



*AIX 5L System Administration I:
Implementation*

(Course Code AU14)

Student Notebook

ERC 7.0

IBM Learning Services
Worldwide Certified 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®	AIXwindows	CICS
Current	DB2	Hummingbird
Language Environment	Magstar	Micro Channel
Network Station	SecureWay	SP
Xstation Manager		

Tivoli™ is a trademark of Tivoli Systems Inc. in the United States, or other countries and or both.

Intel™, Itanium™ are trademarks of Intel Corporation in the United States, other countries and or both.

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

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

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

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

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

June 2001 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. The original repository material for this course has been certified as being Year 2000 compliant.

© Copyright International Business Machines Corporation 1997, 2001. 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	xv
Course Description	xvii
Agenda	xix
Unit 1. Introduction to RS/6000 System Administration	1-1
Objectives	1-2
What Is RISC Technology	1-3
RS/6000 System Bus Types	1-5
Workstation Configuration	1-7
Server Configurations	1-8
PC Connectivity	1-9
Uniprocessor (Uni)	1-10
SMP and SP	1-11
System Management Objectives	1-12
Role of the System Administrator	1-13
Who Can Perform Administration Tasks?	1-14
Activity: su	1-16
Checkpoint	1-21
Unit Summary	1-22
Unit 2. System Management Tools and Documentation	2-1
Objectives	2-2
2.1 SMIT	2-3
Early System Administration	2-4
AIX Administration	2-6
System Management Tools	2-8
System Management Interface Tool (SMIT)	2-9
SMIT Main Menu (ASCII)	2-11
SMIT Main Menu (Motif)	2-12
Dialog Screen	2-13
Output Screen	2-16
SMIT Log and Script Files	2-17
smit Command	2-18
Exercise: Using SMIT	2-20
2.2 Configuring AIX Documentation	2-21
Configuring AIX V5.1 Documentation	2-22
Configuring AIX V5.1 Online Documentation	2-23
Internet and Documentation Services	2-25
Change/Show Default Browser	2-26
Change Documentation and Search Server	2-27
Change Local Documentation and Search Server	2-28
Change Local Documentation and Search Server	2-29

AIX Version 5.1 Documentation	2-30
Print AIX Version 5.1 Documentation	2-32
Search AIX Version 5.1 Documentation	2-33
Activity: Configuring Web-based Documentation	2-34
2.3 WebSM	2-39
Web-based System Manager	2-40
Accessing the Web-based System Manager	2-42
Using the Web-based System Manager (1 of 3)	2-43
Using the Web-based System Manager (2 of 3)	2-44
Using the Web-based System Manager (3 of 3)	2-45
Configuring Client/Server WebSM	2-46
Exercise: Configuring WebSM server	2-48
Checkpoint (1 of 2)	2-49
Checkpoint (2 of 2)	2-50
Unit Summary	2-51
Unit 3. System Startup and Shutdown	3-1
Objectives	3-2
Startup Modes	3-3
Starting System Management Services	3-5
PCI RS/6000 Start Up Process Overview	3-7
AIX Startup on Itanium-based Systems	3-9
alog	3-11
/etc/inittab	3-12
System Resource Controller	3-14
System Resource Controller Syntax	3-15
Stopping Processes	3-16
System Shutdown	3-17
Manage the System Environment	3-19
Manage System Language Environment	3-21
Exercise: System startup and shutdown	3-23
Checkpoint	3-24
Unit Summary	3-25
Unit 4. Devices	4-1
Objectives	4-2
Device Terminology	4-3
Listing of /dev Directory	4-5
Device Configuration Database	4-7
List All Supported Devices	4-8
List All Defined Devices	4-10
Device States	4-12
Self-Configuring Devices	4-14
SMIT Devices Menu	4-15
Device Addressing	4-17
Location Code Format for PCI Devices	4-18
Location Code Example: Non-SCSI	4-21

Location Code Format for SCSI Devices	4-23
Location Code Example for SCSI Device	4-25
Location Code Example: PCI	4-26
Adding an ASCII Terminal	4-27
Attachment	4-28
Add a TTY	4-30
Documenting Hardware Configuration	4-32
Exercise: Devices	4-33
Checkpoint (1 of 2)	4-34
Checkpoint (2 of 2)	4-35
Unit Summary	4-36
Unit 5. System Storage Overview	5-1
Objectives	5-2
Components of AIX Storage	5-3
Traditional UNIX Disk Storage	5-4
Benefits of the LVM	5-5
Physical Storage	5-6
Volume Groups	5-8
Volume Group Descriptor Area	5-9
Volume Group Limits	5-10
Logical Storage	5-12
Uses of Logical Volumes	5-14
What Is a File System?	5-16
Why Have File Systems	5-17
Standard File Systems in AIX	5-18
Let's Review	5-20
/etc/filesystems	5-21
Mount	5-23
Mounting over an Empty Directory	5-24
Mounting over Files	5-25
Listing File Systems	5-26
Listing Logical Volume Information	5-27
Checkpoint (1 of 3)	5-28
Checkpoint (2 of 3)	5-29
Checkpoint (3 of 3)	5-30
Activity: LVM Commands	5-31
Unit Summary	5-33
Unit 6. Working with the Logical Volume Manager	6-1
Objectives	6-2
Logical Volume Manager	6-3
Volume Groups	6-5
SMIT Volume Groups Menu	6-6
Listing Volume Group Information (1 of 4)	6-7
Listing Volume Group Information (2 of 4)	6-8
Listing Volume Group Information (3 of 4)	6-9
Listing Volume Group Information (4 of 4)	6-10

Adding Volume Groups 6-11
 Removing Volume Groups 6-13
 Set Characteristics of a Volume Group 6-14
 Change a Volume Group 6-15
 Logical Track Group Size (LTG) 6-16
 Hot Spare 6-18
 Activate/Deactivate a Volume Group 6-21
 Import/Export a Volume Group 6-22
 Activity: Volume Groups 6-23
 Logical Storage 6-28
 Mirroring 6-29
 Mirror Write Consistency 6-32
 Striping 6-34
 Logical Volume Policies 6-35
 SMIT Logical Volumes Menu 6-37
 Showing Logical Volume Characteristics 6-38
 Add a Logical Volume 6-40
 Remove a Logical Volume 6-41
 Set Characteristics of a Logical Volume 6-42
 Showing LV Characteristics (1 of 2) 6-43
 Showing LV Characteristics (2 of 2) 6-44
 Add/Remove a Logical Volume Copy 6-45
 Reorganize a Volume Group 6-46
 Physical Volumes 6-47
 SMIT Physical Volumes Menu 6-49
 Listing Physical Volume Information (1 of 3) 6-50
 Listing Physical Volume Information (2 of 3) 6-51
 Listing Physical Volume Information (3 of 3) 6-52
 Add/Move Contents of Physical Volume 6-53
Working with lvmstat 6-55
 Migrating Physical Partitions 6-57
 Documenting the Disk Storage Setup 6-59
 Exercise: Logical Volume Manager 6-60
 Checkpoint 6-61
 Unit Summary 6-62

Unit 7. Working with File Systems 7-1
 Objectives 7-2
 Structure of a Journalled File System 7-3
 Structure of an Inode 7-5
 File System Fragmentation 7-6
 Variable Number of Inodes 7-8
 Allocation Group Size 7-10
 JFS vs. JFS2 File Systems 7-12
 File and File System Sizes 7-14
 Compressed File Systems 7-16
 Large Enabled File Systems 7-18
 Activity: Inodes and NBPI 7-19

Journalized Log	7-21
File Systems	7-23
Listing File Systems	7-24
List All Mounted File Systems	7-25
Working with Journalized Files Systems in SMIT	7-27
Add a Standard Journalized File System on a Previously Defined Logical Volume	7-28
Add a Standard Journalized File System	7-29
Add a JFS2 File System on a Previously Defined Logical Volume	7-31
Add an Enhanced Journalized File System	7-32
Mount a File System	7-33
Change/Show Characteristics of a Journalized File System	7-35
Change/Show Characteristics of an Enhanced Journalized File System	7-37
Remove a Journalized File System	7-38
System Storage Review	7-39
Exercise: Working with File Systems	7-40
Checkpoint	7-41
Unit Summary	7-42
Unit 8. Managing File Systems	8-1
Objectives	8-2
Space Management	8-3
Listing Free Disk Space	8-4
Control Growing Files	8-5
skulker	8-7
Listing Disk Usage	8-8
Fragmentation Considerations	8-9
Defragmenting a File System	8-11
Verify a File System	8-12
Documenting File System Setup	8-14
Exercise: Managing File Systems	8-15
Checkpoint	8-16
Unit Summary	8-17
Unit 9. Paging Space	9-1
Objectives	9-2
What Is Paging Space?	9-3
Paging Space	9-4
Sizing Paging Space	9-5
Paging Space Placement	9-7
Paging Space	9-9
Adding Paging Space	9-10
Change Paging Space	9-11
Remove Paging Space	9-13
Problems With Paging Space	9-14
Documenting Paging Space Setup	9-15
Exercise: Paging Space	9-16
Checkpoint	9-17
Unit Summary	9-18

Unit 10. AIX V5.1 Installation	10-1
Objectives	10-2
Installation Methods	10-3
Installation Process - from CD	10-4
Console and Language Definition	10-6
Installation and Maintenance Menu	10-8
Installation Settings	10-9
Method of Installation	10-10
Installation Disks	10-12
Primary Language Environment®	10-13
Advanced Options	10-14
Begin Installation	10-16
Installation Flow Chart - All Systems	10-17
Configuration Assistant Menu	10-18
Activity: Configuration Assistant	10-20
Checkpoint	10-22
Unit Summary	10-23
Unit 11. AIX Software Installation and Maintenance	11-1
Objectives	11-2
AIX Product Offering	11-3
Packaging Definitions	11-4
Fileset Naming	11-5
Bundles	11-7
Software Updates	11-8
Software States	11-10
Software Installation and Maintenance	11-12
Install and Update Software	11-13
Install and Update from LATEST Available Software	11-15
Software Inventory	11-17
List Installed Software	11-18
Software Maintenance and Utilities	11-19
instfix Command	11-20
Exercise: AIX Software Installation	11-22
Checkpoint	11-23
Unit Summary	11-24
Unit 12. Backup and Restore	12-1
Objectives	12-2
Why Backup?	12-3
Why Restore?	12-4
Types of Backup	12-5
Backup Strategy	12-6
Backup Devices - Diskette	12-7
Backup Devices - Tape	12-8
Backup Device - Read/Write Optical Drive	12-10
Backup Menus	12-12
rootvg Backup Process - mksysb	12-13

/image.data File for rootvg	12-15
/bosinst.data File for rootvg	12-17
rootvg Image Backup	12-21
mksysb Image	12-23
Non-rootvg Backup	12-24
Restoring a mksysb (1 of 2)	12-25
Restoring a mksysb (2 of 2)	12-26
Restoring a non-rootvg Volume Group	12-28
Activity: savevg	12-29
backup by File Name	12-32
backup by File Name Examples	12-33
backup a File or a Directory	12-35
backup a File by Inode	12-36
Incremental backup Example	12-38
Backup a File System by Inode	12-40
restore Command	12-41
Restore a File or a Directory	12-43
Exercise: Using backup and restore	12-44
Other UNIX backup Commands	12-45
tar Command	12-46
cpio Command	12-48
dd Command	12-50
Controlling the Tape	12-52
Good Practices	12-54
Optional Exercise: Using tar and cpio	12-55
Checkpoint	12-56
Unit Summary	12-57
Unit 13. Security and User Administration	13-1
Objectives	13-2
13.1 Security Concepts	13-3
Security Concepts	13-4
Groups	13-5
Groups	13-7
User Hierarchy	13-8
Control root's Access	13-9
Security Logs	13-10
File/Directory Permissions	13-12
Reading Permissions	13-14
Changing Permissions	13-15
umask	13-16
Changing Ownership	13-17
Exercise: Security Files	13-18
13.2 User Administration	13-19
Login Sequence	13-20
User Initialization Process	13-22
Security and Users	13-23

SMIT Users	13-24
List All Users	13-25
Add a User to the System	13-26
Change / Show Characteristics of a User	13-28
Remove a User from the System	13-30
Passwords	13-31
Regaining root's Password	13-33
SMIT Groups	13-34
List All Groups	13-35
Add Groups	13-36
Change / Remove Groups	13-37
Message of the Day	13-39
Exercise: User Administration	13-40
13.3 Security Files	13-41
Security Files	13-42
/etc/passwd File	13-44
/etc/security/passwd File	13-46
/etc/security/user File	13-48
/etc/security/user File Continued	13-50
Group Files	13-53
/etc/security/login.cfg File	13-54
Validating the User Environment	13-56
System Management Services	13-57
PCI RS/6000 Passwords	13-58
Documenting Security Policy and Setup	13-59
Checkpoint (1 of 2)	13-60
Checkpoint (2 of 2)	13-61
Activity: Examine the Security Files	13-62
Unit Summary	13-64
Unit 14. Scheduling	14-1
Objectives	14-2
cron Daemon	14-3
crontab Files	14-5
crontab File	14-6
Editing crontab	14-8
The at and batch Commands	14-9
Controlling at Jobs	14-11
Documenting Scheduling	14-12
Exercise: Scheduling	14-13
Checkpoint	14-14
Unit Summary	14-15
Unit 15. Printers and Queues	15-1
Objectives	15-2
AIX 5.1 Printing Environments	15-3
AIX Print Subsystem: Advantages	15-5

System V Print Subsystem: Advantages	15-7
Concepts of Queues	15-9
Printer Data Flow	15-10
System Files Associated with Printing	15-12
qdaemon	15-13
The /etc/qconfig File	15-14
Printer Menu	15-16
Configuring a Printer with a Queue	15-18
Selecting a Printer Type	15-19
Selecting a Printer Type	15-20
Printer Attachment	15-21
Add the Print Queues	15-22
Remote Printing	15-23
Client Authorization	15-24
Start lpd	15-25
Add a Remote Print Queue	15-26
Define the Print Server on the Client	15-27
Let's Review	15-28
Submitting Print Jobs	15-29
Listing Jobs in a Queue	15-31
Change Characteristics of a Queue	15-33
Removing a Queue	15-35
Managing Queues	15-36
Understanding Queue Status	15-37
Bringing Queues Up and Down	15-39
Managing Print Jobs	15-40
Cancelling Print Jobs	15-41
Job Priority Example	15-42
Holding a Job in a Queue	15-43
Moving a Job between Queues	15-44
Printing-related Directories to Monitor	15-45
Printing Problem Checklist	15-46
Exercise: Printers and Queues	15-48
Checkpoint (1 of 2)	15-49
Checkpoint (2 of 2)	15-50
Unit Summary	15-51
Unit 16. Networking Overview	16-1
Objectives	16-2
What Is TCP/IP?	16-3
An Internet	16-5
Names and Addresses	16-7
TCP/IP Network Facilities	16-8
Information Needed to Configure TCP/IP	16-10
Configuring TCP/IP	16-12
Flat Name Resolution	16-14
Identifying the Hostname	16-15
Basic TCP/IP User Functions	16-16

Exercise: Networking16-20
 Checkpoint16-21
 Unit Summary16-22

Appendix A. Command Summary A-1

Startup, Logoff and Shutdown A-1
 Directories A-1
 Files - Basic A-1
 Files - Advanced. A-2
 Editors A-3
 Shells, Redirection and Pipelining A-4
 Metacharacters. A-4
 Physical and Logical Storage A-5
 Variables A-6
 Tapes and Diskettes. A-7
 Transmitting A-8
 System Administration A-8
 Miscellaneous. A-8
 System Files. A-9
 Shell Programming Summary A-9
 Variables A-9
 Commands. A-10
 Miscellaneous. A-11
 vi Editor A-12
 Entering vi A-12
 Read, Write, Exit A-12
 Units of Measure A-12
 Cursor Movements A-13
 Search and Replace A-14
 Adding Text A-14
 Deleting Text A-14
 Replacing Text A-15
 Moving Text A-15
 Miscellaneous. A-15

Appendix B. Sample Shell Scripts Used in Class B-1

Appendix C. MCA Appendix C-1

Key Positions C-1
 Setting/Changing Boot Lists C-1
 Microchannel Location Codes. C-4
 100 - 195 C-5

Appendix D. AIX Control Book Creation D-1

AIX Control Book Creation D-1

Directories to monitor	D-2
Appendix E. Licensing Overview	E-1
Appendix F. Serial Devices	F-1
Appendix G. Itanium-based System Startup.	G-1
Appendix H. The System V Print Subsystem	H-1
Appendix I. Checkpoint Solutions	I-1
Glossary	X-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®	AIXwindows	CICS
Current	DB2	Hummingbird
Language Environment	Magstar	Micro Channel
Network Station	SecureWay	SP
Xstation Manager		

Tivoli™ is a trademark of Tivoli Systems Inc. in the United States, or other countries and or both.

Intel™, Itanium™ are trademarks of Intel Corporation in the United States, other countries and or both.

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

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

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

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

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

Course Description

AIX 5L System Administration I: Implementation

Duration: 5 days

Purpose

The purpose of this course is to enable the students to install, customize and administer the AIX 5L Version 5.1 operating system in a multiuser environment using RS/6000s or Itanium-based systems.

Audience

Anyone responsible for the system administrative duties implementing and managing AIX 5L Version 5.1 operating system on an IBM RS/6000 or an Itanium-based system.

Prerequisites

The student attending this course should be able to:

- Log in to an AIX system and set a user password
- Execute basic AIX commands
- Manage files and directories
- Use the vi editor
- Use redirection, pipes, and tees
- Use the utilities find and grep
- Use the command and variable substitution
- Set and change Korn shell variables
- Write simple shell scripts

These skills can be acquired by taking AIX 5L Version 5.1 Basics or through equivalent AIX/UNIX knowledge.

Objectives

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

- Install the AIX 5L Version 5.1 operating system, software bundles, and filesets.
- Perform system startup and shutdown

- Understand and use the system management tools
- Manage physical and logical devices
- Perform file system management
- Create and manage user and group accounts
- Perform and restore system backups
- Utilize administrative subsystems, including cron to schedule system tasks, and security to implement customized access of files and directories.
- Describe basic networking concepts

Contents

- Introduction (Overview of RS/6000)
- System Management Tools - SMIT and the Web-based System Manager
- Software Installation and Management
- System Startup and Shutdown Devices
- Printers and Queues
- Managing Queues
- System Storage Overview
- Working with the Logical Volume Manager
- Working with File Systems
- Managing File Systems
- Paging Space
- Backup and Restore
- Security
- User Administration
- Scheduling
- Networking Overview

Curriculum relationship

This course should follow the AIX 5L Version 5.1 Basics course. A basic understanding of AIX environment and simple commands is recommended before taking this course.

Agenda

Day 1

Welcome
Unit 1 - Introduction to RS/6000 Systems
Activity: su
Unit 2 - System Management Tools and Documentation
2.1 - SMIT
Exercise: Using SMIT
2.2 - Configuring AIX Documentation
Activity: Configuring Web-based Documentation
2.3 - WebSM
Exercise: Configuring WebSM server
Unit 3 - System Startup and Shutdown
Exercise: System Startup and Shutdown
Unit 4 - Devices
Exercise: Devices

Day 2

Unit 5 - System Storage Overview
Let's Review: LVM Terminology
Unit 5 (Cont)
Activity: LVM Commands
Unit 6 - Working With the Logical Volume Manager
Activity: Volume Groups
Unit 6 (Cont)
Exercise: Logical Volume Manager
Unit 7 - Working with File Systems
Activity: Inodes and NPBI
Unit 7 (Cont)
Exercise: Working with File Systems

Day 3

Unit 8 - Managing File Systems
Exercise: Managing File Systems
Unit 9 - Paging Space
Exercise: Paging Space
Unit 10 - AIX V5.1 Installation
Activity: Configuration Assistant
Unit 11 - AIX Software Installation and Maintenance
Exercise: AIX Software Installation

Unit 12 - Backup and Restore
Activity: savevg
Unit 12 (Cont)
Exercise - Using backup and restore

Day 4

Unit 12 (Cont)
Exercise (optional) - Using tar and cpio
Unit 13 - Security and User Administration
 13.1 - Security Concepts
 Exercise: Security Files
 13.2 - User Administration
 Exercise: User Administration
 13.3 - Security Files
 Activity: Examine the Security Files
Unit 14 - Scheduling
Exercise: Scheduling

Day 5

Unit 15 - Printers and Queues
Let's Review
Unit 15 (Cont)
Exercise: Printers and Queues
Unit 16 - Networking Overview
Exercise: Networking

Unit 1. Introduction to RS/6000 System Administration

What This Unit Is About

This unit introduces basic RS/6000 configurations and describes the roles of the system administrator.

What You Should Be Able to Do

After completing this unit, you should be able to:

- Define the terminology and concepts of the RS/6000
- List common configurations available with the RS/6000
- Describe the roles of the system administrator
- Obtain root access with the **su** command

How You Will Check Your Progress

Accountability:

- Checkpoint questions
- Activity

References

GA23-2674	<i>Exploring IBM RS/6000 Computers</i>
SG24-4690	<i>A Technical Introduction to PCI-Based RS/6000 Servers</i>
SG24-2581	<i>Managing AIX on PCI-Based RISC System/6000 Workstations</i>
SG24-5120	<i>RS/6000 System Handbook</i>
www.rs6000.ibm.com	

Objectives

After completing this unit, students should be able to:

- Define the terminology and concepts of the RS/6000
- List common configurations available with the RS/6000
- Describe the roles of the system administrator
- Obtain root access with the **su** command

Figure 1-1. Objectives

AU147.0

Notes:

What Is RISC Technology

Reduced Instruction Set Computing processors aim to:

- Implement the most used instructions in hardware
- Execute multiple instructions in one cycle
- Provide synergy between hardware and software

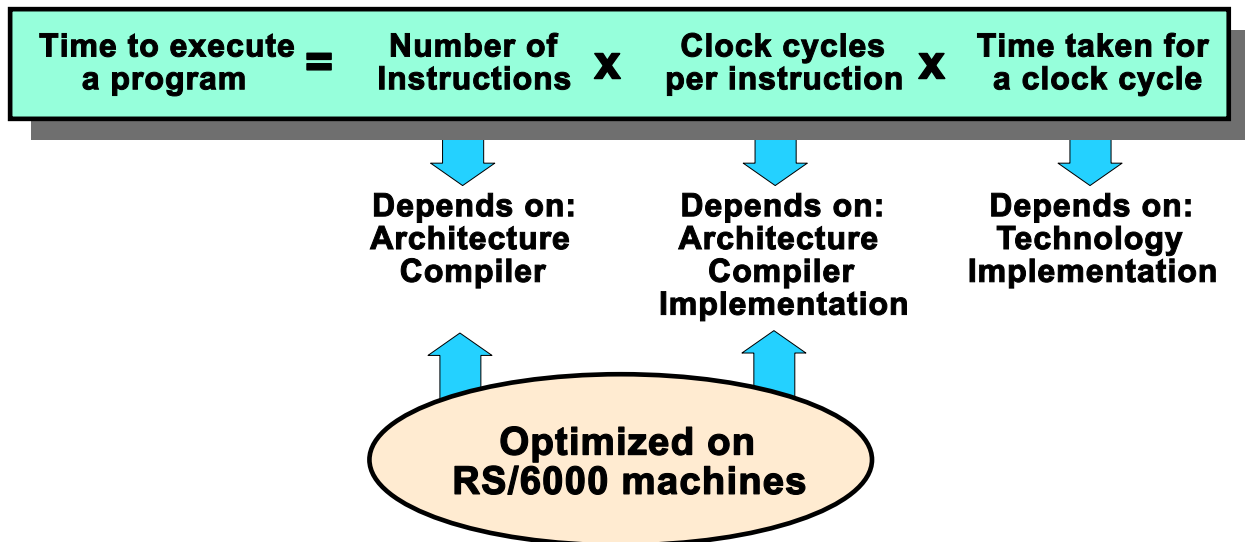


Figure 1-2. What Is RISC Technology

AU147.0

Notes:

Reduced Instruction Set Computing (RISC) architecture was originally developed by IBM in the 1970s. Its basic objective was to provide a reduced instruction set that would execute very fast with maximum efficiency in the hardware. More complex instructions would be implemented in the software.

The simple RISC-based instruction is typically executed in one system clock cycle (or less using superscalar techniques). IBM has enhanced the standard RISC technology by introducing the newer **Performance Optimized With Enhanced Risc (POWER)** architecture. The original POWER architecture has also evolved into the POWER2 and PowerPC architectures.

The POWER architectures are designed with the newest in circuitry engineering and multiprocessor technologies and yield very fast performance.

The instructions are handled in a **superscalar** (parallel) fashion by the processor which further increases the performance offered by a RISC system.

Support for 64-bit architecture is provided by AIX V4.3. This support provides improved performance for specialized applications with:

- Large address spaces (up to 16,384,000 terabytes)
- Access to large datasets for data warehousing, scientific and multimedia applications
- Long integers in computations

A major enhancement in AIX 5.1 is the introduction of the 64-bit kernel. The primary advantage of a 64-bit kernel is the increased kernel address space allowing systems to support increased workloads. This ability is important for a number of reasons:

- Data sharing and I/O device sharing are simplified if multiple applications can be run on the same system
- More powerful systems will reduce the number of systems needed by an organization, thereby reducing the cost and complexity of system administration

Server consolidation and workload scalability will continue to require higher capacity hardware systems that support more memory and additional I/O devices. The 64-bit AIX 5.1 kernel is designed to support these requirements.

RS/6000 System Bus Types

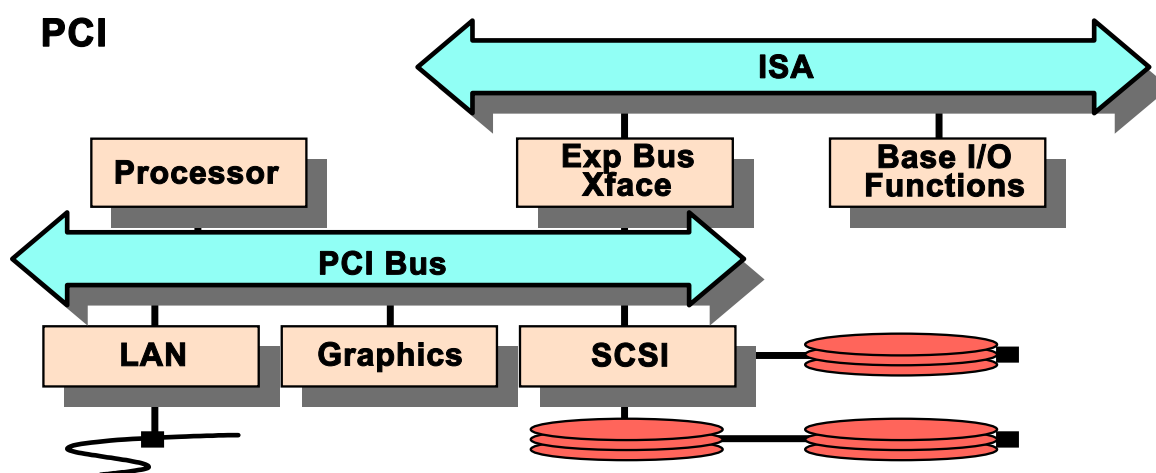


Figure 1-3. RS/6000 System Bus Types

AU147.0

Notes:

The job of the bus is to provide the highway for information to flow between the RS/6000 system elements and the optional I/O feature cards (for example, SCSI adapters, token-ring cards) that are plugged into the adapter slots.

PCI Based RS/6000 Systems

Peripheral Component Interconnect (PCI) buses are an open industry specification which supports complete processor independence. The PCI bus works across multiple operating system platforms. IBM uses this technology in all of its RS/6000s.

RS/6000's also contain an ISA (Industry Standard Architecture) bus for use with some built-in devices like the diskette drive and keyboard.

Some older model PCI systems also contain ISA slots that would accept standard ISA cards. Newer models no longer support this.

The first RS/6000's were based on IBM's MCA (Micro Channel®Architecture). The MCA systems are sometimes referred to as classical systems. These were very popular and still make up a large portion of the installed bases. MCA machines can be easily recognized by

the physical key on the front of the machines. PCI and MCA are basically the same from an administrative viewpoint. There are differences primarily in the startup procedure. For more information, see the MCA section in the appendix.

With AIX 5.1, Intel Itanium architectures are now supported. It is available on a limited basis.

A good source for hardware information is www.ibm.com/servers/eserver/pseries/hardware.

Workstation Configuration

Single-User Graphical Workstation

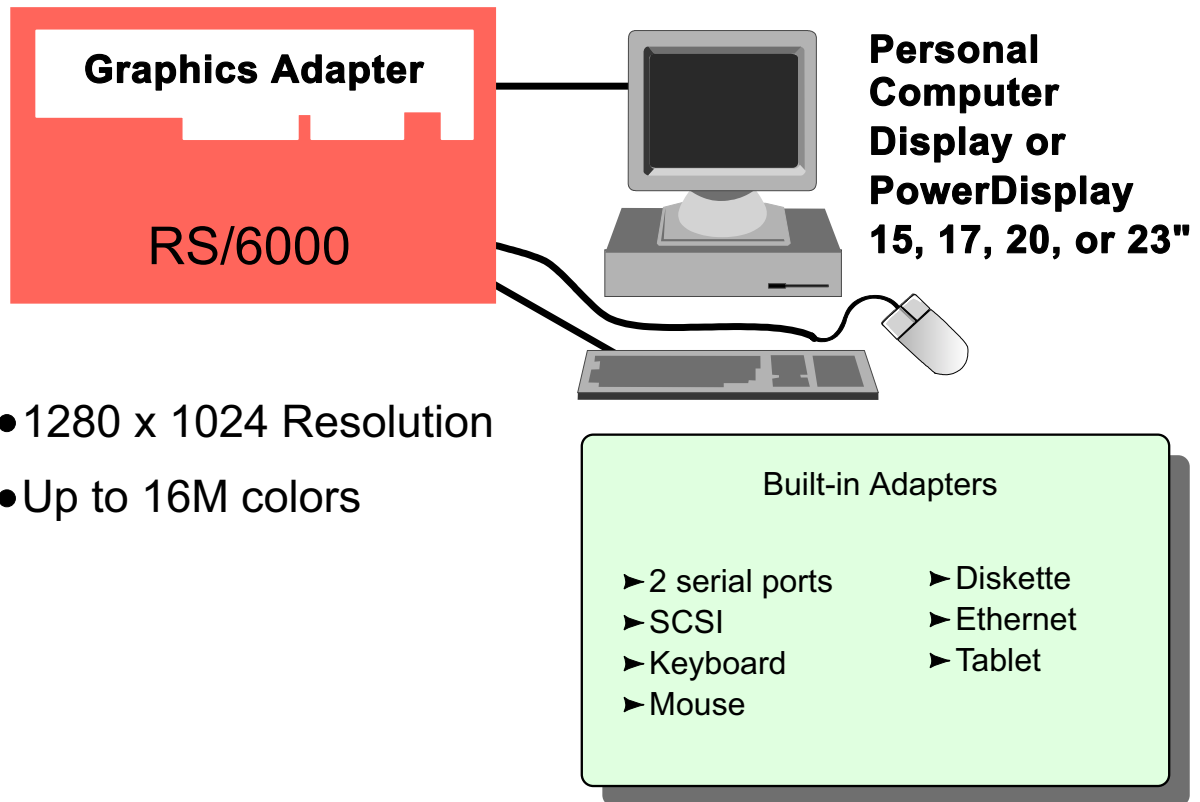


Figure 1-4. Workstation Configuration

AU147.0

Notes:

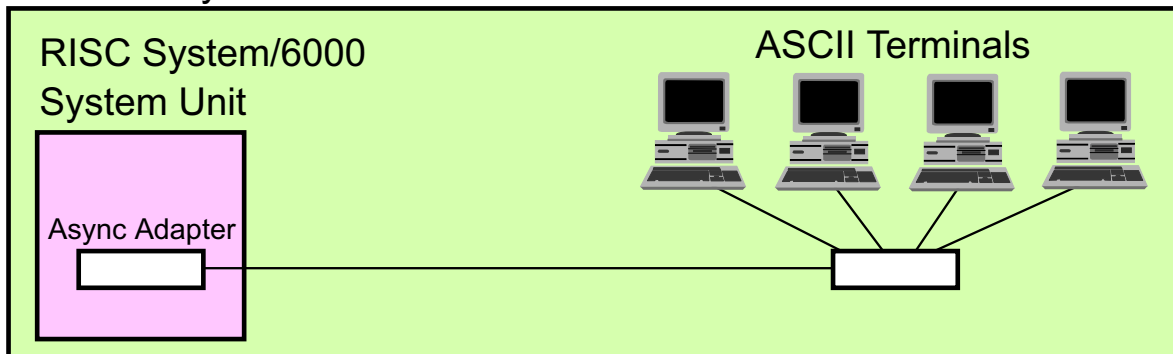
One common configuration for the RS/6000 is as a single-user graphical workstation suitable for graphics applications such as CAD/CAM.

In this configuration the RS/6000 will have a graphics display (referred to as an LFT - Low Function Terminal) which is attached to a graphics adapter inside the system unit. A keyboard, mouse, and optional graphics tablet are plugged into special ports on the system board.

There are a number of graphics cards available for the different RS/6000 models which differ in speed, resolution, number of colors supported, 2D or 3D support, and so forth. There are corresponding displays that can be used from personal computer displays through to the 23-inch PowerDisplay.

Server Configurations

Multuser System



Networked System

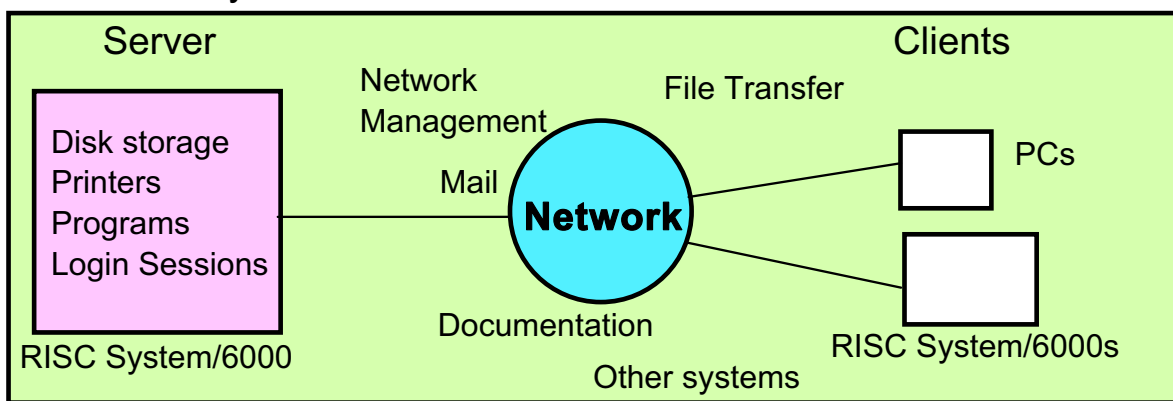


Figure 1-5. Server Configurations

AU147.0

Notes:

Some multiuser systems consist only of ASCII terminals connected locally or over a telephone line by modem. Two ASCII devices can be connected to the serial ports provided on RS/6000s. All further ASCII devices will require an asynchronous adapter card.

More complex systems consist of many RS/6000s and other devices such as PCs connected over a local area network (LAN) like Ethernet or token ring. In this case the RS/6000 will require the appropriate communications adapter card.

PC Connectivity

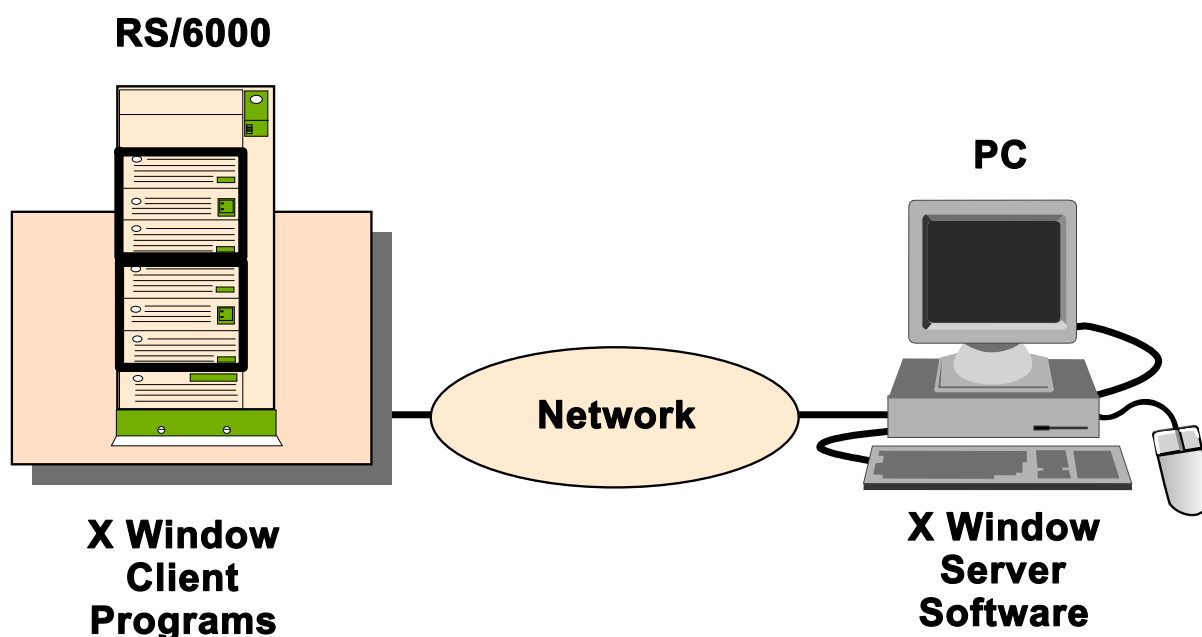


Figure 1-6. PC Connectivity

AU147.0

Notes:

Very commonly, RS/6000's are accessed via a network using PC's.

One way to connect is using **telnet**. Another method, which is growing in popularity, is to install software on the PC to give the PC the capability to function as an X-Window Server. This allows the PC to function as a graphics display station for the RS/6000. There are many commercially available software packages for several different operating systems that provide this functionality.

Uniprocessor (Uni)

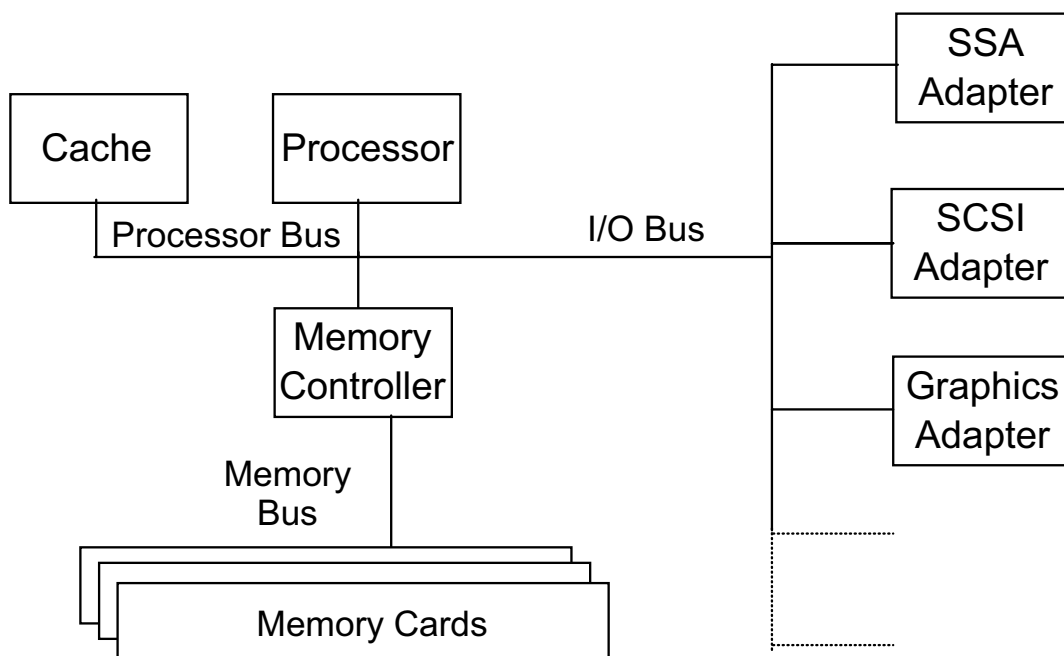


Figure 1-7. Uniprocessor (Uni)

AU147.0

Notes:

Uniprocessor is referring to a machines with only one processor. The processor is connected to the memory and other adapters via the bus. Today, the I/O busses are based on the Peripheral Component Interconnect (PCI) architecture.

SMP and SP

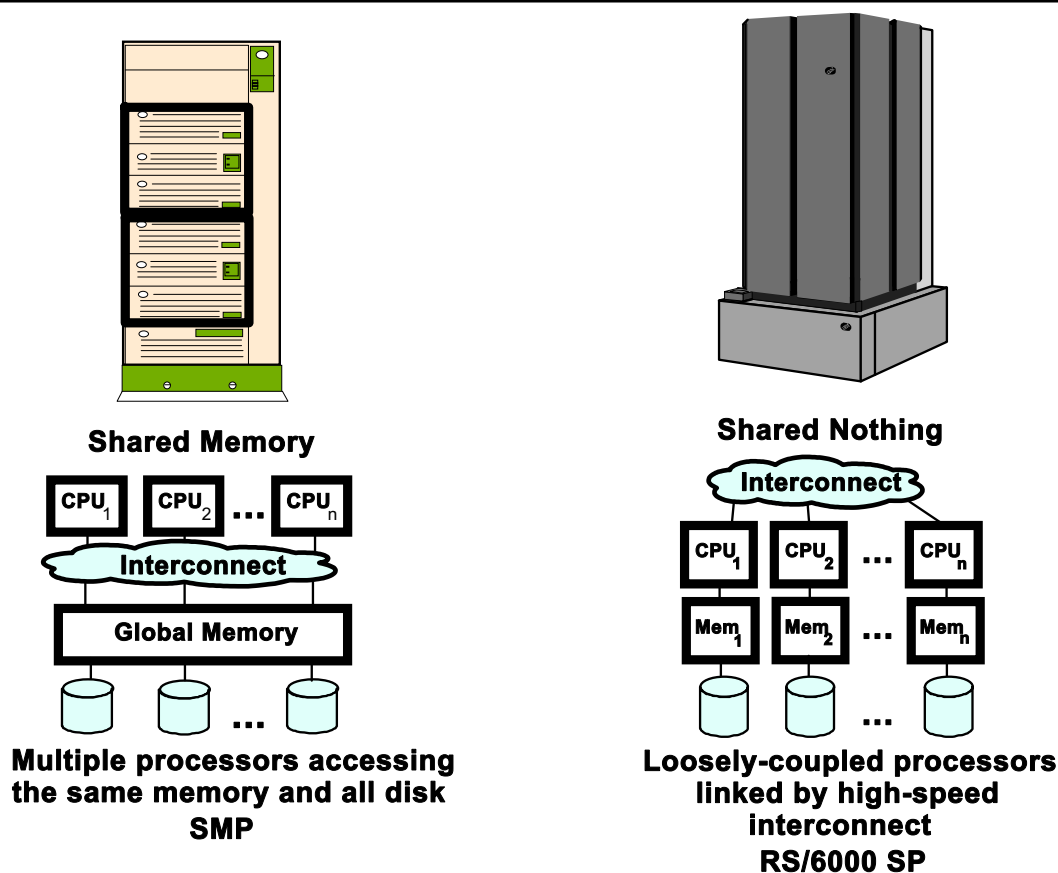


Figure 1-8. SMP and SP

AU147.0

Notes:

The **Symmetric MultiProcessor** (SMP) architecture supports a single copy of the operating system which is shared by all processors. Memory and disk is also shared. RS/6000 SMP models support both PCI and MCA buses and can support up to 24 processors.

The **IBM Scalable PowerParallel** (SP) systems are a set of up to 16 rack-mounted RS/6000 systems, called nodes. The nodes fit in 128 slots on the RS/6000 SP. A “thin node” takes one slot, a “wide node” takes two slots and an SMP “high node” takes four slots.

Each SP node has its own memory, operating system, expansion slots, and disk. A high speed network called the SP Switch (up to 480MB/sec.) is available to connect the nodes together. The SP uses the Parallel System Support Programs (PSSP) to control its environment.

The SP system is ideal for any parallel computing, high CPU-usage (such as modeling and numerical analysis) and I/O-intensive applications (such as Data Mining, OLTP, DB2/PE and Oracle Parallel Query/Server).

System Management Objectives

- Minimize time and resources spent managing systems.
- Maximize productivity, reliability, and performance.
- Provide remote system management solutions.



Figure 1-9. System Management Objectives

AU147.0

Notes:

Minimize time and resource spent managing systems; that is, manage efficiently. AIX helps with tools such as SMIT and the Web-based System Manager.

Maximize productivity, reliability and performance; that is, maximize the productivity of the users. AIX helps with features, such as the logical volume manager, that don't require the system to be brought down for maintenance.

Provide remote system management solutions; AIX supports Web-based technology with the Web-based System Manager. As a result, multiple systems can be managed from one AIX system over the network. This can be done with SMIT using telnet as well.

Role of the System Administrator

- Preinstallation planning of:
 - User accounts/groups
 - Storage allocation/paging space
 - Subsystem (printing, networks...)
 - Standard naming conventions
 - Determine system policies
- Install and configure hardware
- Configure the software
- Configure the network
- System backup
- Create/manage user accounts
- Define and manage subsystems
- Manage system resources (for example, disk space)
- Performance monitoring
- Capacity planning
- Managing licenses for products
- Document system configuration and keep it current

Figure 1-10. Role of the System Administrator

AU147.0

Notes:

There are a number of distinct tasks which the system administrator on a UNIX or AIX system must perform. Often there will be more than one system administrator in a large organization, and the tasks can be divided between the different administrators.

Who Can Perform Administration Tasks?

- Usually exclusive to the **root** user
 - Bypasses any file permissions
 - Very dangerous to login as root
 - Keep root password secure
- Some tasks can be performed by other users in special groups such as system, security, and printq
- **su** command allows you to obtain root's permissions or permissions of any user whose password you know

```
$ su root
```

or

```
$ su - root
```

Figure 1-11. Who Can Perform Administration Tasks?

AU147.0

Notes:

AIX security permissions restrict the performance of administrative tasks to the **root** user (and sometimes other users in special groups; for example, **system** for general tasks, **security** for user administration, **printq** for printer management.) This means that the **root** user's password must be kept secure and only divulged to the few users who are responsible for the system.

A certain amount of discipline is also required when using the **root** ID, because typing errors made as **root** could do catastrophic system damage. For normal use of the system, a non-administrative user ID should be used, and only when superuser privilege is required should the root user ID be used.

To obtain superuser (**root**) privileges while logged in as a normal user, you can use the **su** command. This will prompt you for root's password and then give you a shell with root privileges so that you can perform commands. When you have performed the required tasks you should exit from the **su** command in the same way as exiting from a normal shell (for example, **<ctrl-d>** or the **exit** command.) This will prevent accidents which could damage the system.

The **su** command allows you to assume the permissions of any user whose password you know.

Every time the **su** command is used an entry is placed in the file **/var/adm/sulog** (this is an ASCII text file). This makes it easy to record access as the superuser. Normal logins are recorded in the file **/var/adm/wtmp**. To read the contents of this file use the command: **who /var/adm/wtmp**.

The **su** command can also be specified with the **-** option. The **-** specifies that the process environment is to be set as if the user had logged into the system using the login command. Nothing in the current environment is propagated to the new shell. For example, using the **su** command without the **-** option, allows you to have all of the accompanying permission of root while keeping your own working environment.

Activity: su



Figure 1-12. Activity: su

AU147.0

Working with su

Activity Instructions

In this activity, you will look at several ways to invoke the privileges of the root user.

Direct Logins to root

Start by directly logging in as root.

1. Log in at the user **root**. The password is **ibmaix**.
2. Open a terminal. (if necessary)
3. Determine your current directory and your home directory.
4. To verify that you do, in fact, have root privileges, **cat** out the file **/etc/security/passwd**. This file holds the encrypted passwords for the users on your system. Only root can look at this file. More details on this file will be mentioned later in the course.

Did you see the contents of the file? _____

5. Log out of the system.

Using the su command

6. Log into the system using the login name **team01** and the password **team01**.
7. Open a terminal. (if necessary)
8. Determine your current directory and your home directory.
current directory:
9. Check to see if you have privileges to view **/etc/security/passwd**.
Did you see the contents of the file? _____
10. Change to the **/tmp** directory.
11. Now switch user to **root** without using the - (dash) option.
12. Determine your current directory and your home directory.
Has anything changed? _____
13. Do you have **root** privileges? Check **/etc/security/passwd**. _____
14. Change to the **/etc** directory.
15. Exit back to **team01**.
16. Check you current directory. Does **su** effect you current directory? _____
17. Now try using the **su** command with the - (dash) option. What is your home directory? If you're not sure, check before performing the **su**. _____
18. Determine your current directory and your home directory.
current directory:
Is there any thing different this time? _____
19. Check **/etc/security/passwd** to make sure you have root privileges.
20. Exit back to **team01** and log out.

END

Consider the following questions:

What is the difference between the **su** command used with and without dash?

What situations would it be helpful to use the dash and not use the dash?

Are there disadvantages of logging in directly as root?

Working with su

Activity Instructions with Hints

In this activity, you will look at several ways to invoke the privileges of the root user.

Direct Logins to root

Start by directly logging in as root.

1. Log in as the user **root**. The password is **ibmaix**.

In the box, enter: **root**

In the box, enter root's password: **ibmaix** (You will not see the password appear on the screen)

2. Open a terminal. (if necessary)

On the tool bar at the bottom, click the small triangle above the “pencil and paper” icon (fourth icon from the left). This will open a drawer of icons.

Click ONCE on the Terminal icon. This should bring up a terminal window. All of the commands for this exercise should be type at the command line in this window.

3. Determine your current directory and your home directory.

pwd

echo \$HOME

4. To verify that you do, in fact, have root privileges, **cat** out the file **/etc/security/passwd**. This file holds the encrypted passwords for the users on your system. Only root can look at this file. More details on this file will be mentioned later in the course.

cat /etc/security/passwd

Did you see the contents of the file? _____

5. Log out of the system.

On the blank area of the screen (not in a window), press and hold down the right mouse button. A drop-down menu should appear. Continue holding the right mouse button and drag the cursor to the bottom of the menu to the “**Log out...**” selection. Then, release the mouse button.

Click “**Continue logout**”

A new log in box should appear.

Using the su command

6. Log into the system using the login name **team01** and the password **team01**.

In the box, enter: **team01**

In the box, enter password: **team01** (You will not see the password appear on the screen)

7. Open a terminal. (if necessary)

On the tool bar at the bottom, click the small triangle above the “pencil and paper” icon (fourth icon from the left). This will open a drawer of icons.

Click ONCE on the Terminal icon. This should bring up a terminal window. All of the commands for this exercise should be type at the command line in this window.

8. Determine your current directory and your home directory.

current directory:

```
$pwd
$echo $HOME
```

9. Check to see if you have privileges to view **/etc/security/passwd**.

```
$cat /etc/security/passwd
```

Did you see the contents of the file? _____

10. Change to the **/tmp** directory.

```
$cd /tmp
$pwd (to confirm)
```

11. Now switch user to **root** without using the - (dash) option.

```
$su or $su root
root's Password: ibmaix
```

12. Determine your current directory and your home directory.

current directory:

```
# pwd
# echo $HOME
```

Has anything changed? _____

13. Do you have **root** privileges? Check **/etc/security/passwd**. _____

```
# cat /etc/security/passwd
```

14. Change to the **/etc** directory.

```
# cd /etc
# pwd
```

15. Exit back to **team01**.

```
# exit
$
```

16. Check your current directory. Does **su** effect your current directory? _____

```
$pwd
```

17. Now try using the **su** command with the - (dash) option. What is your home directory? If you're not sure, check before performing the **su**. _____

\$echo \$HOME

\$su - or **su - root**

Note: Make sure you include a space before and after the - (dash)

root's Password: **ibmaix**

#

18. Determine your current directory and your home directory.

pwd

echo \$HOME

Is there any thing different this time? _____

19. Check **/etc/security/passwd** to make sure you have root privileges.

cat /etc/security/passwd

20. Exit back to **team01** and log out.

exit

On the blank area of the screen (not in a window), press and hold down the right mouse button. A drop-down menu should appear. Continue holding the right mouse button and drag the cursor to the bottom of the menu to the “**Log out...**” selection. Then, release the mouse button.

Click “**Continue logout**”

A new log in box should appear.

END

Consider the following questions:

What is the difference between the **su** command used with and without the dash?

What situations would it be helpful to use the dash and not use the dash?

Are there disadvantages of logging in directly as root?

Checkpoint

1. What type of adapter are you likely to require for a single-user graphics workstation?
 - a. Asynchronous
 - b. Communications
 - c. Graphics
2. What is the difference between UP and SMP machines?
3. True or false? The **su** command allows you to get root authority even if you signed on using another userID.

