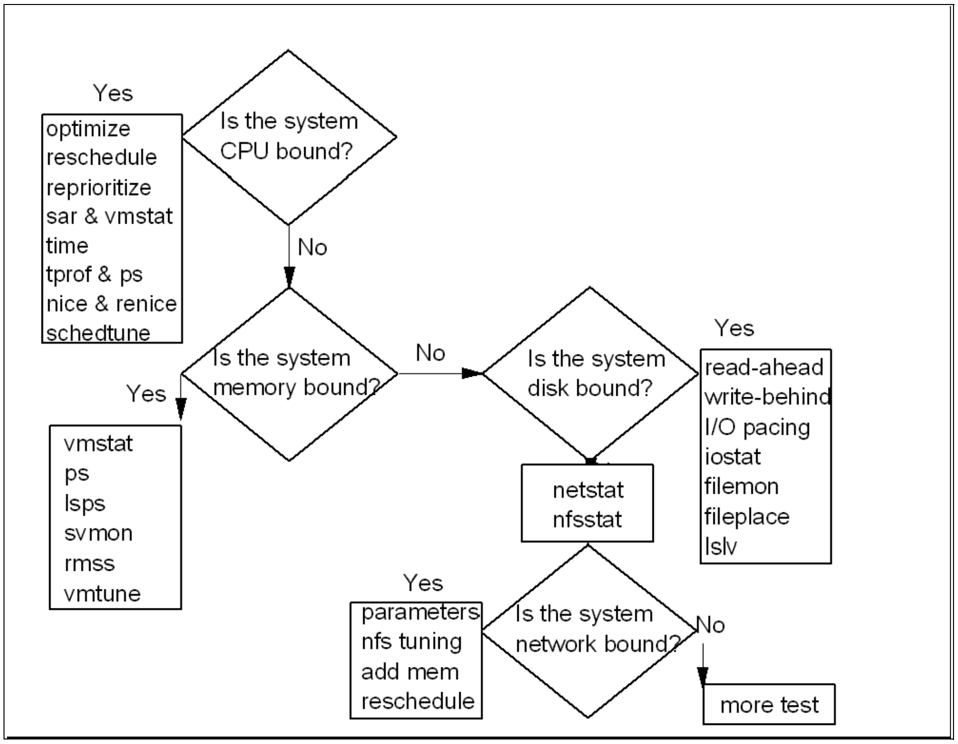


Figure 23. Performance tuning flowchart

Here follows a short summary of some of the commands and their flags used for CPU performance problem determination.

10.5.0.1 The sar command

Collects, reports, or saves system activity information

The syntax of the `sar` command:

```
/usr/sbin/sar [ { -A | [ -a ] [ -b ] [ -c ] [ -d ][ -k ] [ -m ] [ -q ] [ -r
] [ -u ] [ -V ] [ -v ] [
-w ] [ -y ] } ] [ -P ProcessorIdentifier, ... | ALL ] [ -ehh [ :mm [ :ss ]
] ] [ -fFile ] [
-iSeconds ] [ -oFile ] [ -shh [ :mm [ :ss ] ] ] [ Interval [ Number ] ]
```

Some useful `sar` flags:

Table 39. Some useful sar flags

Flags	Description
-u	Display %idle, %sys,%ysr, and%wio

Flags	Description
-P ALL	Reports per-processor statistics for each individual processor, and globally for all processors

10.5.0.2 The ps command

Shows current status of processes

The syntax of the ps command is:

```
X/Open Standards

ps [ -A ] [ -N ] [ -a ] [ -d ] [ -e ] [ -f ] [ -k ] [ -l ] [ -F format] [ -o
Format ] [ -c Clist ] [
-G Glist ] [ -g Glist ] [ -m ] [ -n NameList ] [ -p Plist ] [ -t Tlist ] [
-U Ulist ] [ -u Ulist ]

Berkeley Standards

ps [ a ] [ c ] [ e ] [ ew ] [ eww ] [ g ] [ n ] [ U ] [ w ] [ x ] [ l | s |
u | v ] [ t Tty ] [
ProcessNumber ]
```

Some useful `ps` flags:

Table 40. Some useful ps flags

Flags	Description
-f	Full listing
-l	Long listing
-u	Memory related information
-v	Memory related information