Figure 1-13. Checkpoint

AU147.0

Notes:

Unit Summary

- **Common Configurations**

- Single-user graphics workstation
- Multiuser ASCII
- Networked system
- X Window enabled PC
- SMP
- SP

- **System Administrator's Role:**

- Preinstallation planning
- Install - hardware, software, network
- Manage - user accounts, system resources, licenses
- Backup/recovery
- Define subsystems
- Performance monitoring, capacity planning

Figure 1-14. Unit Summary

AU147.0

Notes:

Unit 2. System Management Tools and Documentation

What This Unit Is About

This unit introduces the tools that are available for system administration.

What You Should Be Able to Do

After completing this unit, you should be able to:

- Outline the benefits of system management tools available with AIX Version 5.1
- Define the functionality of SMIT
- Define how SMIT activity is logged
- Define the use of the Web-based System Manager
- Set up and use AIX Web-based documentation

How You Will Check Your Progress

Accountability:

- Checkpoint questions
- Exercises
- Activity

References

Online	<i>System Management Concepts: Operating System and Devices</i>
Online	<i>System Management Guide: Operating System and Devices</i>
Online	<i>AIX 5L Version 5.1 Web-based System Manager Administration Guide</i>
SG24-5765	<i>AIX 5L Differences Guide Version 5.1 Edition</i>

Objectives

After completing this unit, students should be able to:

- Outline the benefits of the system management tools available with AIX Version 5.1
- Define the functionality of SMIT
- Define how SMIT activity is logged
- Define the use of the Web-based System Manager
- Set up and use AIX Web-based documentation

Figure 2-1. Objectives

AU147.0

Notes:

2.1 SMIT

Early System Administration

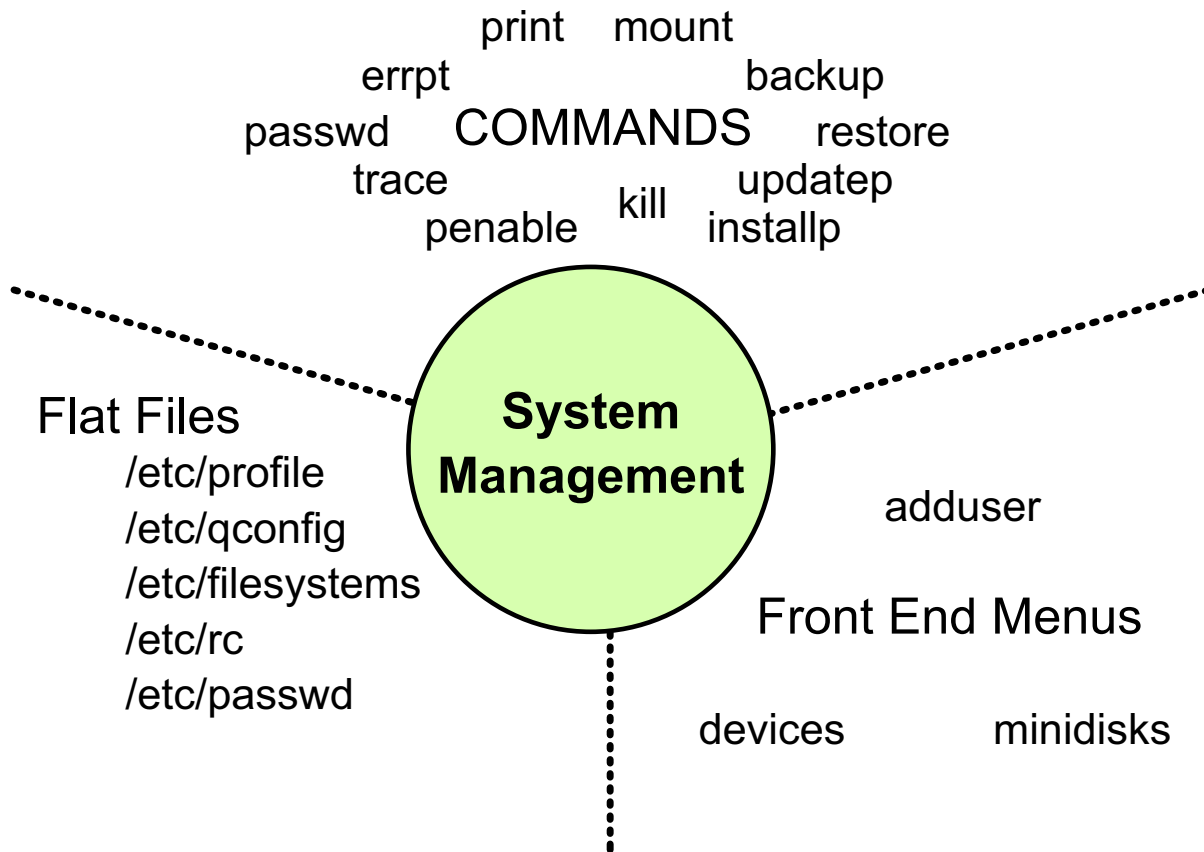


Figure 2-2. Early System Administration

AU147.0

Notes:

The main disadvantages with system administration on UNIX and AIX systems before AIX Version 3 was the fact that there was not a common consistent interface to perform system administration tasks, and the administrator had to be very knowledgeable about how the system worked. The following techniques were used:

- **Commands** — A number of commands were available which performed some system management functions. These had various heritages (for example, from AT&T, Berkeley and IBM) and were not necessarily available on all systems.
- **Front Ends** — A few menu or command driven front ends were available to perform some aspects of system management. Unfortunately, these were not consistent with each other, and also could not be used non-interactively (that is, from a shell script.)
- **Flat Files** — Configuration of some aspects of the system was performed by editing files which were in a variety of different formats. This was very prone to typing errors and also required knowledge of one of the system editors.

The first front ends available for the IBM RT systems were for device handling and user creation. Now there are front ends to perform most basic administrative tasks.

AIX Administration

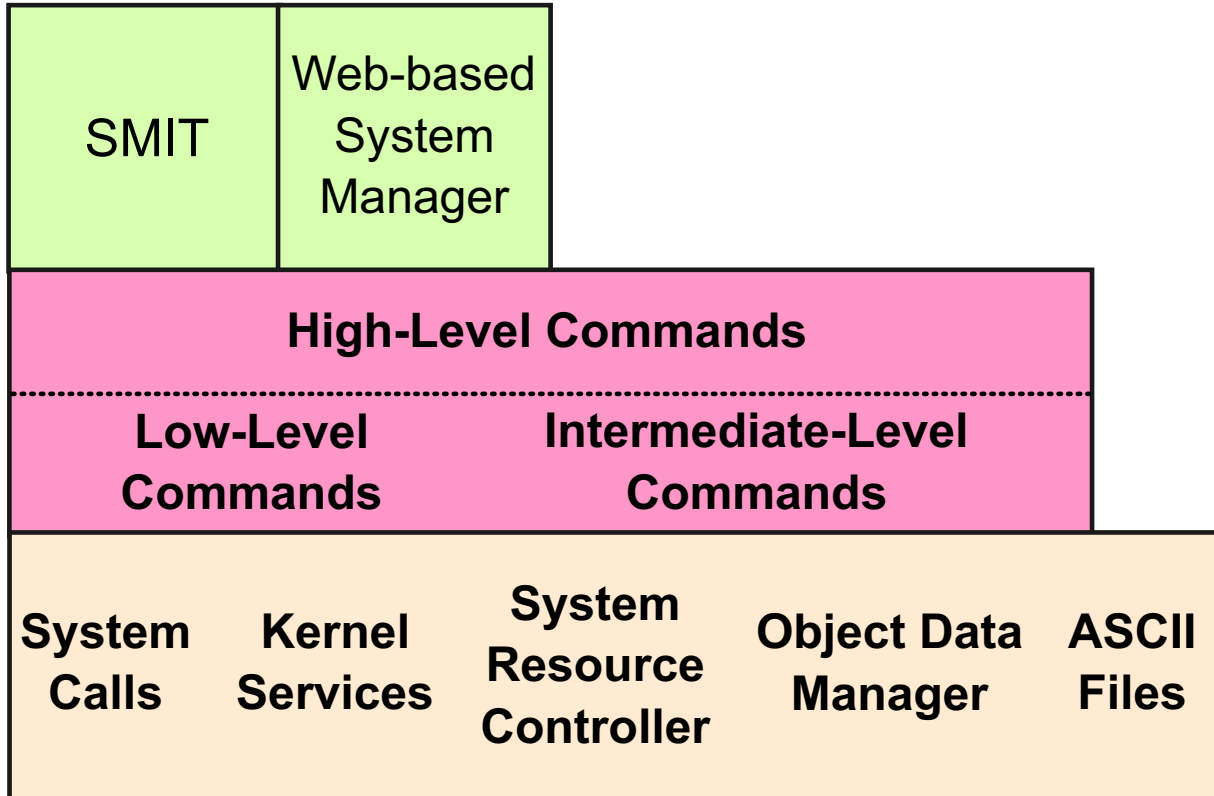


Figure 2-3. AIX Administration

AU147.0

Notes:

The **System Management Interface Tool (SMIT)** provides a menu-driven interface that provides access to most of the common system management functions within one consistent environment.

SMIT does not perform any system management functions directly. It is a user interface that constructs high-level commands from the user's selections and then executes these commands on demand. Those commands could be entered directly by the user to perform the same tasks.

- **High-Level Commands** — These are standard AIX commands (either shell scripts or C programs) which can also be executed by a user. They execute multiple low-level or intermediate-level commands to perform the system administrative functions.
- **Intermediate-Level Commands** — These commands interface with special AIX components such as the System Resource Controller and the Object Data Manager. (These commands are rarely executed directly by a user.)

- **Low-Level Commands**— These are AIX commands which correspond with AIX system calls or kernel services. (They are not normally executed directly by a user.)

SMIT does not cover every possible system management task and occasionally there will be a need to run AIX commands or edit ASCII files directly. However, SMIT does make the most frequent or complex/tedious tasks much easier with a greater degree of reliability.

The **Web-based System Manager** was introduced with AIX V4.3. The Web-based System Manager is an intuitive object-oriented user interface for performing system management tasks. This tool can be run in stand-alone mode or in a client-server environment. The Web-based System Manager will be discussed in further detail later in this unit.

System Management Tools

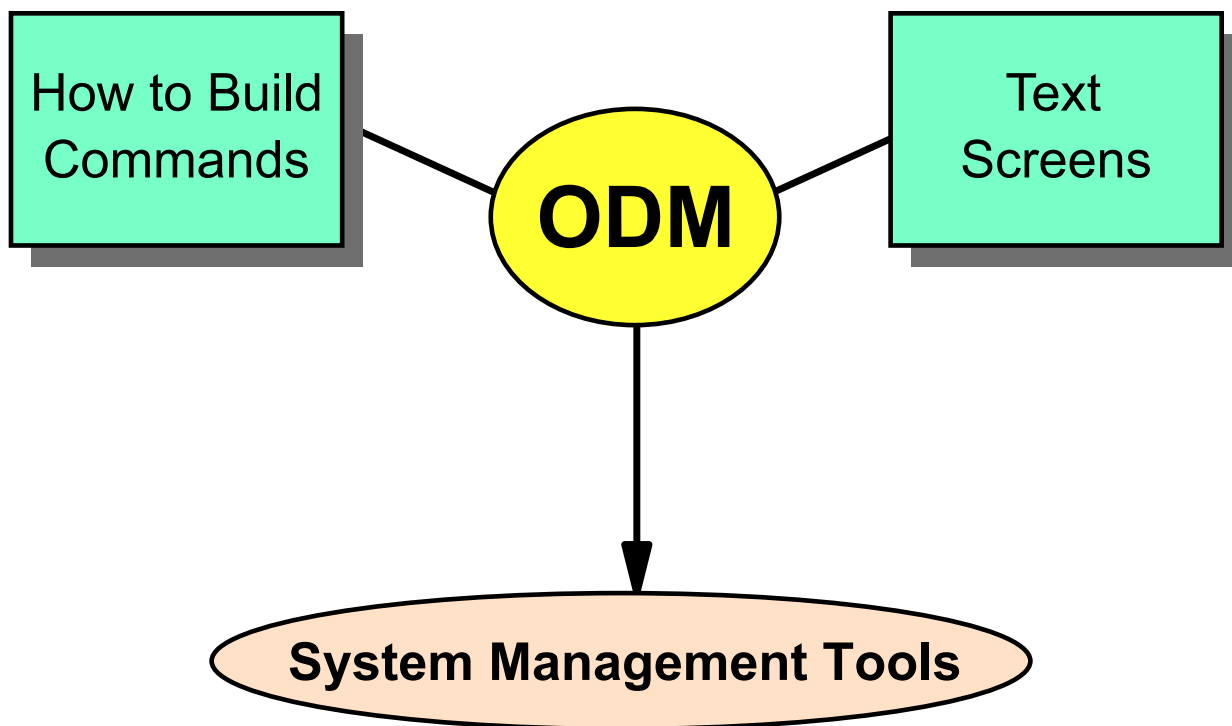


Figure 2-4. System Management Tools

AU147.0

Notes:

AIX contains a database call Object Data Manager (ODM). The ODM is unique to AIX compared to other UNIX environments. It keeps information about devices, software and most other configurations of a machine. It also contains all of the menus, screens and commands that SMIT uses.

When using SMIT, the system configuration in ODM is maintained automatically. Most administrators never have a need to directly manipulate ODM. SMIT and the high-level commands ensure ODM is correct.

System Management Interface Tool (SMIT)

ASCII or AIXWindows (Motif) User Interface Components

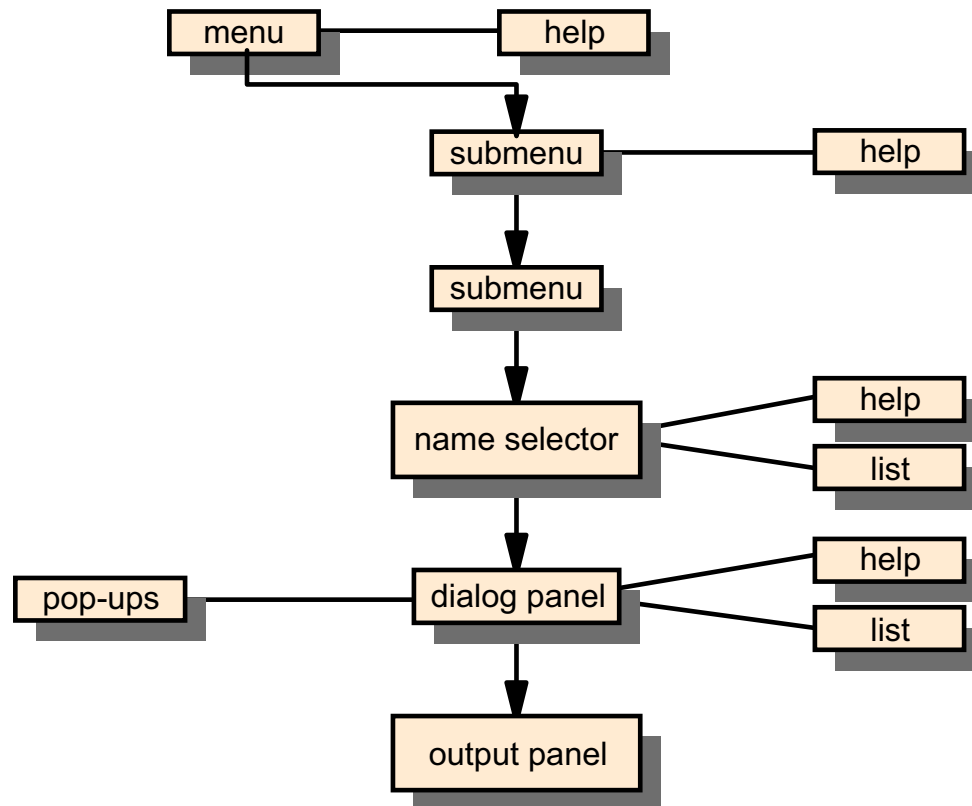


Figure 2-5. System Management Interface Tool (SMIT)

AU147.0

Notes:

SMIT provides a flexible user environment. The user can use an ASCII or AIXWindows-based interface which provide the same facilities but the interaction is slightly different. The user interface consists of a number of components:

- **Menus** - SMIT has a hierarchy of menus which breaks down the typical system management tasks into related areas. Some submenus may appear in multiple places within the hierarchy where appropriate.
- **Selector/Dialog Screens** - A selector screen allows you to select an object on which an action is to be performed (for example a tape drive). Having selected the object, a dialog screen will allow you to control the way in which the task is performed (for example set the attributes for the drive, or install from that drive.)
- **Pop-up Lists** - Where there are a number of possible values for a parameter, you can often request a list of these values and select either a single item or multiple items.

- **Output Panels** - SMIT constructs and runs standard AIX commands. The standard output and standard error from these commands are displayed within a special SMIT output screen, and this output can be reviewed after command completion.
- **Contextual Help** - SMIT provides online help which will guide you through the use of SMIT, and will also provide contextual information about each submenu, dialog screen and also each field within a dialog screen.

SMIT Main Menu (ASCII)

```
# smit
```

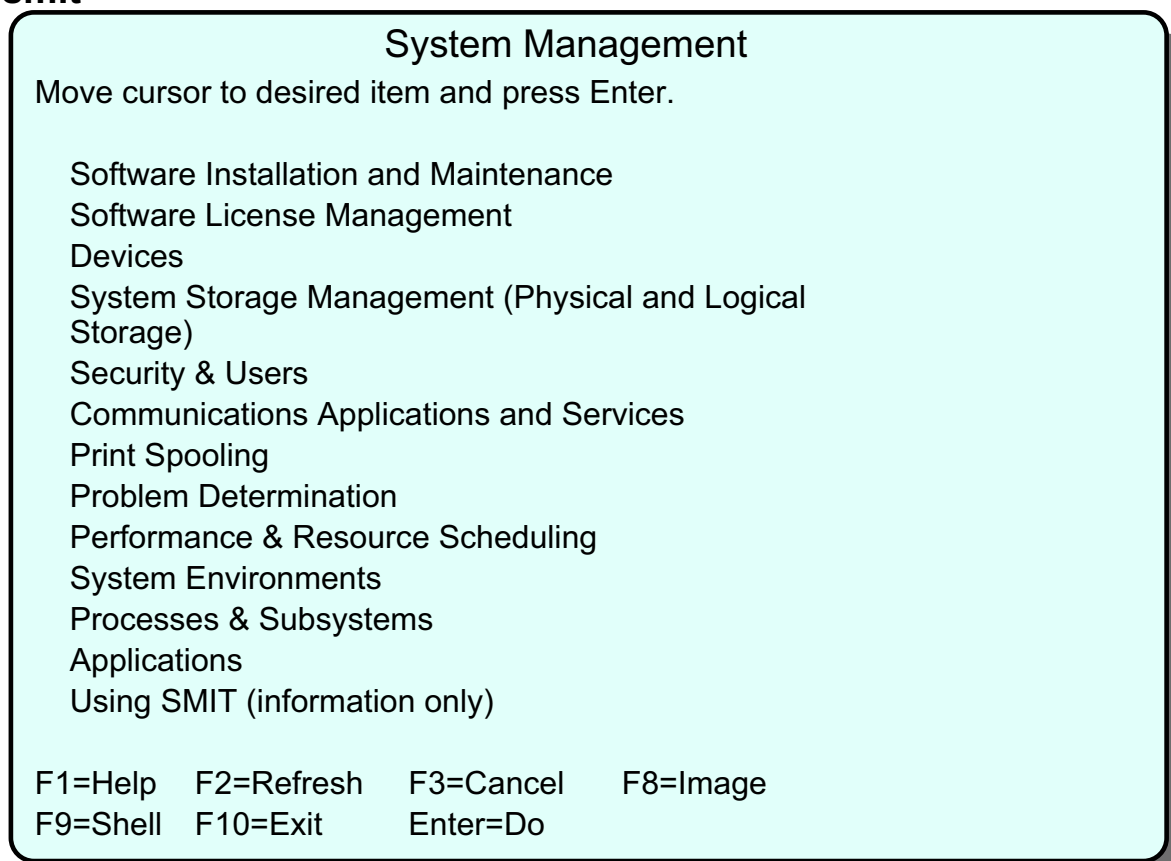


Figure 2-6. SMIT Main Menu (ASCII)

AU147.0

Notes:

The SMIT main menu allows you to select the administrative functions to be performed. You can also select online help on how to use SMIT.

In the ASCII mode, in order to select from the menus, you have to use the up and down arrow keys. This will move a highlighted bar over the menu items. Press **Enter** to select the highlighted item.

You can also use some of the keyboard function keys to perform other functions, such as exiting SMIT or starting a shell.

SMIT Main Menu (Motif)

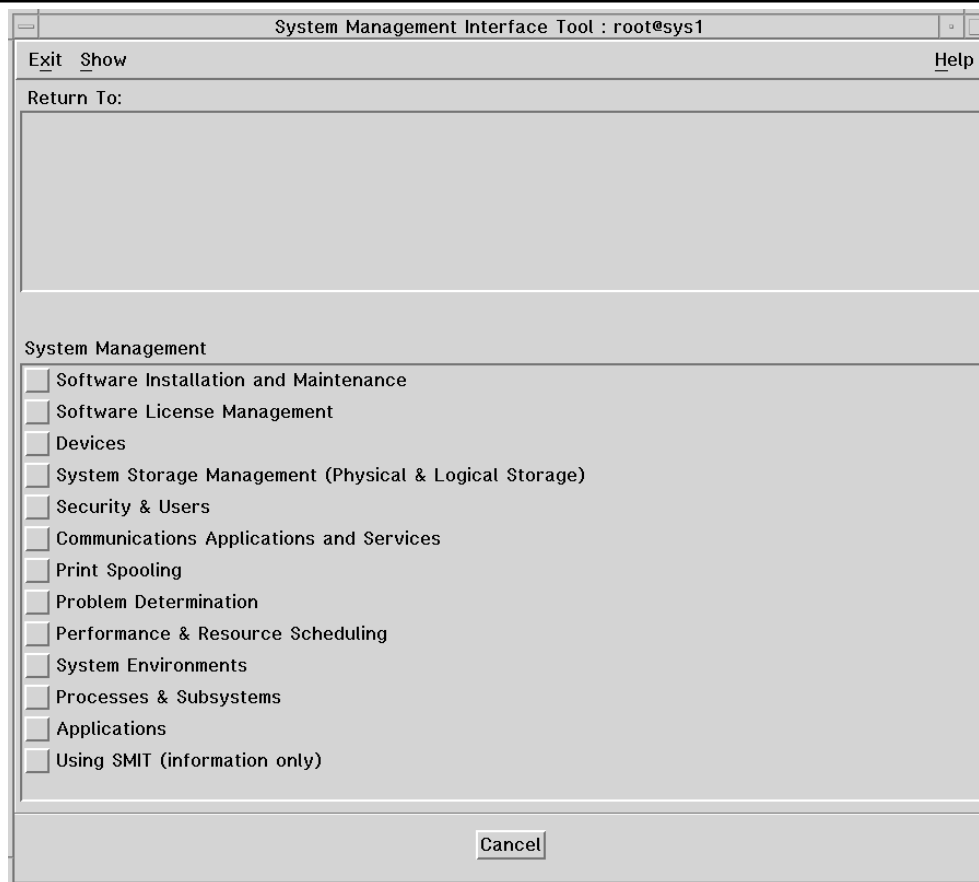


Figure 2-7. SMIT Main Menu (Motif)

AU147.0

Notes:

The graphical (Motif) version of SMIT must be run using a graphical environment like AIXWindow or Common Desktop Environment (CDE). Typing the command `smit` in the graphical environment will automatically call graphical SMIT.

To work with graphical SMIT, use the mouse to point and click your way through the menu system. Clicking the Cancel box at the bottom of the screen, moves you back one screen. Or, you can select the screen title in the Return To section of the screen.

A number of functions are available through pull-down menus on the top of the screen. To exit SMIT, for example, click the Exit pull-down.

The function keys used in the ASCII version of SMIT do not correspond to actions in the graphical SMIT.

Dialog Screen

Schedule a Job

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

	[Entry Fields]	
YEAR	[00]	#
MONTH	[Jun]	+
DAY (1-31)	[1]	#
* HOUR (0-23)	[]	#
* MINUTES (0-59)	[]	#
SHELL to use for job execution	Korn (ksh)	+
* COMMAND or SHELL SCRIPT (full pathname)	[]	/

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

Figure 2-8. Dialog Screen

AU147.0

Notes:

A dialog screen allows you to enter values which determine the operation performed. Some fields will already be filled in from information held in the system. Usually you can change this data from the default values.

A selector screen is a special case of a dialog screen in which there is only one value to change. This usually indicates the object which will be acted upon by the subsequent dialog and AIX command.

To enter data, move the highlighted bar to the value you want to change and then either enter a value or select one from a pop-up list. Fields that you can type in are indicated by square brackets []. Fields that have data that is larger than the space available to display it are indicated by angle brackets <>, to indicate that there is data further to the left or right (or both) of the display area.

Special symbols on the screen are used to indicate how data is to be entered:

- * A required field
- # A numeric value is required for this field.

- / A pathname is required for this field.
- X** A hexadecimal value is required for this field.
- ?** The value entered will not be displayed.
- +** A pop-up list or ring is available.

An * symbol in the leftmost column of a line indicates that the field is required. A value must be entered here before you can commit the dialog and execute the command.

In the ASCII version, a **+** is used to indicate that a pop-up list or ring is available. To access a pop-up list use the **F4** key. A ring is a special type of list. If a fixed number of options are available, the **Tab** key can be used to cycle through the options.

In the Motif version a **List** button is displayed. Either click the button or press **Ctrl-I** to get a pop-up window to select from.

The following keys can be used while in the menus and dialog screens. Some keys are only valid in particular screens. Those valid only for the ASCII interface are marked (A) and those valid only for the Motif interface are marked (M)

- F1 (or ESC-1)** Help - show contextual help information
- F2 (or ESC-2)** Refresh - redraw the display (A)
- F3 (or ESC-3)** Cancel - return to the previous screen (A)
- F4 (or ESC-4)** List - display a pop-up list of possible values (A)
- F5 (or ESC-5)** Reset - restore the original value of an entry field
- F6 (or ESC-6)** Command - show the AIX command that will be executed
- F7 (or ESC-7)** Edit - a field in a pop-up box or select from a multi-selection pop-up list
- F8 (or ESC-8)** Image - save the current screen to a file (A) and Show the current fastpath
- F9 (or ESC-9)** Shell - start a sub-shell (A)
- F9** Reset All Fields (M)
- F10 (or ESC-0)** Exit - exit SMIT immediately (A)
- F10** Go to command bar (M)
- F12** Exit - exit SMIT immediately (M)
- Ctrl-I** List - give a pop-up list of possible values (M)
- PgDn (or Ctrl-v)** Scroll down one page
- PgUp (or ESC-v)** Scroll up one page
- Home (or ESC-<)** Go to the top of the scrolling region
- End (or ESC->)** Go to the bottom of the scrolling region

Enter	Do the current command or select from a single-selection pop-up list
/text	Finds the text in the output
n	Finds the next occurrence of the text

Output Screen

```

Command OK  stdout: yes  stderr: no

Before command completion, additional instructions may appear below.

[TOP]
  UID  PID  PPID    C  STIME  TTY  TIME  CMD
  root   1    0    4  20:15:04  -   1:49  /etc/init
  root 1719    1    0  20:16:14  -   0:10  /etc/syncd 60
  root 2003    1    0  20:16:19  -   0:00  /etc/srcmstr
  root 2233    1    0  17:16:14  -   0:00  /usr/lib/errdemon
  ray 3525    1    0  20:01:28  0   0:00  -ksh
  root 3806  2003    0  19:16:23  -   0:00  /etc/syslogd
  ray 4162  3525    6  20:53:22  0   0:04  smit
  root 5355    1    0  20:16:27  -   0:12  /etc/cron
  root 6649  2003    0  20:16:32  -   0:00  qdaemon
  ray 7303  4162    8  20:09:45  0   0:00  ps -ef

[MORE...6]

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

```

Figure 2-9. Output Screen

AU147.0

Notes:

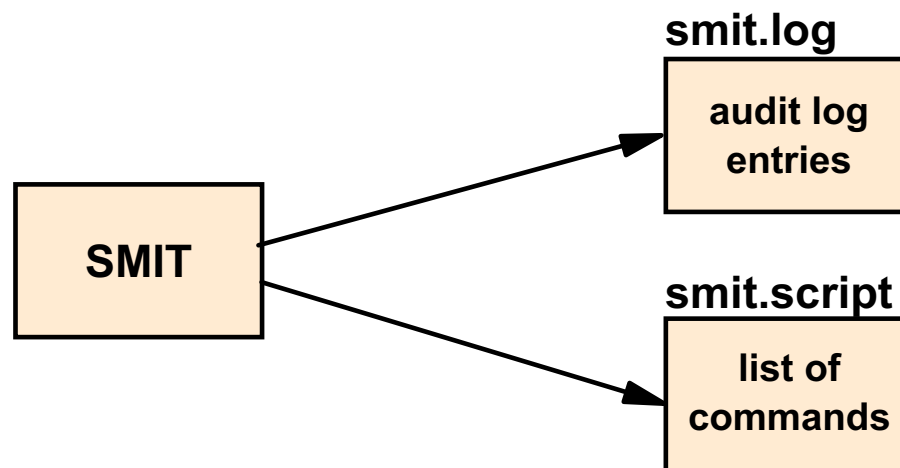
The **Command** field can have the following values: OK, RUNNING and FAILED.

Note that in the Motif version there is a man in the top right hand corner of the screen which is used to indicate the three values.

- **stdout** is the standard output, that is, there is output produced as a result of running the command. The output will be displayed in the body section of this screen.
- **stderr** is the error messages if there are any. In this case there are no error messages.

The body of the screen holds the output/error messages of the command, in this case output.

SMIT Log and Script Files



- **\$HOME/smit.log**

Keeps a log of all menu and dialog screens visited, all commands executed and their output. Also records any errors during the SMIT session.

- **\$HOME/smit.script**

Shell script containing all AIX commands executed by SMIT.

Figure 2-10. SMIT Log and Script Files

AU147.0

Notes:

SMIT creates two files in the **\$HOME** directory of the user running SMIT. If these files already exist, then SMIT will append to them. These files can grow quite large over time, especially during installations, so the user must maintain them and truncate them when appropriate.

The **smit.log** file contains a record of every SMIT screen (menu/selector/dialog) visited, the AIX commands executed, and the output from these commands. When the image key is pressed, the screen image is placed in the smit.log file. If there are any error/warning messages from SMIT or any diagnostic/debugging messages, then these are also appended to the **smit.log** file.

The **smit.script** file just contains the AIX commands executed by SMIT (preceded by the date and time of execution). This file can be used directly as a shell script to perform tasks multiple times, or used as the basis for more complex operations.

smit Command

```
smit [-options] [ FastPath ]
```

- Invoke ASCII Version

```
# smitty
```

- Run no high-level commands

```
# smit -x
```

- Redirect the log file and script file

```
# smit -s /home/team01/smit.script
```

Figure 2-11. smit Command

AU147.0

Notes:

The command **smit** is used to invoke SMIT. It is not common to run smit with any options although a number of them do exist. Here are a sampling of the more commonly used options.

Many administrators prefer the ASCII version of SMIT over the graphical SMIT. If you are working in a graphical environment and want to use the ASCII version of SMIT, use the command **smitty (or smit -C)**. This is commonly used.

Using a SMIT fastpath can be very helpful. Fastpaths are names of individual screens within SMIT. If you want to by-pass the menu system and go straight to a particular screen, use the command **smit fastpath**. When using SMIT, you can view the fastpath screen name by pressing **F8 - Image**.

If you want to explore the menus of SMIT without accidentally running a command, invoke SMIT using **smit -x**. This will log all the normal entries in smit.log and smit.script but will not execute any commands.

Since `smit.log` and `smit.script` are created in the user's HOME directory, this can be a problem if you log in directly as root. Root's HOME directory is `/`. Later you will learn that filling the root area of your disk can cause your machine to crash. You can tell SMIT to log this information elsewhere using **`smit -l`** filename for the `smit.log` and **`smit -s`** filename for `smit.script`.

Exercise: Using SMIT



Figure 2-12. Exercise: Using SMIT

AU147.0

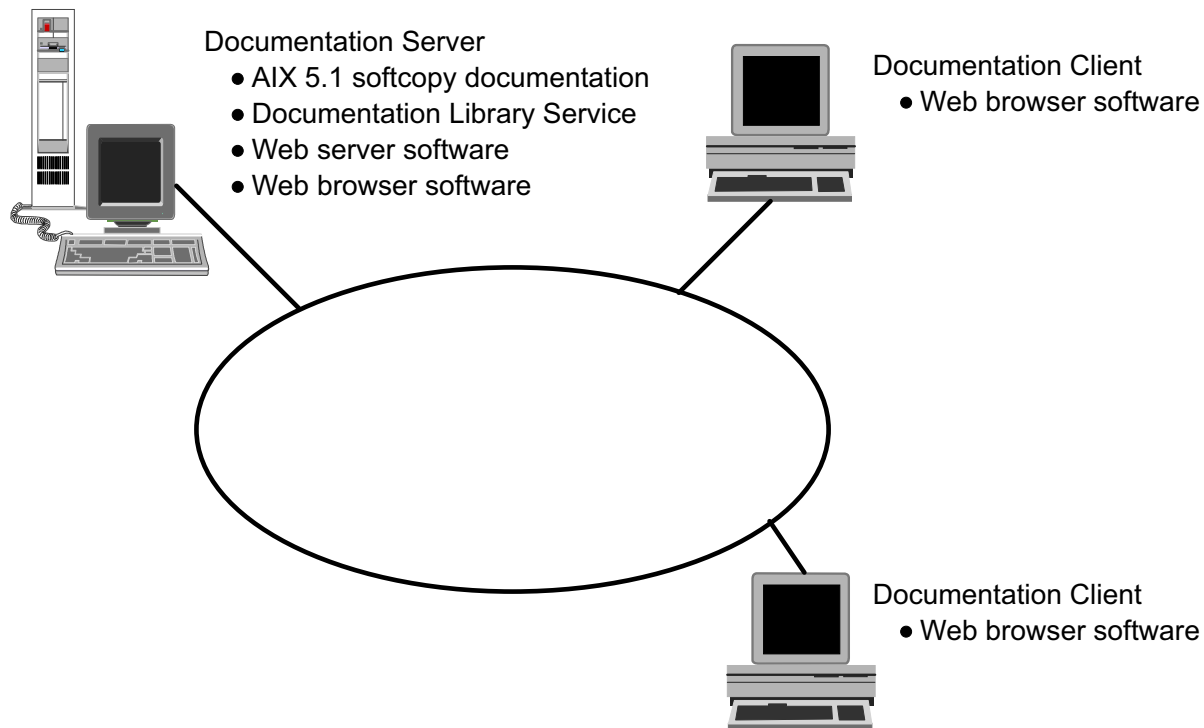
Notes:

This lab will allow you to get familiar with SMIT. In this exercise, you will also be given the chance to use each of the interface AIX provides.

The exercise can be found in your Exercise Guide.

2.2 Configuring AIX Documentation

Configuring AIX V5.1 Documentation



View AIX documentation from anywhere with a browser

Figure 2-13. Configuring AIX V5.1 Documentation

AU147.0

Notes:

In addition to providing SMIT to make system administration tasks easy, AIX also provides Beginning with AIX V4.3, softcopy documentation is loaded on a *documentation server*. Any other computer in the network with appropriate Web-browser software (for example, the Netscape Navigator) can then become a *documentation client*.

When users on a client computer request an AIX document, the request is sent to the Web server on a documentation server which then sends back the requested item. When searches are performed, they are done on the server computer and the results are then sent back to the user on the client computer.

Configuring AIX V5.1 Online Documentation

- Configure TCP/IP
- Install the Web server software
- Configure and start the Web server software
- Install the Web browser software
- Install or mount the AIX documentation
- Configure the Documentation Library Service

Figure 2-14. Configuring AIX V5.1 Online Documentation

AU147.0

Notes:

The steps outlined above are used to configure an AIX V5.1 documentation server or online documentation for a stand-alone RS/6000 system.

1. Configure TCP/IP on the AIX system. This will be discussed later in the course.
2. Install the Web server software. AIX V5.1 includes two products that can be used: the Lite NetQuestion server software and the IBM HTTP Server Web server. Any other Web server software that supports CGI (Common Gateway Interface) can also be used. The Lite NetQuestion server can only be used for local users, not remote users.
3. Configure and start the Web server software. Use IBM HTTP Server Web server for easy set up.
4. Install Web browser software. This is necessary if users on the server wish to access documents. The Netscape Communicator is included with AIX V5.1. Actually, any browser can be used, provided it supports Java 1.3.
5. The AIX V5.1 Documentation includes *User Guides*, *System Management Guides*, *Application Programmer Guides*, *Commands Reference Volumes*, *Files References*,

and *Technical Reference Volumes*. This documentation can be installed to disk or mounted as a CD-ROM file system.

6. Configure the Documentation Library Service (**bos.docsearch**). This is installed by default with the base operating system. To configure it use the **smit web_configure** fastpath or the Web-based System Manager.

Installation of the documentation client involves a subset of the steps outlined above:

1. Install and configure TCP/IP
2. Install the Web browser software
3. Configure the Documentation Library Service. Only the **bos.docsearch** client filesets need to be installed on the clients.

Most of the documentation configuration can be done with the Configuration Assistant. The Configuration Assistant will be discussed in the AIX Installation unit.

Internet and Documentation Services

```
# smit web_configure
```

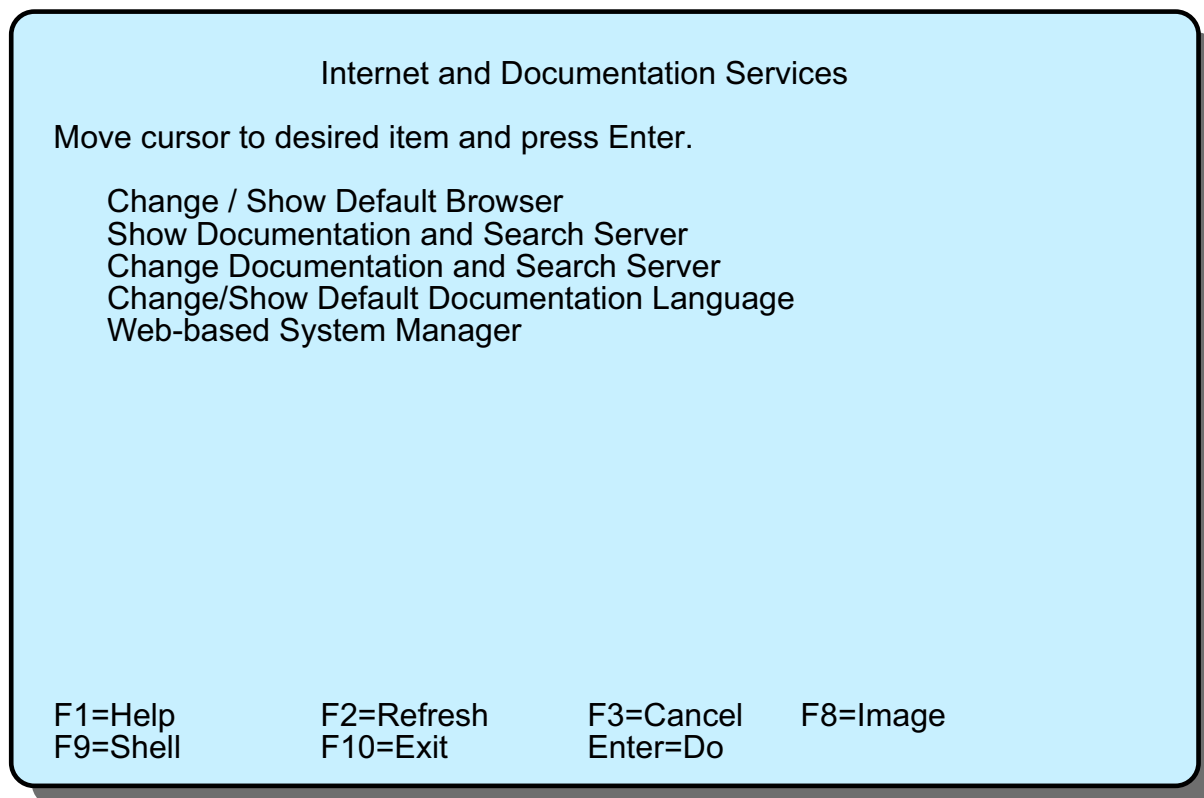


Figure 2-15. Internet and Documentation Services

AU147.0

Notes:

Use the SMIT fastpath **smit web_configure** to access this menu. This menu is also accessed via the **System Environments** option on the main SMIT menu.

Choose the first option, **Change/Show Default Browser** to begin configuration of either a documentation server or client.

The Web-based System Manager can also be used to configure the AIX V5.1 online documentation.

Change/Show Default Browser

Change/Show Default Browser

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

* Default browser LAUNCH COMMAND [Entry Fields]
[netscape]

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

Figure 2-16. Change/Show Default Browser

AU147.0

Notes:

Select Change/Show Default Browser from the smit web-configure screen.

On this screen, type in the command that launches the browser that will be the default browser for users on this system. Indicate the full path name if necessary and any applicable options/flags. Netscape does not require any options/flags.

Use this SMIT screen also on the documentation clients to indicate the default browser.

Change Documentation and Search Server

Change Documentation and Search Server

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

[Entry Fields]
None - disabled

Documentation search server LOCATION

+ _____

Documentation Search Server LOCATION

Move cursor to desired item and press Enter.

None - disabled
Remote computer
Local - this computer

F1=Help F2=Refresh F3=Cancel
 F8=Image F10=Exit Enter=Do
 /=Find n=Find Next

Figure 2-17. Change Documentation and Search Server

AU147.0

Notes:

Select **Change Documentation and Search Server** from smit web-configure screen.

Indicate the location of the documentation server. If configuring the server, choose **Local - this computer**. Choose this option also if using a standalone AIX V5.1 system.

If configuring the client, choose **Remote computer**. If this option is chosen, an additional menu is displayed where the hostname of the server is entered.

Change Local Documentation and Search Server

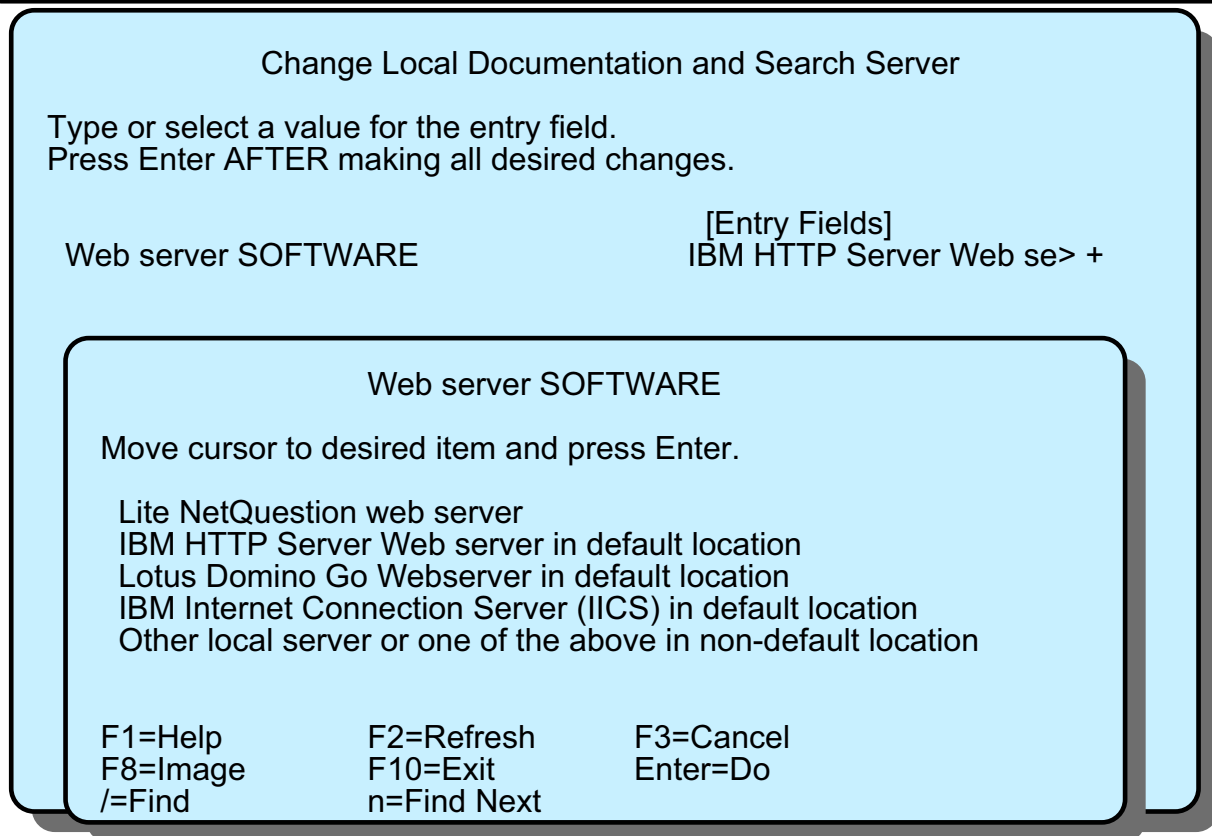


Figure 2-18. Change Local Documentation and Search Server

AU147.0

Notes:

Use this menu to choose the Web server software that is being used. A pop-up menu is available.

AIX Version 5.1 Documentation

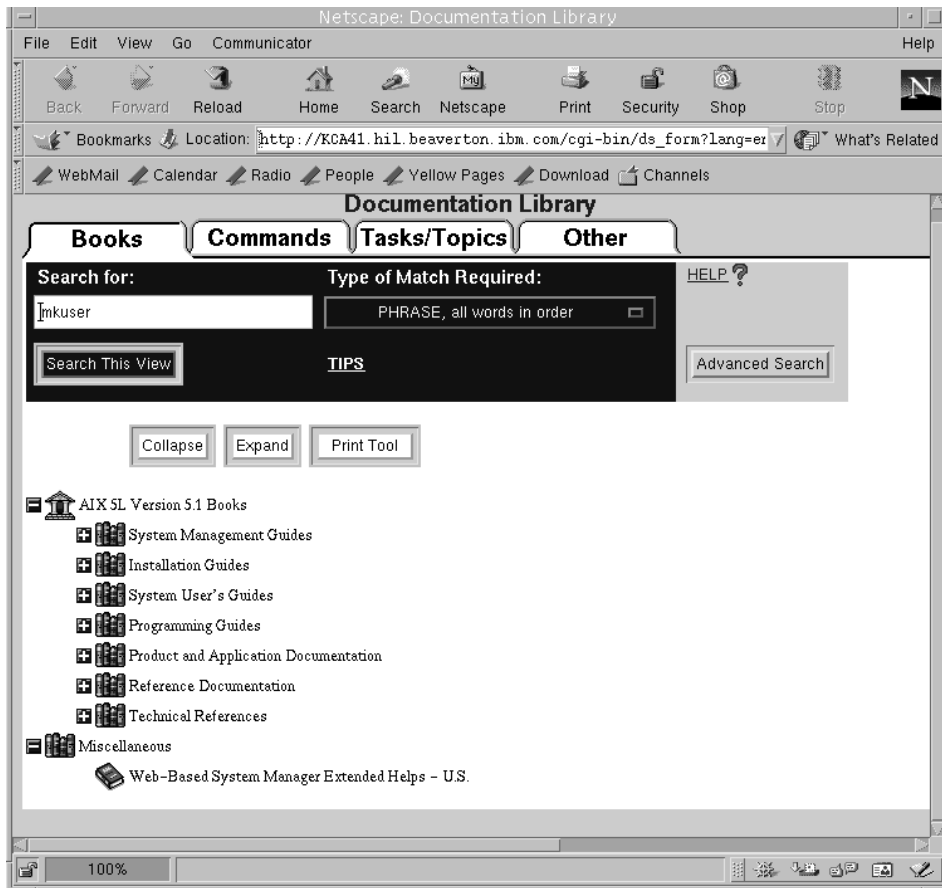


Figure 2-20. AIX Version 5.1 Documentation

AU147.0

Notes:

Once the documentation is set up, it can be accessed with:

- Your web browser (for example, netscape), using the URL:

`http://<hostname>/cgi-bin/ds_form`

The **<hostname>** is the name of the server as configured to TCP/IP. This hostname must be able to be resolved in the `/etc/hosts` file or through DNS.

- The "Search" function from the *Documentation Library* icon using CDE (the Common Desktop Environment)
- The **docsearch** command

Online documentation is also available at: **`http://www.ibm.com/servers/aix/library`**.

On the library home page, near the top, you are given different methods to look at the documentation. You can view the documents by books, look at command documentation or view it by a topics and task list.

Moving down the screen, the next area allows you to perform a search. This is probably the quickest and easiest way to locate information on a specific item. Just type in a key word and let it find the documents for you.

The last part of the screen shows icons representing the books and category of books that are available. You can click the icons to expand their information.

Print AIX Version 5.1 Documentation

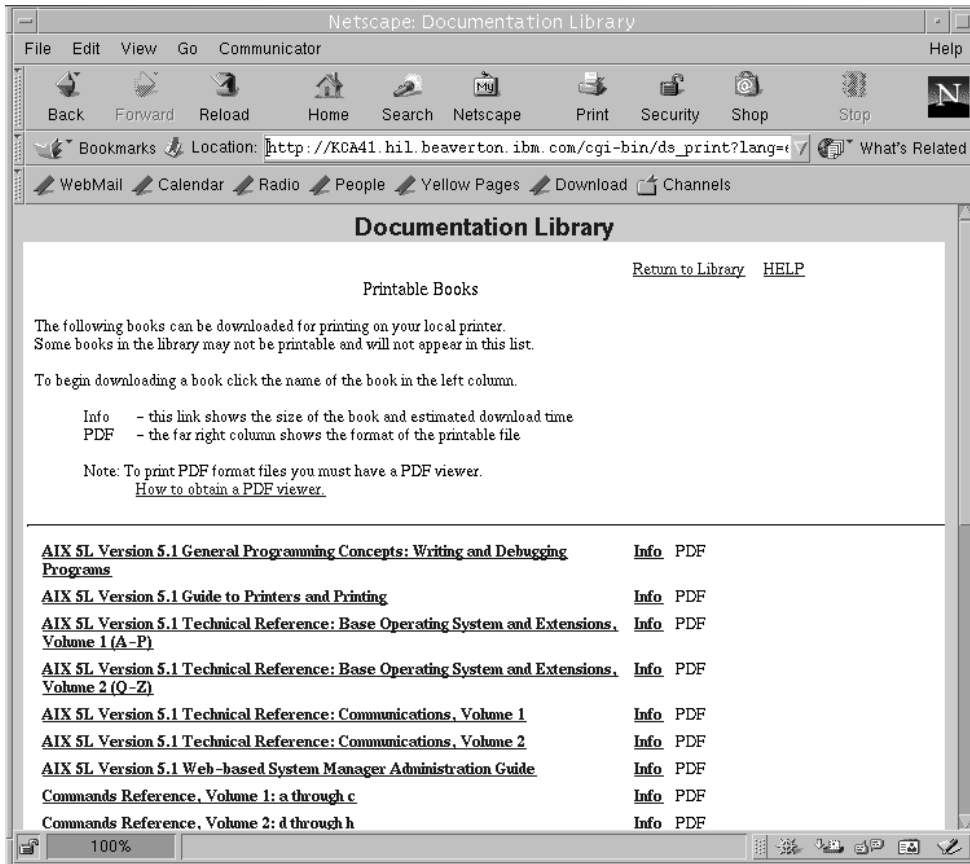


Figure 2-21. Print AIX Version 5.1 Documentation

AU147.0

Notes:

The Documentation Library Service contains a Print Tool button. When you click this button, you see a list of books that can be downloaded in a single printable file. You have the option of customizing this list to include your own book for printing.