10.5.0.3 The netstat command

Shows network status

The syntax for the `netstat` command is:

```
To Display Active Sockets for Each Protocol or Routing Table Information

/bin/netstat [ -n ] [ {  -A -a } |  { -r  -C  -i  -I  Interface } ] [ -f
AddressFamily ] [ -p
 Protocol ] [ Interval ] [ System ]

To Display the Contents of a Network Data Structure
```

```
/bin/netstat [ -m | -s |  -ss |  -u |  -v ] [  -f AddressFamily ] [ -p
Protocol ] [ Interval ] [  System ]

To Display the Packet Counts Throughout the Communications Subsystem

/bin/netstat -D

To Display the Network Buffer Cache Statistics

/bin/netstat -c

To Display the Data Link Provider Interface Statistics

/bin/netstat -P

To Clear the Associated Statistics

/bin/netstat [ -Zc | -Zi | -Zm | -Zs ]
```

Some useful `netstat` flags:

Table 41. Some useful netstat flags

Flags	Description
-i	Interface status
-m	Mbuf information
-Z { c | i | m | s }	Clears the statistics defined by the additional flag
-v	Statistics for each CDLI

10.5.0.4 The nfsstat command

Displays statistical information about the Network File System (NFS) and Remote Procedure Call (RPC) calls.

The syntax of the `nfsstat` command is:

```
/usr/sbin/nfsstat [ -c ] [ -s ] [ -n ] [ -r ] [ -z ] [ -m ]
```

Some useful `nfsstat` flags:

Table 42. Some useful nfsstat flags

Flags	Description
-r	Displays RPC info
-s	Displays server information

Flags	Description
-c	Displays client information

10.6 Quiz

10.6.1 Quiz answers

Chapter 11. Software updates

This chapter will cover the AIX software updates procedures.

11.1 Overview

The biggest dream of all system administrators is to have bug-free operating system and software installed on it. Installation of software fixes is one of the key action an administrator must perform to keep a system error free. Software problems most often occur when changes have been made to the system, and either the prerequisites have not been met, for example, system firmware not at the minimum required level, or instructions have not been followed exactly in order. You, as an system administrator, should carefully choose downtime of your system. System updating and checking procedures take a lot of time and make the system unavailable for use during it.

11.1.1 Terminology

IBM has a special terminology for software packaging:

fileset The smallest individually installable unit. It is a collection of files that provides a specific function. For example, `bos.net.tcp.nfs 4.3.3.0`

fileset update An individually installable update. Fileset updates either enhance or correct a defect in a previously installed fileset. An example of a fileset update is: `bos.net.tcp.nfs 4.3.3.10`

package Contains a group of filesets with a common function. It is a single, installable image. For example `bos.net`

LPP Licensed Program Product (LPP) is a complete software product collection including all packages and filesets. For example the Base Operating System BOS itself is a LPP witch in turn is a collection of packages and filesets.

PTF Program Temporary Fix (PTF). The PTF is an updated, or fixed fileset (or group of filesets). Each fix has an Authorized Program Analysis Report number (APAR).

11.1.2 Software layout

Each software components is divided into three parts to support code serving and diskless workstation:

root The root part of a software product contains the part of the product that cannot be shared. In a client/server environment, these are the files for which there must be a unique copy for each client of a server. Most of the root software is associated with the configuration of the machine or product. In a standard system, the root parts of a product are stored in the root (/) file tree. The `/etc/objrepos` directory contains the root part of an installable software product.

usr The usr part of a software product contains the part of the product that can be shared by machines that have the same hardware architecture. Most of the software that is part of a product usually falls into this category. In a standard system, the usr parts of products are stored in the `/usr` file tree.

share The share part of a software product contains the part of the product that can be shared among machines, even if they have different hardware architectures. This would include nonexecutable text or data files. For example, the share part of a product might contain documentation written in ASCII text or data files containing special fonts.

To verify that the / (root), /usr and /usr/share parts of the system are valid with each other do:

```
lppchk -v
```

This command verifies that all software products installed on the / (root) file system are also installed on the /usr file system and, conversely, all the software products installed in the /usr file system are also installed on the / (root) file system.