Search AIX Version 5.1 Documentation

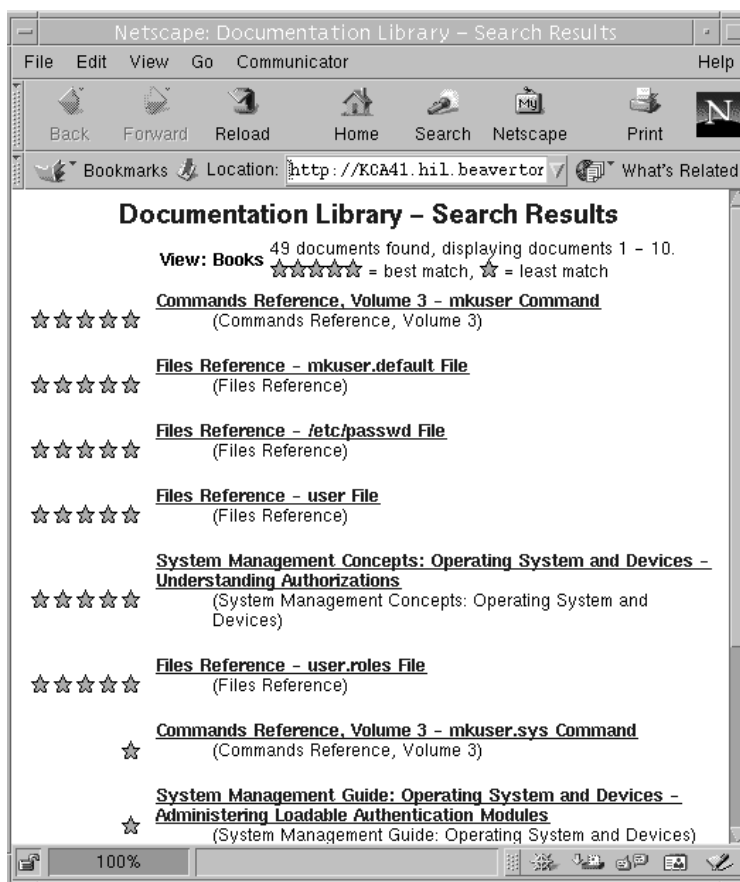


Figure 2-22. Search AIX Version 5.1 Documentation

AU147.0

Notes:

Probably the easiest way to find an answer is to search the documentation using the Search window on the Documentation Library screen.

Above is the results of a search. A star system is used to indicate the documents that best match your keywords. Five stars is the best. Clicking the item will take you to that document.

Activity: Configuring Web-based Documentation

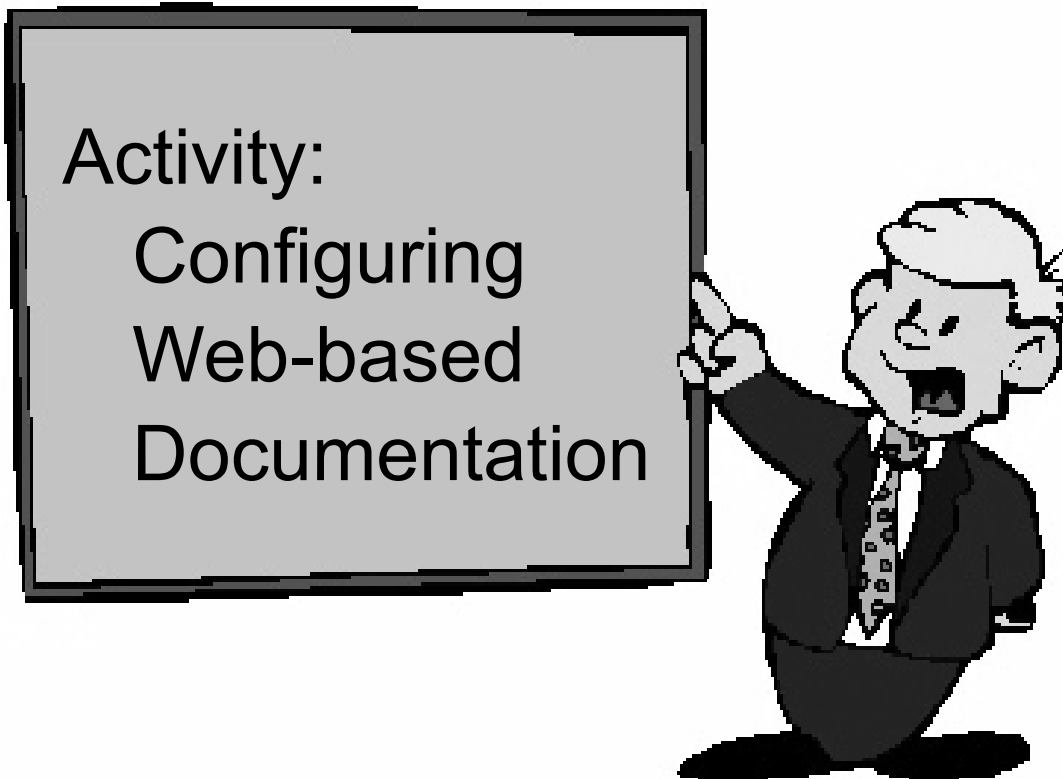


Figure 2-23. Activity: Configuring Web-based Documentation

AU147.0

Activity - Configuring Web-based Documentation

Activity Instructions

Configure the Documentation

1. Log in to AIX as **teamxx** and **su** to **root**.
2. Use SMIT to configure the Documentation. This will allow you to access the AIX V5.1 online documentation. All the necessary software has been installed; all you have to do is perform the configuration so that you can access the online documentation. Use **IBM HTTP Server Web server**.

Verify the AIX online documentation

3. Since you will be accessing the online documentation from a Web server, it is necessary to know your system's TCP/IP host name and IP address. Display this at the command line and record the results. _____
4. Access the AIX V5.1 online documentation.

5. Congratulations! You have configured the AIX online documentation. Be sure to add a bookmark with your browser so you don't need to remember the long URL. When you are done, exit from the browser.
 - On the Netscape toolbar, click **Bookmarks -> Add Bookmarks**.
 - On the Netscape toolbar, click **File -> Exit**.
6. A quick way to locate information in the documentation is to do a search. Use the command (from the command line) that starts the Documentation Library Service. As time permit, get familiar with the Web-based documentation by trying a few searches and looking at some of the documentation. When you are done, log out.

END OF ACTIVITY

Activity Instructions with Hints

Configure the Documentation

1. Log in to AIX as **teamxx** and **su** to **root**.
 - From the login window click **Options** and then click **Command Line Login**.
 - When the unformatted message appears, press **Enter** to get the login prompt.
 - Log in as **teamxx**
 - \$su root**
2. Use SMIT to configure the Documentation. This will allow you to access the AIX V5.1 online documentation. All the necessary software has been installed - all you have to do is perform the configuration so that you can access the online documentation. Use **IBM HTTP Server Web server**.
 - # smit web_configure**
 - Choose **Change/Show Default Browser**. The SMIT screen should show *netscape*. This is the command that is used to launch the web browser. (In your own environment, if you are using a browser other than Netscape, you will need to type in the command that launches that browser, including any applicable options.) Press **Enter**.
 - Press **F3 - Cancel** to return to the **Internet and Documentation Services** menu.
 - Choose **Change Documentation and Search Server**.
 - Press **F4 - List** and then choose **Local - this computer**. Press **Enter**.
 - Press **Enter** again to display the SMIT screen asking for the Web server SOFTWARE. On this screen press F4 and select **IBM HTTP Server Web server**. Press **Enter**.
 - The menu will expand to display additional fields. If you are using the IBM HTTP Server Web server, the fields will already be filled in with the correct values. (In your own environment, if you are using some other Web server software, you will need to fill in the port number being used, the cgi-bin directory and the HTML document directory).
 - Press **enter** to configure the documentation library service. Verify the results and press **F10** to exit SMIT.

Verify the AIX online documentation

3. Since you will be accessing the online documentation from a web server, it is necessary to know your system's TCP/IP host name and IP address. Display this at the command line and record the results. _____
 - # hostname**
 - # host hostname**

4. Access the AIX V5.1 online documentation.

- # **xinit** to bring up AIXWindows
- From a window: # **netscape** to bring up the web browser.
- Click **Accept for the Netscape license agreement**, if asked.
- Once the Netscape window appears, it may be necessary to enlarge the window. If Netscape errors appear, just click **OK** to remove them. At the URL type:

http://<hostname>/cgi-bin/ds_form

The <hostname> in the command should be the name displayed by the **hostname** command. Press **Enter**. The AIX V5.1 Base Documentation screen should appear.

5. Congratulations! You have configured the AIX online documentation. Be sure to add a bookmark with your browser so you don't need to remember the long URL. When you are done, exit from the browser.

- On the Netscape toolbar, click **Bookmarks -> Add Bookmarks**.
- On the Netscape toolbar, click **File -> Exit**

6. A quick way to locate information in the documentation is to do a search. Use the command (from the command line) that starts the Documentation Library Service. As time permit, get familiar with the Web-based documentation by trying a few searches and looking at some of the documentation. When you are done, log out.

•# **docsearch**

END OF ACTIVITY

2.3 WebSM

Web-based System Manager

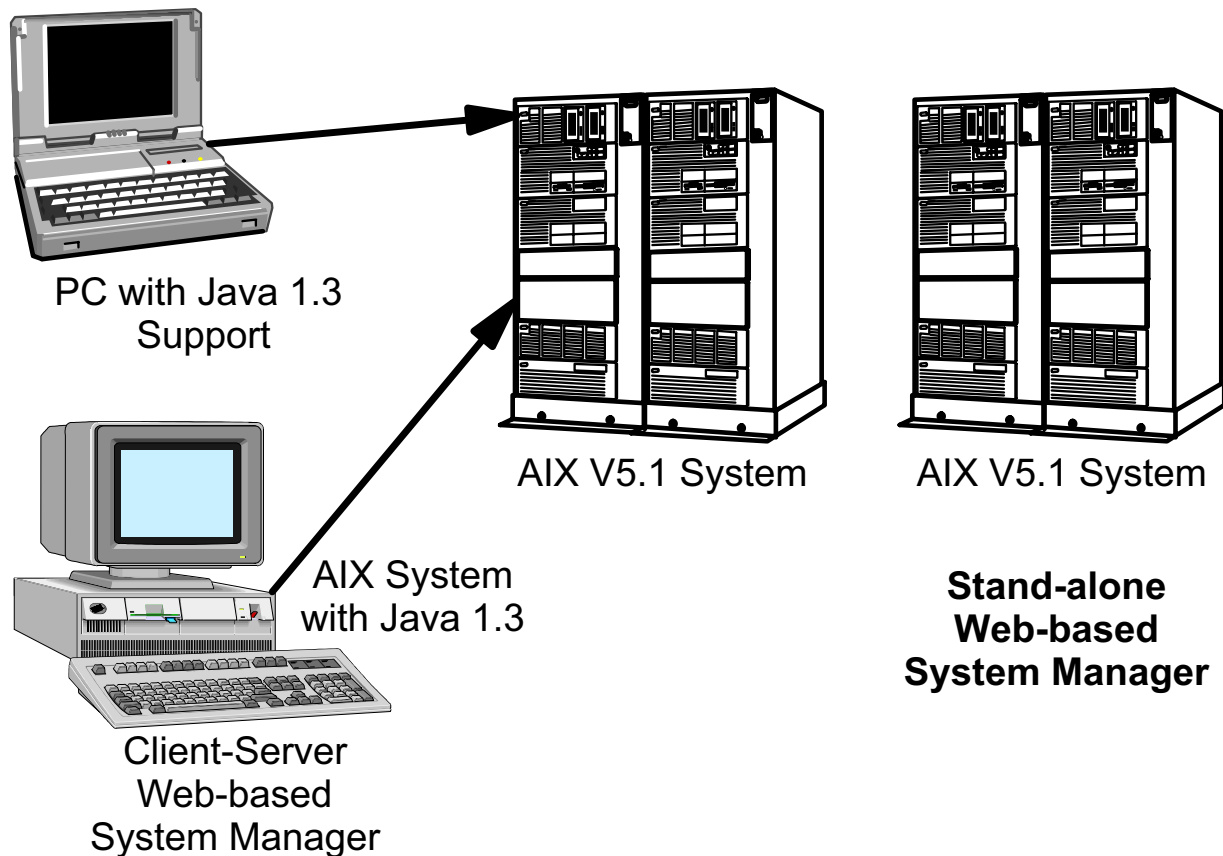


Figure 2-24. Web-based System Manager

AU147.0

Notes:

AIX V4.3 introduced the Web-based System Manager, which is the next step in the evolution of AIX system administration tools. The Web-based System Manager can be run in stand-alone mode, that is, you can use this tool to perform system administration functions on the AIX system you are currently running on. However, the Web-based System Manager also supports a client-server environment. In this environment, it is possible to administer an AIX system from a remote PC or from another AIX system using a graphics terminal. In this environment, the AIX system being administered is the *server* and the system you are performing the administration functions from is the *client*.

The client can operate in either application mode on AIX with Java 1.3 or in applet mode on platforms that support Java 1.3. Thus, the AIX system can be managed from another AIX system or from a PC running Microsoft Windows 95 or Windows NT.

The objectives of the Web-based System Manager are:

- Simplification of AIX administration by a single interface

- Enable AIX systems to be administered from almost any client platform (client must have a browser that supports Java 1.3)
- Enable AIX systems to be administered remotely
- Provide a system administration environment that provides a similar look and feel to the Windows 95/NT and AIX CDE environments

The Web-based System Manager provides a comprehensive system management environment and covers most of the tasks in the SMIT user interface. The Web-based System Manager can only be run from a graphics terminal so SMIT will need to be used in the ASCII environment.

Accessing the Web-based System Manager

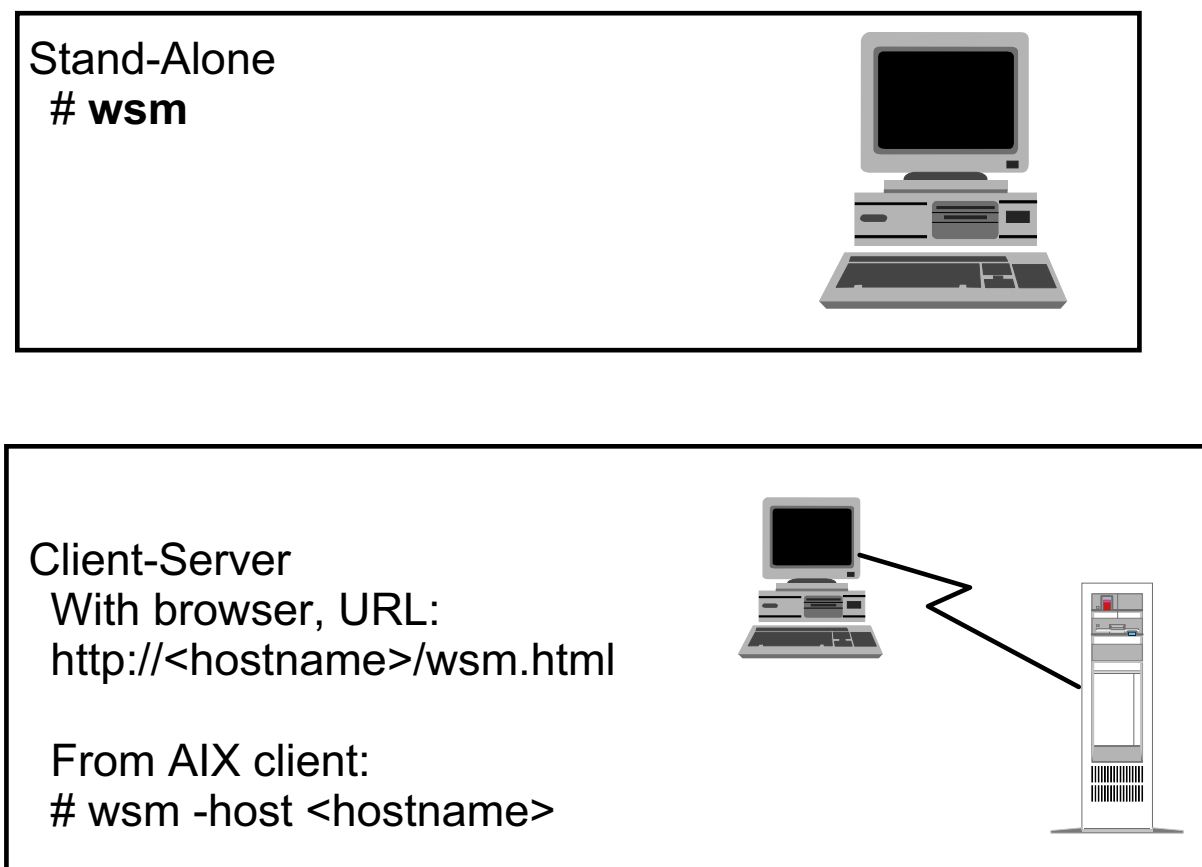


Figure 2-25. Accessing the Web-based System Manager

AU147.0

Notes:

In stand-alone mode, to access the Web-based System Manager use the command **wsm**.

- From the CDE Application Manager, you can also access by icons if you are using CDE. Open the **System Admin** folder and double-click on **Management Console** to view icons for each of the Web-based System Manager applications.

If using the Web-based System Manager in client-server mode:

- If the Web-based System Manager client is running as a Java applet in a browser, or as a stand-alone Java application, use the appropriate URL to access the tool. The default URL is **`http://<hostname>/wsm.html`**
- From an AIX V5.1 client, use the command **`wsm -host <hostname>`**. This will bring up a login box where you enter your ID and password for the remote AIX system.

Using the Web-based System Manager(1 of 3)



Figure 2-26. Using the Web-based System Manager (1 of 3)

AU147.0

Notes:

The graphic above shows the Web-based System Manager Console Window containing two primary panels. The panel on the left displays the machines that you can manage from the Console Window. This panel is referred to as the *Navigation Area*. The panel on the right (the *Contents Area*) displays results based on the item selected in the Navigation Area. You select the machine to perform management operations from the Navigation Area. As you navigate to the desired operation in the Navigation Area, the Contents Area is updated to show the allowable choices.

There is a session log that is a facility of the console. It keeps track of changes made on managed hosts during a Web-based System Manager session.

Using the Web-based System Manager(2 of 3)

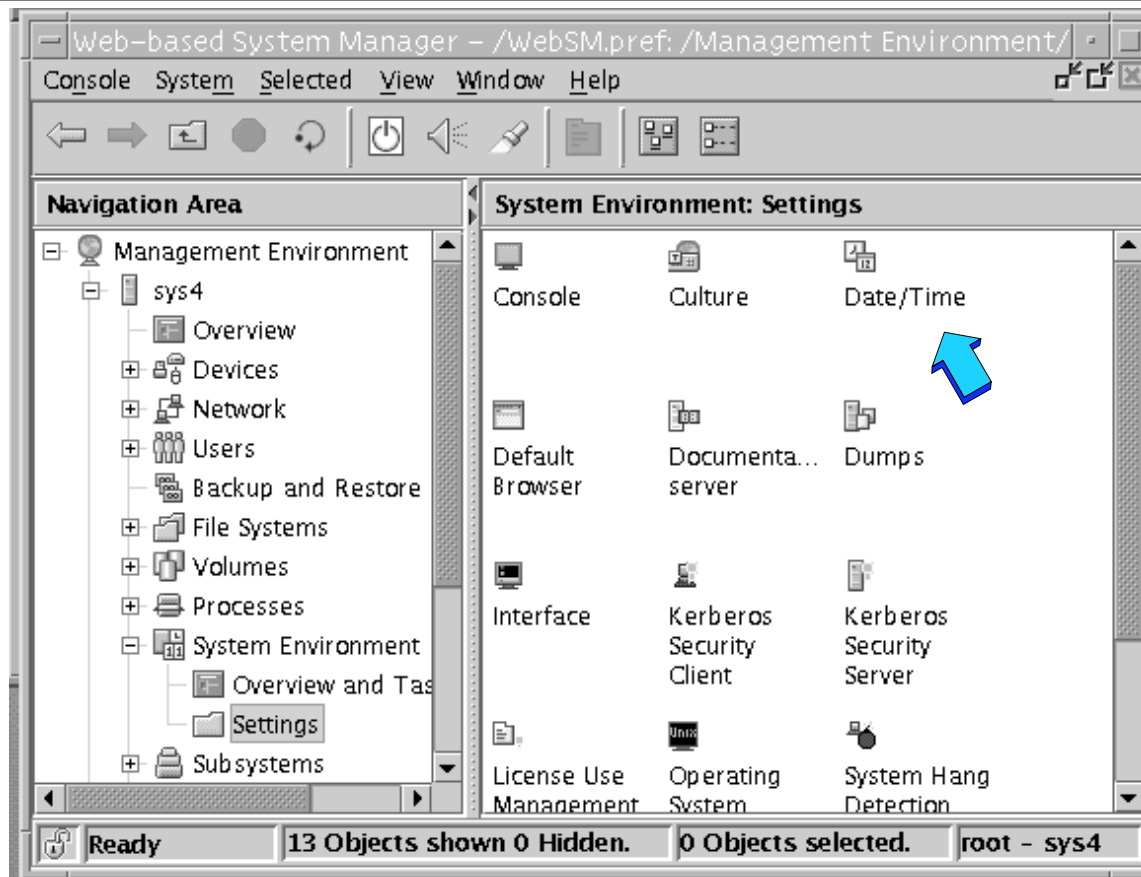


Figure 2-27. Using the Web-based System Manager (2 of 3)

AU147.0

Notes:

The graphic shows the **System Environment: Settings** window of the Web-based System Manager. This window contains a toolbar. From left to right, the symbols support the following functions: Back to previous screen, Forward to next screen, Up one level, Stop reloading, Reload now, Shutdown, Broadcast message, Find, Show properties of highlighted object, Icon (to return to icon mode if currently viewing details), Details (which lists each icon and provides an explanation of each). Most of these functions can also be accessed via the **View** option on the menu bar.

If you select the Date and Time icon, this will allow you to set the date and time on the system.

Using the Web-based System Manager(3 of 3)

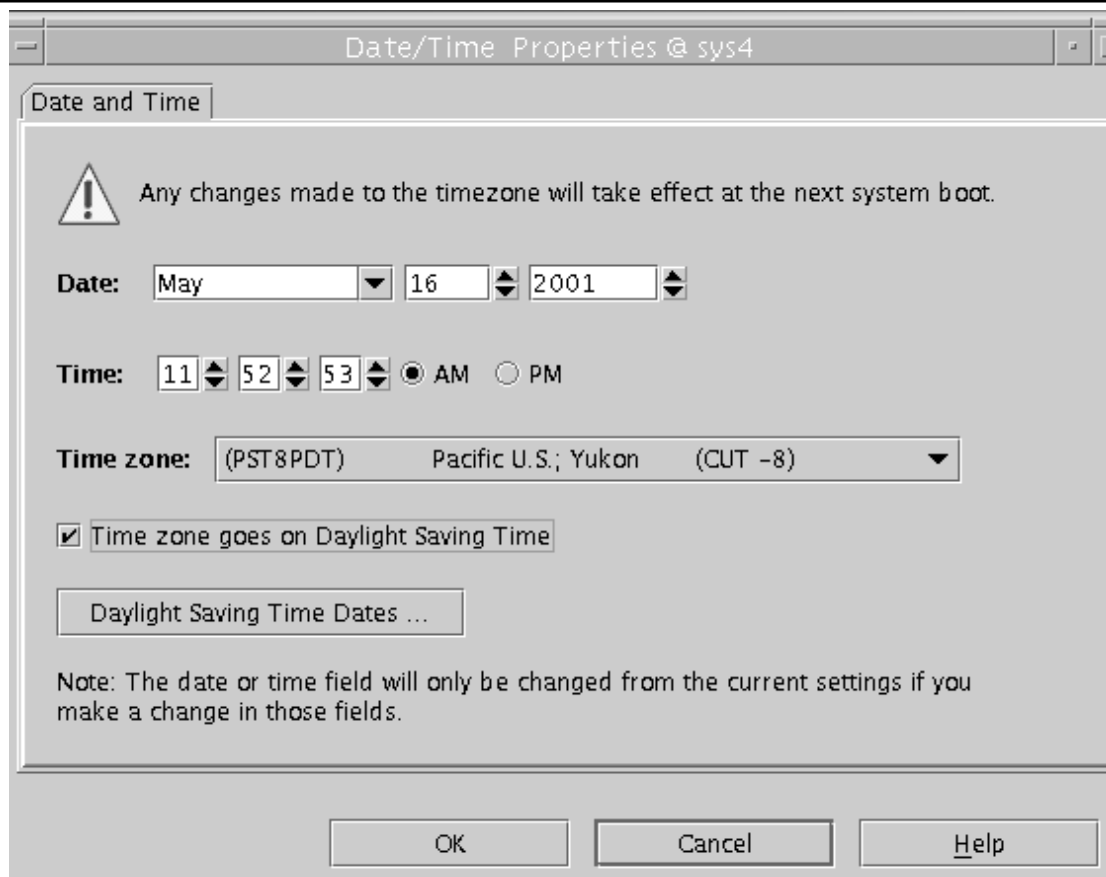


Figure 2-28. Using the Web-based System Manager (3 of 3)

AU147.0

Notes:

Note that the Web-based System Manager supports an easy-to-use point-and-click environment where information can be entered. Use this window to set the system date and time (only the root user can perform this function). When finished, click **OK** to apply your change.

Additional information on the Web-based System Manager can be accessed through the Internet using the URL:

<http://www.rs6000.ibm.com/software/OS/wsm/wsmFAQ.html>

Configuring Client/Server WebSM

- Install the Web server
- Test the Web server
- Install WebSM (usually done by default with the base)
- Define the Web server document directory.

```
# /usr/websm/bin/wsmappletcfg -docdir directory
```

- Enable WebSM server

```
# /usr/websm/bin/wsmserver -enable
```

Figure 2-29. Configuring Client/Server WebSM

AU147.0

Notes:

These are the steps needed to set up the Web server from scratch. If you already have set up the Web-based documentation, the first two steps (Install the Web server and Test the Web server) are already done.

WebSM is installed by default in AIX 5.1. The following filesets are installed from the AIX 5.1 Base Installation media:

```
sysmgt.help.msg.en_US.websm  
sysmgt.msg.en_US.websm.apps  
sysmgt.websm.apps  
sysmgt.websm.diag  
sysmgt.websm.framework  
sysmgt.websm.icons  
sysmgt.websm.rte  
sysmgt.help.en_US.websm
```


To set up the documentation directory, you need to know the location of the document directory for the web server you are using. We will be using the IBM HTTP Server Web server in the classroom. The path needed is **/usr/HTTP Server/htdocs**

Run the following command:

/usr/websm/bin/wsmappletcfg -docdir directory

For example, for IBM HTTP Server Web server, the command would be:

/usr/websm/bin/wsmappletcfg -docdir /usr/HTTPServer/htdocs

This can also be done through smit using the fastpath

smit web_based_system_manager.

Next, enable the WebSM server

/usr/websm/bin/wsmserver -enable

To accessing WebSM from the client machine, use the URL:

http://*hostname*/wsm.html

Exercise: Configuring WebSM server



Figure 2-30. Exercise: Configuring WebSM server

AU147.0

Notes:

This lab will allow you to set up WebSM and learn how to use this interface. If you have other machines in your classroom that are networked together, you can also try to perform remote administration using WebSM.

The exercise can be found in your Exercise Guide.

Checkpoint (1 of 2)

1. Define the SMIT function keys that can be used for the following:
 - List the command that will be run _____
 - List the screen name which can be used for the fastpath _____
 - Take a screen image: _____
 - Break out into a shell: _____
 - Return to the previous menu: _____
2. T/F AIX Web-based documentation can be used to reference information in different ways, such as searching for a command, searching for a task or viewing information in a book like manner.
3. T/F The AIX V5.1 documentation is viewed using a Web browser.
4. WebSM is available for client access automatically after the BOS is installed. True or False?

Notes:

Checkpoint (2 of 2)

5. Which of the statements are true regarding the Web-based System Manager?
 - a. An AIX V5.1 system can be managed from a remote PC with appropriate JAVA and web-browser code installed.
 - b. In stand-alone mode use the **wsm** command to access the Web-based system manager.
 - c. It is possible to manage an AIX V5.1 system from a remote AIX V5.1 system using an ascii terminal.
 - d. The Web-based System Manager includes TaskGuides that direct the user through complex tasks.

Notes:

Unit Summary

- There are a number of system management tools that can be used by the system administrator, such as SMIT, and the Web-based System Manager
- SMIT provides graphics or ASCII support for most system administration tasks
- The Web-based System Manager supports system administration tasks in a stand-alone or client-server environment
- Use a Web browser to access soft copy documentation with AIX

Figure 2-33. Unit Summary

AU147.0

Notes:

Unit 3. System Startup and Shutdown

What This Unit Is About

This unit discusses how the system environment should be managed.

What You Should Be Able to Do

After completing this unit, you should be able to:

- Describe the system startup process
- Describe how to shut down the system
- Describe the contents of the `/etc/inittab` file
- Manage the system environment

How You Will Check Your Progress

Accountability:

- Checkpoint questions
- Exercise

References

Online *System Management Guide: Operating System and Devices*

Objectives

After completing this unit, students should be able to:

- Describe the system startup process
- Describe how to shut down the system
- Describe the contents of the /etc/inittab file
- Manage the system environment

Figure 3-1. Objectives

AU147.0

Notes:

Startup Modes

Normal Mode

- Login prompt
- All processes running
- Multi-user mode

System Management Services

- Not AIX
- Runs from FIRMWARE
- Sets boot list

Maintenance Mode

- Maintenance menu
- Recover root password
- Fix machine that won't boot

Diagnostics

- AIX Diagnostics

Figure 3-2. Startup Modes

AU147.0

Notes:

When you power on your RS/6000, one of the first things it will do is determine which device it should use to boot the machine. By default, the machine will use the “normal” boot list which usually contains one or more hard drives. When the machine does a “normal” boot, it will complete the full AIX boot sequence and start processes, enable terminals and generate a login prompt to make it available for multi-user access. It also activates the disks, sets up access to the files and directories, starts networking and completes other machine specific configurations.

Another boot option for the RS/6000 is to boot machine specific code called the System Management Services (SMS) programs. These programs are not part of AIX. This code is shipped with the hardware and is built-in to the firmware. This can be used to examine the system configuration and set boot lists without dependency on an operating system. It is invoked during the initial stages of the boot sequence using the **F1** key.

If your system will not boot or you have lost the root password, you will need to boot your machine using bootable media other than the hard drive (like an installation CD or bootable backup - mksysb tape). This will boot you into “maintenance” mode. To do this, you need

to ensure that the device that contains your alternate boot media (CD or tape) is in the boot list. When you boot from the new media, you will be given “backdoor” access to your system.

There is one other boot option - that is to boot into diagnostics. This can be accomplished by using bootable media specifically designed for diagnostics or the diagnostic mode is invoked when the hard drive is the boot device during a “service” boot. Diagnostics is discussed in the System Administration II class.

All machines have a normal boot list and one or more service boot lists. The normal boot list is the default boot list. The service boot list is invoked (like SMS) during the initial stages of the boot sequence using function keys (**F5** or **F6**). We will discuss the service boot list more in a later unit.

When connecting to systems via TTYs and with newer models like F80, M80 and H80, you have to use the 1, 5, and 6 keys as there are no **F**-keys.

Starting System Management Services

- Power on the system
- Press F1 when icons appear and tones sound

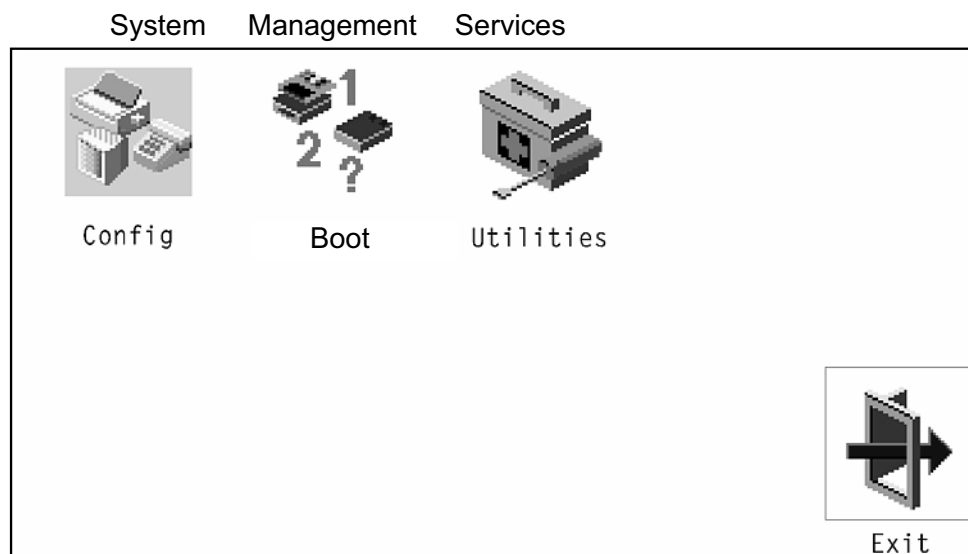


Figure 3-3. Starting System Management Services

AU147.0

Notes:

If you want to set the boot lists or view the system hardware configurations without the aid of AIX, you can use the **System Management Services (SMS) programs**.

To invoke SMS, power on (or reboot) the system. You will hear one beep when the machine first powers on. About 30 to 45 seconds later, you will hear a different tone. This is what you are listening for. Also, you will probably hear the monitor activate. You have about a 15 second time frame to press **F1**. If you hear the music play, you've waited too long. As the monitor warms up, you might see hardware icons appear on the screen. You want to press the **F1** key before it reaches the last hardware device (speaker). Don't wait for the screen to warm up however, because many times as the icons are beginning to appear, the music is sounding and it is too late.

Timing is everything!

Above is an example of a System Management Service main menu. The exact configuration of the menu will vary depending on the model of PCI RS/6000 system being used. The four main services in this example include:

Config - View the hardware configuration of the system

Boot - View or change the boot device order list

Utilities - Set power-on and supervisory passwords, and enabling unattended start mode, view vital product data, view the error log, select active console, and updating the firmware.

Exit - Return to previous screen.

PCI RS/6000 Start Up Process Overview

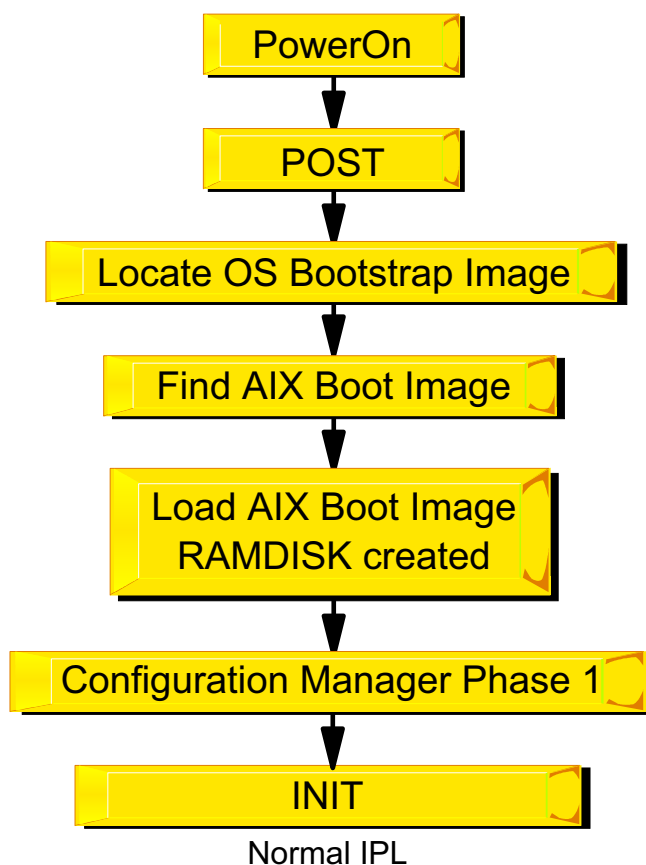


Figure 3-4. PCI RS/6000 Start Up Process Overview

AU147.0

Notes:

During the boot process a number of steps must be completed. The LED panel will provide information on the boot progress. Some values displayed are model specific. These values can be found in the Service Guide for that specific model. Values for the 43P Model 140 can be found in the appendix of the student notebook.

The initial step in booting a machine completes a Power-on Self Test (POST). This step initialized memory, the keyboard, communication adapters, and audio components. The icon related to each device is displayed on the screen. This is the same point where you would press a function key to choose a different boot list. The LED values display during this part are model specific.

Once the POST is completed, the system will locate and load bootstrap code. This part is completed by System ROS (Read Only Storage) stored in the firmware. The bootstrap code, sometimes referred to as Software ROS or level 2 firmware, takes control and builds AIX specific boot information, then locates, loads and turns control over to the AIX boot logical volume (BLV). Because these machines can run different operating systems, the System ROS is generic boot information for the machine and is operating system

independent. The Software ROS is AIX information created based on the machine type and is responsible for completing machine preparation to enable it to start an AIX kernel.

The AIX kernel is then loaded and takes control. The kernel completes the boot process by configuring devices and starting the **init** process. LED codes during this part are generic AIX codes. These will be the same on all AIX systems.

AIX Startup on Itanium-based Systems

IBM POWER processor

Intel Itanium-based processor

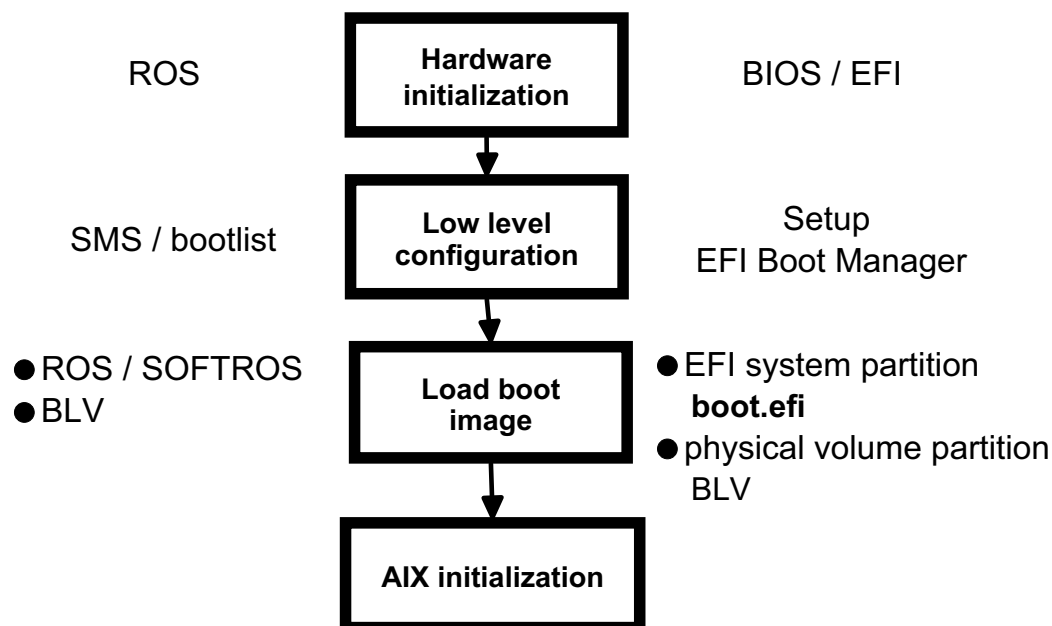


Figure 3-5. AIX Startup on Itanium-based Systems

AU147.0

Notes:

Architecture types

With the addition of support for Intel Itanium architecture, AIX 5.1 now supports four architecture types:

Architecture	Processor	Description
rs6k	POWER	This is the original or “classic” RS/6000 workstation based on the microchannel bus
rspc	POWER	POWER Reference Platform, based on the PCI bus
chrp	POWER	Common Hardware Reference Platform, based on the PCI bus
ia64	Intel Itanium	Intel Itanium architecture

The **bootinfo -p** command returns the system architecture type.

Hardware initialization

When the power is turned on, system firmware performs hardware initialization. Hardware initialization includes: detecting what hardware is present, performing basic functionality tests, and initializing hardware settings.

On Itanium-based systems, this firmware is called BIOS (Basic Input Output System) and EFI (Extensible Firmware Interface).

Low level configuration

After the hardware is initialized, the system provides an opportunity to enter a low level configuration tool. This tool allows you to display system information and perform various configuration tasks, including configuring the boot list. The boot list is an ordered list of possible devices.

On Itanium-based systems, the **Setup** menu provides the ability to display and set hardware configuration. You can manage the boot list using the **EFI Boot Manager**. There is no facility to modify the boot list from AIX.

Load the boot image

Using the information in the boot list, the system locates the boot device and attempts to load the boot image into system memory. The boot image includes the programs and files required to start AIX, including the AIX kernel, the RAM file system (RAMFS), and a copy of the base ODM.

Itanium-based systems

On Itanium-based systems, the boot disk is divided into two partitions:

- EFI system partition
- Physical volume partition

The EFI system partition is a small section of the disk required by the EFI firmware. It contains the boot loader (a program called **boot.efi**).

The physical volume partition contains rootvg, which contains the boot logical volume.

boot.efi loads the boot image from the boot logical volume into memory.

AIX initialization

AIX initialization begins when the boot image has been loaded into memory. AIX initialization is essentially the same for both POWER and Itanium-based systems.

More Information

Complete details on AIX startup on Itanium-based systems can be found in an appendix.

alog

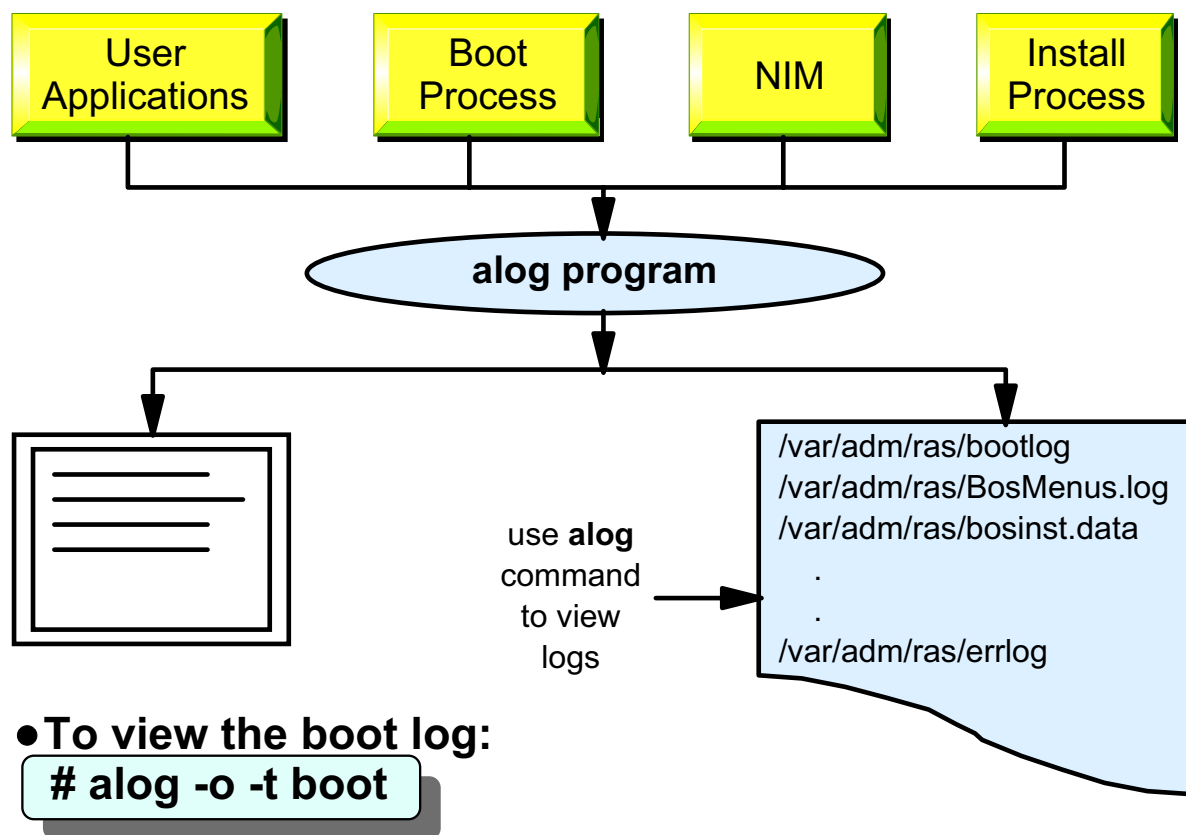


Figure 3-6. alog

AU147.0

Notes:

The **alog** command is a BOS feature that provides a general-purpose logging program that can be used by any application or user to manage a log. The **alog** command reads standard input and writes the output to standard out and copies it to a fixed size file at the same time. The file is treated as a circular log. That means when it is filled, new entries are written over the oldest entries. Log files used by **alog** are specified on the command line or defined in the **alog** configuration database maintained by the ODM. The system-supported log types are **boot**, **bosinst** and **nim**.

Many users start the boot process and then go and get a cup of coffee. Unfortunately, boot messages may appear on the screen, only to be scrolled and lost, never to be seen by the user. In some instances, these messages may be important, particularly if the system did not boot properly. Fortunately, **alog** is used by the **rc.boot** script and the configuration manager during the boot process to log important events. To view the boot information, the command **alog -o -t boot** may be used. If the machine will not boot, boot the machine into maintenance mode and view the **boot** log contents.

/etc/inittab

• Format of the file: id:runlevel:action:command

```

init:2:initdefault:
brc::sysinit:/sbin/rc.boot 3 >/dev/console 2>&1 # Phase 3 of system boot
powerfail::powerfail:/etc/rc.powerfail 2>&1 | alog -tboot > /dev/console # Power
Failure Detection
rc:23456789:wait:/etc/rc 2>&1 | alog -tboot > /dev/console # Multi-User checks
fbcheck:23456789:wait:/usr/sbin/fbcheck 2>&1 | alog -tboot > /dev/console # run
/etc/firstboot
srcmstr:23456789:respawn:/usr/sbin/srcmstr # System Resource Controller
rctcpip:23456789:wait:/etc/rc.tcpip > /dev/console 2>&1 # Start TCP/IP daemons
rcnfs:23456789:wait:/etc/rc.nfs > /dev/console 2>&1 # Start NFS Daemons
rchtcpd:23456789:wait:/etc/rc.httppd > /dev/console 2>&1 # Start HTTP daemon
cron:23456789:respawn:/usr/sbin/cron
piobe:2:wait:/usr/lib/lpd/pio/etc/pioint >/dev/null 2>&1 # pb cleanup
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
l2:2:wait:/etc/rc.d/rc 2
l2:3:wait:/etc/rc.d/rc 3
...
tty0:2:respawn:/usr/sbin/getty /dev/tty0
ctrmc:2:once:/usr/bin/startsrc -s ctrmc > /dev/console 2>&1
cons:0123456789:respawn:/usr/sbin/getty /dev/console

```

Figure 3-7. /etc/inittab

AU147.0

Notes:

The **/etc/inittab** file lists the processes that **init** will start, and it also specifies when to start them. If this file gets corrupted, the system will not boot properly. It is useful to keep a backup of this file.

The fields are:

- **identifier** - Up to 14 characters that identify the process. Terminals use their logical device name as an identifier.
- **runlevel** - Defines what run levels the process is valid for. AIX uses run levels of 0-9. If the **telinit** command is used to change the runlevel, a SIGTERM signal will be sent to all processes that are not defined for the new run level. If after 20 seconds a process hasn't terminated, a SIGKILL signal is sent. The default run level for the system is **2**, which is AIX multiuser mode.
- **action** - How to treat the process. Valid actions are:
 - **respawn**: If the process does not exist, start it
 - **wait**: Start the process and wait for it to finish before reading the next line

- once: Start the process and do not restart it if it stops
 - sysinit: Commands to be run before trying to access the console
 - off: Do not run the command
- **command** - The AIX command to run to start the process

The **telinit** command can be used to cause init to re-read the **/etc/inittab** file. You may need to do this if **init** stops respawning the getty process on a TTY due to line errors.

AIX uses a default run level of 2. This is the “normal” multi-user mode. You may want to perform maintenance on your system without having other users logged in. The command **shutdown -m** will place your machine into a single user mode terminating all log-ins. Once the machine reaches the single user mode, you will be prompted to enter root password. When you are ready to return to normal mode, type **telinit 2**.

Because this file controls part of the boot process, great care should be taken to prevent it from becoming corrupt. Not using **vi** to edit this file is a good place to start. AIX provides several commands to add, change and remove entries from **/etc/inittab**. They are **mkitab**, **chitab**, and **rmitab**. These commands perform syntax checking to ensure there are no invalid lines in this file.

After editing the **/etc/inittab** file, force the system to re-read the file by using the **telinit q** command.

System Resource Controller

- Provides a single interface to control subsystems
- Controls individual or groups of subsystems

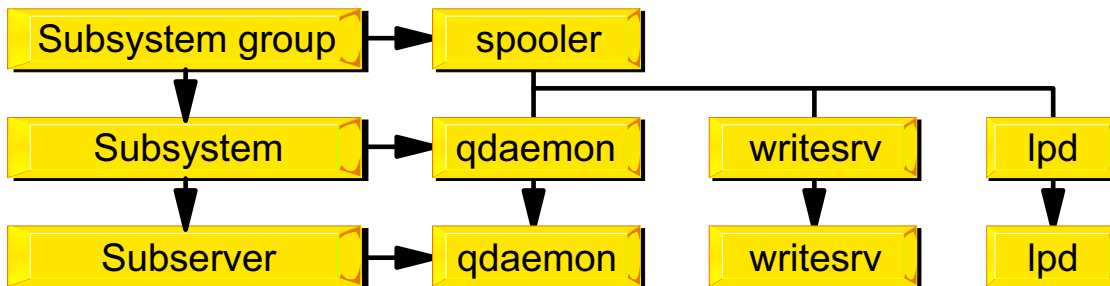


Figure 3-8. System Resource Controller

AU147.0

Notes:

The **System Resource Controller (SRC)** provides a set of commands to make it easier for the administrator to control subsystems. A subsystem is a program (or a set of related programs) designed to perform a function. This can be further divided into subservers. Subservers are similar to daemons. SRC was designed to minimize the need for user intervention since it provides control of individual subsystem or groups of subsystems with a few commands.

The relationship between the group and subsystem is easily seen from the output of **lssrc -a**. The graphic shows the relationship between the spooler subsystem group and its subsystems qdaemon, writesrv, and lpd. Some subsystem have subservers. For example, the tcp/ip group contains a subsystem, inetd, that has several subservers, for example ftp and telnet.

System Resource Controller Syntax

List SRC Status

```
# lssrc -g spooler
subsystem  Group      PID      Status
qdaemon    spooler    8022     active
writesrv   spooler    9558     active
lpd        spooler                    inoperative
```

Start a Subsystem

```
# startsrc -s lpd
0513-059 The lpd Subsystem has been started. Subsystem PID is 12472.
```

Refresh a Subsystem

```
# refresh -s lpd
0513-095 The request for subsystem refresh was completed successfully.
```

Stop a Subsystem

```
# stopsrc -s lpd
0513-044 The lpd Subsystem was requested to stop.
```

Figure 3-9. System Resource Controller Syntax

AU147.0

Notes:

These are examples of SRC commands.

The **lssrc** command is used to show the status of SRC. In the example, we are checking the status of the spooler group using **-g**. To list the status of all processes, the **-a** should be used (**lssrc -a**).

-s and **-g** are options that control subsystems or subsystem groups respectively. These can be used with the SRC commands.

In the remaining examples, we are controlling one subsystem, **lpd** - the daemon that controls the print server. Use **startsrc** to start subsystems or groups. Use **stopsrc** to stop subsystems or groups. The **refresh** command forces the subsystem to reread any of its configuration files.

Stopping Processes

```
# ps -ef
```

UID	PID	PPID	C	STIME	TTY	TIME	CMD
root	1	0	0	May 04	-	0:11	/etc/init
root	2626	1	0	May 04	-	1:17	/usr/sbin/syncd 60
root	4136	1	0	May 04	-	0:00	/usr/sbin/srcmstr
root	4964	4136	0	May 04	-	0:00	/usr/sbin/inetd
root	6734	1	0	May 04	-	0:02	/usr/sbin/cron
root	8022	4136	0	May 04	-	0:00	/usr/sbin/qdaemon
root	9036	1	0	May 04	-	0:00	/usr/sbin/uprintfd
root	9345	1	0	May 04	-	0:02	/usr/bin/program

For process not started by srcmstr

```
# kill 9345
```

For processes started by SRC

```
# stopsrc -s qdaemon
```

Figure 3-10. Stopping Processes

AU147.0

Notes:

Because some processes are started using SRC, they should be stopped using SRC. If you are not sure how it was started, you can run **lssrc** to view what is controlled by SRC. Or, by examining the output from **ps -ef**, you can determine the same information.

In the output above, **srcmstr** has a PID of 4136. Any processes with PPID of 4136 is controlled by SRC. These should be stopped using **stopsrc** - as is the case with **qdaemon**. Processes that do not have a PPID of 4136 are not controlled by SRC and can be stopped with the **kill** command - as is the case with **program**.

System Shutdown

shutdown command

- Gracefully stops all activity on the system and advises all logged on users.
- Warns users of a impending shutdown

```
# shutdown +2 The system will be down until 3AM

Broadcast message from root@localhost (tty) at 1:30:20...

The system will be down until 3AM

shutdown: PLEASE LOG OFF NOW!!!
All processes will be killed in 2 minutes
```

Figure 3-11. System Shutdown

AU147.0

Notes:

The SMIT **shutdown** fastpath or the **shutdown** command is used to shut the system down cleanly.

If used with no options, it will display a message on all enabled terminals (using the **wall** command), then after one minute will disable all terminals, kill all processes on the system, sync the disks and unmount all file systems and then halt the system.

You can also use shutdown with the **-F** option for a fast immediate shutdown (no warning), **-r** to reboot after the shutdown or **-m** to bring the system down into maintenance mode. The **-k** is a “pretend” shutdown. It will appear to all users that the machine is about to shutdown, but no shutdown will actually occur.

To shut down the system to single-user mode:

```
# cd /
# shutdown -m
```

If you need a customized shutdown sequence, you can create a file called **/etc/rc.shutdown**. If this file exists, it is called by the shutdown command and is executed

first. For example, this is useful if you need to close a database prior to a shutdown. If **rc.shutdown** fails (non-zero return code value) the shutdown is terminated.

Flags

- d** Brings the system down from a distributed mode to a multiuser mode
- F** Does a fast shutdown, bypassing the messages to other users and bringing the system down as quickly as possible.
- h** Halts the operating system completely; same as the **-v** flag.
- i** Specifies interactive mode. Displays interactive messages to guide the user through the shutdown.
- k** Avoids shutting down the system.
- m** Brings the system down to maintenance (single user) mode.
- r** Restarts the system after being shutdown with the reboot command.
- t** Restarts the system on the date specified by mmddHHMM [yy] where:
 - mm Specifies the month.
 - dd Specifies the day.
 - HH Specifies the hour.
 - MM Specifies the minute.
- l** In AIX 5.1, creates a new file (/etc/shutdown.log) and appends log output to it. This may be helpful in resolving problems with the shutdown procedure. While the output is generally not extensive, if the root file system is full, the log output will not be captured.

The **-t** option is only supported on systems that have a power supply which automatically turns power off at shutdown and an alarm to allow reboot at a later time. Systems without this capability may hang or may reboot immediately after shutdown.

Manage the System Environment

smit system

System Environments

Move cursor to desired item and press Enter.

Stop the system

Assign the Console

Change / Show Date and Time

Manage Language Environment

Change / Show Characteristics of the Operating System

Change / Show Number of Licensed Users

Manage AIX Floating User Licenses for this Server

Broadcast Message to all Users

Manage System Logs

Change/Show Characteristics of System Dump

Internet and Documentation Services

Change System User Interface

Figure 3-12. Manage the System Environment

AU147.0

Notes:

The System Environment selection in SMIT controls many different aspects of the system.

Stop the System - runs the shutdown command

Assign the Console - allows assignment or reassignment of the system console. A reboot is required for it to take effect.

Change/Show Date and Time - runs the date command to set the date and time. Time zones are also controlled here. Time in AIX is kept in CUT (GMT) time and is converted and displayed using the local time zone.

Manage Language Environments - sets up the language information on your system

Change/Show Characteristics of the Operating System - allows dynamic setting of kernel parameters

Change/Show Number of Licensed Users - shows status of fixed and floating licenses

Manage AIX Floating User Licenses for this Server - sets up floating licenses

Broadcast Message to all Users - issues the wall command

Manage System Logs - displays and cleans up various system logs

Change/Show Characteristics of System Dump - manages what happens when your system panics, crashes and dumps system data

Internet and Documentation Services - controls setting up the web-based documentation

Change System User Interface - determines whether CDE or command-line login is used

Manage System Language Environment

- # smit mlang

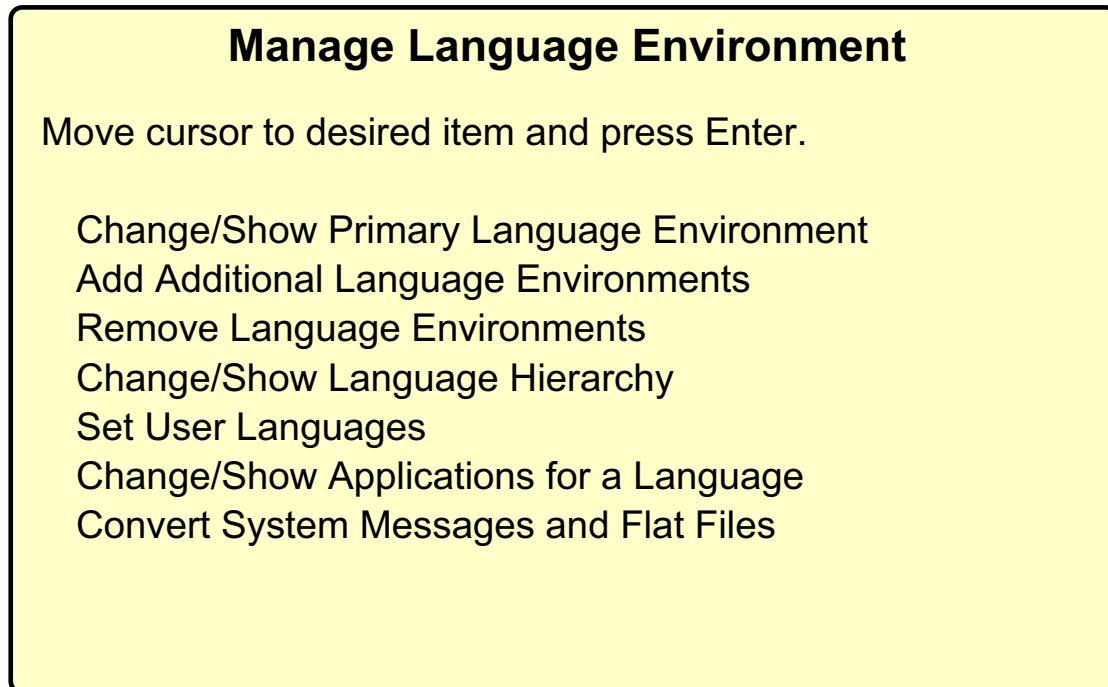


Figure 3-13. Manage System Language Environment

AU147.0

Notes:

The LANG variable specifies the installation default locale. The LANG value is set in the **/etc/environment** file at installation time by the system, based on the information given by the user.

The choice of the language environment affects the means of handling collation, character classification, case conversion, numeric and monetary formatting, date and time formatting, and so forth.

Many language-territory combinations are supported by more than one code set. Be careful when changing the LANG environments to assure the locale chosen matches the user's needs, the keyboard mapping, and font selection.

To change the system National Language (used for accessing online documentation, online help in SMIT and all error messages) use the **chlang** command. For example, **chlang En_GB** for PC850 code pages or **en_GB** for ISO 8859.1 code pages or **chlang C** for POSIX messages. This updates the default setting of the LANG environment variable in **/etc/environment**. You must log off and log in again to the system, for the change to the language environment to become effective.

Industry-standard code sets are provided by means of the ISO8859 family of code sets, which provide a range of single-byte code set support. The Personal Computer (PC) based code sets IBM-850 and IBM-932 are also supported. IBM-850 is a single-byte code set while IBM-932 is a multibyte code set used to support the Japanese locale.

The installation default locale refers to the locale selected at installation. For example, when prompted, a user can specify the English language as spoken in Great Britain during the installation. The code set automatically defaults to the ISO8859-1 code set.

To convert ASCII text files or message catalogs from one code page to another, the **iconv** command or SMIT can be used.

Exercise: System startup and shutdown

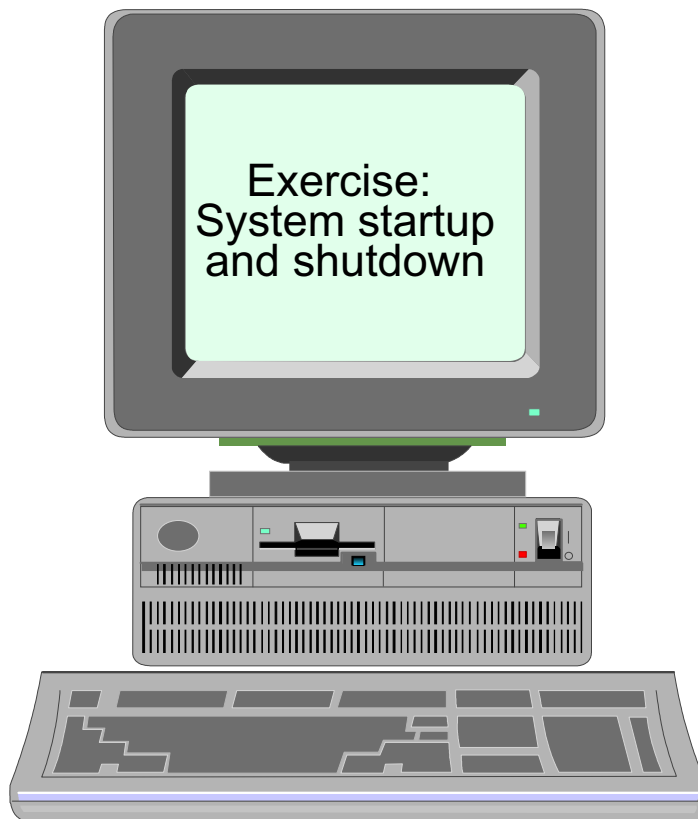


Figure 3-14. Exercise: System startup and shutdown

AU147.0

Notes:

This lab will allow you to become familiar with the startup and shutdown sequences for AIX. The exercise can be found in your Exercise Guide.

Checkpoint

1. What is the first process that is created on the system and which file does it reference to initiate all the other processes that have to be started?

2. Which AIX feature can be used to stop and start groups of daemons or programs?

3. True or false? You can only execute the **shutdown** command from the console.

Figure 3-15. Checkpoint

AU147.0

Notes:

Unit Summary

- When the system boots up it first runs through a number of hardware checks before starting the processes defined in the `/etc/inittab` file
- The LED codes produced during the boot process can be used to identify problems. Alternatively, the boot log file can be accessed to obtain the system messages produced during the boot phase
- Once the system is up, it can be shutdown by an authorized user from any terminal
- SMIT can be used to change common system settings such as the language used, and the date and time used by the system

Notes:

Unit 4. Devices

What This Unit Is About

This unit introduces the concepts of devices, their different states and their location codes.

What You Should Be Able to Do

After completing this unit, you should be able to:

- Describe the difference between logical and physical devices
- Describe the purpose of the ODM predefined and customized databases
- Describe different states of a device
- Describe the format of device location codes
- Use SMIT to Add/Show/Change/Delete devices

How You Will Check Your Progress

Accountability:

- Checkpoint questions
- Exercise

References

Online *System Management Guide: Operating System and Devices*

Objectives

After completing this unit, student should be able to:

- Describe the difference between logical and physical devices
- Describe the purpose of the predefined and customized databases
- Describe different states of a device
- Describe the format of device location codes
- Use SMIT to Add/Show/Change/Delete devices

Figure 4-1. Objectives

AU147.0

Notes:

Device Terminology

- Physical Devices
- Ports
- Device Drivers
- Logical Devices
- /dev Directory

Figure 4-2. Device Terminology

AU147.0

Notes:

In order to attach peripherals such as terminals and printers to an AIX system, we must tell AIX the characteristics of these devices so that the operating system can send the correct signals to the adapter to which the device is connected. A number of pieces of hardware and software must interact correctly for the device to function correctly.

- **Physical Devices** - Actual hardware that is connected in some way to the system.
- **Ports** - The physical connectors/adapters in the system where physical devices are attached. Most ports are programmable by the system software to allow attachment of many different types of devices.
- **Device Drivers** - Software in the kernel that controls the activity on a port and the format of the data that is sent to the device.
- **Logical Devices** - Software interfaces (special files) that present a means of accessing a physical device to the users and application programs. Data appended to logical devices will be sent to the appropriate device driver. Data read from logical devices will be read from the appropriate device driver.

- **/dev** - The directory which contains all of the logical devices that can be directly accessed by the user. (Some of the logical devices defined are only referenced in the ODM customized database and cannot be accessed by users.)

Listing of /dev Directory

```

●# ls -l /dev
brw-rw---- 1 root system 13,0 Jan 24 02:25 fd0
brw-rw---- 1 root system 13,64 Jan 24 02:25 fd1
crw-rw---- 1 root system 13,0 Jan 24 02:25 rfd0
crw-rw---- 1 root system 13,64 Jan 24 02:25 rfd1
:
:
crwxr-xr-x 1 root system 20,0 Jan 24 02:25 rmt0
crwxr-xr-x 1 root system 20,1 Jan 24 02:25 rmt0.1
crwxr-xr-x 1 root system 20,2 Jan 24 02:25 rmt0.2
crwxr-xr-x 1 root system 20,3 Jan 24 02:25 rmt0.3
crwxr-xr-x 1 root system 20,4 Jan 24 02:25 rmt0.4
crwxr-xr-x 1 root system 20,5 Jan 24 02:25 rmt0.5
crwxr-xr-x 1 root system 20,6 Jan 24 02:25 rmt0.6
crwxr-xr-x 1 root system 20,7 Jan 24 02:25 rmt0.7

```

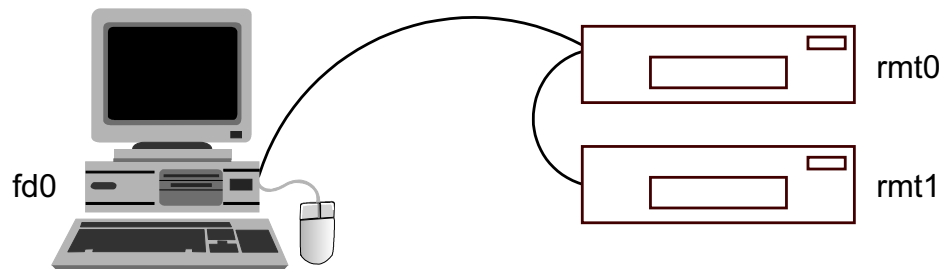


Figure 4-3. Listing of /dev Directory

AU147.0

Notes:

There are a large number of devices that can be configured in the system. Devices can be one of two types:

- Block device is a structured random access device. Buffering is used to provide a block-at-a-time method of access. Usually only disk file systems.
- Character (raw) device is a sequential, stream-oriented device which provides no buffering.

Most block devices also have an equivalent character device. For example, **/dev/hd1** provides buffered access to a logical volume whereas **/dev/rhd1** provides raw access to the same logical volume.

The **ls -l** command allows you to see the type of a file. A special file (in the **/dev** directory) will be indicated by a **b** in the first column for a block device or a **c** for a character device.

Normally the fifth field contains a numeric value indicating the number of bytes in the file. For devices, it shows the major and minor device numbers. The device **rmt0** shown in the listing has a major device number of 20 and a minor device number of 0. This indicates

that the code to handle major device 20 must already be in the kernel, and it must handle device number 0 correctly. In the case of two tape drives, again the major device number of 20 is already in the kernel, and it must handle device number 1 correctly. More precisely, the major number refers to the software section of code in the kernel which handles that type of device, and the minor number to the particular device of that type.

Examples of block devices:

cd0	CD-ROM
fd0, fd0l, fd0h	Diskette
hd1, lv00	Logical Volume
hdisk0	Physical Volume

Examples of character (raw) devices:

console, lft, tty0	Terminal
lp0	Printer
rmt0	Tape Drive
tok0, ent0	Adapter
kmem, mem, null	Memory
rfd0, rfd0l, rfd0h	Diskette
rhd1, rlv00	Logical Volume
rhdisk0	Physical Volume

Device Configuration Database

Predefined Configuration Database			
Name	Type	Subclass	Description
memory	totmem	sys	Memory
tape	4mm4gb	scsi	4.0 GB 4mm Tape Drive
disk	osdisk	scsi	Other SCSI Disk Drive
adapter	22100020	pci	IBM PCI Ethernet Adapter (22100020)
adapter	14101800	pci	IBM PCI Tokenring Adapter (14101800)
adapter	ppa	isa_sio	Standard I/O Parallel Port Adapter
adapter	isa_keyboard	isa_sio	Keyboard Adapter

Customized Configuration Database			
Name	Status	Location	Description
sa0	Available	01-C0	Standard I/O Serial Port 1
sioka0	Available	01-F0	Keyboard Adapter
rmt0	Available	04-C0-00-3,0	4.0 GB 4mm Tape Drive
hdisk0	Available	04-C0-00-4,0	16 Bit SCSI Disk Drive
hdisk1	Available	04-C0-00-5,0	16 Bit SCSI Disk Drive
mem0	Available	00-00	Memory
tok0	Available	04-03	IBM PCI Tokenring Adapter (14101800)
lft	lft	node	Low Function Terminal Subsystem
diskette	fd	siofd	Diskette Drive
printer	ibm4019	parallel	IBM 4019 LaserPrinter

Figure 4-4. Device Configuration Database

AU147.0

Notes:

The predefined and customized databases store information about all of the logical devices in the system and their attributes managed by the ODM (Object Data Manager).

The predefined database contains configuration data for all possible devices supported by the system. The SMIT menus have options to install non-supported drivers. The contents of the predefined database is largely defined at installation time, ensuring that you always have support for devices in your system.

The customized database contains configuration data for all currently defined and configured (available) devices.

The Configuration Manager is a program that automatically configures devices on your system during system boot and run time. The Configuration Manager uses the information from the predefined and customized databases during this process, and updates the customized database afterwards.

List All Supported Devices

PdDv (Predefined Devices)

Isdev -P -H

class	type	subclass	description
memory	totmem	sys	Memory
tape	4mm4gb	scsi	4.0 GB 4mm Tape Drive
disk	osdisk	scsi	Other SCSI Disk Drive
adapter	22100020	pci	IBM PCI Ethernet Adapter (22100020)
adapter	14101800	pci	IBM PCI Tokenring Adapter (14101800)
adapter	ppa	isa_sio	Standard I/O Parallel Port Adapter
adapter	isa_keyboard	isa_sio	Keyboard Adapter
.	.	.	.
.	.	.	.

Isdev -Pc tape

tape	1200mb-c	scsi	1.2 GB 1/4-Inch Tape Drive
tape	150mb	scsi	150 MB 1/4-Inch Tape Drive
tape	3490e	scsi	3490E Autoloading Tape Drive
tape	4mm2gb	scsi	2.0 GB 4mm Tape Drive
.	.	.	.
.	.	.	.

Figure 4-5. List All Supported Devices

AU147.0

Notes:

Default characteristics for known device types are stored in the ODM predefined database.

Devices are classified by Class, Type and Subclass where "Class" indicates what the device does, "Type" indicates what model it is and "Subclass" indicates how it can be attached to the system.

There are also definitions for some unknown devices which can be attached to the system (for example, non-IBM serial or parallel printers or SCSI disk drives). These devices are either intelligent and need little configuration, or the device attachment method is standard (for example, parallel or RS232) and no "features" of the device are assumed when it is added.

To find out what devices are listed in the predefined database, use the SMIT option **List All Supported Devices** which runs the command **Isdev -P**

To find out the attributes of a predefined device, use the SMIT option **Show Characteristics of a Supported Device** which runs the command **Isattr -D**

The **-P** option pulls information from the predefined database in the ODM.

The **-H** option shows the headers for the output.

The **-c** option specifies the class of device.

SMIT is the best way to obtain a listing of currently supported devices for the system:

smit devices -> List Devices -> List All Supported Devices

The devices listed may not physically exist on the system, but device support for them has been installed.

List All Defined Devices

CuDv (Customized Devices)

lsdev -C -H

name	status	location	description
sys0	Available	00-00	System Object
bus0	Available	00-00	PCI Bus
bus1	Available	04-A0	ISA Bus
sa0	Available	01-C0	Standard I/O Serial Port 1
scsi0	Available	04-C0	Wide SCSI I/O Controller
cd0	Available	04-C0-00-2,0	SCSI Multimedia CD-ROM Drive
rmt0	Defined	04-C0-00-3,0	4.0 GB 4mm Tape Drive
hdisk0	Available	04-C0-00-4,0	16 Bit SCSI Disk Drive
hdisk1	Available	04-C0-00-5,0	16 Bit SCSI Disk Drive
mem0	Available	00-00	Memory
ppa0	Available	01-B0	Standard I/O Parallel Port Adapter
tok0	Available	04-03	IBM PCI Tokenring Adapter (14101800)

lsattr -EH -l sys0

attribute	value	description	user_settable
keylock	normal	State of system keylock at boot time	False
realmem	131072	Amount of usable physical memory Kbytes	False
iostat	true	Continuously maintain DISK I/O history	True

lsattr -E -l sys0 -a realmem

realmem	131072	Amount of usable physical memory in Kbytes	False
---------	--------	--	-------

Figure 4-6. List All Defined Devices

AU147.0

Notes:

The devices that have been customized in the system are described in the ODM customized database. Each device has a logical device name, a status, a location and various attributes.

The **lsdev -CH** command provides information on the resource name, its status (or state), the address or location, and a brief description of all devices in the customized database.

This list contains those devices that are found on the system. The status column will contain:

Available: The device is ready and can be used

Defined: The device is unavailable

Devices may appear in a defined state after a restart. If this is the case, it may be due to the fact that the device is powered off or the fact that the device no longer exists on the system.

Devices with a location code are physical devices. Devices without a location code are logical devices. Location codes depend on the type of device and the adapter to which the device is connected.

The **lsattr -E -I [resource name]** command provides detailed information on the effective attributes currently configured for specified devices. In the example, it provides configuration information on the system itself.

The **-C** option for **lsdev** pulls the customized information from the ODM.

The **-E** option for **lsattr** shows the effective attributes.

The **-I** option for both commands is the logical device name.

The **-c** option for both commands is the class of device.

The **-a attribute** option for the **lsattr** command displays information for a specific attribute.

Another command that can be used to list information about devices found in the ODM customized database is **lscfg -v**. The listing is sorted by parent, child and device location. Specific hardware information about devices will be listed such as EC level, FRU number, part number, etc. The output also displays the model architecture and bus type.

Device States

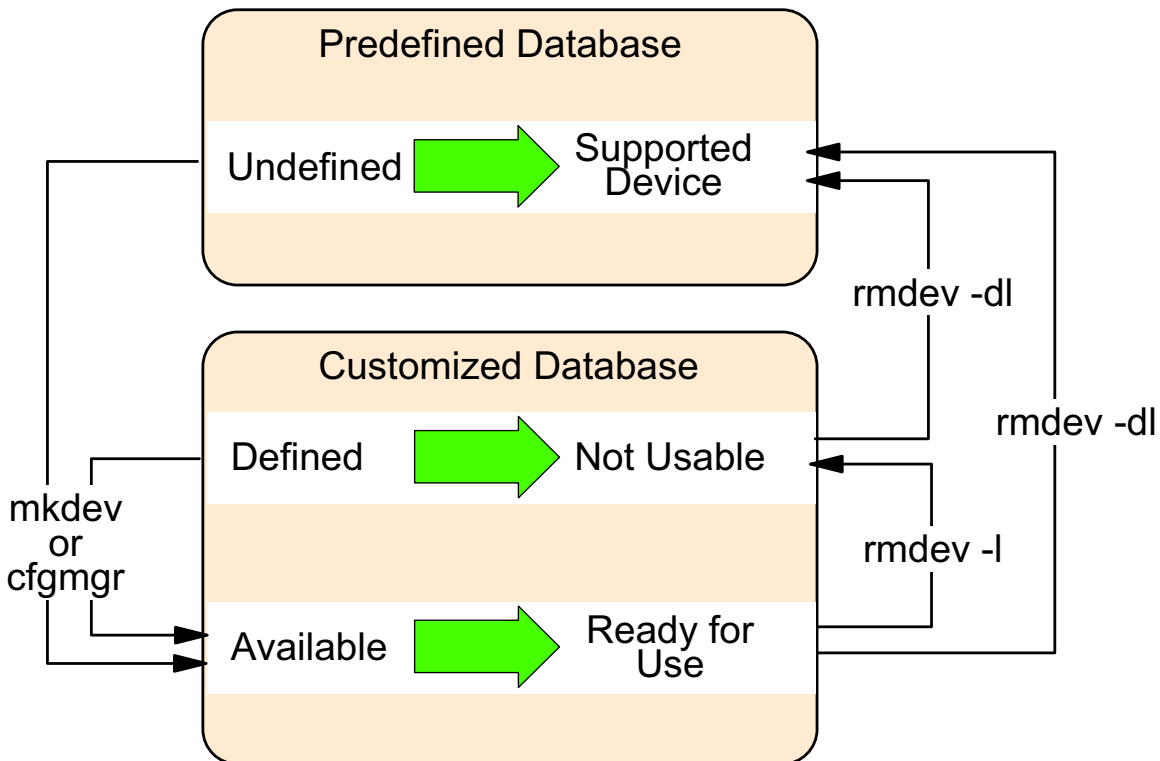


Figure 4-7. Device States

AU147.0

Notes:

The most common device states are:

Undefined - The device is a supported device but is not configured. It does not reside in the customized database.

Defined - The device has been added to the customized database. It has been allocated a logical device name, a location code and attributes have been assigned to it. But, it is still unavailable for use.

Available - The device resides in the customized database. The device is fully configured and is ready for use.

When a device is first identified, it is configured and put into the **Available** state.

If a device that has been configured in the past is powered off and the machine is rebooted, the device will appear in the **Defined** state. This indicates that the system knows it is supposed to be there, but because it was not powered on, it cannot be used.

You can control the device states by using **smit** or the commands **mkdev** and **rmdev**.

To put a defined tape device into an available state:

In the smit devices area, use Configure a Defined Tape Device -or-

mkdev -l rmt0

To move an available tape device to defined:

In the smit devices area, use Remove a Tape Device and set Delete from Database to no
-or-

rmdev -l rmt0

To permanently remove an available or define tape device:

In the smit devices area, use Remove a Tape Device and set Delete from Database to yes
-or-

rmdev -dl rmt0

Remember, most **Defined** devices are the result of not powering on the device before booting. Or, it could be the device was physically removed, but you never ran **rmdev -dl xxxx** to remove the device from the ODM.

Self-Configuring Devices

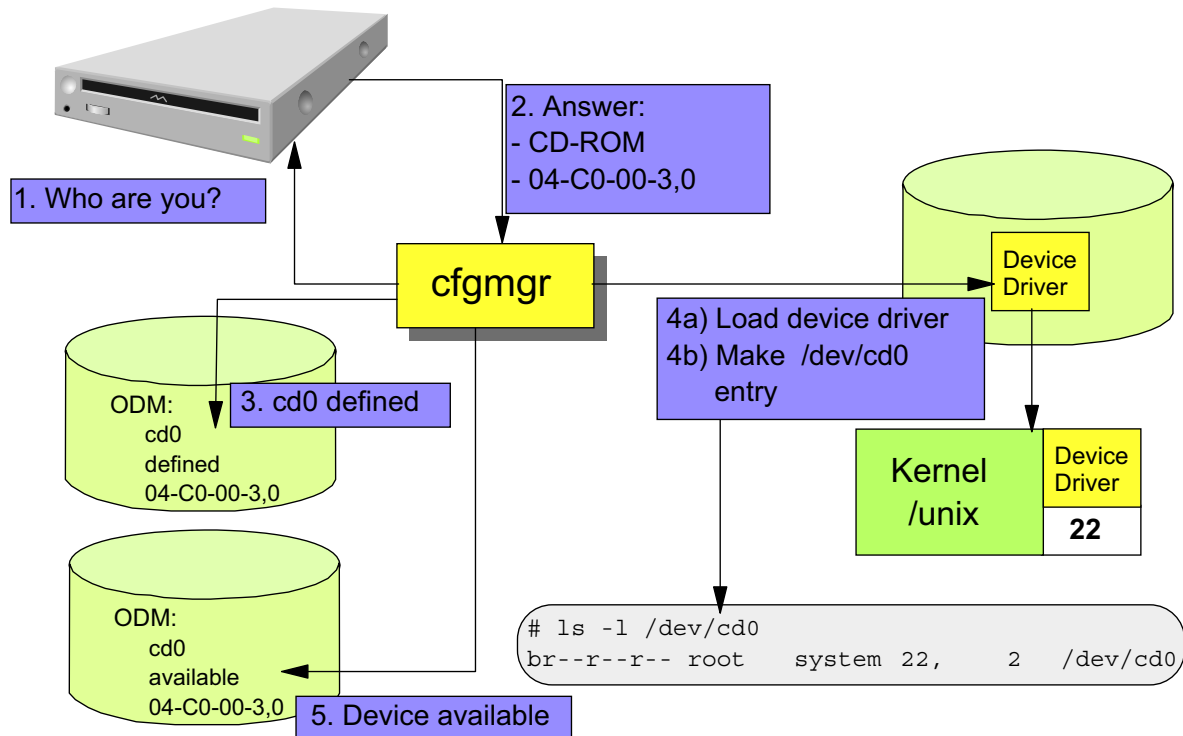


Figure 4-8. Self-Configuring Devices

AU147.0

Notes:

cfmgr is a program that runs during boot that configures devices. It can also be run safely from the command line on a system that is up and running. **cfmgr** will identify all self-configuring devices as long as they are powered on and matches them to the information in the predefined database. It then uses the predefined information to complete a customized database entry for the device.

All devices are self-configuring except for parallel and serial devices. So except for things like printers and ASCII terminals, configuring a device requires only attaching it and power it on before booting the machine. Since **cfmgr** runs during the boot process, no more intervention is required by the administrator.

You will see that for SCSI devices, you will need to set a unique SCSI ID on the device before attaching it. Once that is done, configuration of the device is handled by AIX.

SMIT Devices Menu

smit devices

Devices
Move cursor to desired item and press Enter.

- Install/Configure Devices Added After IPL
- Printer/Plotter
- TTY
- Asynchronous Adapters
- PTY
- Console
- Fixed Disk
- CD ROM Drive
- Read/Write Optical Drive
- Diskette Drive
- Tape Drive
- Communications
- Graphic Displays
- Graphic Input Devices
- Low Function Terminal (LFT)
- SCSI Initiator Device
- Xstation Configuration
- SCSI Adapters
- Asynchronous I/O
- Multimedia
- List Devices
- Install Additional Device Software
- ISA Adapters
- PCMCIA Cards

Figure 4-9. SMIT Devices Menu

AU147.0

Notes:

The SMIT Devices menu (fastpath: **# smit devices**) is used to manage the configuration information about the devices in the system. This information controls the way the kernel and applications behave towards the physical devices attached. The list of devices will vary depending on what you have configured or installed on your system.

Some of the options are submenus which provide the functions to add, change and delete the configuration information, report any errors and trace activity, for specific device types.

- **Install/Configure Devices Added After IPL**

Runs cfgmgr

- **Printer/Plotter**

This submenu allows you to configure printer devices and also queues for local printers and remote printers.

- **TTY**

Any non-printer device attached to a serial port. (For example: terminal, modem, direct connection.)

- **PTY**

A pseudo-terminal device. It provides the appearance of a real ASCII terminal to the application, but does not have any physical port attachment. Used for applications such as AIXWindows and TCP/IP communications.

- **Communication**

Adapters for various types of communications. (For example: Token Ring, Ethernet, MultiProtocol, X.25, 3270, Fiber Optic.)

- **Display Power Management**

For "power management", displays power off (to conserve power) or dim to preserve the display tube. This option is available when accessing Low Function Terminal (LFT) and Graphic Displays.

Devices can also be managed using the Web-based System Manager. To do so, use the fastpath **wsm devices**.

Device Addressing

- Location codes are used for device addressing
- The location code for a device is a path from the adapter in the CPU drawer or system unit, through the signal cables and the asynchronous distribution box (if there is one) to the device
- Location codes consist of up to four fields of information depending on the type of device
- Location codes differ based on model type

Figure 4-10. Device Addressing

AU147.0

Notes:

Every logical device is assigned a location code when it is attached to the system. Location codes depend on the type of device and the adapter to which it connects.

The location code is another way of identifying the physical device.

The location codes exist to provide the system with a method of locating the device and establishing relationships between devices and their adapters. If a hardware failure occurs, the location code is often displayed or referred to in the LED.

The format for location codes is:

AB-CD-EF-GH

The length of the location code depends on the type of device. Two pairs indicate an adapter. Four pairs indicates a device attached to an adapter.

Location Code Format for PCI Devices

AB-CD-EF-GH

AB	00	Resources attached to the processor
	01	Resources attached to the ISA bus
	04	Resources attached to the PCI bus
CD	01-99	For pluggable adapters/cards
	A-Z,0	As position 1 and 2 respectively for integrated adapters
EF		The connector ID
GH		Port identifier, address, memory modules, device, FRU for the device

Figure 4-11. Location Code Format for PCI Devices

AU147.0

Notes:

Knowing how to interpret location codes allows you to quickly locate a device based on the software definition. If you have several of the same type of devices, like hard disk for example, it allows you to easily identify the exact disk that is having the problem.

The actual values used in the location codes vary from model to model. For specific values, you need to reference the *Service Guides* for your model. These can be found online at: www.rs6000.ibm.com/resource/hardware_docs.

In this unit, we will provide a general overview of location codes and specific examples for machine types found in the classrooms.

The location code format above is an example of what you might find on a 43P Model 140 RS/6000 system.

The first set of digits, AB, defines the bus type that devices are attached to:

- **00** defines resources attached to the processor bus, such as the system planar, the processor, memory and the primary PCI bus

- **01** defines resources attached to the ISA bus such as the diskette drive, mouse and keyboard adapters
- **04** defines resources attached to the PCI bus such as the ISA bus, the SCSI controller and devices attached to that SCSI controller. Some PCI RS/6000 systems may use other location codes to define the PCI bus-attached devices. See the hardware documentation for specific PCI location code information

The second set of digits, CD, identify a slot or adapter number.

For pluggable cards, CD will show the actual slot number where the card is attached. If it is plugged into slot 3 in the machine, the location code might be 04-**03**. The CD values can range from 01 through 99.

For integrated devices, like the built in keyboard port, the C position will be a letter A-Z and the D position will be a 0. For example 01-**F0** shows the keyboard adapter is on the ISA bus (01) and is an integrated adapter (F0). The letters are assigned in the order they are discovered during configuration. Each integrated device is assured a unique value.

EF is usually 00. We will show an example of a 128-port async adapter shortly that will show a non-00 EF position.

GH is usually 00 for non-SCSI devices. Multiple diskette drives is one exception. And, 128-port async adapter will also give non-00 GH positions.

*Additional information for classroom machines

As mentioned above, each model has unique location code numbers. You may be using a 43P Model 150 in your classroom. If so, your machine will have different numbers than those discussed in the lecture.

For the AB position you will not see 04 for the PCI bus. These machines contain two PCI buses and they must have two location codes that identify the two buses. Devices on the primary bus will have location codes that start with **10**. Devices on the secondary bus will start with **1P**.

The integrated devices are on the primary PCI bus (start with 10) or on the ISA bus (01). Their CD positions are fixed unlike on the Model 140 where the letters are assigned in the order of discovery. So, for example, **01-D1** is always the integrated diskette drive and is attached on the ISA bus. **10-80** is always the integrated SCSI controller (adapter).

Pluggable cards will be attached to one of the two PCI buses. Slots 2 and 3 are on the primary bus and will start with **10**. Cards in Slots 1, 4 or 5 are on the secondary bus and start with **1P**. Each slot has an assigned location code number. To see the assigned numbers you need to reference the *Service Guide*. To give one example, a card in slot 1 will have an address of **1P-08**.

When you are looking at the location codes on a Model 150, use this chart taken from the *Service Guides* to interpret their meaning:

1P-08 Slot 1

10-b0 Slot 2

10-90 Slot 3

1P-18 Slot 4

1P-10 Slot 5

Location Code Example: Non-SCSI

128-Port Asynchronous Controller

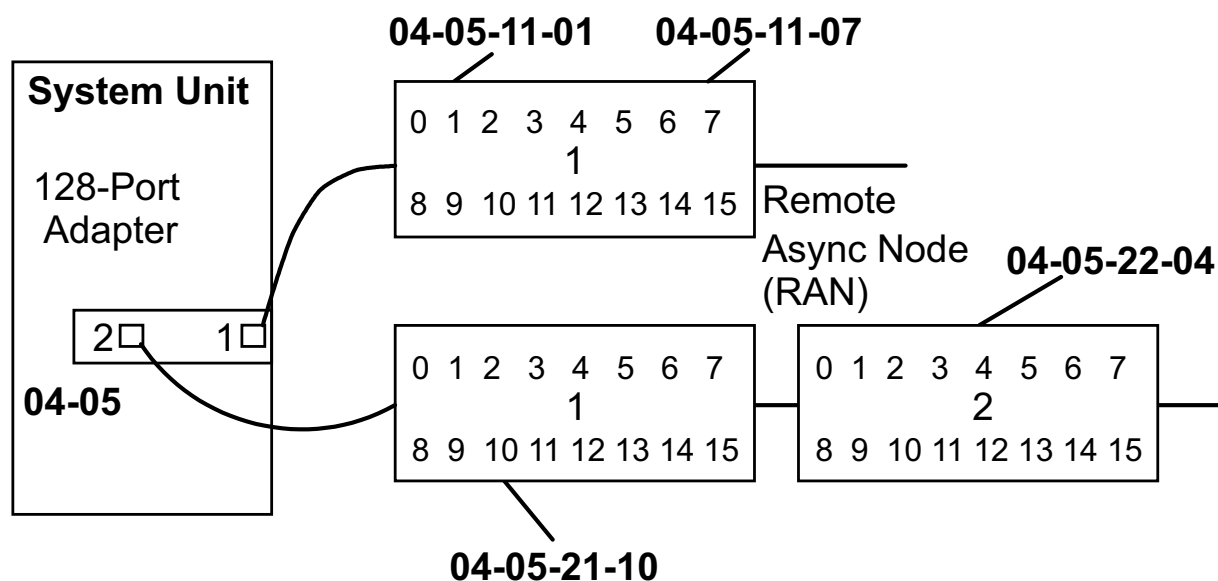


Figure 4-12. Location Code Example: Non-SCSI

AU147.0

Notes:

The above example illustrates non-SCSI device location codes for an RS/6000.

A 128-port asynchronous adapter allows 128 serial devices (like ASCII terminals) to be attached to the adapter. The adapter has two connectors (or ports) on the card. Each connector can support a serial bus.

On each bus, boxes that contain ports are connected to each other. These boxes are called Remote Asynchronous Nodes (RANs). Each of the two connectors can support four RANs. Four RANs on two connectors give a total of eight RANs. Each RAN has 16 ports. That gives a total of 128 ports.

The location code must account for each piece of the puzzle.

AB-CD is the same as previous examples. It provides the adapter card address. In our example, the adapter card is plugged into slot 5 on the PCI bus.

E identifies the connector on the adapter card, 1 or 2.

F identifies the RAN. RANs are numbered in ascending order going away from the adapter, 1-4.

GH is the two-digit port number. For example, port 7 is "07". The range of numbers is 00-15.

Location Code Format for SCSI Devices

AB-CD-EF-G,H

AB-CD	Identifies the bus and the adapter location Same as with non-SCSI devices
EF	For a single SCSI bus - 00 For a dual SCSI bus: Internal bus - 00 External bus - 01
G,H	G - SCSI address (SCSI ID) of the device H = Logical unit number of the device

Figure 4-13. Location Code Format for SCSI Devices

AU147.0

Notes:

The above shows an example of location codes for SCSI devices.

The location code format is slightly different. You'll notice in this format the G and H positions are separated by a comma.

The AB-CD positions contain the same information we have already covered. It indicates where the adapter card (SCSI controller) is attached - the bus and slot number.

The EF position identifies the SCSI bus. If the controller provides only a single SCSI bus, the EF position will be **00**. If the controller provides for dual SCSI buses, each bus must be identified by a unique address. With dual SCSI, the card's internal bus is identified with **00** and the card's external bus is identified with **01**.

The G,H position provides two pieces of information. The G position is the SCSI address or SCSI ID of the device. The SCSI ID is set on the device itself. It is usually accomplished by setting jumpers or switches on the device. Some devices have dials or push buttons that are external that allow an easy method to set the ID. Set the SCSI ID so that it doesn't

conflict with another device on that bus. When **cfgmgr** runs it will recognize the ID that is set on the hardware and set the G position accordingly.

The H is usually a 0. If the SCSI devices has multiple devices within it, then the logical unit number (LUN) is used to uniquely identify each device. Non-zero numbers are used with RAID arrays or some CD jukeboxes.

Location Code Example for SCSI Device

SCSI Devices (Disk, Tape, CD-ROM)

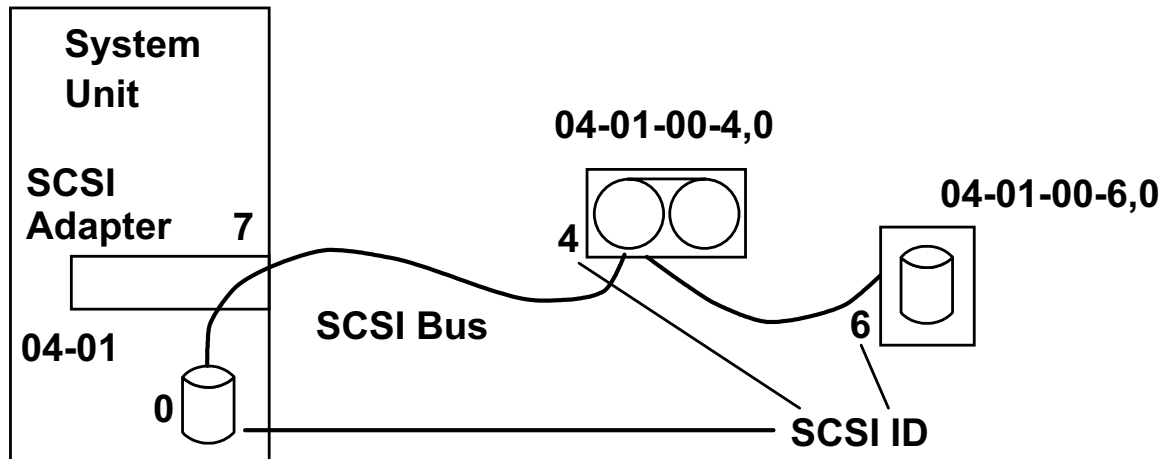


Figure 4-14. Location Code Example for SCSI Device

AU147.0

Notes:

This example shows several SCSI devices attached to a single SCSI bus on a 43P Model 140. This is not a dual SCSI. This is a single bus that has devices that are housed inside and outside the cabinet.

From the device addressing, the adapter is located on slot 1 on the PCI bus. The external disk has a SCSI ID of 6 and the tape device has a SCSI ID of 4.

What would the location code be for the disk with SCSI ID of 0? _____ - _____ - _____ - _____, _____

The SCSI adapter uses a SCSI ID of 7 by default. Normally, you should not set a device to a SCSI ID of 7 for that reason.

Location Code Example: PCI

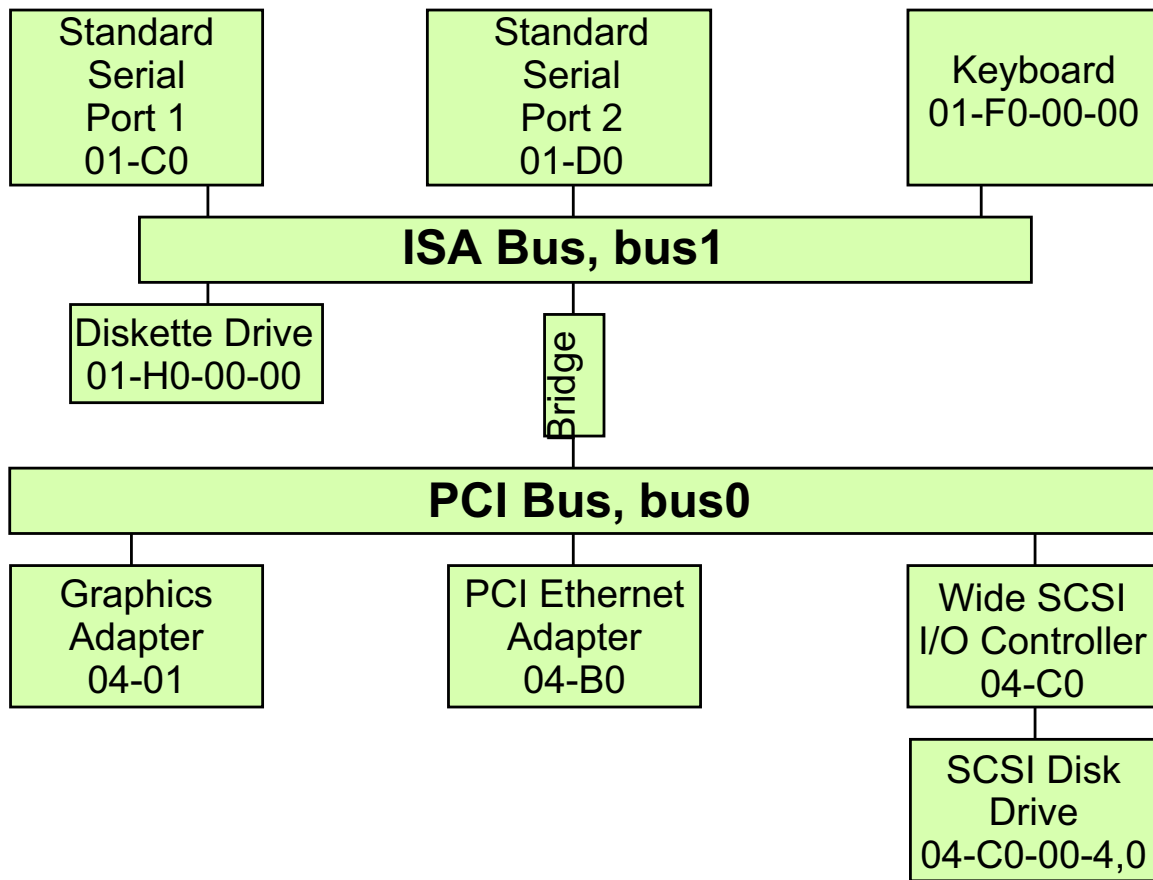


Figure 4-15. Location Code Example: PCI

AU147.0

Notes:

The example shown in the visual is a partial view of a 43p Model 140 system with an attached ISA bus.

All devices attached to the ISA bus are identified with a prefix location code of 01 and the PCI devices in this example are attached with a prefix location code of 04.

Adding an ASCII Terminal

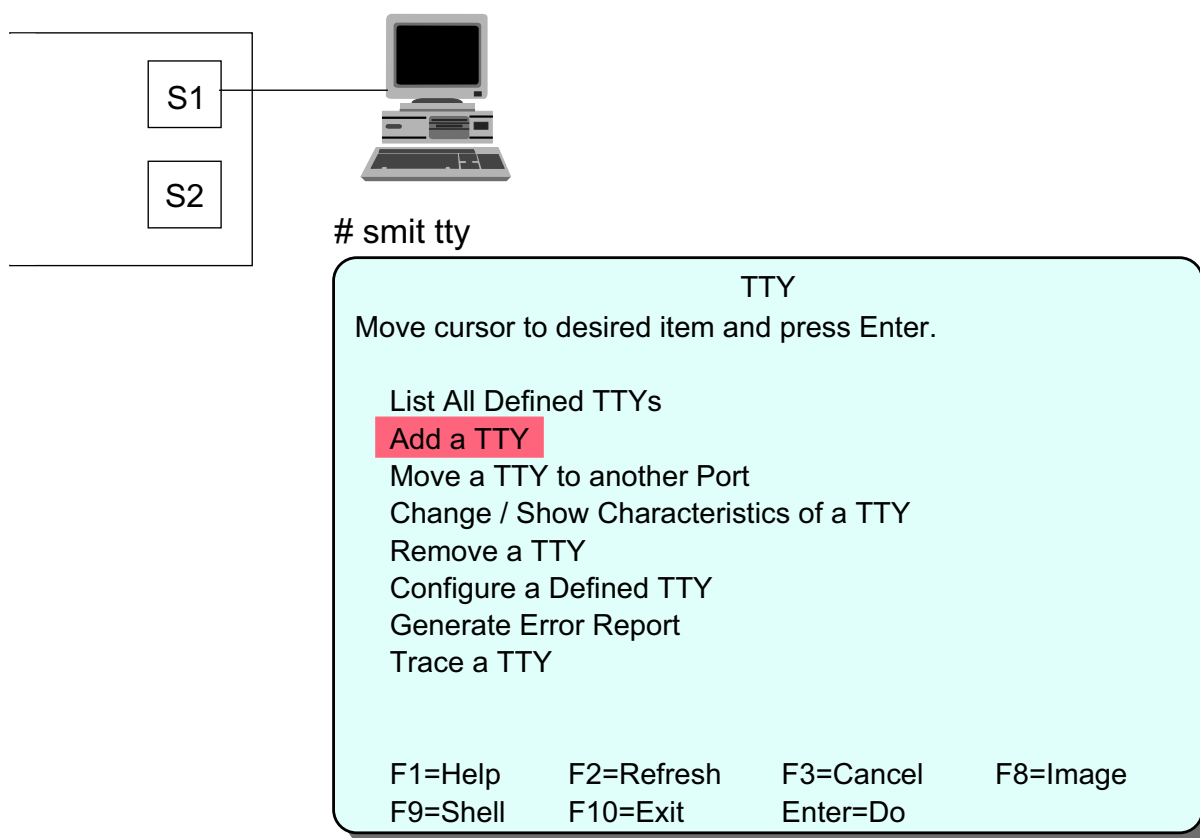


Figure 4-16. Adding an ASCII Terminal

AU147.0

Notes:

Most devices self-configure using **cfgmgr**. One type of device that does not is an ASCII terminal. We will go through the process of adding an ASCII terminal to provide an example of what is required to manually configure a device.

First, physically attach the terminal to the serial port. Be sure to note which serial port it is attached to. We will need that information as we complete this process.

To begin the configuration, use **smit tty**.

This screen is used to manage the configuration of asynchronous devices.

To add the terminal, select **Add a TTY**.

Attachment

TTY Type

Move cursor to desired item and press Enter.

tty	rs232	Asynchronous Terminal
tty	rs422	Asynchronous Terminal

Parent Adapter

Move cursor to desired item and press Enter.

sa0	Available	01-S1 Standard I/O Serial Port 1
sa1	Available	01-S2 Standard I/O Serial Port 2
sa2	Available	04-03-11 16-Port RAN EIA-232 for 128-Port adapter
sa3	Available	04-03-12 16-Port RAN EIA-232 for 128-Port adapter
sa4	Available	04-03-13 16-Port RAN EIA-232 for 128 Port adapter

Figure 4-17. Attachment

AU147.0

Notes:

Once you select **Add a TTY**, you will then be asked the **TTY Type** and which **Parent Adapter** the terminal is attached to.

The choices for TTY type are **rs232** and **rs422**. Rs232 is the most common TTY type.

To select the correct parent adapter, you need to know where the device is physically attached. This is where the serial port is important.

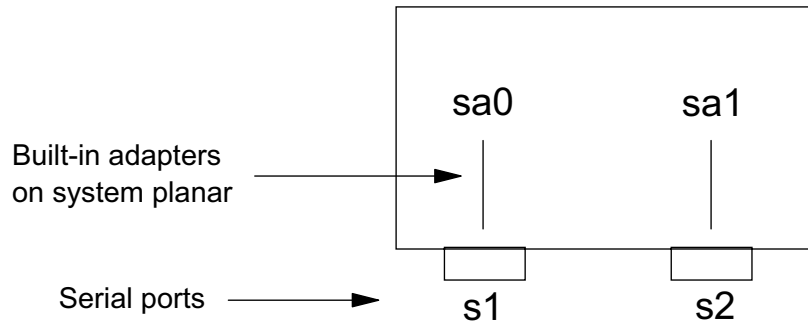
In our example from the previous page, the terminal was attached to serial port 1. Therefore, we select **sa0 - Standard I/O Serial Port 1**.

The location code is also displayed. **01-S1** is, in fact, the location code of serial port 1.

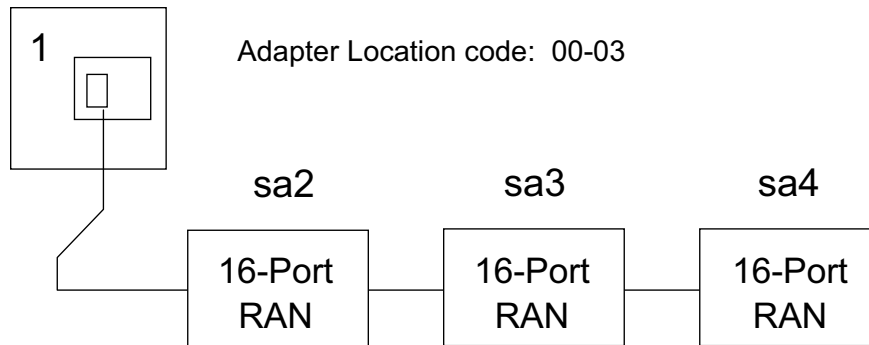
sa2, **sa3**, and **sa4** are remote asynchronous nodes used in conjunction with the 128-port async adapter.

Be careful with the numbering scheme. **sa0** is serial port 1. **sa1** is serial port 2. The **sa** stands for serial adapter. The adapters are devices and device names are numbered starting at 0.