11.1.3 Software states

The installed software or software update can stay in one of the following state:

- applied
- committed

If the service update was not committed during installation, then you must commit it after installation once you have decided that you will not be returning to the previous version of the software. Committing the updated version of the service update deletes all previous versions from the system and recovers the disk space that was used to store the previous version. When you are sure that you want to keep the updated version of the software, you should commit it. To commit fileset `bos.sysmgt.trace` that is currently applied but not committed do:

```
installp -c bos.sysmgt.trace
```

Note

Before installing a new set of updates, you should to consider committing any previous updates that have not yet been committed.

If you decide to return to the previous version of the software, you must reject the updated version that was installed. Rejecting a service update deletes the update from the system and returns the system to its former state. A service update can only be rejected if it has not yet been committed. Once committed, there is no way to delete an update except by removing the entire fileset, or by force-installing the fileset back to a previous level.

When you install a base level fileset, it is automatically committed during installation. If you want to delete a fileset, it must be removed (as opposed to rejected) from the system. A fileset is always removed with all of its updates.

To display the installation and update history information for bos.sysmgt.trace fileset do:

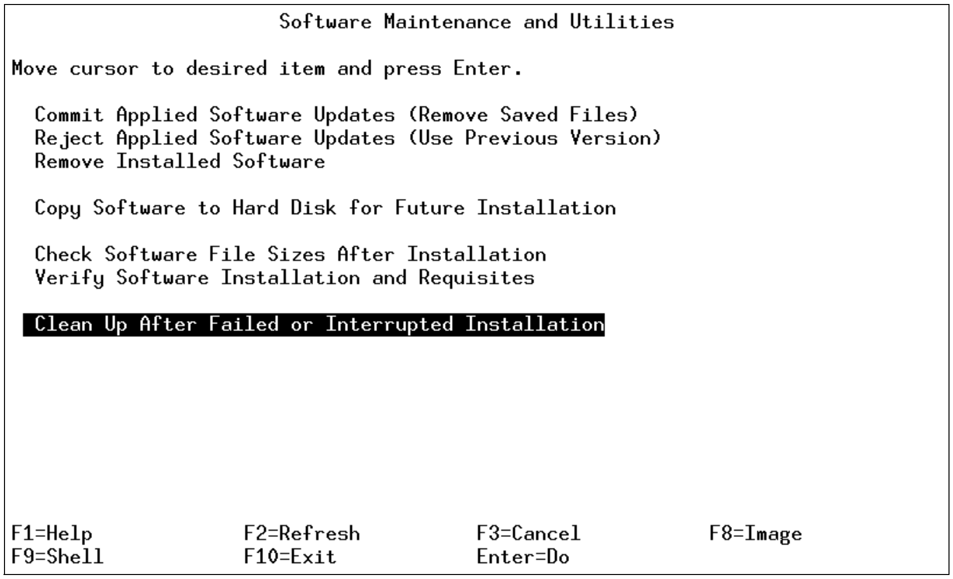
```
# lslpp -h bos.sysmgt.trace
  Fileset         Level     Action       Status       Date         Time

------------------------------------------------------------------------
Path: /usr/lib/objrepos
  bos.sysmgt.trace
                 4.3.3.0   COMMIT       COMPLETE     06/15/00     09:57:28
                4.3.3.11   COMMIT       COMPLETE     06/16/00     11:19:13

Path: /etc/objrepos
  bos.sysmgt.trace
                 4.3.3.0   COMMIT       COMPLETE     06/15/00     09:57:33
                4.3.3.11   COMMIT       COMPLETE     06/16/00     11:19:14
```

As you can see the fileset bos.sysmgt.trace was once updated. Now is in committed state at the fix level 4.3.3.11.

If something goes wrong during the software installation so that the installation is prematurely canceled or interrupted, a cleanup must be run. To do this use smitty maintain_software or use installp command:

```
installp -C
```

Figure 24 on page 250 shows how to clean up after interrupted installation using smitty.



```
                   Software Maintenance and Utilities

Move cursor to desired item and press Enter.

  Commit Applied Software Updates (Remove Saved Files)
  Reject Applied Software Updates (Use Previous Version)
  Remove Installed Software

  Copy Software to Hard Disk for Future Installation

  Check Software File Sizes After Installation
  Verify Software Installation and Requisites

  Clean Up After Failed or Interrupted Installation







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

Figure 24. Smitty Software Maintenance

11.2 Installing a software patch

Once you have AIX installed, you may want to upgrade or enhance the software on your system. To do this, there are two special bundles:

Update bundle collection of fixes and enhancements that update software products on the system. This will include updated fileset. For example fileset may be updated from 4.3.3.0 to 4.3.3.10. Appalling an updated bundle will not change the level of the operating system.

Maintenance level bundle collection of fixes and enhancments that upgrade the operating system to the latest level. For example, a maintenance level bundle can upgrade operating system from 4.3.2 to 4.3.3.

Software fixes are identified using one of the following conventions:

1. *fileset:version.release.modyfication.fix*. Modification level is used to describe functionality support. Fix levels describe a fix change.
2. PTF number such as `U469083`.
3. APAR number such as `IY00301`.

It is simple to obtain software updates. Check the Web page `http://techsupport.services.ibm.com/support/rs6000.support/databases` and download what is required.

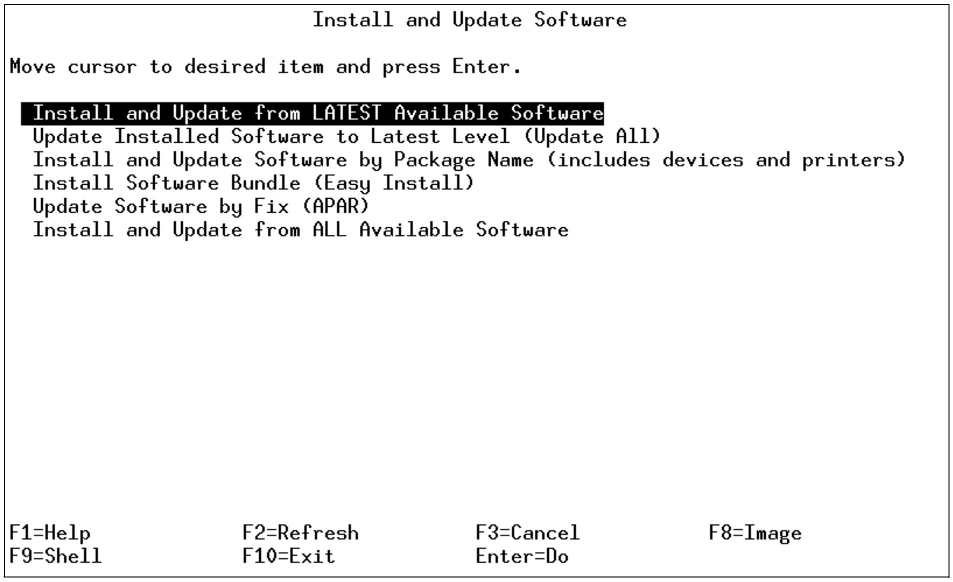
For a more customized approach to downloading AIX fixes, use the AIX application called `FixDist`. As a Web-alternative application, `FixDist` provides more discrete downloads and transparently delivers all required updates with just one click. It can also keep track of fixes you have already downloaded so you can download smaller fix packages the next time you need them. Because `FixDist` utility is a user interface to an anonymous FTP server, check if you can FTP outside through your firewall.

11.2.1 Software patch installation procedure

Before installing optional software or service updates, complete the following prerequisites:

1. AIX BOS must be installed on your system.
2. The software you are installing is available on either CD-ROM, tape, or diskette, or it is located in a directory on your system, or if your computer is a configured Network Installation Management (NIM) client, it is in an available `lpp_source` resource.
3. If you are installing service updates and do not have a current backup of your system, do this before any installation.
4. If file system have been modified, it is a good idea to back them up separately before updates are applied, since it is possible that the update process may replace configuration files.
5. Check if there is enough space in the file system.
6. Log in as root user.

The easiest way to install software update is `smitty`. Use `smitty install_update` to access installation menu. Appropriate menu is shown in the Figure 25 on page 252.



```
                          Install and Update Software

Move cursor to desired item and press Enter.

  Install and Update from LATEST Available Software
  Update Installed Software to Latest Level (Update All)
  Install and Update Software by Package Name (includes devices and printers)
  Install Software Bundle (Easy Install)
  Update Software by Fix (APAR)
  Install and Update from ALL Available Software
















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

Figure 25. Install and update software

`smitty` menu option:

- *Install and Update from LATEST Available Software*. This option allows you to install and/or update software from the latest level software available on installation media
- *Update Installed Software to the Latest Level*: Enables you to update all currently installed fileset to the latest level available on the installation media. This option is also used to update currently installed software to a new maintenance level.
- *Update Software by Fix (APAR)*. Enables you to install fileset updates that are grouped by some relationship and identified by a unique APAR. A fix to an APAR can be made up of one or more fileset updates

If you are more comfortable with a shell, all of this can be done using `installp` or `instfix` command.