For the built-in serial connection the nomenclature looks like this:



For the 128-port adapter the nomenclature looks like this:



Add a TTY

```
# smit mktty
```

Add a TTY

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

[TOP]	[Entry Fields]
TTY type	tty
TTY interface	rs232
Description	Asynchronous Terminal
Parent Adapter	sa0
* PORT number	[] + ←
Enable LOGON	disable + ←
BAUD rate	[9600] +
PARITY	[none] +
BITS per character	[8] +
Number of STOP BITS	[1] +
TIME before advancing to next port setting	[0] +#
TERMINAL type	[dumb] ←
FLOW CONTROL to be used	[xon] +
[MORE ...29]	

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

Figure 4-18. Add a TTY

AU147.0

Notes:

There is only one mandatory field on this screen and that is the PORT number. The F4 key will provide a list of possible port numbers. For the first built-in serial port it is **s1**, for the second it is **s2**. On a 16-port RAN, the choices are **0-15**. Select the one to which the terminal is connected. The combination of the appropriate RAN selected on the Parent Adapter selector screen and the port number shown here provides the system with the correct location code.

You must supply the port number to uniquely locate the device. The value required depends upon the adapter specified. For example:

Built-in serial port S1	s1
Built-in serial port S2	s2
8-Port Adapter	0-7
16-Port Adapter	0-15
Each 16-PORT RAN	0-15

The Enable LOGIN attribute will be set to disable by default. If you are adding a terminal that should have a login prompt, you should change this to enable.

The asynchronous line characteristics must be specified: baud rate, parity, bits per character, stop bits. In a national language environment you must use eight bits with no parity (the default). Set the speed appropriately for the terminal device or modem you are using, up to 38400.

The TERMINAL type attribute is used to assign the TERM environment variable when a user logs in on the device. You must set this to the name of a supported terminal type. The list of supported terminals can be found in directories located in /usr/share/lib/terminfo.

Documenting Hardware Configuration

- **lsdev -CH**

Provides name, status, location, and description of devices

- **lscfg -v**

Provides details of all devices including manufacturer, type and model number and part numbers

- **lsattr -El sys0**

Provides attributes for the name device (for example, sys0)
Run command for all devices

Figure 4-19. Documenting Hardware Configuration

AU147.0

Notes:

Documentation is an important part of the system administrators job. Be sure to document all device configurations for your machines.

lsdev -CH provides a listing all from the customized database. The **H** option supplies headers to the output for easier interpretation.

lscfg -v provides a verbose detailed output of all of the devices on the machines. It includes vital product data (VPD) which has information like manufacturer, type and model, and part numbers. Not all devices have VPD.

lsattr -El sys0 provides attributes for the device. In this example, it is providing the attributes for the kernel. **sys0** is the device name of the kernel. To fully document your system you need to run this command against all devices configured on your machine. For example, to get the attributes of a hard drive, you need to run **lsattr -El hdisk0**. It would probably be helpful to create a shell script to complete this process for you.

Exercise: Devices



Figure 4-20. Exercise: Devices

AU147.0

Notes:

This lab will give you an opportunity to examine the device configuration of the classroom system.

The exercise can be found in your Exercise Guide.

Checkpoint (1 of 2)

lsdev -C -H

name	status	location	description
sys0	Available	00-00	System Object
bus0	Available	00-00	PCI Bus
bus1	Available	04-A0	ISA Bus
ppa0	Available	01-G0	Standard I/O Parallel Port Adapter
lp0	Available	01-G0-00-00	IBM 4039 LaserPrinter
sa0	Available	01-J0	Standard I/O Serial Port 1
tty0	Available	01-J0-00-00	Asynchronous Terminal
mem0	Available	00-00	Memory
scsi0	Available	04-C0	Wide SCSI I/O Controller
rmt0	Defined	04-C0-00-3,0	5.0 GB 8mm Tape Drive
hdisk0	Available	04-C0-00-4,0	SCSI Disk Drive
tok0	Available	04-04	IBM PCI Tokenring Adapter

Figure 4-21. Checkpoint (1 of 2)

AU147.0

Notes:

Checkpoint (2 of 2)

1. Is it possible to use SCSI ID 7 for a new tape drive?

Use the output on the previous slide (**lsdev -C -H**) to answer the following four questions.

2. What will happen if we attempt to add another device with the SCSI address set to 4?

3. Can the 8 mm tape drive be currently used? Why?

4. Where is the printer connected?

5. The token-ring adapter is installed in what slot?

Figure 4-22. Checkpoint (2 of 2)

AU147.0

Notes:

Unit Summary

- A physical device is the actual hardware attached to the system. A logical device is the software interface used by programs and users to access a physical device.
- Device information is stored in the ODM in two databases: customized and predefined
- Devices can exist in a number of different states: unavailable, defined, available and stopped
- Location codes are used to describe exactly where a device is connected into the system
- Device attributes can be modified through SMIT
- To create, modify, or remove device definitions, it is sometimes necessary to use commands such as **mkdev**, **chdev** and **rmdev**.

Notes:

Unit 5. System Storage Overview

What This Unit Is About

This unit is an overview of AIX system storage.

What You Should Be Able to Do

After completing this unit, you should be able to:

- Describe the terminology and the concepts associated with:
 - Physical Volumes
 - Volume Groups
 - Logical Volumes
 - Physical Partitions
 - Logical Partitions
- Describe how file systems and logical volumes are related

How You Will Check Your Progress

Accountability:

- Checkpoint questions
- Activity

References

Online *System Management: Operating System and Devices*

Objectives

After completing this unit, students should be able to:

- Describe the terminology and concepts associated with:
 - Physical Volumes
 - Volume Groups
 - Logical Volumes
 - Physical Partitions
 - Logical Partitions
- Describe how file systems and logical volumes are related.

Figure 5-1. Objectives

AU147.0

Notes:

Components of AIX Storage

- Files
- Directories
- File Systems
- Logical Storage
- Physical Storage
- Logical Volume Manager

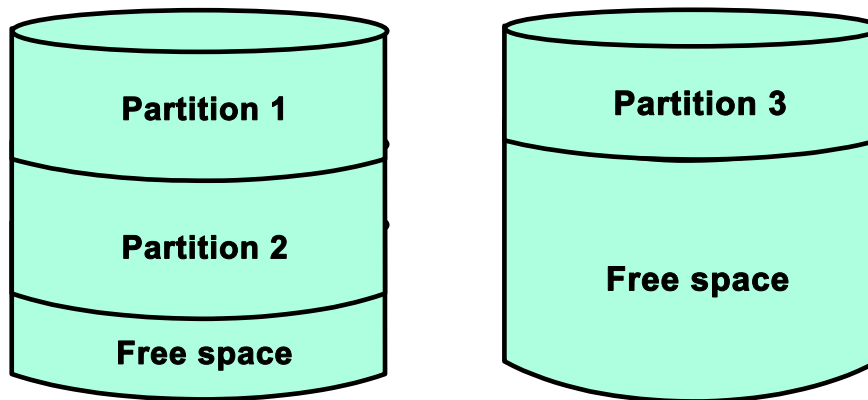
Figure 5-2. Components of AIX Storage

AU147.0

Notes:

These are the basic components or building blocks of AIX storage. As a user you work with files and directories. As a system administrator you will work with the others as well.

Traditional UNIX Disk Storage



PROBLEMS:

- Fixed partitions
- Expanding size of the partition
- Limitation on size of a file system and a file
- Contiguous data requirement
- Time and effort required in planning ahead

Figure 5-3. Traditional UNIX Disk Storage

AU147.0

Notes:

Traditionally, disk partitioning has been implemented via partitions. Customers had to select the correct size for each partition before the system could be installed.

Each file system sits on a partition on the hard disk.

Changing the size of the partition and thus the file system is no easy task. It involves backing up the file system, removing the partition, creating new ones and restoring the file system.

A major limitation to partitions is that each partition has to consist of contiguous disk space. This characteristic limits the partition to reside on a single physical drive. It cannot span multiple hard disks. Since file systems are always contained within a partition, no file system can be defined larger than the largest physical drive. This means that no single file can exist larger than the largest physical drive.

Benefits of the LVM

- Logical volumes solve non-contiguous space problems
- Logical volumes can span disks
- Dynamically increase logical volume size
- Logical volumes can be mirrored
- Hard disks easily added to a system
- Logical volumes can be relocated
- Volume group and logical volume statistics can be collected

These tasks can be performed dynamically!

Figure 5-4. Benefits of the LVM

AU147.0

Notes:

The constraints with traditional UNIX disk storage have been virtually eliminated in AIX with the addition of the Logical Volume Manager.

Note that the tasks listed above can be performed while users are on the system.

Physical Storage

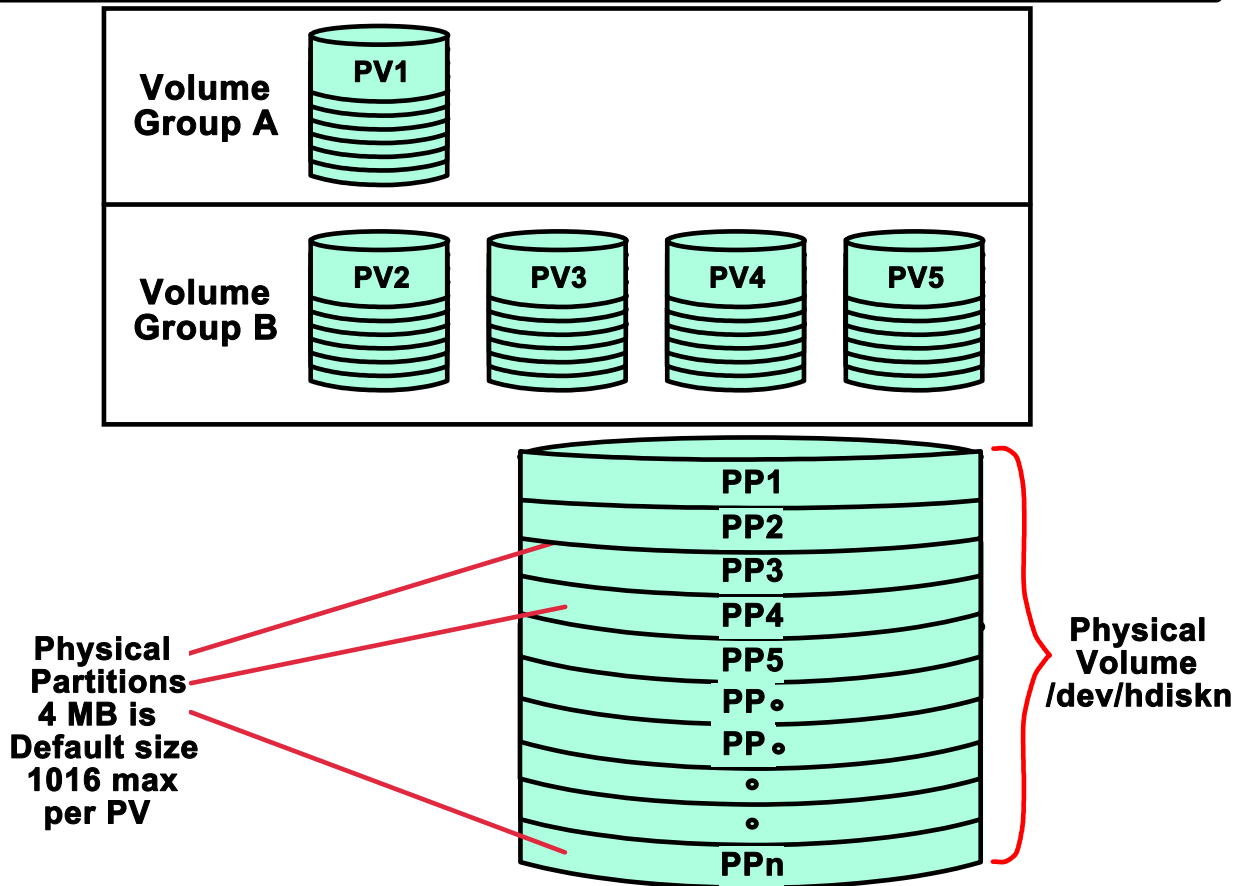


Figure 5-5. Physical Storage

AU147.0

Notes:

Physical Volumes

A physical volume (PV) is the name for an actual disk or hard drive. A PV can be internally attached or externally attached.

When a physical volume is added to a system, a file called `hdiskn` is added to the `/dev` directory. *n* is a number allocated by the operating system. It is usually the next available number. This file may be used to access the device directly but this is not often done.

For a disk to be used by logical volume manager, the disks must be added to a volume group or a new volume group must be set up for it.

A PV can only belong to one VG.

Volume Group

A volume group (VG) is the largest unit of storage allocation. A VG consists of a group of one or more physical volumes (disks) all of which are accessed under one VG name. The combined storage of all the physical volumes make up the total size of the VG. This space can be used by other storage entities like file systems and logical volumes.

Volume groups are portable and can be disconnected from one system and connected to another system. All disks in the VG must move together.

A volume group is broken down and manipulated as a collection of physical partitions (PP) which lie on the physical volumes within the volume group. The size of the PP's within a VG is constant.

Physical Partitions

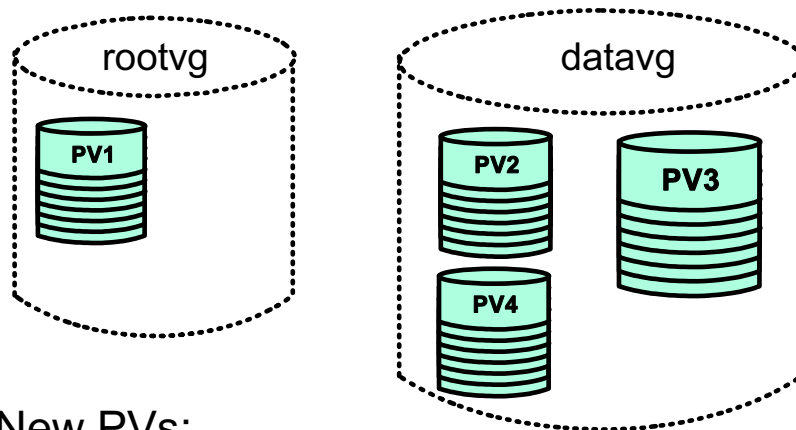
A physical partition (PP) is a division of a physical volume. It is the basic unit of disk space allocation.

Since most disks are larger than 4GB, the PP size will normally need to be changed. PP size can be changed in increments of the power of 2 up to 1024 MB.

All PPs within a volume group are the same size and cannot be changed dynamically. The default size of the PPs is 4 MB. The default maximum number of PPs per PV is 1016. A volume group that contains a physical volume larger than 4.064 gigabytes needs either a physical partition size greater than 4 MB or the physical volume must be allocated more physical partitions. The physical partition size cannot be changed dynamically. However, the number of physical partitions per physical volume can be changed dynamically in multiples of 1016 (that is 1016, 2032, 4064 and so forth up to a maximum of 130,048).

Be aware that if you choose to have more than 1016 PPs per PV, you will decrease the maximum number of physical volumes supported in the volume group. Details on this will be discussed shortly.

Volume Groups



New PVs:

- Add to existing VGs
- Create new VG

Why create new volume groups?

- Separate user data from operating system files
- Disaster recovery
- Data portability
- Data integrity and security

Figure 5-6. Volume Groups

AU147.0

Notes:

When the system is installed, the root volume group (**rootvg**) is created. This is where the AIX operating system files will be contained.

Additional disks can either be added to **rootvg** or a new volume group can be created for them. There can be up to 255 VGs per system.

If you have external disks, it is recommended that they be placed in a separate volume group. By maintaining the user file systems and the operating system files in distinct volume groups, the user files are not jeopardized during operating system updates, reinstallations, and crash recoveries.

Maintenance is easier because you can update or reinstall the operating system without having to restore user data.

For security, you can make the volume group unavailable using **varyoffvg**.

Volume Group Descriptor Area

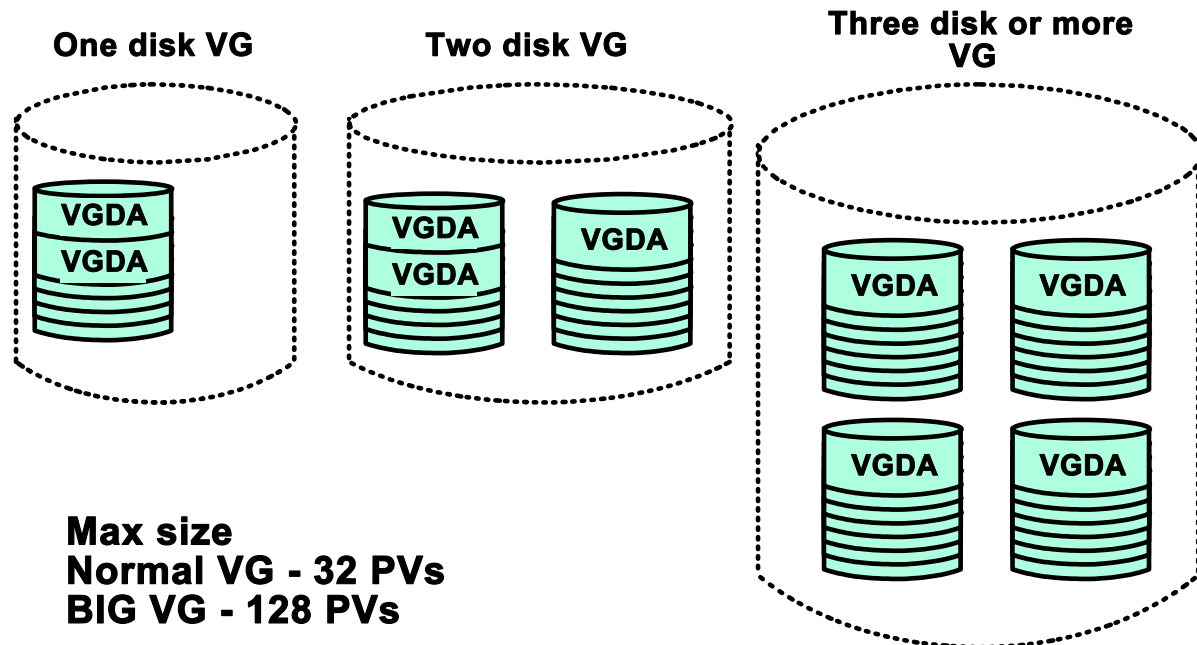


Figure 5-7. Volume Group Descriptor Area

AU147.0

Notes:

VGDA The Volume Group Descriptor Area (VGDA) is an area of disk, at least one per PV, containing information for the entire VG. It contains administrative information about the volume group (for example, a list of all logical volume entries, a list of all the physical volume entries and so forth). There is usually one VGDA per physical volume. The exceptions are when there is a volume group of either one or two disks (illustrated).

Quorum There must be a quorum of VGDA's available to activate the volume group and make it available for use (**varyonvg**). A quorum of VGDA copies is needed to ensure the data integrity of management data that describes the logical and physical volumes in the volume group. A quorum is equal to 51% or more of the VGDA's available.

A system administrator can force a volume group to vary on without a quorum. This is not recommended and should only be done in an emergency.

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)

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

mkvg -t

Figure 5-8. Volume Group Limits

AU147.0

Notes:

With AIX 4.3.2, a second type of Volume Group was added - Big Volume Groups. This increased the number of physical volumes allowed in a volume group.

- **Normal volume groups:** When creating a volume group with SMIT or using the **mkvg** command, normal volume groups are the default.
- **Big volume groups:** This volume group was introduced with AIX 4.3.2. A big volume group must be created using the command line command **mkvg -B**. Creating big volume groups is not supported with SMIT.

The other physical limitation on the volume group is the number of physical partitions on a physical volume within the volume group. In AIX 4.3.1, the ability to increase the number of physical partitions (known as factor) was added. This is done using the command **mkvg -t #** where the "#" is a multiplier for the number of PPs desired. The number is multiplied by 1016 and that becomes the new limit for PPs on PVs with a VG. For example, **mkvg -t 2** would allow 2032 (1016*2) PPs per PV.

Changing the factor will effect the number of PVs in a volume group. The above charts show the relationship between the factor and the number of PVs in both types of VGs. If in our example of **mkvg -t 2**, this would allow us 2032 PPs per PV and the maximum number of PVs is 16 (normal VG) or 64 (big VG).

Logical Storage

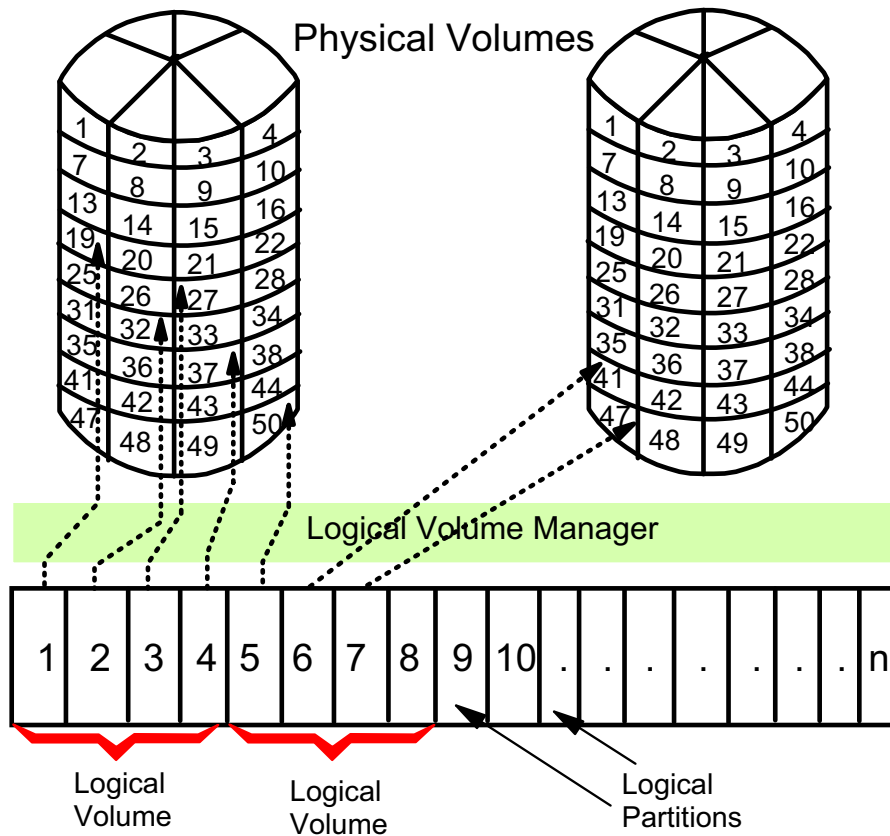


Figure 5-9. Logical Storage

AU147.0

Notes:

A **physical partition** is the smallest unit of allocation of disk. Each logical partition maps to a physical partition which physically stores the data.

Obviously, the logical partitions within a volume group are the same size as the physical partitions within that volume group.

A **logical volume** consists of one or more logical partitions within a volume group.

Logical volumes may span physical volumes if the volume group consists of more than one physical volume. Logical volumes do not need to be contiguous within a physical volume because the logical partitions within the logical volume are maintained to be contiguous. The view the system sees is the logical one. Thus, the physical partitions they point to can reside anywhere on the physical volumes in the volume group.

Logical volumes may be increased in size at any time, assuming that there are sufficient free physical partitions within the volume group. This can be done dynamically through SMIT even when users are doing work in that logical volume. However, logical volumes

cannot easily be decreased and require a file system backup and restore to a recreated smaller logical volume.

Logical volumes consist of a number of logical partitions, so when a logical volume is being created the size requested is increased to the next logical partition boundary. Typically, the logical/physical partition size is 4 MB so a logical volume will be a multiple of 4 MB in size. Logical/physical partition sizes range from 1 MB - 1024 MB.

A volume group is where the physical and logical views of storage meet. It is both a physical view and a logical view.

The following table shows the limitations for logical storage management.

MAXPVS: 32 (128 big volume group)

MAXLVS: 255 (512 big volume group)

Logical Entity	Limitation
Volume group	255 per system
Physical volume	(MAXPVS / volume group factor) per volume group
Physical partition	(1016 x volume group factor) per physical volume up to 1024 MB each in size
Logical volume	MAXLVS per volume group
Logical partition	(MAXPVS * 1016) per logical volume

The **Logical Volume Manager** (LVM) consists of the logical volume device driver (LVDD) and the LVM subroutine interface library. The LVM controls disk resources by mapping data between a more simple and flexible logical view of storage space and the actual physical disks. The LVM does this using a layer of device driver code that runs above traditional disk device drivers.

Uses of Logical Volumes

A logical volume may contain one of the following, and only one at a time:

- Journalled or Enhanced journalled file system (for example: /dev/hd4)
- Paging space (/dev/hd6)
- Journal log (/dev/hd8)
- Boot Logical Volume (/dev/hd5)
- Nothing (raw device)

Figure 5-10. Uses of Logical Volumes

AU147.0

Notes:

When you install the system, you automatically create one volume group (**rootvg**) which consists of a base set of logical volumes required to start the system. **rootvg** contains such things as paging space, the journal log, and boot data, each usually in its own separate logical volume.

You can create additional logical volumes with the **mkiv** command or go through the SMIT menus. This command allows you to specify the name of the logical volume and to define its characteristics.

The theoretical maximum number of user-defined logical volumes per volume group is 255 (512 for big volume groups), but the true limit is determined by the total size of the combined physical volumes assigned to the volume group.

The native file system on AIX is the journalled file system (JFS), or the enhanced journalled file system (JFS2). They use database journaling techniques to maintain consistency. It is through the file system's directory structure that users access files, commands, applications, and so forth.

Paging space is fixed disk storage for information that is resident in virtual memory but is not currently being accessed.

The journal log is the logical volume where changes made to the file system structure are written until such time as the structures are updated on disk. Journalized file systems and enhanced journalized file systems will be discussed in greater detail later in the course.

The boot logical volume is a physically contiguous area on the disk which contains the boot image.

A raw device is simply an empty logical volume. Sometimes an application, for example a database package, may require a raw device.

When you install the operating system, the dump device is automatically configured for you. By default, the primary device is **/dev/hd6**, which is the paging logical volume, and the secondary device is **/dev/sysdumpnull**. For systems migrated from versions of AIX earlier than 4.1, the primary dump device is what it formerly was, **/dev/hd7**.

What Is a File System?

- A file system is:
 - Method of storing data
 - Hierarchy of directories
- Four types supported:
 - Journaled File System (jfs)
 - Enhanced Journaled File System (jfs2)
 - CD-ROM File System (cdrfs)
 - Network File System (nfs)
- Different file systems are connected together via directories to form the view of files users see.

Figure 5-11. What Is a File System?

AU147.0

Notes:

AIX supports four file system types:

jfs	Journaled File System which exists within a Logical Volume on disk
jfs2	Enhanced Journaled File System which exists within a Logical Volume on disk
cdrfs	CD-ROM File System on a Compact Disc
nfs	Network File System accessed across a network

Although these are physically different, they appear the same to users and applications.

A file system is a directory hierarchy for storing files. It has a root directory and subdirectories. In an AIX system, the various file systems are joined together so that they appear as a single file tree with one root. Many file systems of each type can be created.

Because the available storage is divided into multiple file systems, data in one file system could be on a different area of the disk than data of another file system. Because file systems are of a fixed size, "file system full" errors can occur when that file system has become full. Free space in one file system cannot automatically be used by an alternate file system that resides on the same physical volume.

Why Have File Systems ?

- Can strategically place it on disk for improved performance
- Some tasks are performed more efficiently on a file system than on each directory within the file system, for example, back up, move, secure an entire file system
- Can limit disk usage of users by file system (quotas)
- Maintain integrity of the entire file system structure, for example, if one file system is corrupted, the others are not affected
- Special security situations
- Organize data and programs into groups for ease of file management and better performance

Figure 5-12. Why Have File Systems

AU147.0

Notes:

A file system is a structure that allows you to organize your data. It is one level in the hierarchy of your data. By placing data in separate file systems, it allows for ease of control and management of the data.

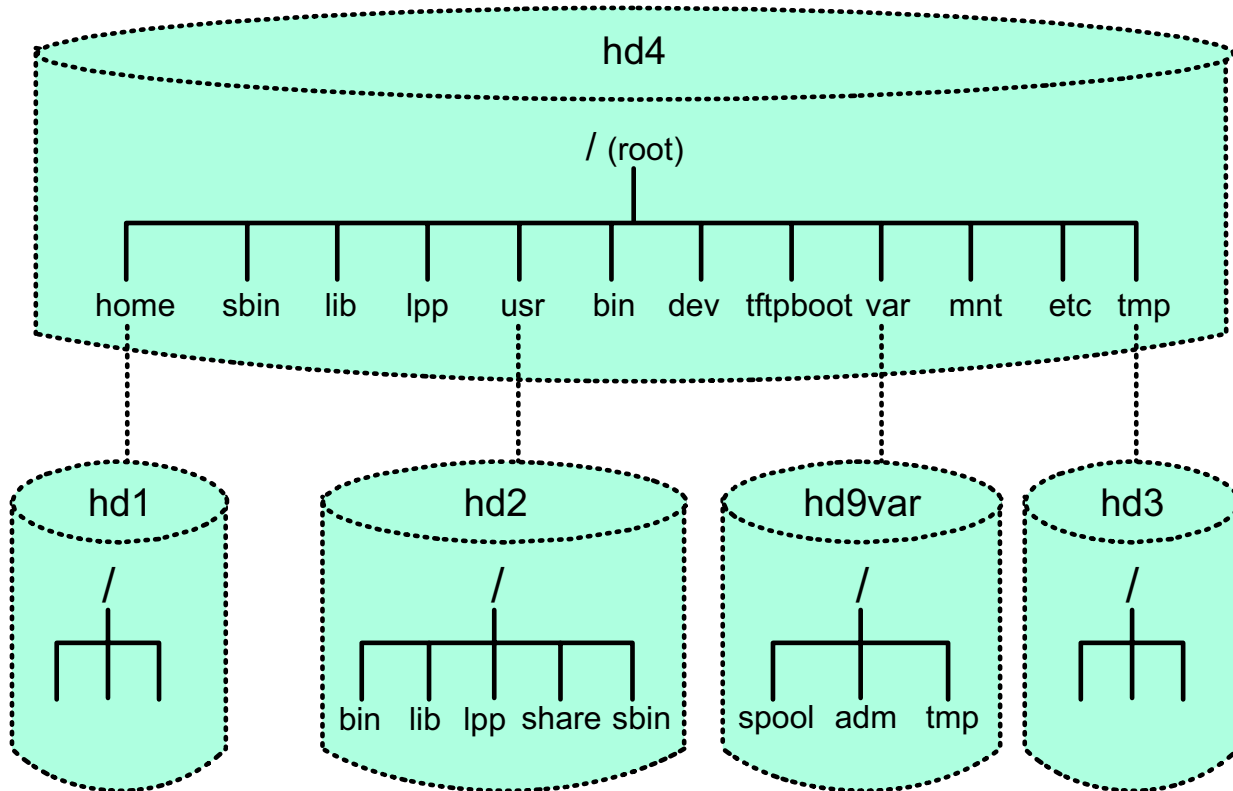
File systems can be placed on the disk in areas that will provide the best performance.

Many times, backups and recoveries are done at a file system level.

Since the administrator determines the size of the file system, users are allocated only a certain amount of shared disk space. This helps to control disk usage. The administrator can also impose more granular control over that disk space by limiting how much space an individual user can use in a file system. This is known as file system quotas.

By having several different file systems, all of your data is not in one place. If a file system ever becomes corrupted, the other file systems will not be effected. Also, administrators can take a file system off-line without effecting other file systems. This is helpful when performing back ups or when limiting user's access to the file system for security reasons.

Standard File Systems in AIX



NOTE: The drawing depicts logical not physical devices

Figure 5-13. Standard File Systems in AIX

AU147.0

Notes:

When AIX is first installed on a stand-alone system there will be only six journaled file systems in existence:

- / (root) = /dev/hd4
At the top of the hierarchical file tree. It contains the files and directories critical for system operations including the device directory and programs that complete the boot process.
- /usr = /dev/hd2
Operating system commands, libraries and application programs. Can be shared across the network.
- /var = /dev/hd9var
Variable spool and log files. The files in this file system vary considerably depending on system activity.

- /home = /dev/hd1
Users' home directories (was /u in earlier versions of AIX). This is traditionally where user data files are stored.
- /tmp = /dev/hd3
Space accessible to all users for temporary files and work space. Should be cleared out frequently.
- /proc = /proc
Special file system to support threads, or light weight processes. This file system is not designed to store user files.

Let's Review

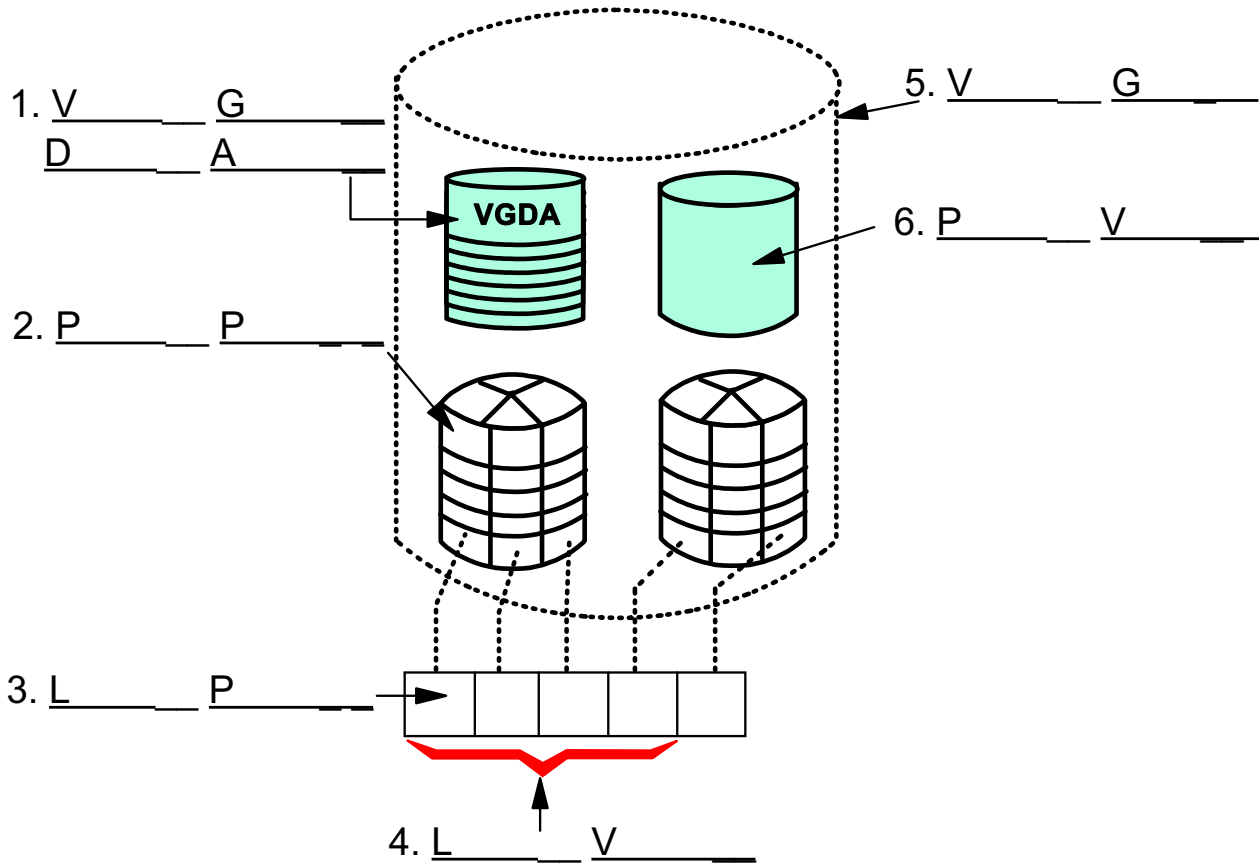


Figure 5-14. Let's Review

AU147.0

Notes:

Label the items shown in the picture above.

/etc/filesystems

```
/:
  dev      = /dev/hd4
  vol      = root
  mount    = automatic
  check    = false
  vfs      = jfs
  log      = /dev/hd8
  type     = bootfs
```

```
/home:
  dev      = /dev/hd1
  vol      = "/home"
  mount    = true
  check    = true
  vfs      = jfs2
  log      = /dev/hd8
```

Figure 5-15. /etc/filesystems

AU147.0

Notes:

The **/etc/filesystems** file documents the layout characteristics, or attributes, of file systems. It is in a stanza format which means a resource is named followed by a colon and a listing of its attributes in the form of attributes = value.

Each stanza in the **/etc/filesystems** file names the directory where the file system is normally mounted.

The file system attributes specify all the parameters of the file system. They are as follows:

- check** used by the **fsck** command to determine the default file systems to be checked. **True** enables checking
- dev** for local mounts identifies either the block special file where the file system resides, or the file or directory to be mounted
- mount** used by the **mount** command to determine whether a file system should be mounted by default

Possible values are:

automatic	file system mounted automatically at system startup
true	file system mounted by the mount all command. This command is issued during system initialization to automatically mount such file systems
false	file system will not be automatically mounted

type used to group together related file systems which can all be mounted with the **mount -t** command

vfs specifies the type of mount. For example, **vfs=jfs2**

vol used by the **mkfs** command when initiating the label on a new file system

log the device to which log data is written, as this file system is modified. (This option is only valid for journaled file systems)

Mount

- **mount**: the glue that logically connects file systems to the directory hierarchy.
- File systems are associated with devices represented by special files in /dev - the logical volume.
- When a file system is mounted, the logical volume and its contents are connected to a directory in the hierarchical tree structure.

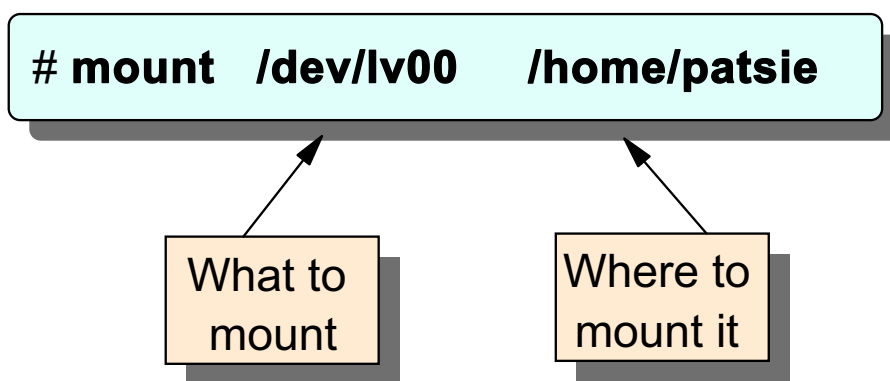


Figure 5-16. Mount

AU147.0

Notes:

A file system has to be mounted in order for it to be available for use. Use the **mount** command or SMIT to do this. The file system can also be unmounted using the **umount** or **unmount** command, or SMIT. These commands can be executed by either the root user or a member of the system group.

Full path names must be used when specifying the mount point.

It is possible to have file systems automatically mounted at boot time. This can be specified in the **/etc/filesystems** file using the **mount=automatic** or **mount=true** parameters.

If SMIT is used to create the file system, the mount point is created automatically.

Mounting over an Empty Directory

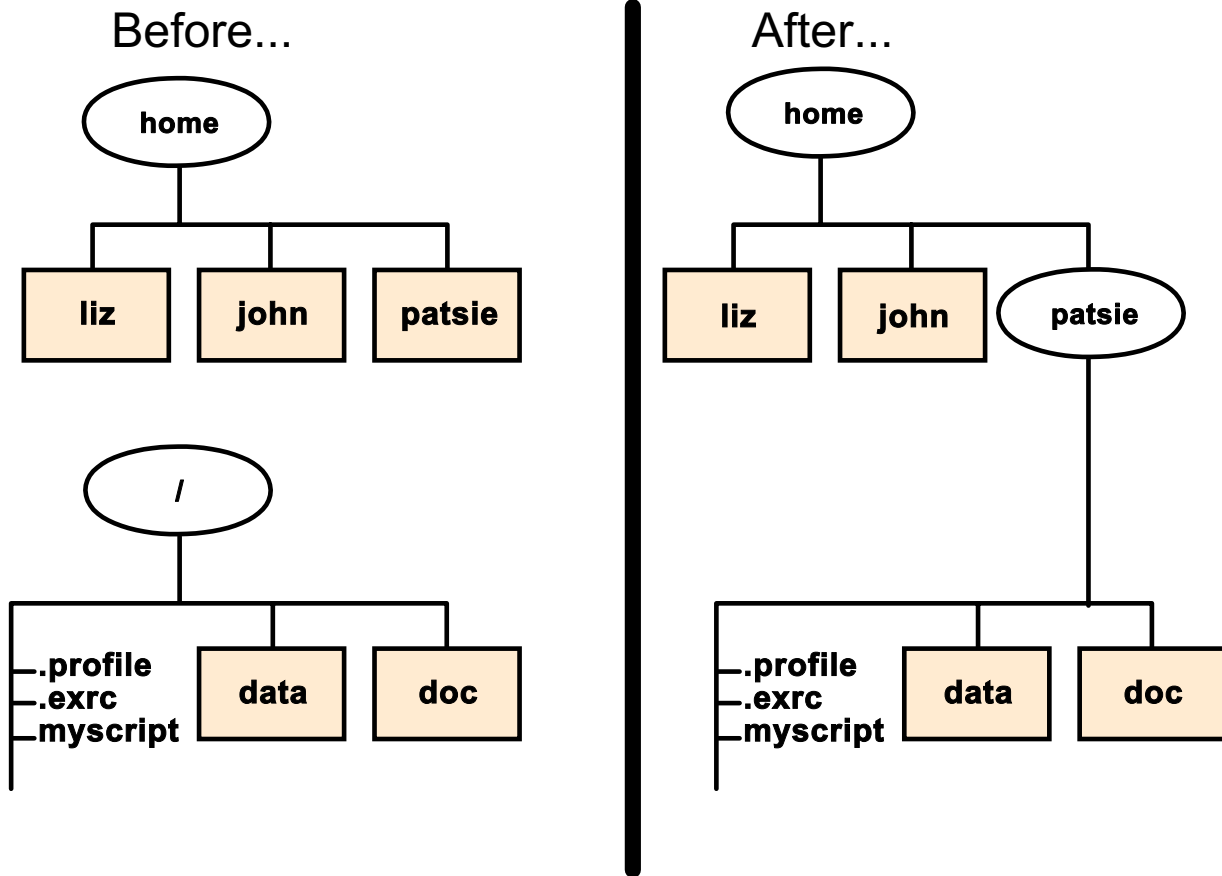


Figure 5-17. Mounting over an Empty Directory

AU147.0

Notes:

In order for users to get access to the data contained in a file system, it must be mounted. When the file system is mounted, it becomes a part of the hierarchical tree structure of files and directories. From the users perspective, there is no way to tell where one file system ends and another begins.

Mounting over Files

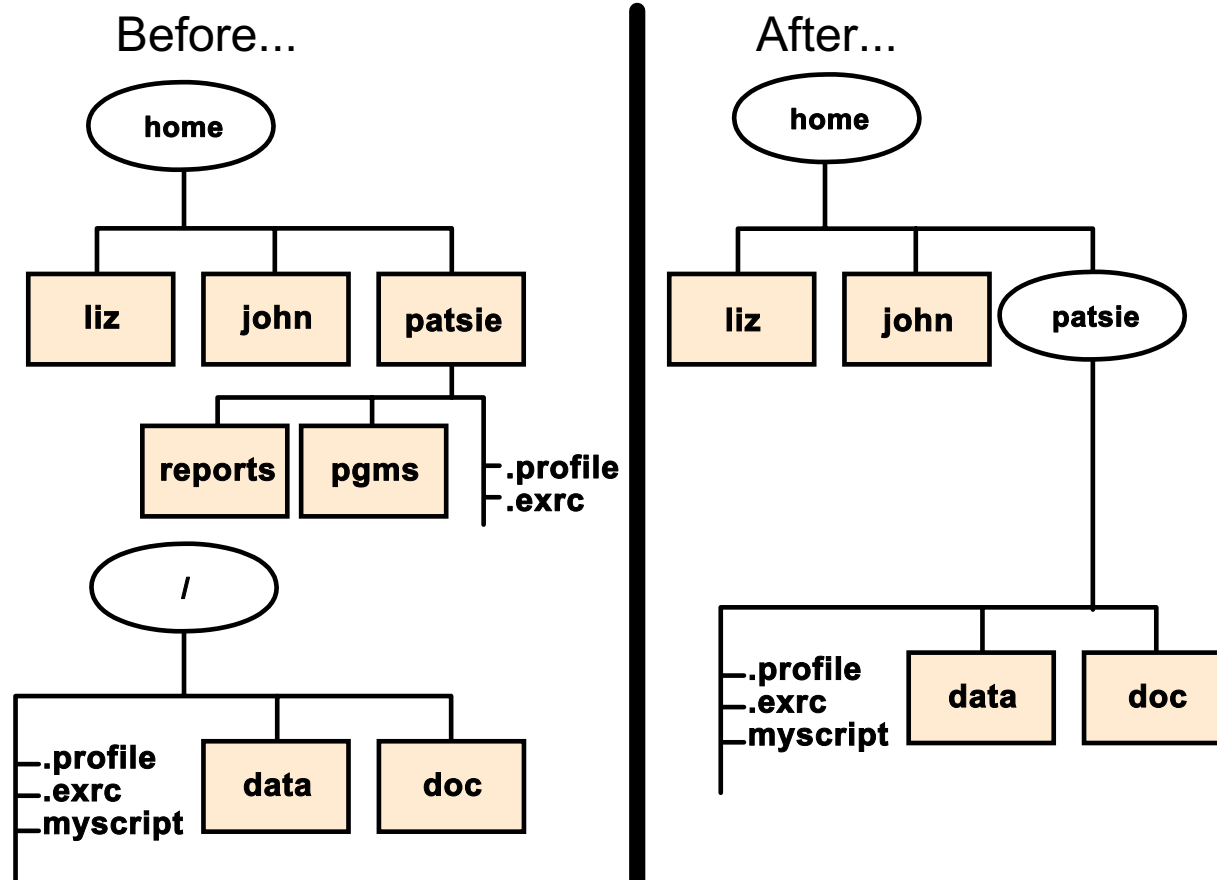


Figure 5-18. Mounting over Files

AU147.0

Notes:

It is possible to mount "over" files and subdirectories. The result is that the files and subdirectories that have been mounted over are now "hidden" from the users, that is, inaccessible. They have not been lost though. They will again be accessible when the **umount** command has been executed on the covering file system.

Not everyone has the authority to mount file systems randomly. Authority is based on two things: what the default mount point is, as specified in the file **/etc/filesystems**, and whether the user has write authority to that mount point. Users can issue file or directory mounts provided they belong to the system group and have write access to the mount point. **root** can mount anywhere under any set of permissions.

Listing File Systems

lsfs

Name	Nodename	Mount Pt	VFS	Size	Options	Auto
/dev/hd4	----	/	jfs	16384	----	yes
/dev/hd1	----	/home	jfs2	90112	----	yes
/dev/hd2	----	/usr	jfs	1277952	----	yes
/dev/hd9var	----	/var	jfs	8192	----	yes
/dev/hd3	----	/tmp	jfs	24576	----	yes
/dev/cd0	----	/infocd	cdrfs		r/o	yes
/dev/lv00	----	/home/john	jfs2	8192	r/w	yes

Figure 5-19. Listing File Systems

AU147.0

Notes:

You can list the various file systems that are defined using the **lsfs** command. This command will display information from **/etc/filesystems** and from the logical volumes in a more readable format.

lsfs will also display information about CD-ROM file systems and remote NFS file systems.

lsfs [-q] [-c | -l] [-v vfstype | -u mountgrp] [file system]

The data may be presented in line and colon (**-c**) or stanza (**-l**) format. It is possible to list only the file systems of a particular virtual file system type (**-v**), or within a particular mount group (**-u**). The **-q** option queries the superblock for the fragment size information, compression algorithm and the number of bytes per inode.

The SMIT fastpath to get to the screen which will accomplish the same task as the **lsfs** command is: **# smit fs**.

Listing Logical Volume Information

- List all Logical Volumes by Volume Group:
lsvg -l rootvg

LVNAME	TYPE	LPs	PPs	PVs	LV STATE	MOUNT POINT
hd6	paging	64	64	1	open/syncd	N/A
hd5	boot	1	1	1	closed/syncd	N/A
hd8	jfslog	1	1	1	open/syncd	N/A
hd4	jfs	2	2	1	open/syncd	/
hd2	jfs	156	156	1	open/syncd	/usr
hd9var	jfs	1	1	1	open/syncd	/var
hd3	jfs	3	3	1	open/syncd	/tmp
hd1	jfs2	1	1	1	open/syncd	/home
lv00	jfs2	2	2	1	open/syncd	/home/john

Figure 5-20. Listing Logical Volume Information

AU147.0

Notes:

lsvg -l rootvg

Provides information about the logical volumes in the rootvg volume group.

lslv lvname

This provides status information about the selected logical volume within the volume group. For example,

lslv hd6

Checkpoint (1 of 3)

1. How many different PP sizes can be set within a single VG?

2. By default, how big are PPs?

3. How many VGs can a PV belong to?
 - a. Depends on what you specify through SMIT
 - b. Only one
 - c. As many VGs as exist on the system
4. T/F All VGDA information on your system is identical, regardless of how many VGs exist.

Using the output listing the file systems shown below, answer the questions on the next page:

lsfs

Name	Nodename	Mount Pt	VFS	Size	Options	Auto
/dev/hd4	--	/	jfs	8192	--	yes
/dev/hd1	--	/home	jfs	90112	--	yes
/dev/hd2	--	/usr	jfs	507904	--	yes
/dev/hd9var	--	/var	jfs	8192	--	yes
/dev/hd3	--	/tmp	jfs	16384	--	yes
/dev/cd0	--	/infocd	cdrfs		ro	yes
/dev/lv00	--	/home/john	jfs	8192	rw	yes

Figure 5-21. Checkpoint (1 of 3)

AU147.0

Notes:

Checkpoint (2 of 3)

5. With which logical volume is the /home file system associated?

6. What type of file systems are being displayed?

7. What is the mount point for the file system located on the /dev/lv00 logical volume?

8. Which are the system supplied logical volumes and their associated file systems?

9. Which file system is used primarily to hold user data and home directories?

Figure 5-22. Checkpoint (2 of 3)

AU147.0

Notes:

Checkpoint (3 of 3)

lsvg -l rootvg

LVNAME	TYPE	LPs	PPs	PVs	LV State	MOUNT POINT
hd6	paging	8	8	1	open/syncd	N/A
hd5	boot	1	1	1	closed/syncd	N/A
hd8	jfslog	1	1	1	open/syncd	N/A
hd9var	jfs	1	1	1	open/syncd	/var
hd3	jfs	2	2	1	open/syncd	/tmp
lv00	jfs2	1	1	1	closed/syncd	/home/john

Using the output listing of logical systems above, answer the following:

10. Which of the logical volumes above are examples of logical volumes with journaled file systems on them?
-

Notes:

Activity: LVM Commands



Figure 5-24. Activity: LVM Commands

AU147.0

System Storage Overview Activity

This activity will give you a chance to look at the file system configuration on your machine and introduce you to some commands that will be covered in depth during the lecture.

1. Log in as **team01** and **su** to **root**.

```
$ su  
password: ibmaix
```

2. View the contents of **/etc/filesystems** and list the file systems on your machine.

```
# more /etc/filesystems
```

3. Try running the **lsfs** command. What does it show? _____

```
# lsfs
```

4. Try running **lsvg**. What does it show? _____

lsvg

5. Try running **lspv**. What does it show? _____

lspv

6. Try running **lsvg -l rootvg**. What does it show? _____

lsvg -l rootvg

END

All of the above commands, plus lots more, will be covered later in the course.

Unit Summary

- The LVM is organized as follows:
 - A **VG** consists of one or more **PVs**
 - Each **PV** is divided into **PPs**
 - A **LV** is made up of **LPs**
 - **LPs** are mapped to **PPs**
- Logical Volumes are used to contain:

JFS or JFS2	Paging Spaces
Dump Space	Journal Log
Boot Logical Volume	Raw Space
- The most common use of logical volumes is to contain JFS or JFS2.

Notes:

Unit 6. Working with the Logical Volume Manager

What This Unit Is About

This unit provides information on how to work with logical volumes, physical volumes and volume groups.