1. To install all filesets within the bos.net software package (located in the `/tmp/install.images` directory) and expand file systems if necessary, enter:

   ```
   installp -aX -d/tmp/install.images bos.net
   ```

2. To install all filesets associated with fix IX38794 from the CD-ROM, enter:

   ```
   instfix -k IX38794 -d /dev/cd0
   ```

Note

If you choose to apply the updates during installation (rather than committing them at installation time), you can still reject those updates later. If a particular update is causing problems on your system, you can reject that update without having to reject all the other updates that you installed. Once you are convinced that the updates cause no problems, you may want to commit those updates to retrieve the disk space that is used to save the previous levels of that software.

After you installed new fix use `lppchk` command to check if the installation was successful. The `lppchk` command verifies that files for an installable software product (fileset) match the Software Vital Product Data database information for file sizes, checksum values, or symbolic links. The useful flags are shown in the Table 43 on page 253.

Table 43. Flags for lppchk command

Flag	Description
-c	Performs a checksum operation on the input FileList items and verifies that the checksum and the file size are consistent with the SWVPD database.
-f	Checks that the File List items are present and that the file size matches the SWVPD database
-l	Verifies symbolic links for files as specified in the SWVPD database.

If you been installing software using `smitty`, the screen returns to the top of the list of messages that are displayed during installation. You can review the message list as described in the next step, or you can exit `smitty` and review the `$HOME/smit.log` file.

After you check that installation went OK you should create new boot image using `bosboot` command:

```
bosboot -ad /dev/hdiskX
```

11.3 Software inventory

After all installations you can check what do you really installed. The most useful command are `instfix` and `lslpp`.

1. To display most recent level, state, description and all updates of the bos.sysmgt.trace fileset run command:

```
# lslpp -La bos.sysmgt.trace
Fileset                      Level  State  Description

------------------------------------------------------------------------
  bos.sysmgt.trace           4.3.3.0    C    Software Trace Service Aids
                            4.3.3.11    C    Software Trace Service Aids
...
```

2. To display whether fix IX78215 is installed, information about each fileset associated with a it run command:

```
# instfix -ik IX78215 -v
IX78215 Abstract: trace allocates too much memory

    Fileset bos.sysmgt.trace:4.3.1.1 is applied on the system.
    All filesets for IX78215 were found.
```

3. To list maintenance level updates, enter:

```
# instfix -i -tp
    All filesets for 4.3.1.0_AIX_ML were found.
    All filesets for 4.3.2.0_AIX_ML were found.
    All filesets for 4.3.1.0_AIX_ML were found.
    All filesets for 4.3.2.0_AIX_ML were found.
    All filesets for 4.3.3.0_AIX_ML were found.
```

or just run `instfix -i | grep ML`

11.4 Commands

For a complete reference of the following command use the *AIX Version 4.3 Command Reference* or the online man pages.

11.4.1 lslpp

Display information about installed filesets or fileset updates. The command has the following syntax:

```
lslpp { -f | -h | -i | -L } ] [ -a ] [ FilesetName ... | FixID ... | all ]
```

Table 44. Commonly used flags of the lslpp command

Flag	Description
-a	Displays all the information about filesets specified when combined with other flags. Displays all the information about filesets specified when combined with other flags.

Flag	Description
-f	Displays all the information about filesets specified when combined with other flags.
-h	Displays the installation and update history information for the specified fileset
-i	Displays the product information for the specified fileset.
-L	Displays the name, most recent level, state, and description of the specified fileset. Part information (usr, root, and share) is consolidated into the same listing.
-w	Lists fileset that owns this file

11.4.2 installp

Installs available software products in a compatible installation package.