What You Should Be Able to Do

After completing this unit, you should be able to:

- Add/Change/Delete Volume Groups
- Add/Change/Delete Logical Volumes
- Add/Change/Delete Physical Volumes
- Describe mirroring
- Describe striping

How You Will Check Your Progress

Accountability:

- Checkpoint questions
- Exercise
- Activity

References

Online	<i>System Management Guide: Operating System and Devices</i>
GG24-4484	<i>AIX Storage Management</i>

Objectives

After completing this unit, students should be able to:

- Add/Change/Delete Volume Groups
- Add/Change/Delete Logical Volumes
- Add/Change/Delete Physical Volumes
- Describe mirroring
- Describe striping

Figure 6-1. Objectives

AU147.0

Notes:

Logical Volume Manager

```
# smit lvm
```

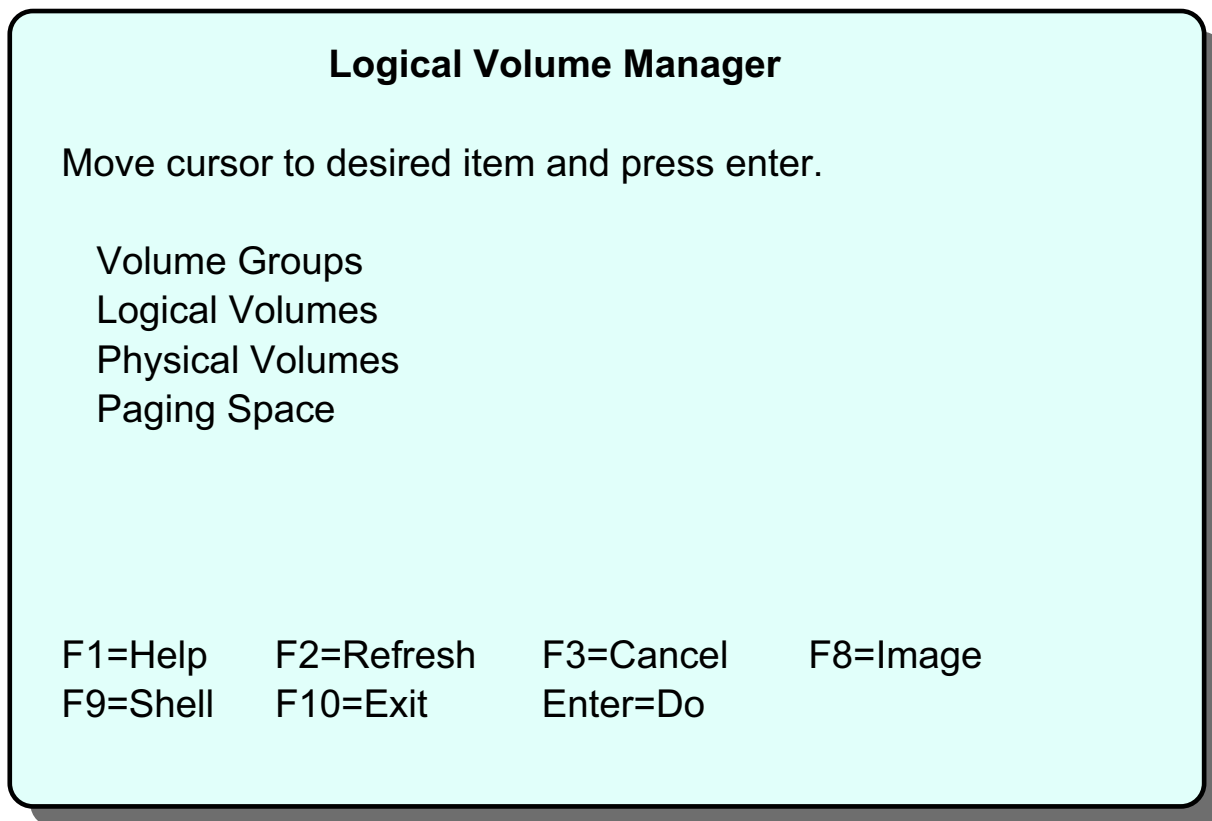


Figure 6-2. Logical Volume Manager

AU147.0

Notes:

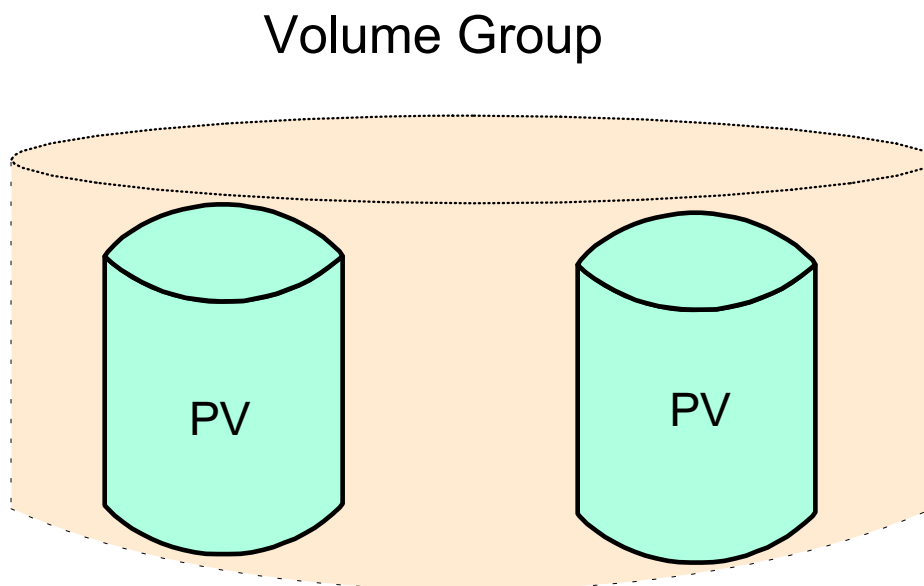
The Logical Volume Manager menu is used to manage many aspects of the system's storage.

- **Volume Groups** - This menu provides facilities to manipulate the volume groups in the system. The "Define a Fixed Disk to the Operating System" menu duplicates some items on this menu.
- **Logical Volumes** - This menu provides facilities to manipulate the logical volumes in the system. Logical volumes which contain journaled file systems, paging space or dump volumes can also be manipulated from their respective menus. However, the facilities on this menu give a much lower level of control over the characteristics of the logical volume. For example, features such as partition allocation policy and mirroring for a logical volume, can only be set using this menu. This menu is also used when a logical volume which will not contain an AIX file system, and so forth, is being manipulated.

- **Physical Volumes** - This option allows the user to configure the physical volumes (fixed disks) in the system. This menu duplicates options on the Fixed Disks menu of Devices.
- **Paging Space** - This option will allow a user to add, delete, activate and list the paging spaces available.

The Web-based System Manager can also be used to manage the Logical Volume Manager.

Volume Groups



- Physical Volume (PV)
- Volume Group (VG)

hard disk
collection of related
disks (PVs)

Figure 6-3. Volume Groups

AU147.0

Notes:

Physical Volume - hard disk. There is a limit of up to 128 PVs per volume group.

A **Volume Group** is a collection of related PVs on a processor that:

- Are not members of another VG
- Share a single physical partition size

When you install your AIX system, one volume group called **rootvg** is automatically created.

There can be a maximum of 255 VGs per system.

A PV that supports removable media should be assigned to a VG containing itself and no other members.

SMIT Volume Groups Menu

smit vg

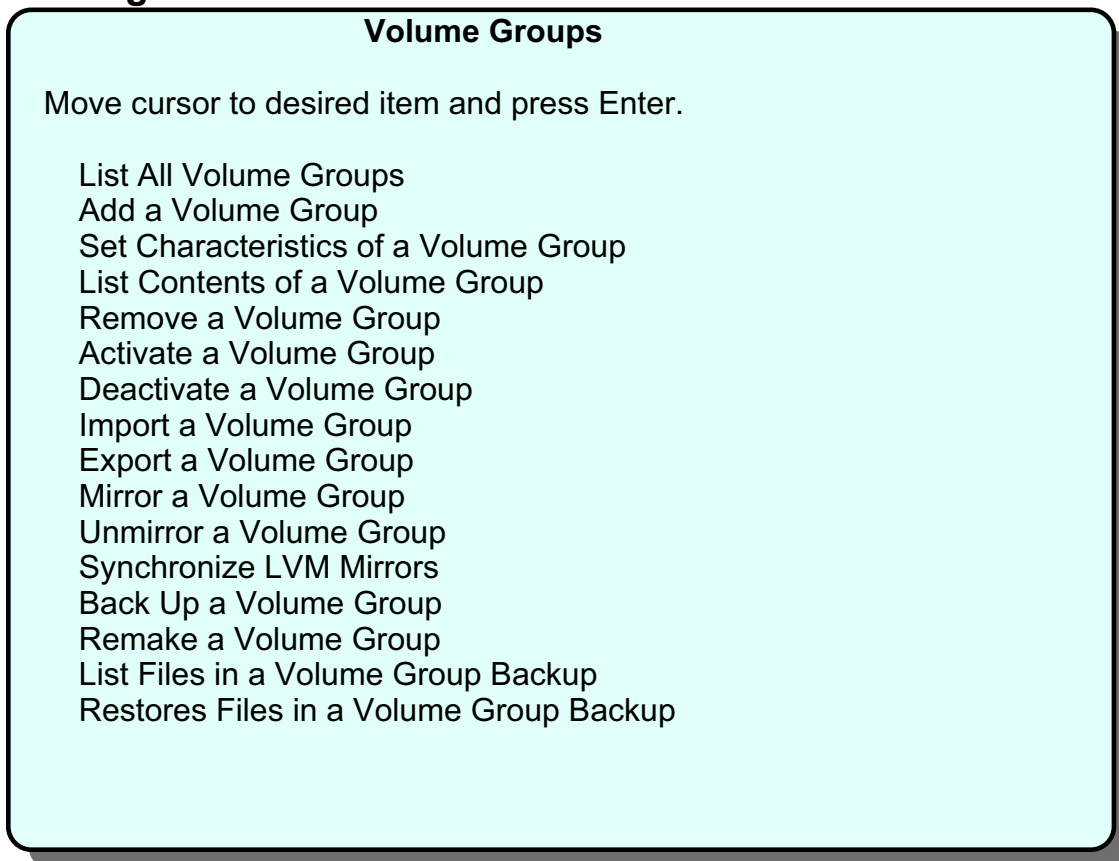


Figure 6-4. SMIT Volume Groups Menu

AU147.0

Notes:

This is the SMIT screen that allows for the configuration of volume groups. We will describe these items throughout the course.

Listing Volume Group Information (1 of 4)

List All Volume Groups

```
# lsvg  
  
rootvg  
payrollvg  
  
# lsvg -o  
  
rootvg
```

Figure 6-5. Listing Volume Group Information (1 of 4)

AU147.0

Notes:

The **lsvg** command is used to list the volume groups in the system.

It can be used to list the names of all volume groups (default) or only those that are varied on/active (**-o**).

Listing Volume Group Information (2 of 4)

List Contents of a Volume Group

```
# lsvg rootvg
VOLUME GROUP: rootvg   VGIDENTIFIER:000bc6fd00004c00000000e10fdd7f52
VG STATE: active       PP SIZE: 16 megabyte(s)
VG PERM: read/write    TOTAL PPs: 1084 (17344 megabytes)
MAX LVs: 256           FREE PPs: 1032 (16512 megabytes)
LVs: 11                USED PPs: 52 (832 megabytes)
OPEN LVs: 10           QUORUM: 2
TOTAL PVs: 2           VG DESCRIPTORS: 3
STALE PVs: 0           STALE PPs: 0
ACTIVE PVs: 2          AUTO ON: yes
MAX PPs per PV: 1016  MAX PVs: 32
LTG size: 128 kbytes   AUTO SYNC: no
HOT SPARE: no
```

Figure 6-6. Listing Volume Group Information (2 of 4)

AU147.0

Notes:

The **lsvg** command can be used to list information about the status and content of a particular volume group, for example **lsvg Volumegroup**.

The output provides status information about the volume group. The most useful information here is:

- State (active or inactive/complete if all PVs are active)

- PP size (4 MB by default)

- Total number of PPs

- Number of free PPs

Listing Volume Group Information (3 of 4)

Physical Volumes

```
# lsvg -p rootvg

rootvg:
PV_NAME    PV_STATE  TOTAL PPs  FREE PPs  FREE DISTRIBUTION
hdisk0     active   159        52        24..00..00..00..28
hdisk1     active   159        78        32..02..00..12..32
```

Figure 6-7. Listing Volume Group Information (3 of 4)

AU147.0

Notes:

lsvg -p *Volumegroup*

This gives information about all of the physical volumes within the volume group. The information given is: PV Name, PV State (active or inactive), Total number of PPs, Number of free PPs and how the free space is distributed across the disk (outer edge..outer middle..center..inner middle..inner edge).

Free distribution is the number of physical partitions allocated within each section of the physical volume: outer edge, outer middle, center, inner middle, inner edge

lsvg -l *Volumegroup*

This gives information about all of the logical volumes within the volume group. The details given are: LV name, Type of LV (for example, file system, paging), number of LPs, number of PPs, number of PVs, LV state and, if the LV contains a journaled file system the mount point is also given.

Listing Volume Group Information (4 of 4)

Logical Volumes

```
# lsvg -l rootvg

rootvg:
LVNAME      TYPE    LPs    PPs    PVs    LV STATE  MOUNT POINT
hd6         paging  8      8      1      open/syncd  N/A
hd5         boot    1      1      1      closed/syncd  N/A
hd8         jfslog  1      1      1      open/syncd  N/A
hd9var      jfs     1      1      1      open/syncd  /var
hd4         jfs     1      1      1      open/syncd  /
hd2         jfs     77     77     1      open/syncd  /usr
hd3         jfs     3      3      1      open/syncd  /tmp
hd1         jfs2   11     11     1      open/syncd  /home
lv00        jfs2    1      2      2      open/syncd  /home/john
lv01        jfs2    4      4      2      open/syncd  /home/fred
```

Figure 6-8. Listing Volume Group Information (4 of 4)

AU147.0

Notes:

lsvg -l *Volumegroup*

This gives information about all of the logical volumes within the volume group. The details given are: LV name, Type of LV (for example, file system, paging), number of LPs, number of PPs, number of PVs, LV state and, if the LV contains a journaled file system the mount point is also given.

Adding Volume Groups

```
# smit mkvg
```

Add a Volume Group

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

	[Entry Fields]	
Volume GROUP name	[]	+
Physical partition SIZE in megabytes	4	+
* PHYSICAL VOLUME names	[]	+
Activate Volume group AUTOMATICALLY at system restart	yes	
Volume Group MAJOR NUMBER	[]	+#
Create VG Concurrent Capable?	no	+
Auto-varyon in Concurrent Mode?	no	+
LTG Size in kbytes	128	+

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

Figure 6-9. Adding Volume Groups

AU147.0

Notes:

The **mkvg** command is used to create a volume group. A new volume group must contain at least one physical volume. The **-y** option is used to indicate the name for the new volume group. If this is not specified, a system generated name is used. The **-s** option is used to specify the physical partition size in MB which must be a power of 2. The default is 4 MB.

The **-n** option means that the volume group is not automatically activated at system startup. This should be done for external disks that may not always be available to the system.

Example of the **mkvg** command: to create a volume group named *newvg* created with a physical partition size of 2MB:

```
mkvg -s 2 -t 2 -y newvg hdisk1
```

The Volume Group MAJOR NUMBER on the SMIT dialog screen is used by the kernel to access that volume group. This field is most often used for High Availability Network File System (HANFS) and High Availability Cluster Multi-Processing (HACMP) applications.

The two items on the SMIT dialog screen referring to concurrent mode operation have no meaning on systems without HACMP installed. These items are valid on AIX V4.2 and later.

Removing Volume Groups

smit reducevg2

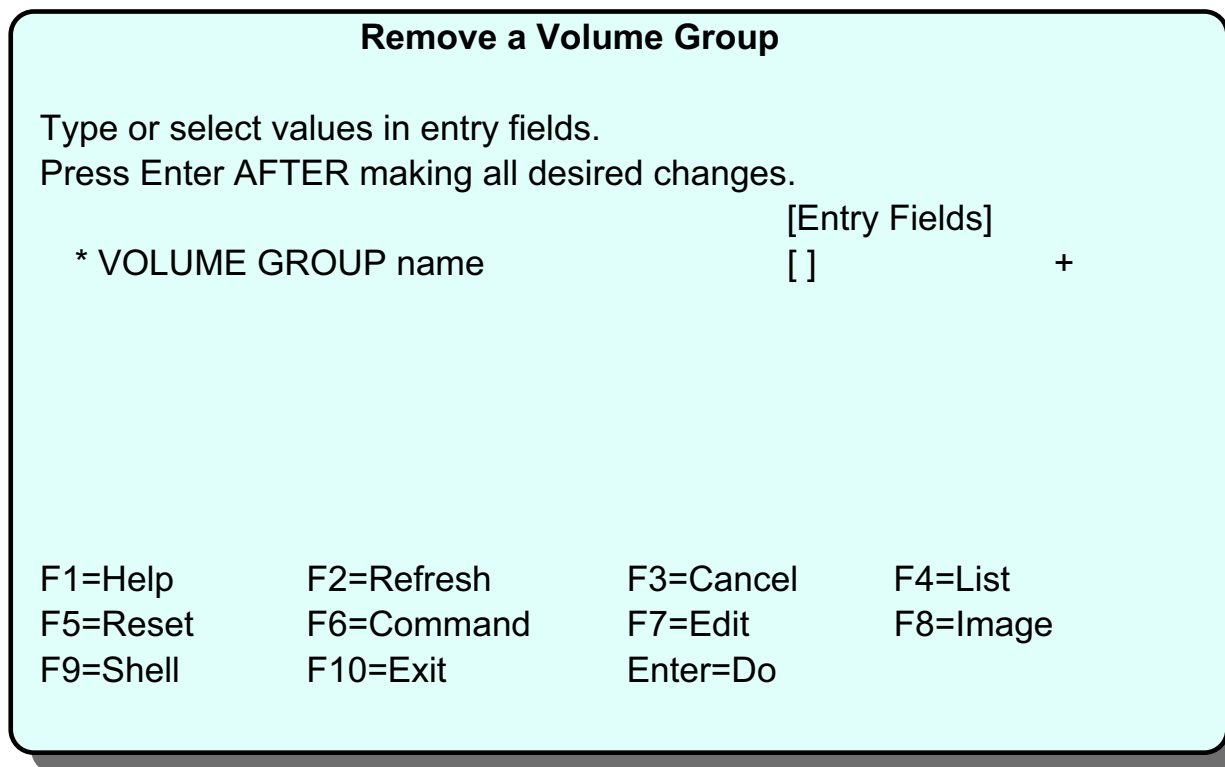


Figure 6-10. Removing Volume Groups

AU147.0

Notes:

If there are no physical volumes in a volume group, you can use the **smit reducevg2** fastpath to remove the volume group.

The **Remove a Volume Group** menu item does not have a corresponding high-level command. The correct way to remove a volume group is to use the **Remove a Physical Volume from a Volume Group** option (**reducevg** command). This will remove the volume group when you remove the last physical volume within it.

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

reducevg [-d] [-f] VolumeGroup PhysicalVolume

Set Characteristics of a Volume Group

smit vgsc

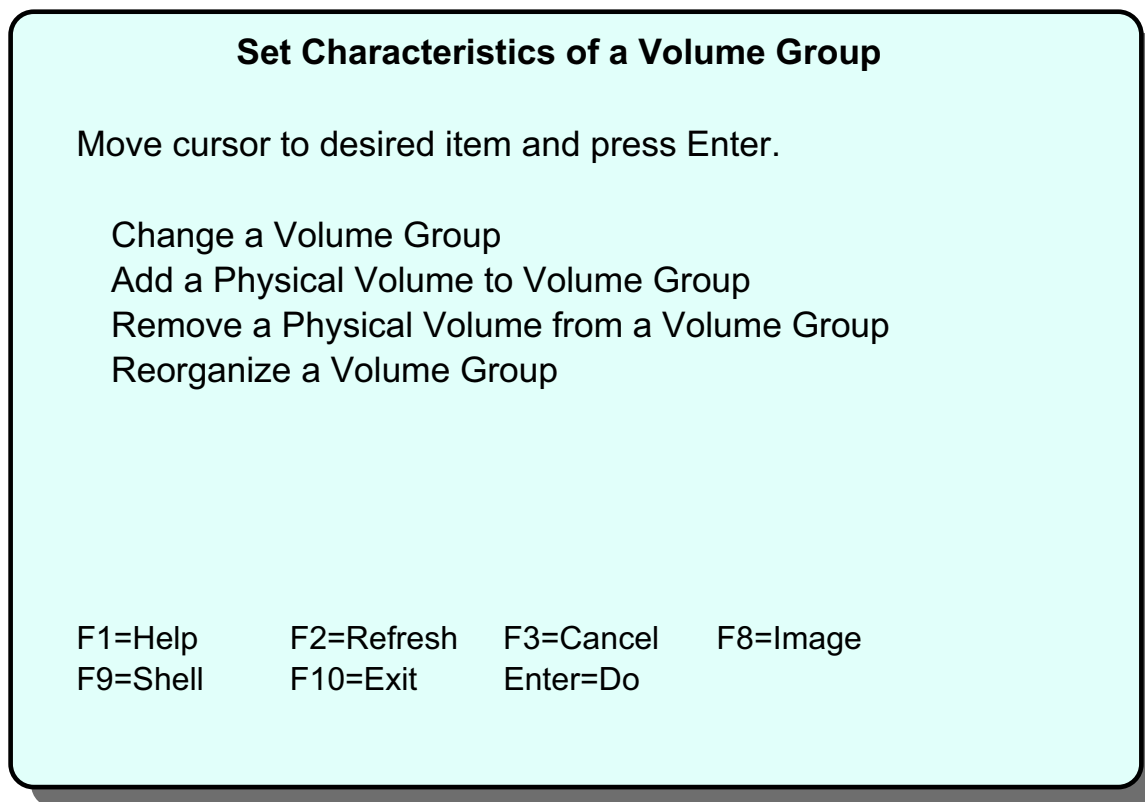


Figure 6-11. Set Characteristics of a Volume Group

AU147.0

Notes:

Add a Physical Volume to a Volume Group

extendvg -f Volumegroup hdiskn

The **extendvg** command is used to add a new physical volume to an existing volume group. The fixed disk must be installed in the system or connected to it externally, and must be powered on.

If the existing data on the disk shows that it is part of another volume group, the **-f** option forces the addition of the disk to the volume group without requesting confirmation. Use this option when adding a disk which has been previously used but which contains no data that is required any further.

Remove a Physical Volume from a Volume Group

reducevg [-d] Volumegroup hdiskn

The **reducevg** command is used to remove a physical volume from a volume group. If it is the last physical volume, the volume group will be removed.

Change a Volume Group

smit chvg

Change a Volume Group

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

	[Entry Fields]	
* VOLUME GROUP name	[]	
* Activate Volume group AUTOMATICALLY at system restart?	yes	+
* A QUORUM of disks required to keep the Volume group on-line?	yes	+
Convert this VG to Concurrent Capable?	no	+
* Autovaryon VG in Concurrent Mode?	no	+
LTG Size in kbytes	128	+
Set hotspare characteristics	n	+
Set synchronization characteristics of stale partitions	n	+

Figure 6-12. Change a Volume Group

AU147.0

Notes:

The **chvg** command is used to change the startup characteristics of a volume group. The **-a y** option sets the volume group to be used at startup. The **-a n** option resets this characteristic to no.

Logical Track Group Size (LTG)



Flexible LTG size for better performance

- Find the **maximum LTG size**:

```
# /usr/sbin/lquerypv -M hdisk0  
256
```

- Set the **LTG size per volume group** using **mkvg** or **chvg**

Figure 6-13. Logical Track Group Size (LTG)

AU147.0

Notes:

Logical track group size

The logical track group size corresponds to the maximum allowed transfer size for disk I/O. In previous versions of AIX, the only supported logical track group size was 128 KB. Many disks today support sizes larger than 128 KB. To take advantage of these larger transfer sizes and achieve better disk I/O performance, AIX 5.1 accepts the following values for the logical track group size:

- 128 KB
- 256 KB
- 512 KB
- 1024 KB

Find the LTG size

You can find the maximum supported logical track size for each physical disk. Use the `lquerypv` command with the `-M` argument. The output gives the maximum logical track group size in KB. Here is an example:

```
# /usr/sbin/lquerypv -M hdisk0
256
```

Set the LTG size

The default logical track group size is 128 KB. The maximum allowed value is the smallest maximum transfer size supported by all disks in a volume group. This value can be changed by using either of the following commands:

SMIT Fastpath	Description
<code>mkvg</code>	Shows all four values (128 KB, 256 KB, 512 KB, and 1024 KB) in the selection dialog for the logical track group size. The <code>mkvg</code> command will fail with an appropriate error message if you try to create a volume group with a LTG size larger than the physical volume will support.
<code>chvg</code>	Shows only the values for the logical track group size supported by the member disks in the volume group.

To change the LTG size, the volume group must be varied on, the logical volumes must be closed, and file systems must be unmounted.

Hot Spare

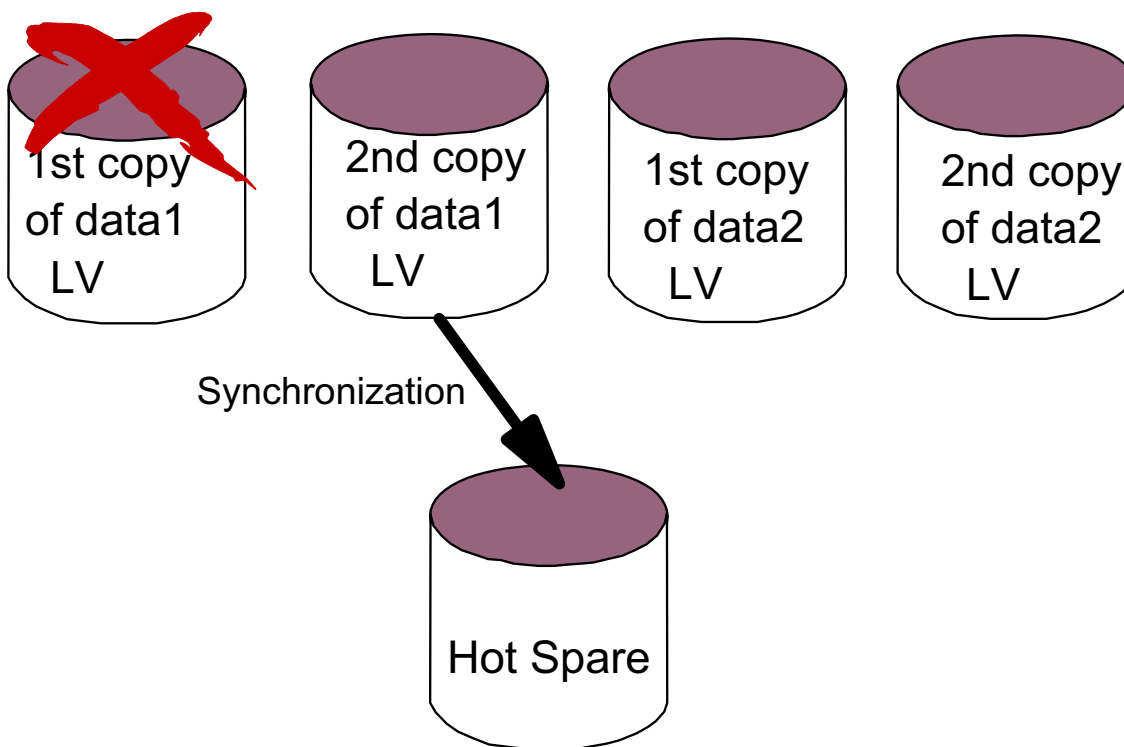


Figure 6-14. Hot Spare

AU147.0

Notes:

What is an LVM hot spare?

A hot spare is a disk or group of disks used to replace a failing disk. LVM marks a physical volume missing due to write failures. It then starts the migration of data to the hot spare disk.

Minimum hot spare requirements

The following is a list of minimal hot sparing requirements enforced by the operating system.

- Spares are allocated and used by volume group
- Logical volumes must be mirrored
- All logical partitions on hot spare disks must be unallocated

- Hot spare disks must have at least equal capacity to the smallest disk already in the volume group. Good practice dictates having enough hot spares to cover your largest mirrored disk.

Hot spare policy

The `chpv` and the `chvg` commands are enhanced with a new `-h` argument. This allows you to designate disks as hot spares in a volume group and to specify a policy to be used in the case of failing disks.

The following four values are valid for the hot spare policy argument (`-h`):

Argument	Description
<code>y</code> (lower case)	Automatically migrates partitions from one failing disk to one spare disk. From the pool of hot spare disks, the smallest one which is big enough to substitute for the failing disk will be used.
<code>Y</code> (upper case)	Automatically migrates partitions from a failing disk, but might use the complete pool of hot spare disks.
<code>n</code>	No automatic migration will take place. This is the default value for a volume group.
<code>r</code>	Removes all disks from the pool of hot spare disks for this volume group.

Synchronization policy

There is a new `-s` argument for the `chvg` command that is used to specify synchronization characteristics.

The following two values are valid for the synchronization argument (`-s`):

Argument	Description
<code>y</code>	Automatically attempts to synchronize stale partitions.
<code>n</code>	Will not automatically attempt to synchronize stale partitions. This is the default value.

Examples

The following command marks `hdisk1` as a hot spare disk:

```
# chpv -hy hdisk1
```

The following command sets an automatic migration policy which uses the smallest hot spare that is large enough to replace the failing disk, and automatically tries to synchronize stale partitions:

```
# chvg -hy -sy testvg
```

How to set up hot sparing

The following table summarizes the steps required to set up hot sparing.

Step	Command	Action
1	-	Decide which volume groups with mirrored logical volumes require high availability
2	-	Decide how many hot spare disks are required, and how large the hot spare disks must be, based on the existing disks in the volume group
3	<code>extendvg</code>	Add the hot spares to the volume groups which they are to protect
4	-	Decide which hot spare policy will be most effective for your volume groups
5	<code>chpv</code>	Designate the selected disks as hot spares
6	<code>chvg</code>	Decide which synchronization policy meets the business needs, and set the policy
7	-	Sleep well at night!

Activate/Deactivate a Volume Group

- **Activate a Volume Group (make it available for use)**

```
varyonvg [ -f ] Volumegroup
```

```
# varyonvg datavg
```

- **Deactivate a Volume Group (make it unavailable for use)**

```
varyoffvg Volumegroup
```

```
# varyoffvg datavg
```

Figure 6-15. Activate/Deactivate a Volume Group

AU147.0

Notes:

The **varyonvg** command is used to activate a volume group that is not activated at system startup (or has been added to the system since startup.)

The **-f** option is used to force a volume group online. This is needed if you have lost quorum but still want to varyon the VG. If you are using no quorum, then it is needed to bring the VG on-line if any VGDA is lost.

The **varyoffvg** command is used to deactivate a volume group. No logical volumes should be open when this command is issued. Removing a disk without deactivating the volume group could cause errors and loss of data in the volume group descriptor areas and the logical volumes within that volume group.

Import/Export a Volume Group

smit importvg

Import a Volume Group

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

	[Entry Fields]	
VOLUME GROUP name	[]	
* PHYSICAL VOLUME name	[]	+
Volume Group MAJOR NUMBER	[]	+#
Make this VG Concurrent Capable?	no	+
Make default varyon of VG Concurrent?	no	+

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

Figure 6-16. Import/Export a Volume Group

AU147.0

Notes:

If you have a volume group on one or more removable disks that you want to access on another system, you must first export the volume group from the current system using the **exportvg** command. This removes any information about the volume group from the system. To export a volume group it must be inactive.

To access an exported volume group on a system, it must be imported to the system using the **importvg** command. Do not attempt to import a rootvg. Also, unless instructed to do so by support personnel, never interrupt an LVM command.

Activity: Volume Groups



Figure 6-17. Activity: Volume Groups

AU147.0

Working with the Logical Volume Manager - Activity

In this activity, you will add/remove a disk from a volume group and create/remove a new volume group. If you are sharing the machine with someone, only one of you can perform these steps!

Instructions

1. Log in as team01 and switch to the root user account.
2. Ensure you have a hard disk available to create a new volume group. What is the name of the disk that is currently available? _____
3. From the disk information that you determined in the step above, add the free disk into **rootvg**.
4. Check to see if the disk is now associated with **rootvg**.
5. Take the same disk, out of **rootvg**. Make sure you remove the correct disk.
6. Verify that the disk is no longer associated with **rootvg**.
7. Using the free disk, create a new volume group called **newvg**.

8. Verify that the new volume group was created. What are the maximum number of PVs that can be added to this VG?
9. Convert **newvg** into a Big Volume Group.
10. What is the maximum number of PVs that can be added to the converted **newvg**?
11. Remove **newvg**.
12. Verify that **hdiskn** is not associated with any volume group. If it is not, repeat the steps above. You will need to have this disk free for a later exercise.

END

Working with the Logical Volume Manager Activity with Hints

1. Log in as team01 and switch to the root user account.

```
$ su
```

2. Ensure you have a hard disk available to create a new volume group. What is the name of the disk that is currently available? _____

```
# lspv
```

3. From the disk information that you determined in the step above, add the free disk into **rootvg**.

Note: You need to substitute your free disk name when **hdiskn** is referenced.

```
# smit vg
```

Select **Set Characteristics of a Volume Group**.

Select **Add a Physical Volume to a Volume Group**.

```
* VOLUME GROUP name           [rootvg] +
* PHYSICAL VOLUME names       [hdiskn] +
OK or ENTER
```

Then use <F10> to exit from SMIT.

OR

```
# extendvg -f rootvg hdiskn
```

4. Check to see if the disk is now associated with rootvg.

```
# lspv
```

5. Take the same disk, out of **rootvg**. Make sure you remove the correct disk!

```
# smit vg
```

Select **Set Characteristics of a Volume Group**.

Select **Remove a Physical Volume from a Volume Group**.

```
* VOLUME GROUP names           [rootvg] +
* PHYSICAL VOLUME names       [hdiskn] +
FORCE deallocation of all partitions on  no      + this physical volume?
OK or ENTER
```

Then use <F10> to exit from SMIT.

OR

```
# reducevg rootvg hdiskn
```

6. Verify that the disk is no longer associated with **rootvg**.

```
# lspv
```

7. Using the free disk, create a new volume group called **newvg**.

smit vgSelect **Add a Volume Group**

VOLUME GROUP name	[newvg]	
Physical partition SIZE in megabytes	16	+
* PHYSICAL VOLUME names	[hdiskn]	+
OK or ENTER		

Then, use <F3> to return to the Volume Group menu.

OR

mkvg -f -y "newvg" -s 8 hdiskn

8. Verify that the new volume group was created. What are the maximum number of PVs that can be added to this VG?

Select **List All Volume Groups**

List only the ACTIVE volume groups?	no	+
OK or ENTER		

Then, use <F3> to return to the Volume Group menu.

Select **List Contents of a Volume Group**

* VOLUME GROUP name	[newvg]	+
List OPTIONS	status	+
OK or ENTER		

Then, use <F10> to exit SMIT.

OR

lsvg**# lspv****# lsvg newvg**

9. Convert **newvg** into a Big Volume Group.

chvg -B newvg

Answer "y" to the prompt.

10. What is the maximum number of PVs that can be added to the converted **newvg**?

smit vgSelect **List Contents of a Volume Group**

* VOLUME GROUP name	[newvg]	+
List OPTIONS	status	+
OK or ENTER		

Then, use <F3> to return to the Volume Group menu.

OR

lsvg newvg

11. Remove **newvg**.

There are two Volume Group menu items in SMIT that will let you remove a volume group. You can select **Set Characteristics of a Volume Group** and remove the disk from the volume group. This is the same method to remove a disk from a volume group. Since there is only one disk in the volume group, the VG is automatically deleted at the same time the last disk is removed.

The other menu choice is the obvious choice. Let's pick that item.

Select **Remove a Volume Group**

* VOLUME GROUP name **[newvg]** +

OK or ENTER

Then, use <F10> to exit SMIT.

OR

reducevg rootvg hdiskn

Note: There is no such command as **rmvg**.

12. Verify that **hdiskn** is not associated with any volume group. If it is not, repeat the steps above. You will need to have this disk free for a later exercise.

lspv

END

Logical Storage

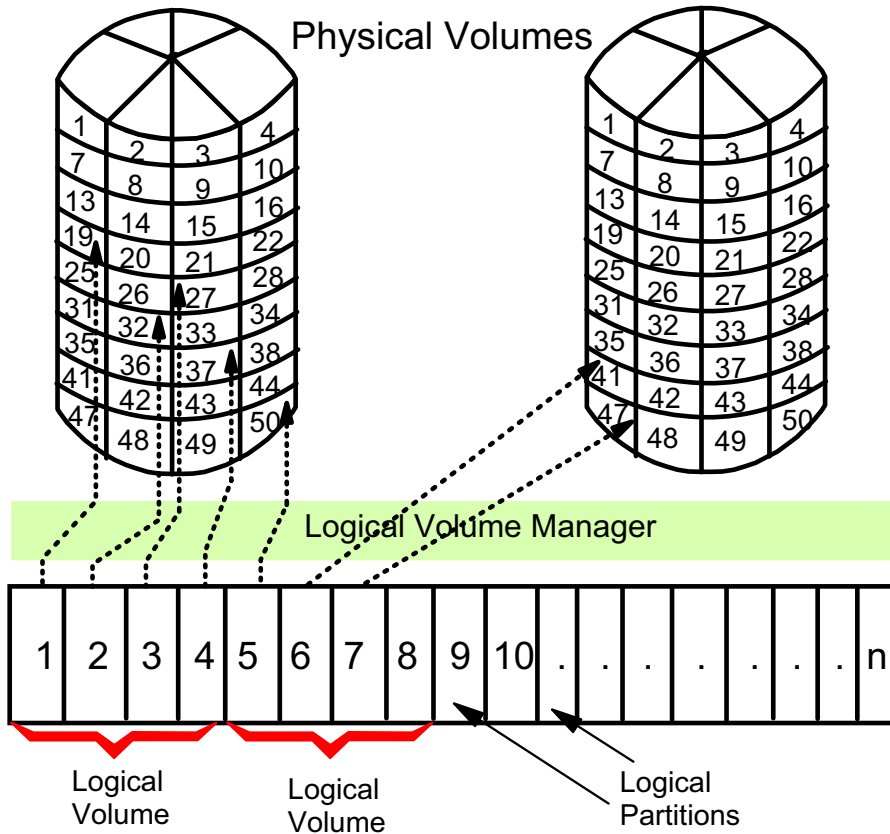


Figure 6-18. Logical Storage

AU147.0

Notes:

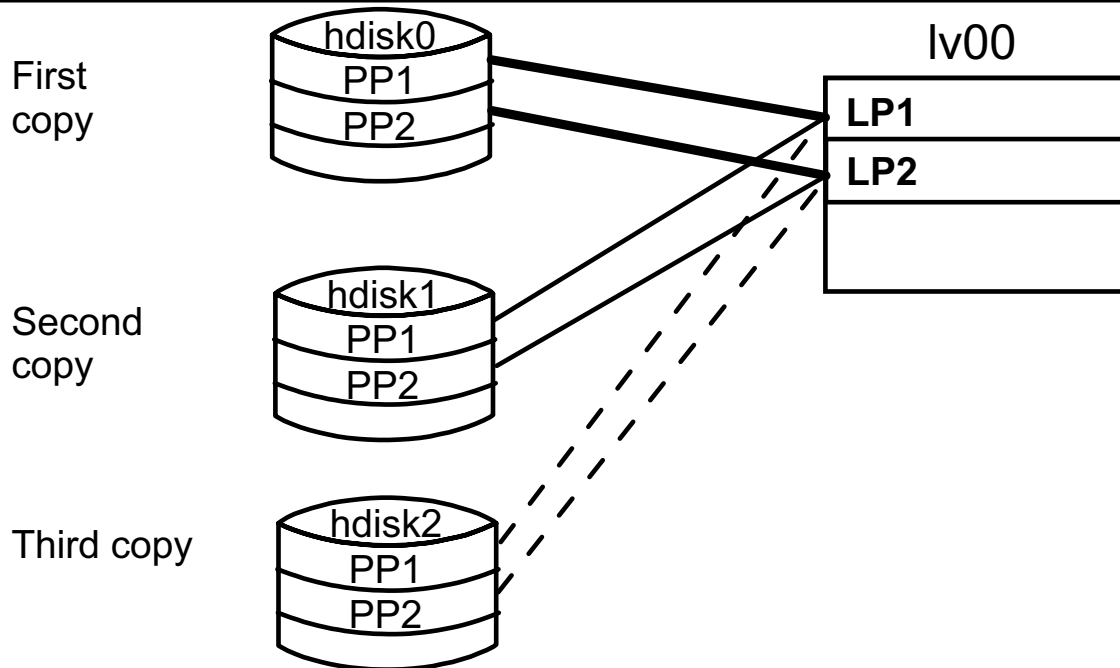
Logical Volumes

A logical volume is a set of logical partitions which may span physical volumes (as long as the physical volumes are in the same volume group). A file system sits on top of a logical volume (LV). A logical volume can be dynamically extended. There is a theoretical limit of 512 LVs per VG.

Logical Partitions

Logical partitions are mapped one-to-one to physical partitions unless there is mirroring.

Mirroring



- Mirroring is when a logical partition maps to more than one physical partition of the same volume group
- Scheduling Policy:

Parallel	PPs written simultaneously
Sequential	PPs written in sequence

Figure 6-19. Mirroring

AU147.0

Notes:

When creating a logical volume you can implement mirroring of the logical partitions of the logical volume. This means that either two or three copies are kept on the disk of the logical partitions so that the data is still intact and available in the event of a disk failure. Normally, each copy must reside on a separate disk, but this restriction can be removed if required. Physical partitions do not need to be contiguous.

Implementing mirroring will inhibit the performance of the logical volume. However, this can be partly overcome by setting the scheduling policy for logical partition copies. The scheduling policy determines how reads and writes are conducted to a mirrored logical volume. The following table describes the four possible scheduling policies.

Policy	Reads	Writes
Parallel	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.	Initiated concurrently The write request returns when the copy that takes the longest time to update completes. Control is passed back to the application after the slowest disk has completed its write. This means there is a chance that the data integrity could be lost if a disk failure occurred while the copies were being updated. To overcome this problem, the mirror write consistency option can be set on.
Parallel/sequential	Always initiates reads on the primary copy.	Initiated concurrently
Parallel/round robin	Alternates between the copies. This results in equal utilization for reads even when there is more than one I/O outstanding at a time	Initiated concurrently
Sequential	Always initiates reads on the primary copy	Initiated serially, first to the primary disk; only when that is completed is the second write initiated to the secondary disk.

Mirroring scheduling policies, such as parallel and parallel/round-robin, can allow performance on read-intensive mirrored configurations to be equivalent to non mirrored ones.

The sequential policy is used to make certain no writes are lost.

Typically, performance on write-intensive mirrored configurations is less than non-mirrored.

When turning on mirroring for an existing logical volume, the copies have to be synchronized so the new copy contains a perfect image of the existing copy at that point in time. This can be done by using the **-k** option on the **mkivcopy** command at the time mirroring is turned on or with the **syncvg** command at a later time. Until the copies are synchronized, the new copy will be marked stale.

Mirror Write Consistency

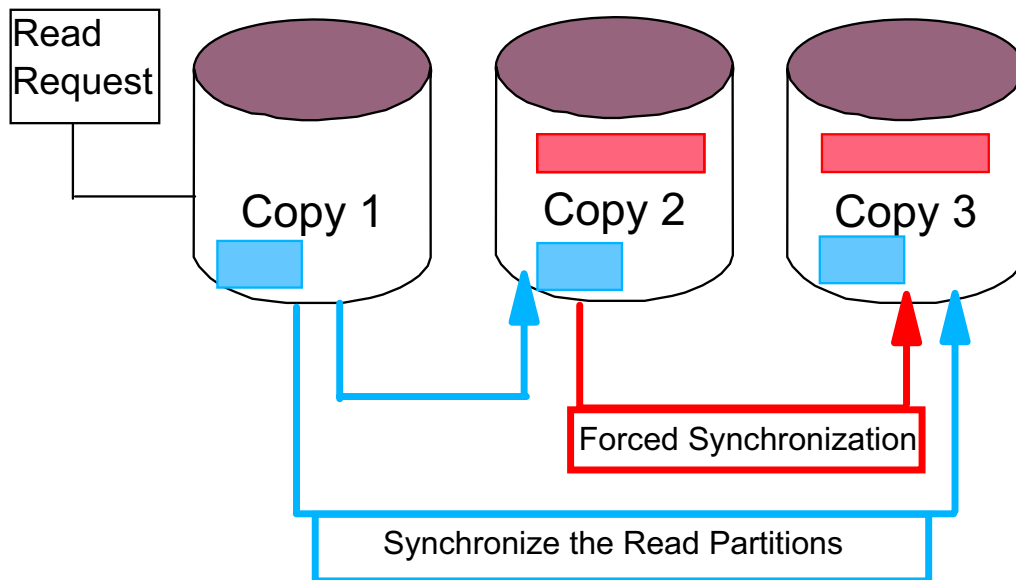


Figure 6-20. Mirror Write Consistency

AU147.0

Notes:

Introduction

Mirror Write Consistency (MWC) ensures data consistency on logical volumes in case a system crash occurs during mirrored writes. The active method achieves this by logging when a write occurs. LVM makes an update to the MWC log that identifies what areas of the disk are being updated before performing the write of the data. This results in a performance degradation during random writes.

With AIX 5.1, there are now two ways of handling MWC:

- Active, the existing method
- Passive, the new method

The purpose of the passive method

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.

Using MWC

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
```

Here is a description of the MWC arguments:

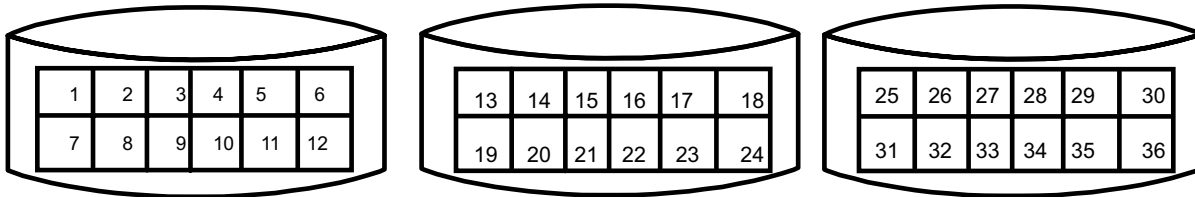
Argument	Meaning	Description
y or a	Yes or Active	Each write is logged to the MWC log. When the volume group is varied back online, the log is used to make logical partitions consistent. This is the default for mirrored logical volumes.
p	Passive	The volume group logs that the logical volume has been 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 synchronization is in progress by propagating the blocks being read to the other mirrors in the logical volume.
n	No	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.

Support for the passive option

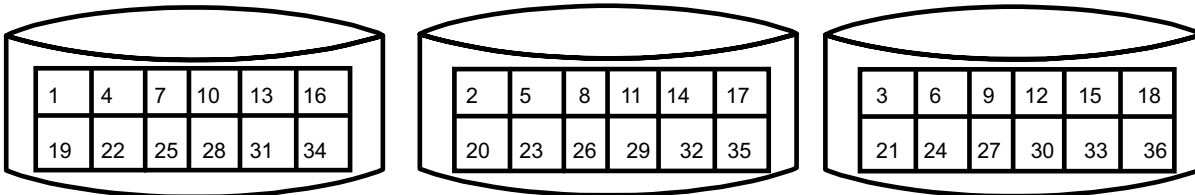
The passive method is only available on volume groups with the big volume group format since they have space to store a flag for each logical volume. Big volume groups allow up to 512 logical volumes and 128 physical volumes per volume group.

Striping

Normal flow of data blocks when a logical volume is spread across physical volumes.



The layout of stripe units when a logical volume is set up to stripe.



- Consecutive stripe units are created on different physical volumes
- Striping increases read/write sequential throughput by evenly distributing partitions among disks
- Stripe unit size is specified at creation time

Figure 6-21. Striping

AU147.0

Notes:

Striping is a technique for spreading the data in a logical volume across several disks such that the I/O capacity of the disk drives can be used in parallel to access data on the logical volume. The primary objective of striping is geared toward enabling very high-performance reading and writing of large sequential files.

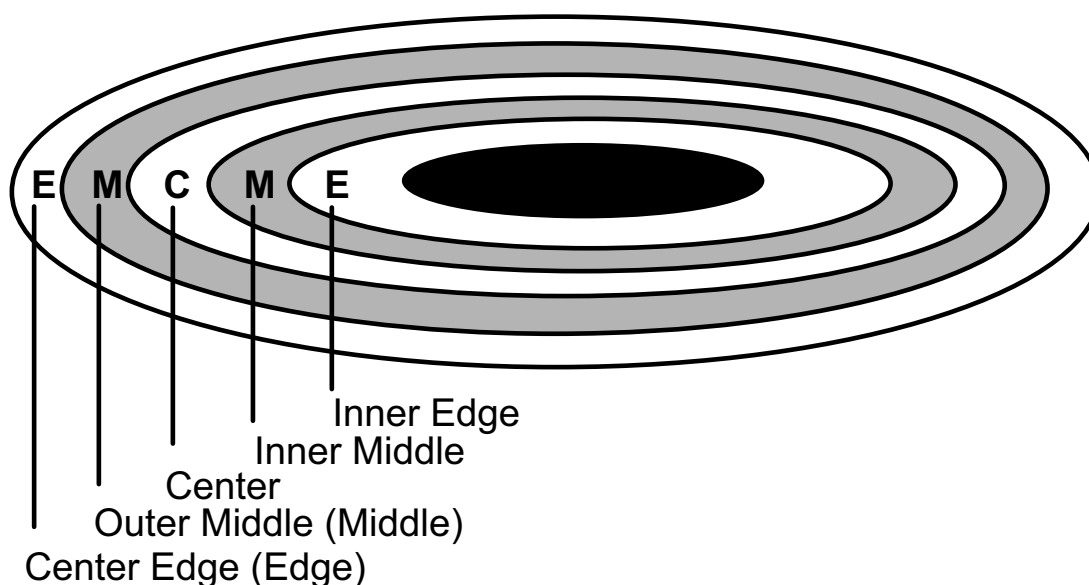
In non-striped logical volumes, data is accessed using addresses to data blocks within physical partitions. In a striped logical volume, data is accessed using addresses to stripe units. The size of the stripe unit is specified at creation time. It is specified as a power of two in the range of 4 KB to 128 KB.

The limitations are as follows:

- The number of physical partitions allocated to a striped logical volume must be able to be evenly distributed among the disks
- At least two physical volumes are required
- Use as many adapters as possible
- Create on a volume group dedicated to striped logical volumes

Logical Volume Policies

Intra-physical volume allocation policy:



Inter-physical volume allocation policy:

- ▶ Maximum number of PVs to use
- ▶ Range of PVs to use

Figure 6-22. Logical Volume Policies

AU147.0

Notes:

When creating/changing a logical volume you can set the way the Logical Volume Manager decides on which physical partitions to allocate to the logical volume. This will affect the performance of the logical volume.

The **intra-physical** volume allocation policy indicates where on the physical volume partitions will be allocated to the logical volume. The choices are: center, middle, edge, inner edge, and inner middle. Location of the data can impact performance. To determine the area with the best performance you will need to check the documentation with your disks. The center area generally was the area with the best performance on older disks. But, that may not be true with newer disks.

The **inter-physical** volume allocation policy indicates how many physical volumes can be used to contain the physical partitions of the logical volume. The maximum number of physical volumes that can be used by the logical volume can be specified (this is normally set to the number of physical volumes in the volume group). The range of volumes used can be: minimum (only allocate partitions on one physical volume, or as many as there are

copies) or maximum, (allocate partitions across all physical volumes up to the maximum number of physical volumes.)

SMIT Logical Volumes Menu

smit lv

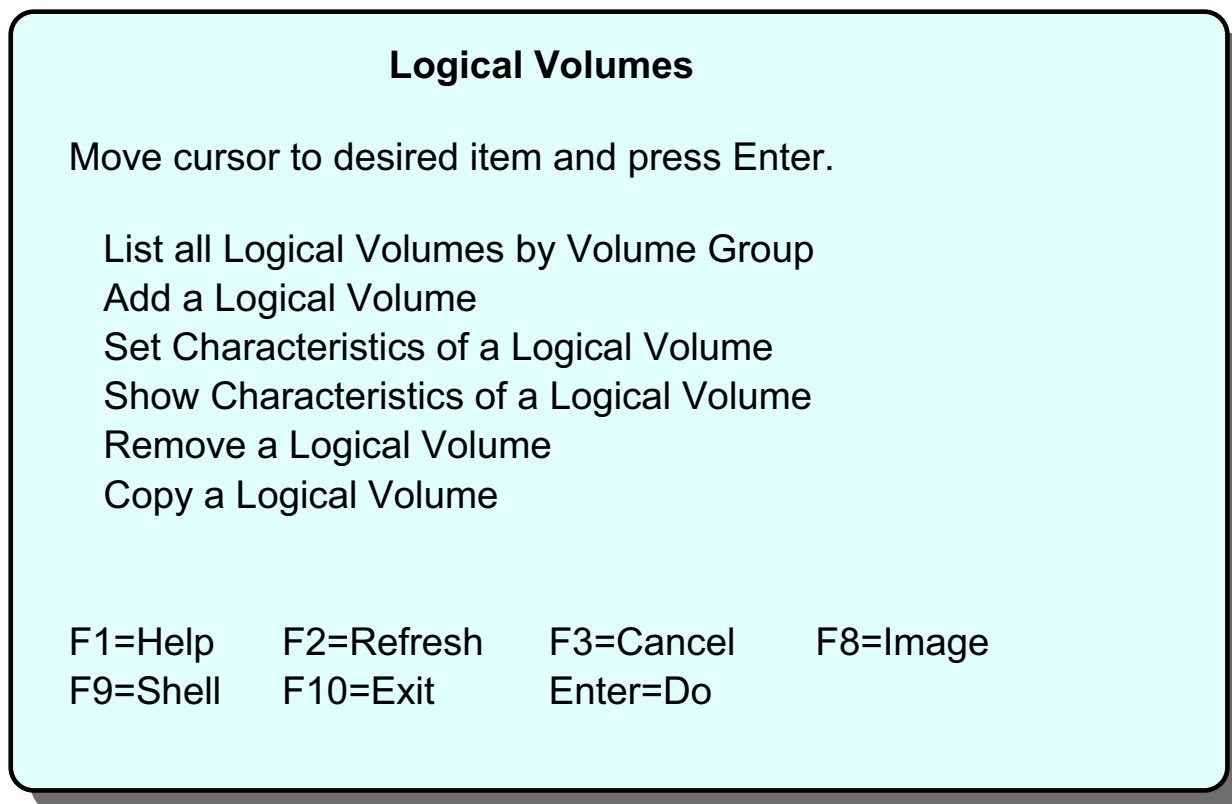


Figure 6-23. SMIT Logical Volumes Menu

AU147.0

Notes:

This is the top-level SMIT menu for logical volumes. The next few pages will discuss these items.

Showing Logical Volume Characteristics

- Show Characteristics of a Logical Volume:

Physical Volume map:

```
# Islv -l lv00
```

```
lv00:/home/john
```

PV	COPIES	IN BAND	DISTRIBUTION
hdisk0	010:000:000	70%	000:000:007:003:000

Logical Partition map:

```
# Islv -m lv00
```

```
lv00:/home/john
```

LP	PP1	PV1	PP2	PV2	PP3	PV3
0001	0134	hdisk0				
0002	0135	hdisk0				
0003	0136	hdisk0				

Figure 6-24. Showing Logical Volume Characteristics

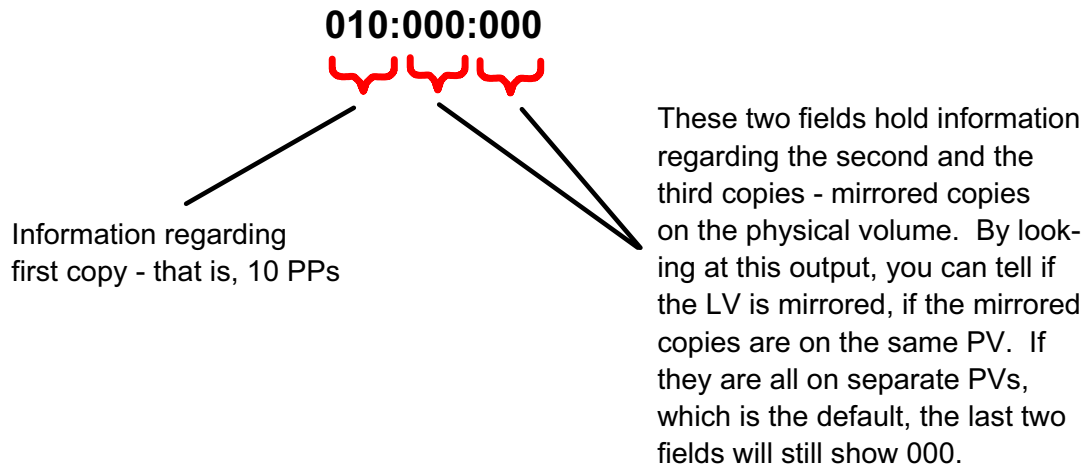
AU147.0

Notes:

Islv -l lvname

This gives information about the distribution of a particular logical volume's logical partitions for each physical volume. The information includes the number of logical partitions on the disk and its copies, if any, on that disk; the percentage of physical partitions which match the intra-physical volume allocation policy; the distribution of physical partitions on the physical volume (outer edge, outer middle, center, inner middle, inner edge).

Copies can be interpreted as:



The **IN BAND** attribute shows the percentage of the physical partitions on the physical volume that belong to the logical volume and were allocated within the region specified by the intra-allocation policy.

Distribution

There is a relationship between the numbers 000:000:007:003:000 and 010:000:000 whereby the 007:003 numbers indicate the distribution of the 010. The interpretation is: of the 10 PPs, 7 PPs are located in the center and 3 PPs in the inner-middle of the disk, respectively.

lslv -m lvname

This gives a map of which physical volumes contain which physical partitions for the logical partitions of the logical volume. Three columns are given, one for each copy of a logical partition.

Add a Logical Volume

```
# 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	[]	
* Volume GROUP name	rootvg	
* Number of LOGICAL PARTITIONS	[]	#
PHYSICAL VOLUME names	[]	+
Logical Volume TYPE	[]	
POSITION on physical volume	middle	+
RANGE of physical volumes	minimum	+
MAXIMUM NUMBER of PHYSICAL VOLUMES	[]	#
to use for allocation		
Number of COPIES of each logical partition	1	+
Mirror Write Consistency	yes	+
Allocate each logical partition copy on a SEPARATE physical volume?	yes	+

[MORE... 10]

Figure 6-25. Add a Logical Volume

AU147.0

Notes:

The **mklv** command creates a logical volume. The name of the logical volume can be specified or alternatively a system-generated name will be used. The volume group the logical volume will belong to and size (in logical partitions) must be specified. Other characteristics that can be set are the allocation policy, copies (mirroring), scheduling policy and striping.