Table 45. Commonly used flags of the installp command

Flag	Description
-ac	commit
-g	includes requisites
-N	overrides saving of existing files
-q	quiet mode
-w	does not place a wildcard at end of fileset name
-X	attempts to expand file system size if needed
-d	input device
-l	list of installable filesets
-c	commit an applied fileset
-C	clean up after an failed installation
-u	uninstall
-r	reject an applied filset
-p	preview of installation
-e	define an installation log
-F	forced overwrite of same or newer version

11.4.3 instfix

Installs filesets associated with keywords or fixes. The command has the following syntax:

```
instfix [ -T ] [ -s String ] [ -k Keyword ] [ -d Device ] [ -i ]
```

Table 46. Commonly used flags of the instfix command

Flag	Description
-d *device*	Specifies the input device
-i	Displays whether fixes or keywords are installed.
-k *keyword*	Specifies an APAR number or keyword to be installed.
-s string	Searches for and displays fixes on media containing a specified string
-T	Displays the entire list of fixes present on the media

11.4.4 lppchk

Verifies files of an installable software product. The command has the following syntax:

```
lppchk { -c | -f | -l | -v } [ -O { [ r ] [ s ] [ u ] } ] [ ProductName [
FileList ... ] ]
```

Table 47. Commonly used flags of the lppchk comman

Flag	Description
-c	Performs a checksum operation on the FileList items and verifies that the checksum and the file size are consistent with the SWVPD database.
-f	Checks that the FileList items are present and the file size matches the SWVPD database.
-l	Verifies symbolic links for files as specified in the SWVPD database.
-O {[r][s][u]}	Verifies the specified parts of the program. The flags specify the following parts: root, share, usr.

11.5 References

The following publications contain more information about system software updates.

- *AIX Version 4.3 Installation Guide*, SC23-4112

- *AIX Version 4.3 Commands Reference, Volume 3*, SC23-4117
- *AIX Version 4.3 Commands Reference, Volume 4*, SC23-4118

11.6 Quiz

11.6.1 Answers

11.7 Exercises

1. Use the various flags of the `lppchk` command to verify the checksum, the file sizes, the symbolic links, and the requisites of the software products installed.
2. Use the `lslpp` command to find out which fileset is used to package a given command.
3. Use the `instfix` command to list fixes installed on your system.
4. Use `FixDist` utility to download AIX fixes.
5. Use `lslpp` command to display state, description and all updates of the different filesets.

Appendix A. Using the additional material

This redbook also contains additional material in CD-ROM or diskette format, and/or Web material. See the appropriate section below for instructions on using or downloading each type of material.

A.1 Using the CD-ROM or diskette

The CD-ROM or diskette that accompanies this redbook contains the following:

File name	*Description*
????????.cpp	????Code Samples????
????????.html	????HTML Documents????
????????.prz	????Presentations????

A.1.1 System requirements for using the CD-ROM or diskette

The following system configuration is recommended for optimal use of the CD-ROM or diskette.

Hard disk space:	????MB minimum????
Operating System:	????Windows/UNIX/S390????
Processor:	???? or higher????
Memory:	????MB????
Other:	????CD-ROM drive????

A.1.2 How to use the CD-ROM or diskette

You can access the contents of the CD-ROM or diskette by pointing your Web browser at the file ????index.html???? in the CD-ROM or diskette root directory and following the links found there. Alternatively, you can create a subdirectory (folder) on your workstation and copy the contents of the CD-ROM or diskette into this folder.

A.2 Locating the additional material on the Internet

The CD-ROM, diskette, or Web material associated with this redbook is also available in softcopy on the Internet from the IBM Redbooks Web server. Point your Web browser to:

```
ftp://www.redbooks.ibm.com/redbooks/SG24????
```

Alternatively, you can go to the IBM Redbooks Web site at:

ibm.com/redbooks

Select the **Additional materials** and open the directory that corresponds with the redbook form number.

A.3 Using the Web material

The additional Web material that accompanies this redbook includes the following:

File name	*Description*
????????.zip	????Zipped Code Samples????
????????.zip	????Zipped HTML Documents????
????????.zip	????Zipped Presentations????

A.3.1 System requirements for downloading the Web material

The following system configuration is recommended for downloading the additional Web material.

Hard disk space:	????MB minimum????
Operating System:	????Windows/UNIX/S390????
Processor:	???? or higher????
Memory:	????MB????

A.3.2 How to use the Web material

Create a subdirectory (folder) on your workstation and copy the contents of the Web material into this folder.

Appendix B. Special notices

This publication is intended to help ???who?????? to ??do what????? The information in this publication is not intended as the specification of any programming interfaces that are provided by ?????????product name(s)????????????. See the PUBLICATIONS section of the IBM Programming Announcement for ???????product name(s)???????????? for more information about what publications are considered to be product documentation.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

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

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Any pointers in this publication to external Web sites are provided for convenience only and do not in any manner serve as an endorsement of these Web sites.