Set Characteristics of a Logical Volume

```
# smit lvsc
```

Set Characteristics of a Logical Volume

Move cursor to desired item and press Enter.

Change a Logical Volume
Rename a Logical Volume
Increase the Size of a Logical Volume
Add a Copy to a Logical Volume
Remove a Copy from a Logical Volume

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

Figure 6-27. Set Characteristics of a Logical Volume

AU147.0

Notes:

The **chlv** command is used to change the characteristics of a logical volume. Characteristics that can be changed are the allocation and scheduling policies and the permissions. (When a logical volume is created it always has read/write permission, but this can be changed to read-only later.)

You can change the name of a logical volume using the **chlv** command with the **-n** option. No other **chlv** options can be specified if **-n** is used.

The size of a logical volume may be increased at any time, assuming that there is sufficient space in the volume group. To do this the **extendlv** command is used. You can set the allocation policies for the new partitions to different values than used by the original logical volume.

The size of a logical volume may not be decreased automatically. To make a logical volume smaller, back it up, delete it, create a new logical volume of the desired size and restore the data.

Showing LV Characteristics (1 of 2)

List all Logical Volumes by Volume Group

```
# lsvg -o | lsvg -i -l
```

LVNAME	TYPE	LPs	PPs	PVs	LV STATE	MOUNT POINT
hd6	paging	8	8	1	open/syncd	N/A
hd5	boot	1	1	1	closed/sync	N/A
hd8	jfslog	1	1	1	open/syncd	N/A
hd9var	jfs	1	1	1	open/syncd	/var

Figure 6-28. Showing LV Characteristics (1 of 2)

AU147.0

Notes:

List all **Logical Volumes by Volume Group** uses **lsvg -o** to find out the active volume groups and then **lsvg -il** to list the logical volumes within them. The **-i** option of **lsvg** reads the list of volume groups from standard input.

The SMIT option **Show Characteristics of a Logical Volume** uses the **lslv lvname** to show status information about the selected logical volume.

Showing LV Characteristics (2 of 2)

Show Characteristics of a Logical Volume

```
# lslv lv02

LOGICAL VOLUME: lv02                VOLUME GROUP: course
LV IDENTIFIER: 000000000004c0000000e5cf75106f.4
PERMISSION:    read/write
VG STATE:     active/complete      LV STATE:      opened/syncd
TYPE:         jfs2                 WRITE VERIFY:  off
MAX LPs:      128                  PP SIZE:      4 megabyte(s)
COPIES:       1                    SCHED POLICY: parallel
LPs:          10                    PPs:          10
STALE PPs:    0                    BB POLICY:    relocatable
INTER-POLICY: minimum              RELOCATABLE:  yes
INTRA-POLICY: middle               UPPER BOUND:  32
MOUNT POINT:  /home/malcolm        LABEL:        /home/malcolm
MIRROR WRITE CONSISTENCY: on
EACH LP COPY ON A SEPARATE PV ?:  yes
```

Figure 6-29. Showing LV Characteristics (2 of 2)

AU147.0

Notes:

Write Verify

Specifies whether to verify all writes to the logical volume with a follow-up read.

Bad Block

Indicates whether the LVM should try to relocate a bad block if one is encountered.

Add/Remove a Logical Volume Copy

smit mklvcopy

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	lv00	
* NEW TOTAL number of logical partition copies	2	+
PHYSICAL VOLUME names	[]	+
POSITION on physical volume	middle	+
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	+

Figure 6-30. Add/Remove a Logical Volume Copy

AU147.0

Notes:

The **mklvcopy** command is used to add copies (mirroring) to a logical volume that has none or to increase the copies from two or three. Specify the logical volume to change the desired total number of copies. This will only succeed if there are enough physical partitions to satisfy the requirements on the physical volumes that are specified to be used (that is, if all copies are to be on different physical volumes).

Also, in order for the copies to match, the logical volume has to be synchronized using the **syncvg** command. This can be done with the **-k** option when the copy is originally started. It can be done later using the **syncvg** command.

The **rmlvcopy** command is used to reduce the total number of copies for a logical volume. Specify the desired total number (for example, two if you are reducing the number of copies from three to two). The **rmlvcopy** command allows you to specify which disk to remove the copy from.

Once a logical volume has been created, striping cannot be imposed or removed.

Reorganize a Volume Group

```
# smit reorgvg
```

Reorganize a Volume Group

Type or select values in entry fields.

Press Enter AFTER making all desired changes.

		[Entry Fields]	
* VOLUME GROUP name		vg3	+
LOGICAL VOLUME names		[lv04 lv07]	+

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

Figure 6-31. Reorganize a Volume Group

AU147.0

Notes:

If the intraphysical volume policy (center, middle, edge, and so forth) is changed after the LV is created, the physical partition will not relocate automatically.

The **reorgvg** command is used to redistribute the physical partitions of the logical volumes of a volume group according to their preferred allocation policies. This should improve disk performance. Preference is given in the order listed on the command line.

```
reorgvg volumegroup [lvname]
```

```
# reorgvg vg3 lv04 lv07
```

In AIX V4.2 and later, if you enter the **reorgvg** command with the volume group name and no other arguments, the entire volume group is reorganized.

Physical Volumes

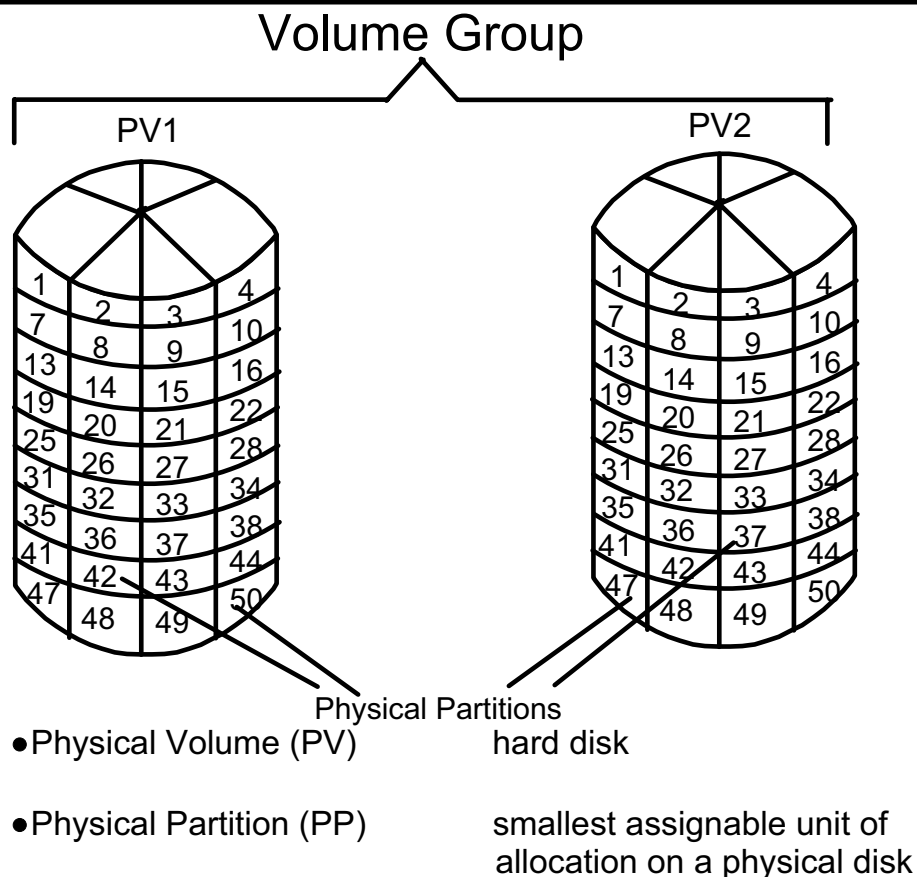


Figure 6-32. Physical Volumes

AU147.0

Notes:

A **Physical Partition** is a fixed size, contiguous set of bytes on a physical volume (PV). Physical partitions (PP) must be the same size across an entire VG. However, there may be multiple VGs on a single system, each having a different PP size. A PP can be 1 - 1024 MB.

AIX V4.3.2 provides support for more than 1016 physical partitions per physical volume. This support provides for multiples of 1016 PPs per PV. Using more than 1016 PPs per PV in this manner will reduce the total number of disks that can exist in the volume group by the same fraction as indicated in the following table:

Reg VG # of PVs	# of PPs	Big VG # of PVs
32	1016	128
16	2032	64
8	4064	32
4	8128	16
2	16256	8

Big Volume Groups are available with 4.3. They can be created with **mkvg -B** or **chvg -B**.

SMIT Physical Volumes Menu

smit pv

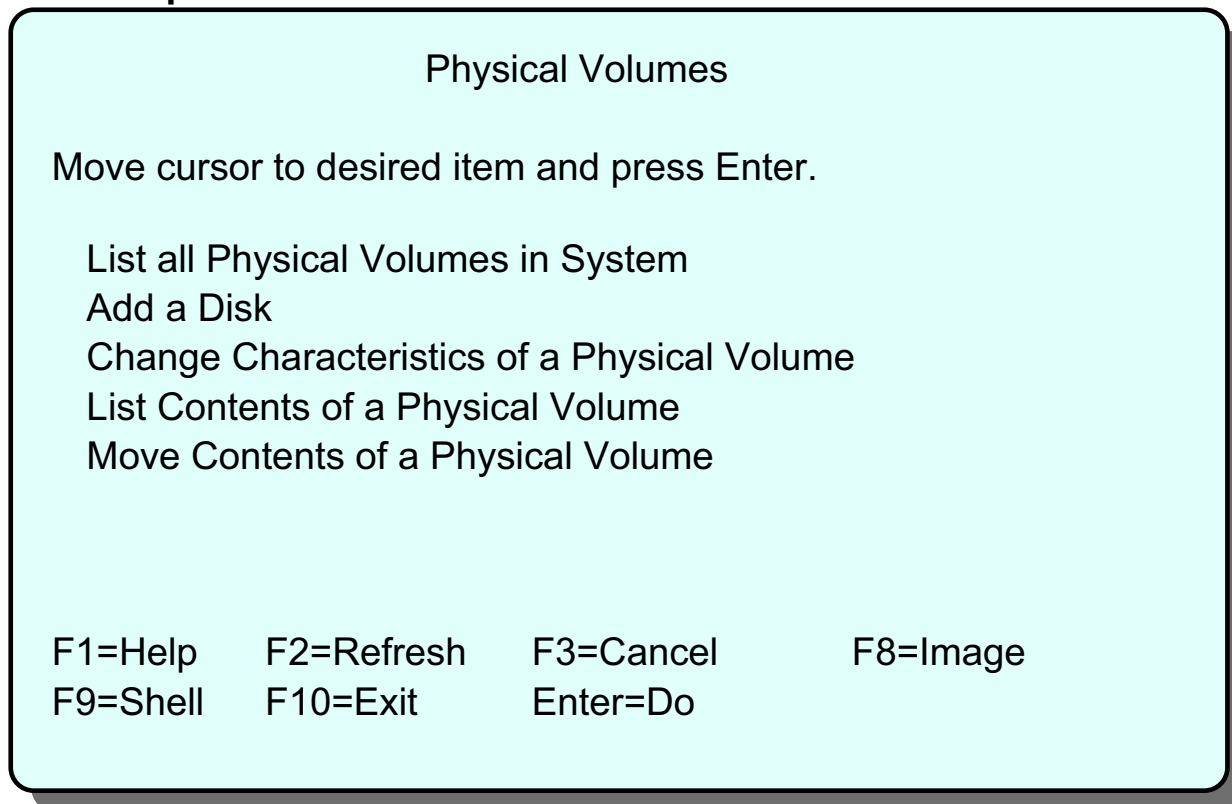


Figure 6-33. SMIT Physical Volumes Menu

AU147.0

Notes:

This is the top-level menu for physical volume. We will explain each of these items in the following pages.

Listing Physical Volume Information (1 of 3)

- List all Physical Volumes in System:

```
# lspv

hdisk0      da1c923411d52ec91cd600802eda72c9      rootvg
hdisk1      bebc8000000000000000000000802evg79c9      rootvg
```

- List Contents of a Physical Volume:

```
# lspv hdisk0

PHYSICAL VOLUME:  hdisk0                VOLUME GROUP:  rootvg
PV IDENTIFIER:    da1c923411d52ec91cd600802eda72c9
VG IDENTIFIER:    000bc6fd00004c00000000e10fdd7f52
PV STATE:        active
STALE PARTITIONS: 0                    ALLOCATABLE:   yes
PP SIZE:         4 megabyte(s)          LOGICAL VOLUMES: 6
TOTAL PPs:      95 (380 MBs)            VG DESCRIPTORS: 2
FREE PPs:       3 (12 megabytes)        HOT SPARE:     no
USED PPs:       92 (368 megabytes)
FREE DISTRIBUTION: 00..03..00..00..00
USED DISTRIBUTION: 19..16..19..19..19
```

Figure 6-34. Listing Physical Volume Information (1 of 3)

AU147.0

Notes:

List All Physical Volumes in System actually uses the undocumented command **getlvodm -C** to list the physical volumes in the system.

The **lspv** command with no parameters can be used to list the physical volume name, PV identifier and volume group for all physical volumes in the system.

The **lspv pvname** command gives status information about the physical volume. The most useful information here is: state (active or inactive), number of PP copies that are stale (are not up to date with other copies), total number of PPs, number of free PPs and distribution of free space on the PV.

Listing Physical Volume Information (2 of 3)

Logical Volumes

```
# lspv -l hdisk0
```

```
hdisk0:
```

LV NAME	LPs	PPs	DISTRIBUTION	MNT POINT
hd1	12	12	00..00..00..12..00	/home
hd3	3	3	00..03..00..00..00	/tmp
hd2	29	29	00..00..17..12..00	/usr
hd4	13	13	00..00..13..00..00	/
hd8	1	1	00..00..01..00..00	N/A
hd6	8	8	00..00..00..08..00	N/A
hd5	1	1	01..00..00..00..00	N/A

Figure 6-35. Listing Physical Volume Information (2 of 3)

AU147.0

Notes:

lspv -l pvname lists all the logical volumes on a physical volume including number of logical partitions, physical partitions and distributions on the disk.

Listing Physical Volume Information (3 of 3)

Physical Partition Map

```
# lspv -p hdisk0
hdisk0:
PP RANGE   STATE REGION   LV NAME TYPE  MNT   POINT
1-1        used  outer edge   hd5          boot  N/A
2-31       used  outer edge   hd2          jfs   /usr
32-32      free  outer edge
33-40      used  outer middle hd6          paging N/A
41-64      free  outer middle
65-65      used  center       hd8          jfslog N/A
66-66      used  center       hd4          jfs   /
67-73      used  center       hd2          jfs   /usr
74-74      used  center       hd9var       jfs   /var
75-76      used  center       hd3          jfs   /tmp
77-77      used  center       hd1          jfs2  /home
78-84      used  center       hd2          jfs   /usr
85-92      used  center       paging00     paging N/A
93-95      used  center       hd10opt     jfs   /opt
96-159     used  inner middle  hd2          jfs   /usr
```

Figure 6-36. Listing Physical Volume Information (3 of 3)

AU147.0

Notes:

lspv -p pvname lists all the logical volumes on a disk and the physical partitions to which its logical partitions are mapped. It is listed in physical partition order and shows what partitions are free and which are used, as well as the location; that is, outer edge, outer middle, center, and so forth.

Add/Move Contents of Physical Volumes

- **Add a disk:**

The device can either be added through SMIT or it can be configured through configuration manager when the system boots up.

- **Move the contents of a Physical Volume:**

`migratepv [-l lvname] sourcePV targetPV ..`

```
# migratepv -l lv02 hdisk0 hdisk6
```

Figure 6-37. Add/Move Contents of Physical Volume

AU147.0

Notes:

To add a physical volume to the system, the option performed is the **Add a Disk** option from the **Fixed Disks** menu under **Devices**. This adds the disk and assigns it an hdisk number. Once the disk has been added, it needs to be added to a volume group so that it can be used. Refer to the **Volume Groups** or **Define a Fixed Disk to the Operating System** menus.

The alternative method is to power down the system, connect the new disk to the system, power up the system, and in so doing **cfgmgr** will be invoked, which will pick up the new device (if it is a detectable device).

In AIX V4.3.1 and later, if you wish to add a disk that exceeds the 1016 PP/PV limitation to a pre-existing volume group, first convert the volume group so that it can hold multiples of 1016 partitions per disk. This is done using the **chvg -t (factor)** command, where *factor* is a value between 1 and 16. Thus, the maximum number of physical partitions per physical volume for this volume group changes to factor X 1016.

The **migratepv** command can be used to move all partitions (or partitions from a selected LV) from one physical volume to one or more other physical volumes in the same volume

group. This would be used if the physical volume is about to be taken out of service and removed from the machine or to balance disk usage.

Working with lvmstat

```
# lvmstat -e -v rootvg
# lvmstat -v rootvg
Logical Volume      iocnt      Kb_read     Kb_wrtn     Kbps
hd3                  1873        240          8480         0.00
hd8                  1764         0           7056         0.00
hd4                  1675         20          8464         0.00
hd9var              1412         81          7387         0.00
hd2                  893          628         3544         0.00
hd10opt             31           89           32           0.00
hd6                  20           0            80           0.00
hd1                   1            4            0            0.00
hd5                   0            0            0            0.00

# lvmstat -l hd3
Log_part  mirror#    iocnt     Kb_read   Kb_wrtn   Kbps
      1      1       137         4        740     0.00
      1      2       136         0        740     0.00
      2      1         0          0          0     0.00
      2      2         0          0          0     0.00
```

Figure 6-38. Working with lvmstat

AU147.0

Notes:

LVM hot spot management

You have a hot spot problem if some of the logical partitions on your disk have so much disk I/O that your system performance deteriorates. Two new commands, **lvmstat** and **migratelp**, help you to identify and remedy hot spot problems within your logical volumes.

Gather statistics

The gathering of statistics must be enabled with the **lvmstat** command for either a logical volume or an entire volume group. By default, no statistics are gathered. Enabling statistics collection for a volume group enables statistics collection for all the logical volumes in that volume group.

Overview of lvmstat output

The first report generated by **lvmstat** provides statistics from the time the system was booted. Each subsequent report covers the time since the previous report. All statistics are reported each time **lvmstat** runs. The report consists of a header row followed by a line of statistics for each logical partition or logical volume depending on the arguments specified.

Interpreting lvmstat output

The output for a volume group displays the I/O activity per logical volume. **lvmstat** displays the number of I/O requests (*iocnt*), the kilobytes read and written (*Kb_read* and *Kb_wrtn*, respectively) and the transferred data in KB/s (*Kbps*). If you request the information for a logical volume **lvmstat** displays it for each logical partition in the logical volume. If you have mirrored logical volumes, you receive statistics for each of the mirrors. The output is sorted in decreasing order on the *iocnt* column.

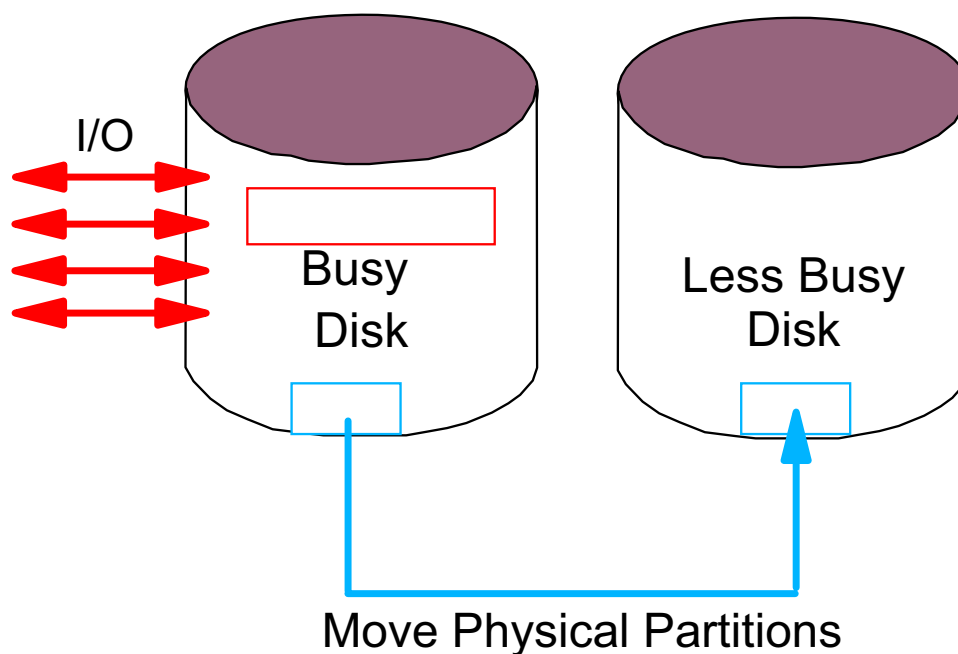
lvmstat syntax

```
lvmstat {-l|-v} Name [-e|-d] [-F] [-C] [-c Count] [-s] [Interval [Iterations]
```

The meaning of the arguments is as follows:

Argument	Description
-l <i>Name</i>	Specifies the name of a logical volume to work on.
-v <i>Name</i>	Specifies the name of a volume group to work on. You can also enable a volume group and selectively disable some logical volumes afterwards.
-e	Enables the gathering of statistics.
-d	Disables the gathering of statistics.
-F	Separates the output of the statistics by colons. This makes it easier for parsing by other scripts.
-C	Clears the counter for the specified logical volume or volume group. The first use of lvmstat , after enabling, displays the counter values since last reboot. Each usage thereafter displays the difference from the last call.
-c	Specifies how many lines from the top you want to display.
-s	Suppresses the header lines for subsequent outputs if you are using the interval and iteration arguments.
<i>Interval</i>	Indicates the amount of time in seconds between each report.
<i>Iterations</i>	Specifies the number of iterations to run. If no <i>iterations</i> argument is specified, lvmstat generates reports continuously.

Migrating Physical Partitions



```
# migratelp hd3/1 hdisk1/109
```

Figure 6-39. Migrating Physical Partitions

AU147.0

Notes:

Purpose of migratelp

With the output of the **lvmstat** command, it is easy to identify the logical partitions with the heaviest traffic. If you have several logical partitions with heavy usage on one physical disk and want to balance these across the available disks, you can use the new **migratelp** command to move these logical partitions to other physical disks.

Syntax

migratelp *lvname* /*lpartnum* [/*copynum*] *destpv* [/*ppartnum*]

Argument	Description
lvname	Specifies the name of the logical volume to work with.
lpartnum	Specifies the number of the logical partition as it is displayed in the lvostat output.
copynum	Specifies a mirror copy. This argument is optional. If not specified the first mirror copy is used.
destpv	Specifies the target physical volume for the move.
ppartnum	Specifies a target physical partition number. This argument is optional.

Example output

When **migratelp** completes successfully the output appears similar to this:

```
# migratelp hd3/1 hdisk1/109
migratelp: Mirror copy 1 of logical partition 1 of logical volume
hd3 migrated to physical partition 109 of hdisk1.
```

If the destination physical partition is already allocated **migratelp** will fail with an error message.

Striped logical volumes

The **migratelp** command will not work with partitions of striped logical volumes.

Documenting the Disk Storage Setup

- List of the disks on the system (PVID and VG)

lspv

- List the VGs

lsvg

- List what LVs are contained in each VG

lsvg -l *vgname*

- List the LVs on each disk

lspv -l *pvname*

Figure 6-40. Documenting the Disk Storage Setup

AU147.0

Notes:

Exercise: Logical Volume Manager



Figure 6-41. Exercise: Logical Volume Manager

AU147.0

Notes:

This lab will have you set up a new volume group and a new logical volume. You will use this volume group and logical volumes in future exercises.

The exercise can be found in your Exercise Guide.

Checkpoint

1. True or false? An LV can span more than one physical volume.
2. True or false? An LV can span more than one volume group.
3. True or false? The contents of a PV can be divided between two VGs.
4. True or false? If mirroring LVs, it is not necessary to perform a backup.
5. True or false? SMIT can be used to easily increase or decrease the size of a logical volume.
6. True or false? Striping is done at a logical partition level.

Figure 6-42. Checkpoint

AU147.0

Notes:

Unit Summary

- SMIT or high-level commands can be used to Add/Change/Delete volume groups, physical volumes and logical volumes
- Mirroring is a way to have two or three copies of a logical volume for high availability requirements
- Disk striping is used to provide high performance in large, sequentially accessed file systems

Notes:

Unit 7. Working with File Systems

What This Unit Is About

This unit provides a more in-depth discussion on the concepts and structure of AIX file systems.

What You Should Be Able to Do

After completing this unit, you should be able to:

- Identify the components of an AIX file
- Add an enhanced journaled file system
- Change characteristics of a file system

How You Will Check Your Progress

Accountability:

- Checkpoint questions
- Exercise
- Activity

References

Online	<i>System Management Concepts: Operating System and Devices</i>
Online	<i>System Management Guide: Operating System and Devices</i>
GG24-4484	<i>AIX Storage Management Guide</i>

Objectives

After completing this unit, students should be able to:

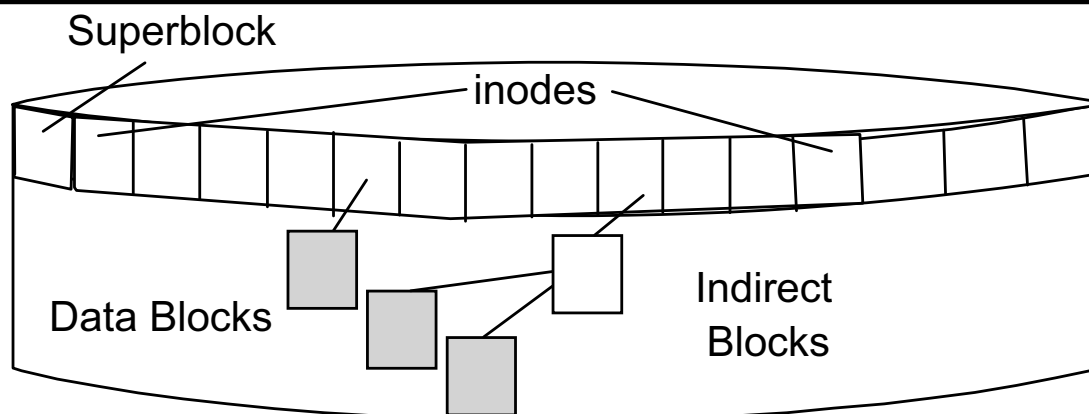
- Identify the components of an AIX file system
- Add an enhanced journaled file system
- Change characteristics of a file system

Figure 7-1. Objectives

AU147.0

Notes:

Structure of a Journalled File System



- **Superblock**

- File system size and identification
- Free list, fragment size, nbpi

- **inodes**

- File size, ownership, permissions, times
- Pointers to data blocks

- **Blocks**

- Data blocks - contain data
- Indirect blocks - contain pointers to data blocks

Figure 7-2. Structure of a Journalled File System

AU147.0

Notes:

AIX-journalled file systems are built within logical volumes. Because journalled file systems exist within logical volumes, the size of the file system will always be multiples of the logical partition size for that logical volume (for example, 4 MB).

An individual file within a file system will by default have units allocated to it in blocks of 4096 bytes. (This may change if you have implemented fragmentation or large files - to be discussed later.)

Some AIX commands often report file sizes in units of 512 bytes to remain compatible with other UNIX file systems. This is independent of the actual unit of allocation.

The first addressable logical block on the file system is the superblock. The superblock contains information such as the file system name, size, number of inodes, date/time of creation.

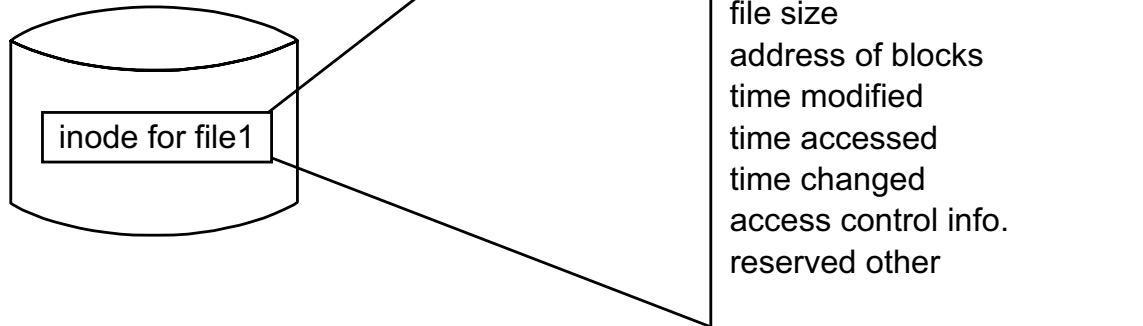
The superblock is critical to the file system and if corrupted will prevent the file system from mounting. For this reason a backup copy of the superblock is always written in block 31.

Immediately following the superblock are inodes which contain identifying information for files such as the file type, size, permissions, user/group/owner, create/modification and last access dates. They also contain pointers to the data block for fragment addresses which hold the data.

For larger files the system creates sets of indirect blocks filled with data block addresses to point to the data block or fragments which hold the data.

Structure of an Inode

- Contents of an inode



- This information can be seen with `ls -li`:

```
$ ls -li /home/team01
2132    drwxr - xr - x   2  team01 staff   512   May 2 14:33   c
2136    drwxr - xr - x   2  team01 staff   512   May 2 14:33  doc
2141    -rw-r - - r - -   1  team01 staff    28   May 16 10:11 Manuals
```

Figure 7-3. Structure of an Inode

AU147.0

Notes:

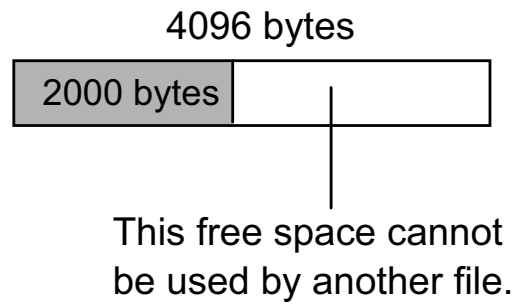
Each file is represented by a single inode. The inode contains information about that file such as:

- ownership
- access permissions
- type
- creation, modification and access times
- number of links to the file
- size
- addresses of data blocks on disk

File System Fragmentation

No Fragmentation

File size = 2000 bytes



Fragmentation Enabled

File size = 2000 bytes

Fragment size = 1024 bytes

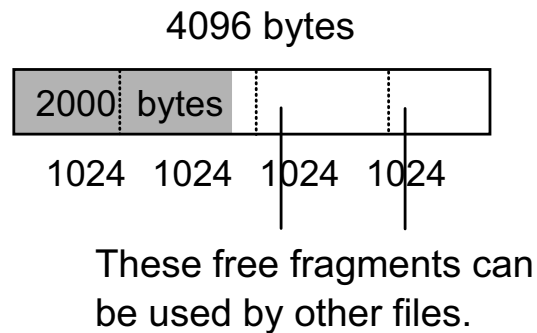


Figure 7-4. File System Fragmentation

AU147.0

Notes:

Fragmentation provides a way to allocate pieces (or fragments) of a 4 KB logical block to files and directories. Fragment support is helpful for small user files and directories. Fragment support applies to the last direct block of small user files and directories and long symbolic links. Fragment size is specified for a file system at creation time. The allowable fragment size for JFS file systems are 512, 1024, 2048 and 4096 bytes. The default fragment size is 4096 bytes.

Different file systems can have different fragment sizes, but only one fragment size can be used within a single file system. Different fragment sizes can also coexist on a single system so that administrators can select a fragment size which is most appropriate for each file system.

JFS fragment support provides a view of the file system as a contiguous series of fragments rather than logical disk blocks.

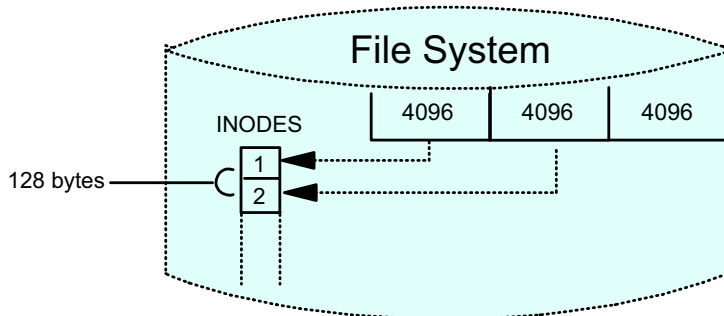
Both operational overhead (additional disk seeks and allocation activity) and better utilization of disk space increase as the fragment size for a file system decreases. In order

to maintain the optimum balance between increased overhead and increased usable disk space, the following factors apply to JFS fragment support:

- Disk space allocations of 4096 bytes of fragments are maintained for a file or directory's logical blocks where possible.
- Only partial logical blocks for files and directories less than 32 KB in size can be allocated less than 4096 bytes of fragments.

Variable Number of Inodes

With the default nbpi = 4096 an inode will be created for every 4096 bytes of file system.



Using the value nbpi = 1024 an inode will be created for every 1024 bytes of file system.

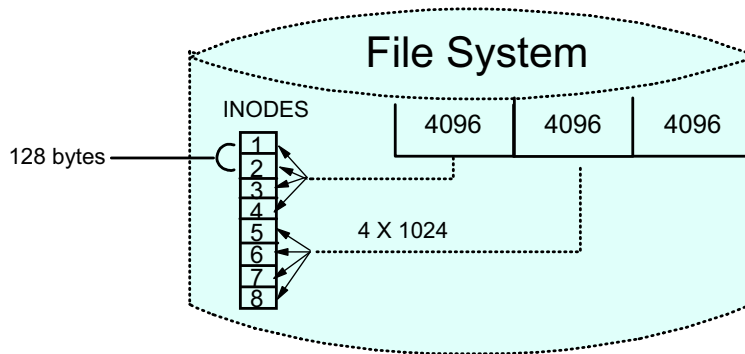


Figure 7-5. Variable Number of Inodes

AU147.0

Notes:

In all UNIX implementations, when a file system is created, inodes are written to disk. For each file or directory one such data structure is used which describes information pertaining to the file or directory. JFS also reserves a number of inodes for files and directories in each file system that is created.

In earlier versions of JFS, the number of inodes created for a file system was fixed. An inode was generated for every 4 KB of disk space that was allocated to the file system being created. In a 4 MB file system this would result in 1024 inodes being generated. As long as files and directories are allocated at a minimum of 4 KB, this would suffice.

However, since fragment support optimizes disk utilization, it increases the number of small files and directories that can be stored within a file system. Since each file or directory requires a disk inode, there needs to be a way to specify the number of inodes needed. JFS allows the number of disk inodes created within a file system to be specified in case more or less than the default number of disk inodes is desired. This number can be specified at file system creation as the number of bytes per inode (NBPI). For example, an

NBPI value of 1024 causes a disk inode to be created for every 1024 bytes of file system space. A small NBPI value results in a large number of inodes and vice versa.

The decision of fragment size and how many inodes to create for a file system should be based on the projected number of files contained by the file system and their size.

With JFS2 it is no longer necessary to project the number of files contained by the file system and their size. JFS2 dynamically allocates space for inodes as needed, and frees the space when it is no longer required.

Allocation Group Size

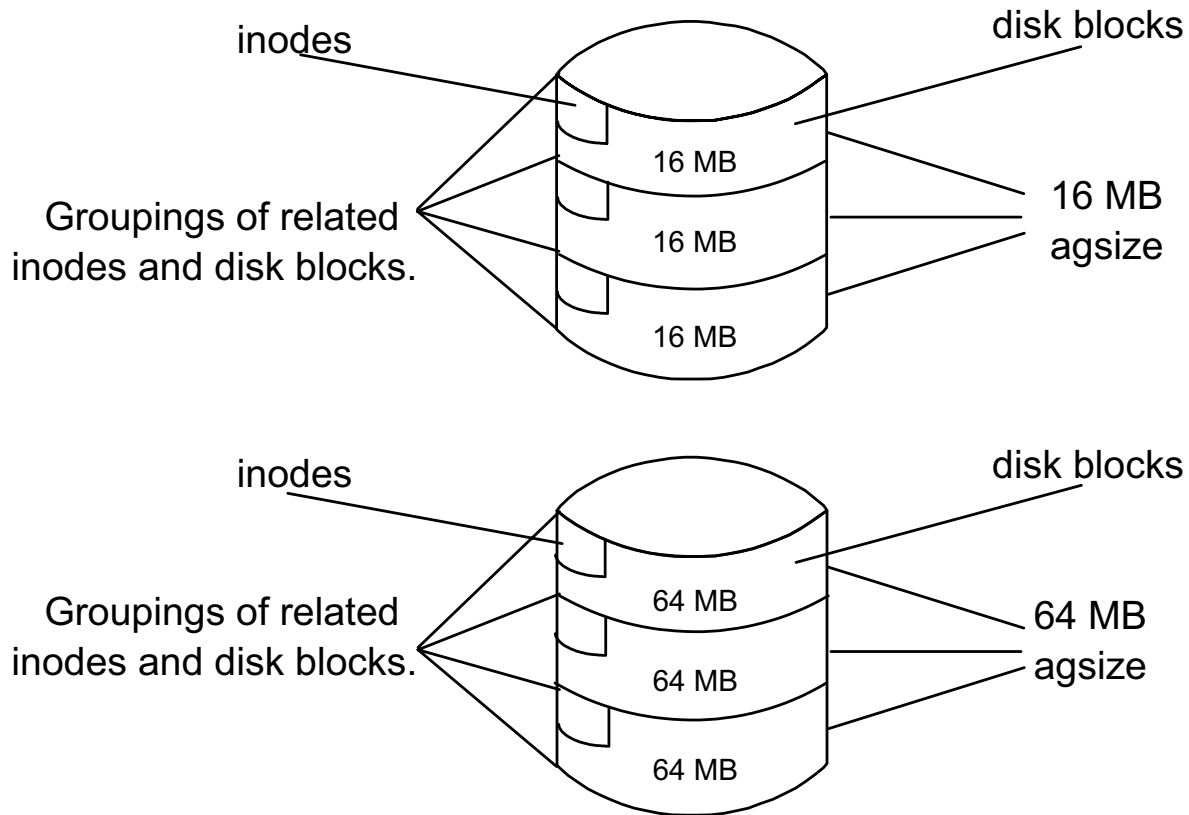


Figure 7-6. Allocation Group Size

AU147.0

Notes:

Allocation Group Size is supported by AIX V4.2 and later and is used to increase the efficiency of the file system. The inodes with the corresponding data blocks are further grouped in logical units of 8, 16, 32, or 64 MB within the file system. Building a relationship between the placement of the data blocks and related inode information reduces the physical action required by the drive heads when I/O operations are performed.

The allocation group size (AGS or agsize) value is a JFS configuration parameter which along with the NBPI and fragment size determine the overall characteristics of the file system.

The allowable set of NBPI values are also dependent on the allocation group size (agsize). For example, for an agsize value of 8MB the only allowable NBPI values are 512, 1024, 2048, 4096, 8192 and 16384 bytes. If you were to double the agsize from 8 MB to 16 MB the range of NBPI values also doubles to 1024, 2048, 4096, 8192, 16384 and 32768 bytes respectively.

Refer to the table for more details.

Allocation Group Size	NBPI
8 MB	512,1024,2048,4096,8192,16384
16 MB	1024,2048,4096,8192,16384,32768
32 MB	2048,4096,8192,16384,32768,65536
64 MB	4096,8192,16384,32768,65536,131072

JFS vs. JFS2 File Systems

	JFS	JFS2
Maximum File Size Architectural / Tested	64 Gigabytes / 64 Gigabytes	4 Petabytes / 1 Terabyte
Maximum File System Size Architectural / Tested	1 Terabyte / 1 Terabyte	4 Petabytes / 1 Terabyte
Inode size	128 Bytes	512 Bytes
Number of inodes	Fixed, set at creation	Dynamic
Fragments / Block Size	512 / 4096	512 / 4096
Online defragmentation	Yes	Yes
Available on Itanium	No	Yes
Available on POWER	Yes, default	Yes



JFS2 uses extent based allocation for **high performance** and **large file size**.

Figure 7-7. JFS vs. JFS2 File Systems

AU147.0

Notes:

Introduction to JFS2

Enhanced **J**ournaled **F**ile **S**ystem (JFS2) is a new file system type in AIX 5.1. It is based on JFS. JFS2 is the only standard file system for the Itanium-based platform. JFS and JFS2 are both available on the POWER platform.

Extent based allocation

JFS2 uses extent based allocation. An extent is an address-length pair, which identifies the starting block address and the length of the extent in blocks. This allows multiple adjacent blocks to be addressed. The advantages of extent based allocation are high performance and large file size.

Dynamic inodes

The traditional approach of reserving a fixed amount of space for inodes at file system creation time required accurate estimates of the number of files that would reside in the file system. If the estimate was high, disk space was wasted. If the estimate was low, no files could be added until the file system was expanded. JFS2 dynamically allocates space for inodes as needed, and frees the space when it is no longer required.

Migration

JFS file systems can co-exist on the same system with JFS2 file systems. However, to fully utilize the JFS2 features, the following steps will be necessary:

- Backup JFS file system data
- Create new JFS2 file systems
- Restore JFS file system data to new JFS2 file systems

File and File System Sizes

•Maximum Individual File Size Limits

AIX Version 4.1 (JFS)	2 GB
AIX Version 4.2 and later (JFS)	64 GB
AIX Version 5 (JFS2)	1 TB

•JFS File System Limit Calculation

Maximum file system size limit in bytes = fragment size * 2^{28}

or

Maximum file system size limit in bytes = nbpi * 2^{24}

Figure 7-8. File and File System Sizes

AU147.0

Notes:

The absolute maximum size of a file system in AIX V4.2 and later is 1 Terabyte. However, the actual maximum for JFS will be determined by that file system's NBPI and fragment size. The actual maximum will be determined when the file system is created by the smaller of (fragment size * 2^{28}) or (nbpi * 2^{24}).

The first calculation, (fragment size * 2^{28}) is based on the fact that 2^{28} , which is equal to 256 MB, is the maximum number of fragments that a file system can have.

The second calculation, (nbpi * 2^{24}) is based on the fact that 2^{24} , which is equal to 16 MB, is the maximum number of inodes that a file system can have.

The larger the NBPI and fragment size are, the larger the file system can be. Fortunately, when SMIT is used to create a file system, it will do the calculations for you to determine how large that file system can be.

For example, if a file system is created with a fragment size of 4096 bytes and an NBPI of 4096 bytes, the calculations would be:

$$(2^{12}) * (2^{28}) = (2^{40}) = 1 \text{ TB (Where } 2^{12} = 4096)$$

$$(2^{12}) * (2^{24}) = (2^{36}) = 64 \text{ GB}$$

The smaller of the two is going to be the limiting factor. Therefore, the answer is 64 GB and the number of inodes is the limiting factor.

Aside from this, there are many issues that have to be evaluated in order to implement an ideal file system. Existing systems offer historical data regarding file size, use and growth characteristics, but on a new system this data may not be readily available. Conscientious planning on the outset can save time and effort and possibly avoid costly file system conversion and migration procedures down the road.

Compressed File Systems

compression = LZ (yes)

fragment size = 1024

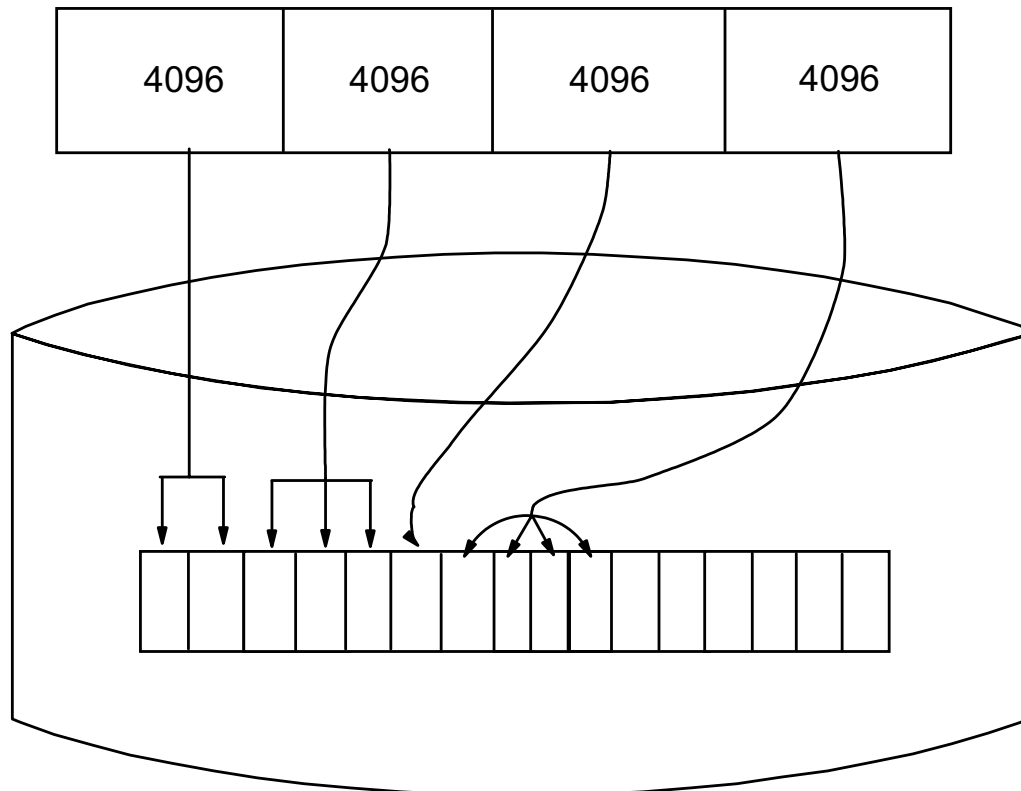


Figure 7-9. Compressed File Systems

AU147.0

Notes:

JFS supports fragmented and compressed file systems. Both types of file systems save disk space by allowing a logical block to be stored on the disk in units or fragments smaller than the full block size of 4096 bytes. In a fragmented file system only the last logical block of files no larger than 32 KB are stored in this manner, so that fragment support is only beneficial for the file systems containing numerous small files. Data compression however, allows all logical blocks of any-sized file to be stored as one or more contiguous fragments. On average, data compression saves disk space by about a factor of 2. JFS2 does not support file system compression.

The use of fragments and data compression does, however, increase the potential for fragmentation of the disk's free space. Fragments allocated to a logical block must be contiguous on the disk. A file system experiencing free space fragmentation may have difficulty locating enough contiguous fragments for a logical block's allocation, even though the total number of free fragments may exceed the logical block's requirements. JFS and JFS2 alleviate free space fragmentation by providing the **defragfs** utility which defragments a file system by increasing the amount of contiguous space. This utility can be used for fragmented and compressed file systems.

Warning: The root file system must not be compressed. Compression of the `/usr` file system is not recommended.

In addition to increased disk I/O activity and free space fragmentation problems, file systems using data compression have the following performance considerations:

- Degradation in file system usability arising as a direct result of the data compression/decompression activity. If the time to compress and decompress data is quite lengthy, it may not always be possible to use a compressed file system, particularly in a busy commercial environment where data needs to be available immediately.
- All logical blocks in a compressed file system, when modified for the first time, will be allocated 4096 bytes of disk space, and this space will subsequently be reallocated when the logical block is written to disk. Performance costs are, therefore, associated with this allocation, which does not occur in non-compressed file systems.
- In order to perform data compression, approximately 50 CPU cycles per byte are required and about 10 CPU cycles per byte for decompression. Data compression, therefore, places a load on the processor by increasing the number of processor cycles.