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

IBM ®

The following terms are trademarks of other companies:

Tivoli, Manage. Anything. Anywhere.,The Power To Manage., Anything. Anywhere.,TME, NetView, Cross-Site, Tivoli Ready, Tivoli Certified, Planet Tivoli, and Tivoli Enterprise are trademarks or registered trademarks of Tivoli Systems Inc., an IBM company, in the United States, other countries, or both. In Denmark, Tivoli is a trademark licensed from Kjøbenhavns Sommer - Tivoli A/S.

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

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

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

PC Direct is a trademark of Ziff Communications Company in the United States and/or other countries and is used by IBM Corporation under license.

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

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

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

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

Appendix C. Related publications

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

C.1 IBM Redbooks

For information on ordering these publications see “How to get IBM Redbooks” on page 267.

- *????full title???????*, xxxx-xxxx
- *????full title???????*, xxxx-xxxx
- *????full title???????*, xxxx-xxxx

C.2 IBM Redbooks collections

Redbooks are also available on the following CD-ROMs. Click the CD-ROMs button at **ibm.com**/redbooks for information about all the CD-ROMs offered, updates and formats.

CD-ROM Title	Collection Kit Number
System/390 Redbooks Collection	SK2T-2177
Networking and Systems Management Redbooks Collection	SK2T-6022
Transaction Processing and Data Management Redbooks Collection	SK2T-8038
Lotus Redbooks Collection	SK2T-8039
Tivoli Redbooks Collection	SK2T-8044
AS/400 Redbooks Collection	SK2T-2849
Netfinity Hardware and Software Redbooks Collection	SK2T-8046
RS/6000 Redbooks Collection (BkMgr)	SK2T-8040
RS/6000 Redbooks Collection (PDF Format)	SK2T-8043
Application Development Redbooks Collection	SK2T-8037
IBM Enterprise Storage and Systems Management Solutions	SK3T-3694

C.3 Other resources

These publications are also relevant as further information sources:

- *????full title???????*, xxxx-xxxx
- *????full title???????*, xxxx-xxxx
- *????full title???????*, xxxx-xxxx

C.4 Referenced Web sites

These Web sites are also relevant as further information sources:

- `http://????????.???.???/` description
- `http://????????.???.???/` description
- `http://????????.???.???/` description

How to get IBM Redbooks

This section explains how both customers and IBM employees can find out about IBM Redbooks, redpieces, and CD-ROMs. A form for ordering books and CD-ROMs by fax or e-mail is also provided.

- **Redbooks Web Site** **ibm.com**/redbooks

  Search for, view, download, or order hardcopy/CD-ROM Redbooks from the Redbooks Web site. Also read redpieces and download additional materials (code samples or diskette/CD-ROM images) from this Redbooks site.

  Redpieces are Redbooks in progress; not all Redbooks become redpieces and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

- **E-mail Orders**

  Send orders by e-mail including information from the IBM Redbooks fax order form to:

	e-mail address
In United States or Canada	pubscan@us.ibm.com
Outside North America	Contact information is in the “How to Order” section at this site: http://www.elink.ibmlink.ibm.com/pbl/pbl

- **Telephone Orders**

United States (toll free)	1-800-879-2755
Canada (toll free)	1-800-IBM-4YOU
Outside North America	Country coordinator phone number is in the “How to Order” section at this site: http://www.elink.ibmlink.ibm.com/pbl/pbl

- **Fax Orders**

United States (toll free)	1-800-445-9269
Canada	1-403-267-4455
Outside North America	Fax phone number is in the “How to Order” section at this site: http://www.elink.ibmlink.ibm.com/pbl/pbl

This information was current at the time of publication, but is continually subject to change. The latest information may be found at the Redbooks Web site.

IBM Intranet for Employees

IBM employees may register for information on workshops, residencies, and Redbooks by accessing the IBM Intranet Web site at http://w3.itso.ibm.com/ and clicking the ITSO Mailing List button. Look in the Materials repository for workshops, presentations, papers, and Web pages developed and written by the ITSO technical professionals; click the Additional Materials button. Employees may access MyNews at http://w3.ibm.com/ for redbook, residency, and workshop announcements.

IBM Redbooks fax order form

Please send me the following:

Title	Order Number	Quantity

First name Last name

Company

Address

City Postal code Country

Telephone number Telefax number VAT number

□ Invoice to customer number

□ Credit card number

Credit card expiration date Card issued to Signature

We accept American Express, Diners, Eurocard, Master Card, and Visa. Payment by credit card not available in all countries. Signature mandatory for credit card payment.

Abbreviations and acronyms

IBM	International Business Machines Corporation
ITSO	International Technical Support Organization

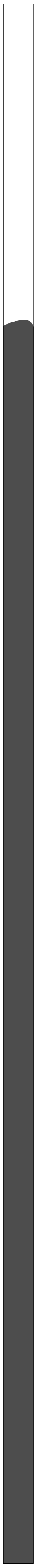
Index

IBM Redbooks review

Your feedback is valued by the Redbook authors. In particular we are interested in situations where a Redbook "made the difference" in a task or problem you encountered. Using one of the following methods, **please review the Redbook, addressing value, subject matter, structure, depth and quality as appropriate.**

- Use the online **Contact us** review redbook form found at **ibm.com**/redbooks
- Fax this form to: USA International Access Code + 1 914 432 8264
- Send your comments in an Internet note to redbook@us.ibm.com

Document Number **Redbook Title**	SG24-6185-00 IBM Certification Study Guide AIX Version 4.3 Problem Determination
Review	
What other subjects would you like to see IBM Redbooks address?	
Please rate your overall satisfaction:	O Very Good O Good O Average O Poor
Please identify yourself as belonging to one of the following groups:	O Customer O Business Partner O Solution Developer O IBM, Lotus or Tivoli Employee O None of the above
Your email address: The data you provide here may be used to provide you with information from IBM or our business partners about our products, services or activities.	 O Please do not use the information collected here for future marketing or promotional contacts or other communications beyond the scope of this transaction.
Questions about IBM's privacy policy?	The following link explains how we protect your personal information. **ibm.com**/privacy/yourprivacy/



(0.1”spine)
0.1”<->0.169”
53<->89 pages

IBM
Redbooks
IBM Certification Study Guide AIX Version 4.3 Problem Determination

(0.2”spine)
0.17”<->0.473”
90<->249 pages

(0.5” spine)
0.475”<->0.875”
250 <-> 459 pages

(1.0” spine)
0.875”<->1.498”
460 <-> 788 pages

(1.5” spine)
1.5”<-> 1.998”
789 <->1051 pages

To determine the spine width of a book, you divide the paper PPI into the number of pages in the book. An example is a 250 page book using Plainfield opaque 50# smooth which has a PPI of 526. Divided 250 by 526 which equals a spine width of .4752". In this case, you would use the .5” spine. Now select the Spine width for the book and hide the others: Special>Conditional Text>Show/Hide>SpineSize(-->Hide:)>Set

IBM Certification Study Guide AIX Version 4.3

(2.0” spine)
2.0” <-> 2.498”
1052 <-> 1314 pages

IBM Certification Study Guide AIX Version 4.3

(2.5” spine)
2.5”<->nnn.n”
1315<-> nnnn pages

To determine the spine width of a book, you divide the paper PPI into the number of pages in the book. An example is a 250 page book using Plainfield opaque 50# smooth which has a PPI of 526. Divided 250 by 526 which equals a spine width of .4752". In this case, you would use the .5” spine. Now select the Spine width for the book and hide the others: Special>Conditional Text>Show/Hide>SpineSize(-->Hide:)>Set

IBM Certification Study Guide AIX Version 4.3

The AIX & RS/6000 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 and RS/6000. A complete set of professional certifications are available. They include:

IBM Certified AIX User
IBM Certified Specialist - RS/6000 Solution Sales
IBM Certified Specialist - AIX V4.3 System Administration
IBM Certified Specialist - AIX V4.3 System Support
IBM Certified Specialist - RS/6000 SP
IBM Certified Specialist - AIX HACMP
IBM Certified Specialist - Domino for RS/6000
IBM Certified Specialist - Web Server for RS/6000
IBM Certified Specialist - Business Intelligence for RS/6000
IBM Certified Advanced Technical Expert - RS/6000 AIX

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

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by IBM's 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

SG24-6185-00 ISBN