Large File Enabled File Systems

File = 132 MB

$$\begin{array}{rcl}
 (1024 * 4 \text{ KB blocks}) + & (1024 * 128 \text{ KB blocks}) & = & 132 \text{ MB} \\
 4 \text{ MB} & + & 128 \text{ MB} & = & 132 \text{ MB}
 \end{array}$$

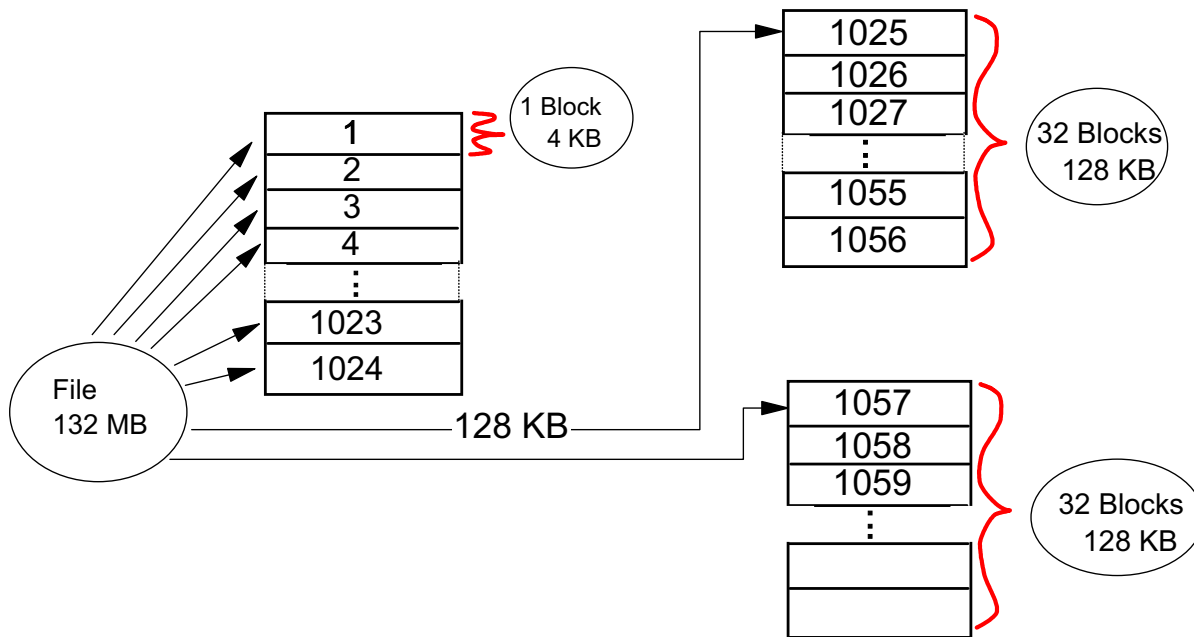


Figure 7-10. Large Enabled File Systems

AU147.0

Notes:

AIX V4.2 and later JFS supports large file enabled file systems. Only file systems enabled for large files can support files with a size greater than 2 GB.

In a file system enabled for large files, the data stored before the 4 MB file offset is allocated in 4096 byte blocks. File data stored beyond the 4 MB file offset is allocated with large disk blocks of 128 KB in size. The large disk blocks are actually 32 contiguous 4096 byte blocks. In the example above, a 132 MB file in a file system enabled for large files has 1024 4 KB disk blocks and 1024 128 KB disk blocks for a total of 2048 blocks.

In a regular standard file system the 132 MB file would require 33 single indirect blocks (each filled with 1024 4 KB disk addresses). However, the large file geometry requires only two single indirect blocks for the 132 MB file.

It is not necessary to use large enabled file systems in JFS2, since large file and file system support is built in by default.

Activity: Inodes and NBPI

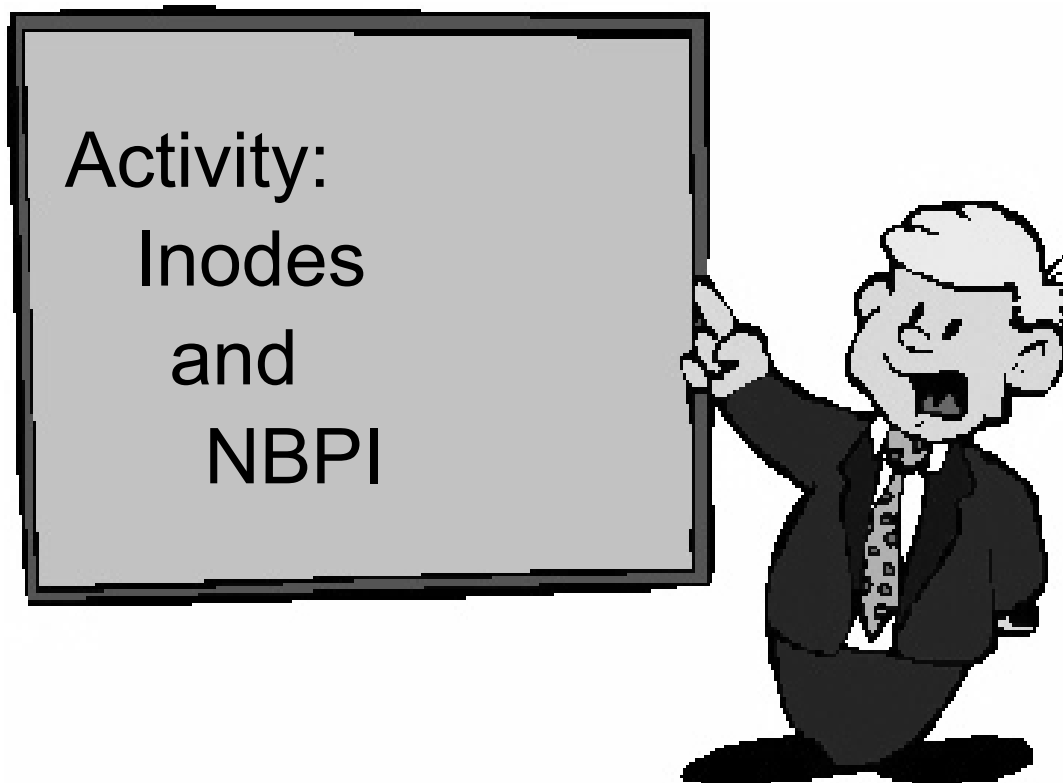


Figure 7-11. Activity: Inodes and NBPI

AU147.0

Activity

Complete the following questions regarding Fragmentation, NBPI and file system types.

1. If you are creating a JFS file system intended to store files that are smaller than 512 bytes in size, what would you to choose for the following values?

Fragment Size _____ NBPI _____

Regular or Large-file enabled _____

2. If you are creating a JFS file system intended to store files about 8 K in size, what would you choose for the following values?

Fragment Size _____ NBPI _____

Regular or Large-file enabled _____

3. If you are creating a JFS file system intended to store files about 2 MB in size, what would you choose for the following values?

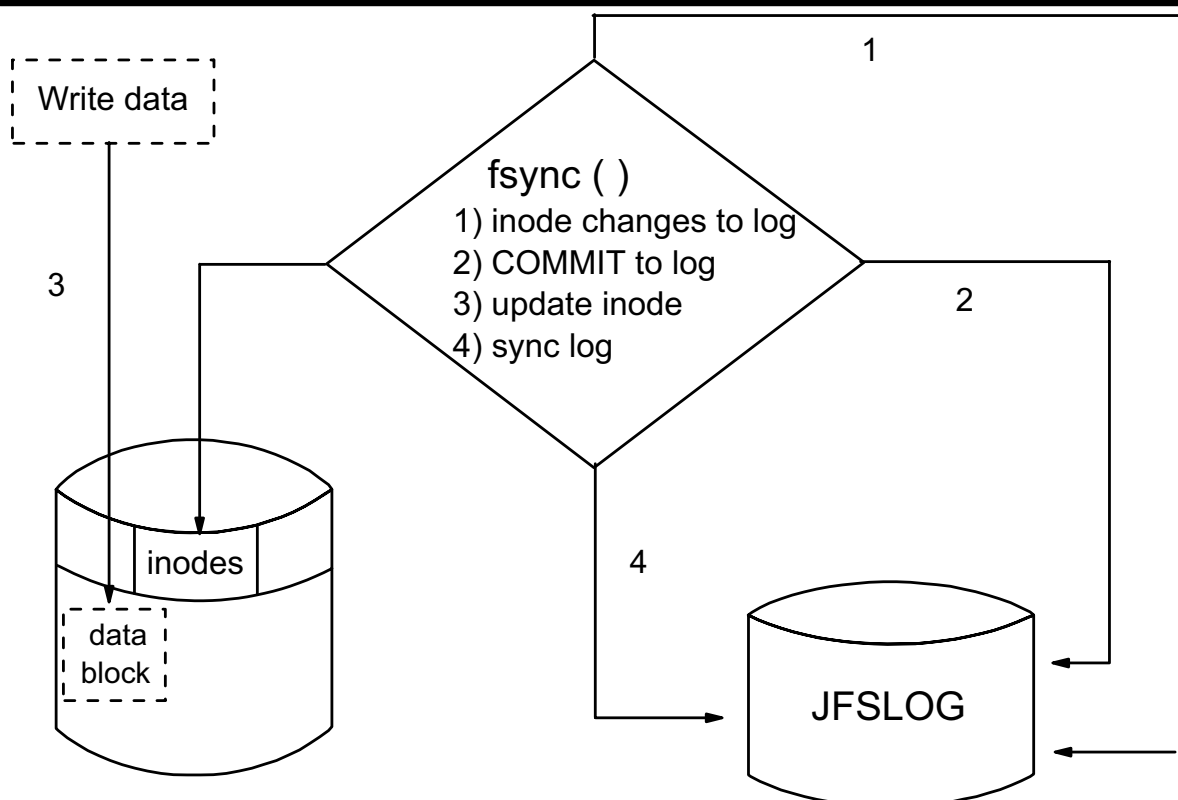
Fragment Size _____ NBPI _____

Regular or Large-file enabled _____

4. When should you use large file-enable file systems?

5. When should you use compressed file systems?

Journal Log



- No journaling of data blocks - only journals inode information (and indirect block information.)

Figure 7-12. Journaled Log

AU147.0

Notes:

AIX memory maps files in current use. Any writes to files are done first in memory and at a later stage are written out to disk when the **sync** system call runs - every minute.

The jfslog (**/dev/hd8**) is a circular log. It is created the size of one physical partition - one per each volume group. The jfslog ensures file system integrity by writing all metadata information to the jfslog immediately. It does this in the form of transactions as illustrated in the diagram. File system metadata consists of changes to the structure itself such as changes to the inodes and the free list.

When the data is written out to disk a sync point is indicated in the log and new transactions are written from that point forward.

By default, a single logical volume per volume group is used to contain the file system journal logs. When you create a new file system, the journal will be added to the existing journal log logical volume. With default log logical volumes, the entire volume group depends on a single log logical volume.

User-created logs override the default log placement and put the file system log on a specific logical volume.

An **inline** log is a new feature specific to JFS2 file systems that allows you to log directly to the file system. The default inline log size is 0.4 % of the logical volume size.

The following table lists the three logging options and which file system type supports them.

Option	JFS	JFS2
Default volume group log	Yes	Yes
Specific user-created log	Yes	Yes
Log directly to the file system	No	Yes

File Systems

smit fs

File Systems

Move cursor to desired item and press Enter

- List All File Systems
- List All Mounted File Systems
- Add/Change/Show/Delete File Systems
- Mount a File System
- Mount a Group of File Systems
- Unmount a File System
- Unmount a Group of File Systems
- Verify a File System
- Backup a File System
- Restore a File System
- List Contents of a Backup

Figure 7-13. File Systems

AU147.0

Notes:

File systems can also be managed using the Web-based System Manager.

Listing File Systems

lsfs

Name	Nodename	Mount Pt	VFS	Size	Options	Auto
/dev/hd4	—	/	jfs	16384	—	yes
/dev/hd1	—	/home	jfs2	90112	—	yes
/dev/hd2	—	/usr	jfs	1277952	—	yes
/dev/hd9var	—	/var	jfs	8192	—	yes
/dev/hd3	—	/tmp	jfs	24576	—	yes
/proc	—	/proc	procfs		ro	yes
/dev/hd10opt	—	/opt	jfs	—	—	yes
/budget	sys4	/reports	nfs	—	bg,hard,intr	

Figure 7-14. Listing File Systems

AU147.0

Notes:

You can list the various file systems that are defined using the **lsfs** command. This command will display information from **/etc/filesystems** and from the logical volumes in a more readable format.

lsfs will also display information about CD-ROM file systems and remote NFS file systems.

lsfs [-q] [-c | -l] [-v vfstype | -u mountgrp | file system]

The data may be presented in line and colon (**-c**) or stanza (**-l**) format. It is possible to list only the file systems of a particular virtual file system type (**-v**), or within a particular mount group (**-u**). The **-q** option queries the superblock for the fragment size information, compression algorithm, and the number of bytes per inode.

The SMIT fastpath to get to the screen which will accomplish the same task as the **lsfs** command is **smit fs**.

List All Mounted File Systems

mount

<u>node</u>	<u>mounted</u>	<u>mounted over</u>	<u>vfs</u>	<u>date</u>	<u>options</u>
	/dev/hd4	/	jfs	Jul 11 20:14	rw,log=/dev/hd8
	/dev/hd2	/usr	jfs	Jul 11 20:15	rw,log=/dev/hd8
	/dev/hd9var	/var	jfs	Jul 11 20:15	rw,log=/dev/hd8
	/dev/hd3	/tmp	jfs	Jul 11 20:15	rw,log=/dev/hd8
	/dev/hd1	/home	jfs2	Jul 11 20:16	rw,log=/dev/lv02
sys4	/budget	/reports	nfs	Jul 11 20:16	rw,hard,bg,intr

Figure 7-15. List All Mounted File Systems

AU147.0

Notes:

The **mount** command, when used with no parameters, is used to list all the file systems which are currently mounted within the overall file system structure.

File systems must be mounted to be accessed, that is, make the file system available for read or write access from your system.

The **mount** command when used with a number of parameters, is also used to perform the mount operation.

There are two types of file systems, system-created and user-created. System-created file systems are expected to be there by the system and by many applications. User-created file systems contain user applications and data.

Standard device names include:

- hd4 /
- hd1 /home
- hd2 /usr

- hd3 /tmp
- hd9var /var

SMIT can also be used to obtain this information. From SMIT you want to select **List all Mounted File Systems** under **File Systems**.

Working with Journalled File Systems in SMIT

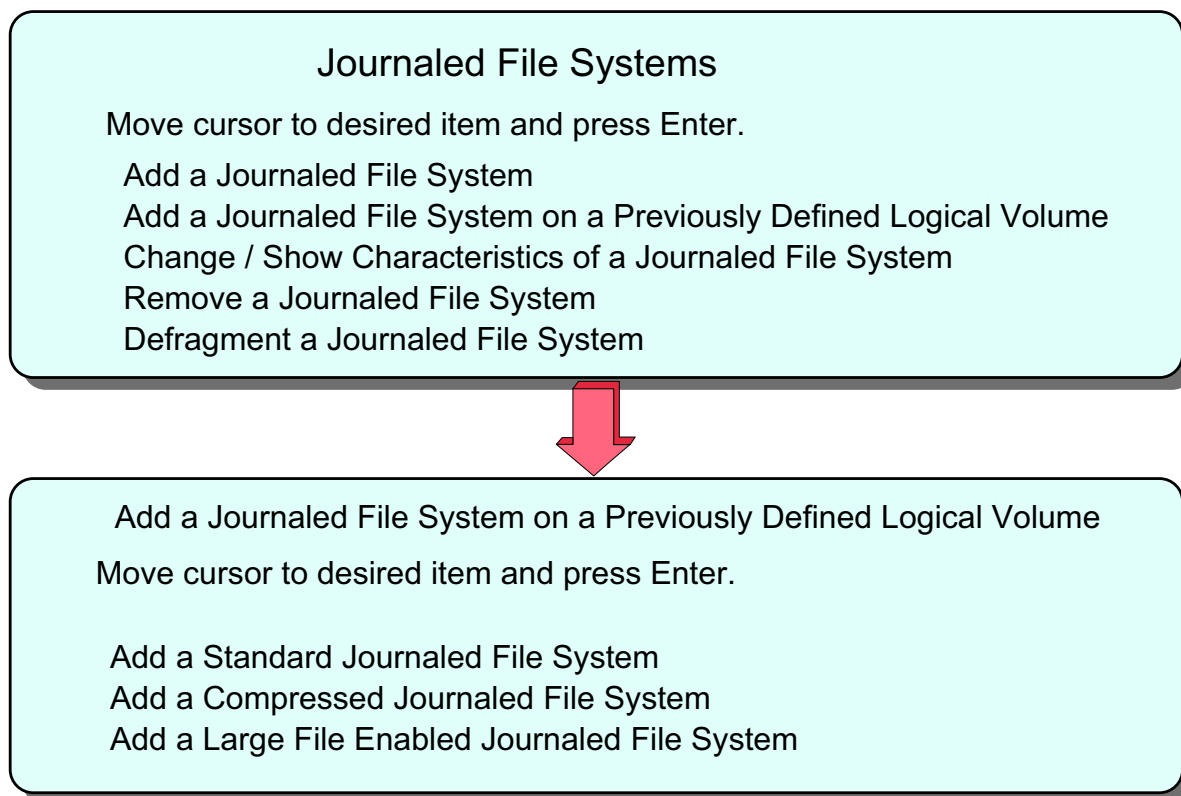


Figure 7-16. Working with Journalled Files Systems in SMIT

AU147.0

Notes:

When choosing to add a JFS or JFS2 file system, there are two options.

- If you choose to **Add a Journalled File System** or **Add an Enhanced Journalled File System**, SMIT will use defaults to create the logical volume in which the file system sits.
- If you choose to **Add a Journalled File System on a Previously Defined Logical Volume** or **Add an Enhanced Journalled File System on a Previously Defined Logical Volume**, this assumes that the logical volume has already been created according to your specifications. The size of the file system will be the size of the logical volume.

The diagram shows the SMIT menu displayed if using the **smit jfs** fastpath. The SMIT menu displayed if using the **smit jfs2** fastpath is identical except that the options are for Enhanced Journalled File Systems (JFS2). Also, if using JFS2, the menu on the bottom of the visual will not be displayed.

In AIX V4.2 and later, the second SMIT menu above will be displayed no matter which option is chosen for adding a JFS file system.

Add a Standard Journalled File System on a Previously Defined Logical Volume

Add a Standard Journalled File System

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

	[Entry Fields]	
* LOGICAL VOLUME name		+
* MOUNT POINT	[]	
Mount AUTOMATICALLY at system restart?	no	+
PERMISSIONS	read/write	+
Mount OPTIONS	[]	+
Start Disk Accounting ?	no	+
Fragment Size (bytes)	4096	+
Number of bytes per inode	4096	+
Allocation Group Size (MBytes)	8	+

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

Figure 7-17. Add a Standard Journalled File System on a Previously Defined Logical Volume

AU147.0

Notes:

When a logical volume is created it is simply an empty container waiting to be formatted for use. The journalled file system is the most common way of using it. Thus, adding a file system to a previously created logical volume formats the logical volume for use as a file system. Adding a file system in this way provides you with the greatest level of control over where the file system will reside on disk.

The SMIT fastpath for this screen is **smit crjfslvstd**.

Add a Standard Journalled File System

Add a Standard Journalled File System

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

	[Entry Fields]	
Volume group name	rootvg	+
* SIZE of file system (in 512-byte blocks)	[]	#
* MOUNT POINT	[]	
Mount AUTOMATICALLY at system restart?	no	+
PERMISSIONS	read/write	+
Mount OPTIONS	[]	+
Start Disk Accounting ?	no	+
Fragment Size (bytes)	4096	+
Number of bytes per inode	4096	+
Allocation Group Size (MBytes)	8	+

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

Figure 7-18. Add a Standard Journalled File System

AU147.0

Notes:

Use the smit fastpath **smit crjfsstd** to access this menu.

The **crfs** command is the high-level command to create a file system.

Note: Do not confuse the **crfs** command with the **mkfs** command which purely builds the file system structure within a logical volume. **crfs** does a lot more: It creates the logical volume if necessary using **mklv**, it builds the file system structure on that logical volume using **mkfs**, and then it makes all appropriate changes to the ODM and **/etc/filesystems** for that logical volume and file system.

There are many parameters which can be set as a JFS file system is being created. The most important of these are given below:

Volume group (**-g volgrp**); i.e., the volume group within which a new logical volume is to be created. The volume group must have sufficient free physical partitions for the new logical volume.

The size (**-a size=size**) in 512 byte blocks. The size of the file system will be rounded up to the nearest logical partition boundary. This attribute specifies the minimum file system and can't be decreased dynamically after the file system has been successfully created.

The mount point (**-m mntpt**). The name of a directory within the overall file system on which the new file system will normally be mounted. The mount point must exist before the file system can be mounted and accessed. Under most circumstances the mount point should be empty.

A file system may be mounted at any other valid directory rather than its normal mount point. In this case, the mount is performed by the administrator, and it is usually for some type of maintenance activity.

Mount automatically at boot time? (**-A yes|no**). The new file system may be listed to mount automatically when the system boots. This will place the **mount=true** line in the **/etc/filesystems** file and will cause the file system to be mounted automatically at its default mount point (above) when the system is restarted. If set to no then **mount = false** is added to the **/etc/filesystems** file.

Permissions (**-p rw|ro**). A mounted file system may be mounted in read-only (**ro**) or read-write (**rw**) mode. This permission setting will be used for the file system if it is mounted automatically, or if it is mounted without providing over-riding permissions.

The permissions setting for a mounted file system may not be by-passed regardless of the authority of the user and the permission bits associated with the file or directory on the file system.

Mount options specify security related mount options. Possible values are: **nosuid** which prevents the execution of setuid and setgid programs and **nodev**, which prevents open system calls of devices from this mount.

The fragment size (**-a fragment=size**) specifies the JFS fragment size in bytes. A file system fragment is the smallest unit of disk storage that can be allocated to a file. This variable must be set to either 512, 1024, 2048 or 4096, the default value being 4096 bytes.

The number of bytes per inode (**-a nbpi=value**) affects the total number of inodes on the file system. The variable must be either 512, 1024, 2048, 4096, 8192 or 16384, default value being 4096.

The compression attribute (**-a compress={no | LZ}**) specifies the data compression algorithm LZ, which stands for the IBM version of the compression algorithm Lempel-Ziv. If you do not want data compression, set this attribute value to no, which is the default value.

The allocation group size (**-a ag= 8 | 16 | 32 | 64**) is a grouping of inodes and disk blocks within the file system. The default agsize is 8 MB. This attribute only applies to AIX V4.2 and later.

Add a JFS2 File System on a Previously Defined Logical Volume

Add an Enhanced Journaled File System

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

	[Entry Fields]	
* LOGICAL VOLUME name		+
* MOUNT POINT	[]	
Mount AUTOMATICALLY at system restart?	no	+
PERMISSIONS	read/write	+
Mount OPTIONS	[]	+
Block size (bytes)	4096	+
Inline Log?	no	+
Inline Log size (MBytes)	[]	#

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

Figure 7-19. Add a JFS2 File System on a Previously Defined Logical Volume

AU147.0

Notes:

The SMIT fastpath for this screen is **crjfs2lvstd**.

The block size parameter refers to the aggregate block size, which is the smallest piece of disk which can be assigned to a file system. It has the same function as the fragment size in JFS.

If inline log is set to yes, then the journal log is placed within the file system structure. The inline log size can also be set through this menu.

Add an Enhanced Journaled File System

Add an Enhanced Journaled File System

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

	[Entry Fields]	
Volume group name	rootvg	+
* SIZE of file system (in 512-byte blocks)	[]	#
* MOUNT POINT	[]	
Mount AUTOMATICALLY at system restart?	no	+
PERMISSIONS	read/write	+
Mount OPTIONS	[]	+
Block size (bytes)	4096	+
Inline Log?	no	+
Inline Log size (MBytes)	[]	#

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

Figure 7-20. Add an Enhanced Journaled File System

AU147.0

Notes:

Mount a File System

Mount a File System

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

	[Entry Fields]	
FILE SYSTEM name	[]	+
DIRECTORY over which to mount	[]	+
TYPE of file system		+
FORCE the mount?	no	+
REMOTE NODE containing the file system to mount	[]	
Mount as a REMOVABLE file system?	no	+
Mount as a READ-ONLY system?	no	+
Disallow DEVICE access via this mount?	no	+
Disallow execution of SUID and sgid programs in this file system?	no	+

Figure 7-21. Mount a File System

AU147.0

Notes:

The files within a file system can only be accessed when the file system is mounted within the overall file system structure. Either an individual file system or a group of file systems can be mounted.

File systems defined with the **mount=true** or **mount=automatic** attribute in the **/etc/filesystems** file will be mounted automatically during system startup.

The syntax of the **mount** and **umount** commands are:

mount [-t Type | Device | Node: Directory]Directory

mount /home/george/myfs

umount|unmount [FileSystem | -t Type]

unmount /home/george/myfs

File systems are usually mounted at startup and are unmounted as part of the **shutdown** procedure.

However, the administrator or members of the security group may issue **mount** commands at any time, and assuming that the user has write permission to the mount point and read permission on the root directory of the file system to be mounted, the command will be effective.

Normal users: can mount a file system provided they belong to the system group and have write access to the mount point

root: can mount anywhere under any set of permissions

The **mount** command has many options which may be issued by the user. The default values for these options are set by the system, or are contained within **/etc/filesystems**.

With the **unmount** command there are many other options that can be used other than **-t** Type or file system name.

Change/Show Characteristics of a Journaled File System

Change/Show Characteristics of a Journaled File System

Type or select values in entry fields.

Press Enter AFTER making all desired changes.

	[Entry Fields]	
File system name	/var	
NEW mount point	[/var]	
SIZE of file system (in 512-byte blocks)	[8192]	
Mount GROUP	[]	
Mount AUTOMATICALLY at system restart ?	yes	+
PERMISSIONS	read/write	+
MOUNT OPTIONS	[]	+
Start Disk Accounting ?	no	+
Fragment Size (bytes)	512	
Number of bytes per inode	4096	
Compression algorithm	no	
Large File Enabled	false	
Allocation Group Size (MBytes)	4	

Figure 7-22. Change/Show Characteristics of a Journaled File System

AU147.0

Notes:

A Journaled File System may have some of its characteristics changed both while it is in use (mounted) and when it is not in use. To do this, use the **chfs** command.

Many characteristics may be changed. The most important of these are: The mount point (**-m mntpnt**). The default mount point may be changed while the file system is in use but the change is only effective when the file system is next mounted. The size of a Journaled File System may be **increased** while it is in use (**-a size=newsize**). The size of a file system may not be decreased at any time, so it is often better to create a new file system and mount it at an appropriate point within the existing file system than to increase the size if it is suspected that the increased size is only temporarily required.

Increasing the size of the file system extends the logical volume, so the new size will be rounded up to the next logical partition boundary. If you extend the logical volume directly, the partitions will be added, but the file system is not changed. Extending the file system will use those added partitions. The mount group of a file system may be changed (**-u mntgrp**), and the change will be effective the next time the new mount group is referenced.

Mount automatically a system restart? Whether a file system is automatically mounted at system startup may be changed (**-A yes|no**) and the change will be effective at the next startup.

The permissions associated with the file system may be changed (**-p ro|rw**) and the change will be effective the next time the file system is mounted.

Change/Show Characteristics of an Enhanced Journaled File System

Change/Show Characteristics of an Enhanced Journaled File System

Type or select values in entry fields.

Press Enter AFTER making all desired changes.

	[Entry Fields]	
File system name	/home	
NEW mount point	[/home]	
SIZE of file system (in 512-byte blocks)	[32768]	
Mount GROUP	[]	
Mount AUTOMATICALLY at system restart ?	yes	+
PERMISSIONS	read/write	+
MOUNT OPTIONS	[]	+
Block size (bytes)	4096	+
Inline Log?	no	+
Inline Log size (MBytes)	[]	#

Figure 7-23. Change/Show Characteristics of an Enhanced Journaled File System

AU147.0

Notes:

An Enhanced Journaled File System (JFS2) may have some of its characteristics changed both while it is in use (mounted) and when it is not in use. To do this, use the **chfs** command.

Many characteristics may be changed. See the notes with the last visual on changing the characteristics of Journaled File Systems.

Remove a Journalled File System

Remove a Journalled File System

Type or select values in entry fields.

Press Enter AFTER making all desired changes.

[Entry Fields]

FILE SYSTEM name		+
Remove mount point	no	+

Figure 7-24. Remove a Journalled File System

AU147.0

Notes:

The **rmfs** command is the high-level command used to remove a file system.

In order to remove a file system, it must be unmounted from the overall file tree, and this cannot be done if the file system is in use, that is, some user or process is using the file system or has it as a current directory.

rmfs will remove any information for the file system from the ODM and **/etc/filesystems**. When the file system is removed, the logical volume on which it resides is also removed.

JFS2 file system removal works the same way.

System Storage Review

Logical Volume Structure

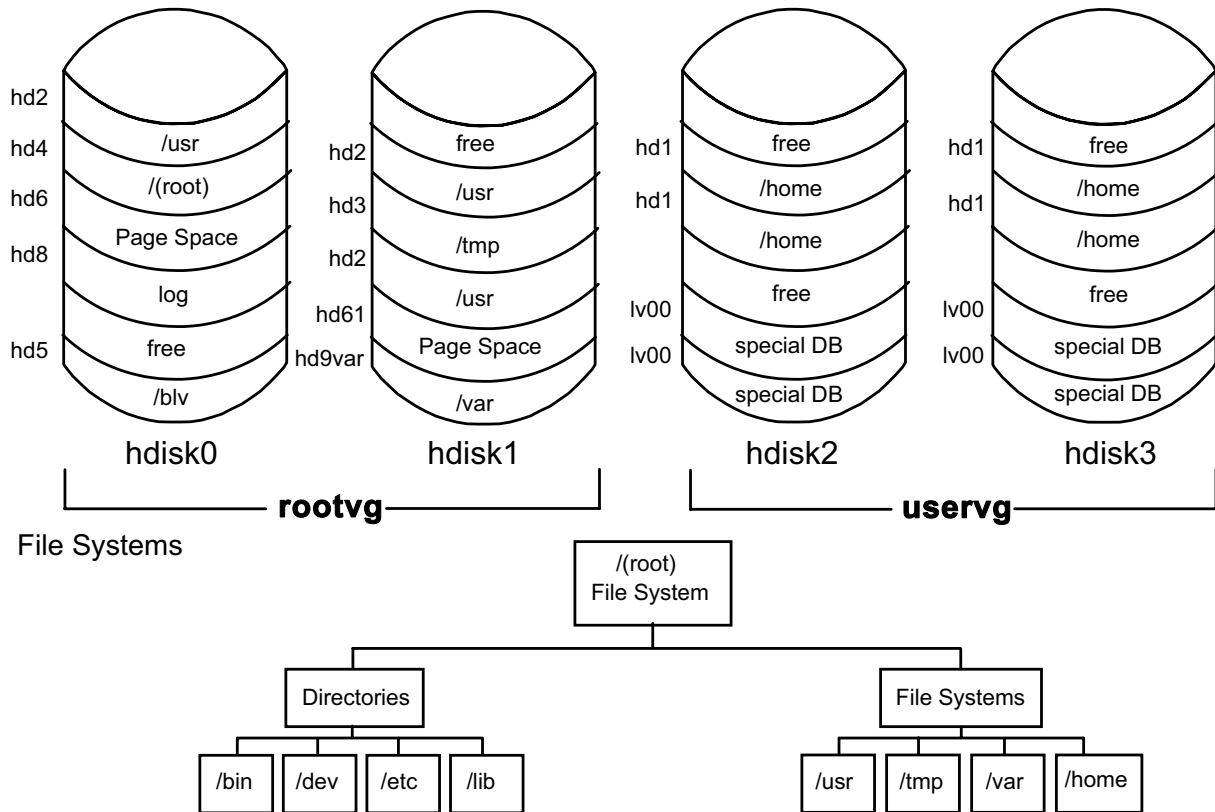


Figure 7-25. System Storage Review

AU147.0

Notes:

It is important to understand the difference between a file system and a directory. A file system is a section of disk that has been allocated to contain files. This section of disk is the logical volume. The section of disk is accessed by mounting the file system over a directory. Once the file system is mounted, it looks like any other directory structure to the user.

The directories on the right of the bottom portion of the visual are all file systems. These file systems are all mounted on the directories `/usr`, `/tmp`, `/var` and `/home`. Notice the corresponding logical volume in the graphic at the top of the visual.

The directories on the left of the bottom portion of the visual are strictly directories that contain files and are part of the `/(root)` file system.

Exercise: Working with File Systems



Figure 7-26. Exercise: Working with File Systems

AU147.0

Notes:

This lab will have you build on the logical volume you created in the last exercise. It will also give you an opportunity to create a file system and learn to increase the size of both the logical volume and file system.

The exercise can be found in your Exercise Guide.

Checkpoint

1. Will the size of the file system change when the size of the logical volume it is on is increased?

2. If a file system is the same size as the logical volume on which it sits, will the size of the logical volume increase when the size of the file system that is sitting on it increases?

3. If you remove a logical volume, will the file system that is sitting on it be removed as well?

Figure 7-27. Checkpoint

AU147.0

Notes:

Unit Summary

- The components of an AIX file system are the superblock, inodes, data blocks and indirect blocks
- Important issues to consider when creating a journaled file system are: fragment size, NBPI, allocation group size, compression and whether it should be large file enabled
- JFS2 supports large files, large file systems, and improves performance
- File systems can be added and removed from the system, and their characteristics can also be changed, all through SMIT

Notes:

Unit 8. Managing File Systems

What This Unit Is About

This unit illustrates the methods that can be used to manage the AIX file systems.

What You Should Be Able to Do

After completing this unit, you should be able to:

- Monitor file system growth and control growing files
- Manage file system disk space usage
- Implement basic file system integrity checks

How You Will Check Your Progress

Accountability:

- Checkpoint questions
- Exercise

References

Online *System Management Guide: Operating System and Devices*

Objectives

After completing this unit, students should be able to:

- Monitor file system growth and control growing files
- Manage file system disk space usage
- Implement basic file system integrity checks

Figure 8-1. Objectives

AU147.0

Notes:

Space Management

- File systems expand upon notice, NOT automatically
- To keep from running into problems:
 - Monitor file system growth
 - Determine causes
 - Control growing files
 - Manage file system space usage
 - Control user disk usage
 - Defragment file system

Figure 8-2. Space Management

AU147.0

Notes:

Although AIX provides for dynamic expansion of a file system, it does not expand the file system on the fly. The system administrator must continually monitor the file system growth to expand it before the file system gets full. If the file system becomes 100% full then the users will receive "out of space" messages on file system expansion.

For example, the **df** command can be run via **cron** (the job scheduler) to perform a regular check of the space available in the file system and produce a report. **cron** will be covered in a later unit.

Listing Free Disk Space

- The **df** command displays information about total space and available space on a file system

```
# df
```

Filesystem	512-blocks	Free	%Used	lused	%lused	Mounted on
/dev/hd4	16384	7600	53%	1243	30%	/
/dev/hd2	1630208	101648	93%	22217	10%	/usr
/dev/hd9var	24576	22360	9%	257	8%	/var
/dev/hd3	24576	21520	12%	144	3%	/tmp
/dev/hd1	24576	5160	79%	518	16%	/home
/dev/lv00	24576	3172	86%	620	22%	/home/john

Figure 8-3. Listing Free Disk Space

AU147.0

Notes:

This is an important command to be aware of and use frequently. If you run out of space in a file system (especially / or **/tmp**), system corruption could occur.

The **df** command lists the free space on all mounted file systems.

df -I shows space used

df -k reports free space in 1KB blocks

Control Growing Files

- ▶ /var/adm/wtmp
- ▶ /var/spool/*/*
- ▶ \$HOME/smit.log
- ▶ \$HOME/smit.script
- ▶ \$HOME/websm.log
- ▶ \$HOME/websm.script
- ▶ /etc/security/failedlogin
- ▶ /var/adm/sulog

Figure 8-4. Control Growing Files

AU147.0

Notes:

Growing files should be monitored and cleaned out periodically. These are some of the files that grow.

If accounting is turned on, **/var/adm/wtmp** will be kept to a reasonable size.

/var/adm/wtmp, **/etc/security/failedlogin** and **/var/adm/sulog** are needed because they contain historical data regarding login activity. Thus, these files should always have a few days worth of login activity kept in them. If accounting is not turned on, to capture the data to archive it, use **who -a** on **/var/adm/wtmp** and **/etc/security/failedlogin** and redirect the output to a save file. Then the file can be purged by using **cat /dev/null** redirected to either **/var/adm/wtmp** or **/etc/security/failedlogin**. **/var/adm/sulog** can be edited directly.

/var/spool contains cron entries, the mail, and other items that grow on an ongoing basis, along with printer files. If there is a problem you can try to clear the queueing subsystem by executing the following commands:

1. `stopsrc -s qdaemon`
2. `rm /var/spool/lpd/qdir/*`

3. `rm /var/spool/lpd/stat/*`
4. `rm /var/spool/qdaemon/*`
5. `startsrc -s qdaemon`

skulker

- The **skulker** command cleans up file systems by removing unwanted or obsolete files
- Candidate files include (can use file aging as criteria):
 - those in /tmp directory
 - a.out file
 - core files
 - ed.hup files
- **skulker** is normally invoked daily by the **cron** command as part of the root's crontab file
- Modify the skulker shell script to suit local needs for the removal of files

Figure 8-5. skulker

AU147.0

Notes:

The following is an example of the types of entries that are in the **/usr/sbin/skulker** program. To analyze the commands that are executed for each type of entry, print out or view the contents of the **/usr/sbin/skulker** file. Use care if modifying the file!

Removes all:

- old primary.output that got lost
- old **qdir** files
- files that are left in the mail queues
- files in **/tmp** older than 24 hours and not accessed or modified in the past 24 hours
- files in **/var/tmp**
- news items older than 45 days
- ***.bak, *.bak, a.out, core, proof, galley, ed.hup** files that are more than one day old
- anything in a **.putdir** directory more than a day old

Listing Disk Usage

- The **du** command can be used to list the number of blocks used by a file or a directory

```
# du /home | sort -r -n
```

```
624 /home
392 /home/fred
98 /home/tom
54 /home/mary
52 /home/liz
23 /home/suzy
2 /home/guest
1 /home/steve
```

- To view individual file sizes, use the **ls -l** command

Figure 8-6. Listing Disk Usage

AU147.0

Notes:

There may be a number of files or users that are causing the increase of use in a particular file system. The **du** command will help to determine the cause.

du gives information in 512-byte blocks, by directory. Use the **-k** option to display sizes in 1 KB units. With the **-a** option, output is displayed by file rather than directory. If used with **sort** on the first column in descending order, it can be an aid in determining which files/directories are the largest. Then using an **ls -l**, you can determine the file/directory's owner.

The **-x** option is also very useful. When using **du -ax**, the report will only show information from the specified file system. This is the best way to determine what file is filling a particular file system.

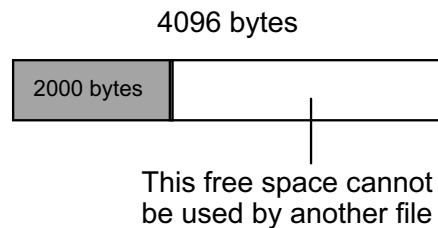
The **find** command is useful to locate files that are over a certain size. For example, to find all files that have greater than 1,000,000 characters and then list them use:

```
# find . -size +1000000c -exec ls -l {} \;
```

Fragmentation Considerations

Without fragmentation

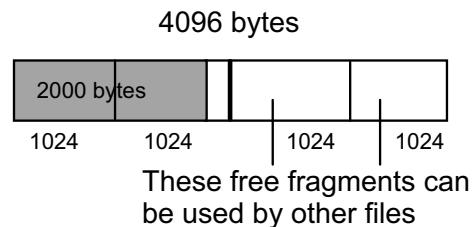
File size = 2000 bytes



With fragmentation

File size = 2000 bytes

Fragment size = 1024 bytes



Considerations to be made:

- Disk space allocation
- Disk space utilization
- I/O activity
- Free space fragmentation
- Fragment allocation map

Figure 8-7. Fragmentation Considerations

AU147.0

Notes:

In JFS, as many whole fragments as necessary are used to store a file or directory's data. Consider that we have chosen to use a JFS fragment size of 4 KB and we are attempting to store file data which only partially fills a JFS fragment. Potentially, the amount of unused or wasted space in the partially filled fragment can be quite high. For example, if only 500 bytes are stored in this fragment then 3596 bytes will be wasted. However, if a smaller JFS fragment size, say 512 bytes was used, the amount of wasted disk space would be greatly reduced to only 12 bytes. It is, therefore, better to use small fragment sizes if efficient use of available disk space is required.

Although small fragment sizes can be beneficial in reducing wasted disk space, this can have an adverse effect on disk I/O activity. For a file with a size of 4 KB stored in a single fragment of 4 KB, only one disk I/O operation would be required to either read or write the file. If the choice of the fragment size was 512 bytes, a 4 KB file would only be allocated a 4 KB block if one were available. If a single 4 KB block were not available, 512 byte fragments would be used, with a potential to allocate eight fragments for this file. If fragments are used, for a read or write to complete, several additional disk I/O operations (disk seeks, data transfers and allocation activity) would be required. Therefore, for file

systems which use a fragment size of 4 KB, the number of disk I/O operations will be far less than for file systems which employ a smaller fragment size.

For file systems with a fragment size smaller than 4 KB, there is likely to be an increase in allocation activity when the size of existing files or directories are extended.

Free space fragmentation can occur much more within a file system that uses smaller fragment sizes.

The fragment allocation map, used to hold information about the state of each fragment for each file system, is held on the disk and in virtual memory. The use of smaller fragment sizes in file systems results in an increase in the length of these maps and therefore requires more resources to hold.

In JFS2 the block size has a similar function to the JFS fragment size. The default is 4096 and can be altered by the system administrator.

Defragmenting a File System

- The **defragfs** command increases a file system's contiguous free space
- The file system must be mounted

```
/usr/sbin/defragfs [-q | -r] filesystem
```

Options:

- q** Reports the current state of the file system
- r** Reports the current state of the file system and the state that would result if the **defragfs** command is run without either **-q** or **-r**

Figure 8-8. Defragmenting a File System

AU147.0

Notes:

Some of the information that is returned with the **defragfs** command is:

- Number of Fragments Moved: Tells you how many data blocks need to be moved.
- Number of Logical Blocks Moved: Tells how many non-contiguous blocks that are in the system currently and how many that will be relocated, if possible.
- Number of Allocation Attempts: This is the required number of calls to the allocation routine to defragment the file system.
- Number of Exact Matches: This is the number of exact matches, based on file sizes, that are in the allocation thus allowing the file to be rewritten contiguously.

Note: Sometimes the estimates provided when running **defragfs** with the **-q** or **-r** options will return different results than what is actually done when running **defragfs** without any options.

Verify a File System

- Command syntax:

```
fsck [-p | -y | -n] [-f] [ file system ]
```

- Checks Journal Log
- Checks inodes, indirect blocks, data blocks, free lists
- If no file system name is specified, the **fsck** command will check all filesystems which have the **check=true** attribute set in the **/etc/filesystems**
- Orphan files are placed in the **/lost+found** directory

Figure 8-9. Verify a File System

AU147.0

Notes:

A file system can be verified using the **fsck** (file system check) command.

This check consists of a number of stages, including:

- Check the journal log for errors
- Check the blocks to ensure that each block is either allocated to a single file or is in the free list
- Check file sizes
- Check directory entries

The **-p** option (preen) is used to check a file system making only minor changes without bothering the user. The command when run under SMIT uses this option.

If **fsck** encounters errors it reports them to the screen. The **-y** option (yes) or **-n** (no) option is used to indicate a yes or no answer to all questions. The yes option is typically used to recover a badly damaged file system. Using the **-y** option will allow **fsck** to discard some badly damaged files. Note, however, that mounted file systems will not be repaired.

If any files are found that are not allocated to a directory anywhere, then **fsck** creates an entry for that data in the **lost+found** directory in the / directory of that file system. If the **lost+found** directory for a file system does not exist, it can be created using the AIX command **mklost+found**.

The **fsck** command also executes each time the system boots up (from the **/etc/rc** file).

Documenting File System Setup

- Run the **lsfs** command
- Get the contents of the **/etc/filesystems** file
- Run the **df** command to check free space
- Check all the mounted file systems by running the **mount** command

Figure 8-10. Documenting File System Setup

AU147.0

Notes:

Exercise: Managing File Systems



Figure 8-11. Exercise: Managing File Systems

AU147.0

Notes:

The lab will allow you to get some experience with the file system management tools. It also allows you to build and test file systems with different characteristics.

This exercise can be found in your Exercise Guide.

Checkpoint

1. What command can you use to determine if a file system is full?

2. What two commands can be used to find the files and users that are taking the most disk space?

3. True/False It is good practice to run `fsck -y` on all file systems, even if they are mounted.

Figure 8-12. Checkpoint

AU147.0

Notes:

Unit Summary

- File system management does not just happen on the system. File systems need to be regularly monitored to ensure that they do not run out of space
- To ensure the integrity of file systems, checks have to be carried out whenever file system corruption is suspected

Figure 8-13. Unit Summary

AU147.0

Notes:

Unit 9. Paging Space

What This Unit Is About

This unit outlines the concepts of paging space.

What You Should Be Able to Do

After completing this unit, you should be able to:

- Define why paging space is required in AIX
- List and monitor the paging space utilization of the system
- Perform corrective actions to rectify too little or too much paging space scenarios

How You Will Check Your Progress

Accountability:

- Checkpoint questions
- Exercise

References

Online	<i>System Management Concepts: Operating System and Devices</i>
Online	<i>System Management Guide: Operating System and Devices</i>
Online	<i>Performance Tuning Guide</i>
SG24-5765	<i>AIX 5L Differences Guide Version 5.1 Edition</i>

Objectives

After completing this unit, students should be able to:

- Define why paging space is required in AIX
- List and monitor the paging space utilization of the system
- Perform corrective actions to rectify too little or too much paging space scenarios

Figure 9-1. Objectives

AU147.0

Notes:

What Is Paging Space?

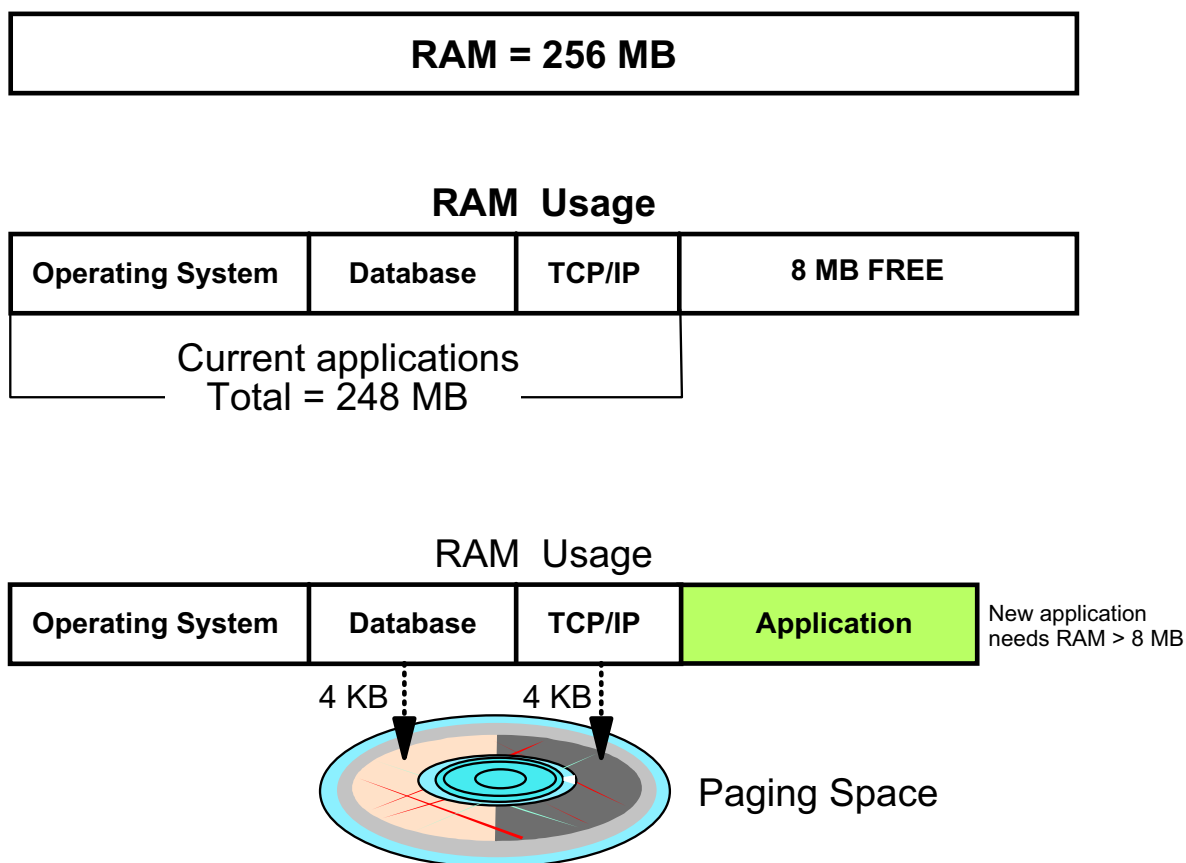


Figure 9-2. What Is Paging Space?

AU147.0

Notes:

For a process to be actively running, it must be loaded into memory. When it is loaded into memory, it is assigned a number of 4 KB areas called page frames. As more processes are loaded into memory, memory may become full. Not everything that resides in memory is active. When memory is full, memory is scanned to locate those page frames that are least-recently used. When one is located, a 4 KB block or "page" of disk space is allocated and the data from the page frame is moved to disk. This area on disk is called paging space.

The paging space is a reserved area on disk that can contain data that resided in memory but was inactive and was moved to make room for processes that are active. If a "paged-out" process is needed in memory again, the page is retrieved and brought back into memory or "paged-in".

In the AIX environment, paging and virtual storage is managed by the Virtual Memory Manager (VMM).

Paging Space

- Is a secondary storage area for over-committed memory
- Holds inactive 4 K "pages" on disk
- Is NOT a substitute for real memory

Figure 9-3. Paging Space

AU147.0

Notes:

Paging space is disk storage information that is resident in virtual memory, but is not currently being accessed. As memory fills, inactive pages are moved to the paging area on disk.

It is very important to remember that paging is a temporary holding area for inactive pages; it is not a substitute for real memory. If your machine has many active processes, it will require more real memory. You must make sure the machine has enough memory to maintain all the active processes. If you run out of memory, your machine will reach a constant state of paging called "thrashing". As it attempts to make room in memory, it completes a page-out; as soon as the page reaches the disk, it is needed again because it is still active. Your machine's resources will be wasted performing only paging activity and no real work will get done.

Increasing the amount of paging space when your machine is thrashing will not solve the problem. Thrashing is result of not enough real memory.

Sizing Paging Space

- Created at installation up to twice the size of real memory
- Amount needed is dependent on applications
- Monitor paging space : **lsps -a**
- Running low on paging space is bad

```
#  
ksh: cannot fork no swap space
```

Figure 9-4. Sizing Paging Space

AU147.0

Notes:

Paging space is created during installation. The initial size is dependent on the amount of RAM in your system. If RAM is greater than or equal to 64 MB, paging space is RAM + 16 MB. If RAM is less than 64 MB, paging space is twice the size of RAM.

This is just a starting point. This is not necessarily the amount of the paging space that is right for your machine. The number and types of applications will dictate the amount of paging space needed. Many sizing "rules of thumb" have been published, but the only way to correctly size your machine's paging space is to monitor the amount of paging activity.

Monitoring the activity is done with the command **lsps -a**. This command and its output will be covered shortly.

If your system runs low on paging space, a message will be sent to the console and sometimes to users as well. At this point the system will be unable to start any new processes until some running processes are terminated or release allocated memory. This situation should obviously be avoided. If any of the following messages appear on the console or in response to a command on any terminal, it indicates a low paging space.

"INIT: Paging space is low"

"ksh: cannot fork no swap space"

"Not enough memory"

"Fork function failed"

"fork () system call failed"

"Unable to fork, too may processes"

"Fork failure - not enough memory available"

"Fork function not allowed. Not enough memory available."

"Cannot fork: Not enough space"

The situation can get worse. If paging space continues to fill, non-system processes will be terminated and the system may even crash. Make sure you have enough paging space.

Paging Space Placement

- Only one paging space per disk
- Use disks with the least activity
- Paging spaces roughly the same size
- Do not extend paging space to multiple PV's
- Use multiple disk controllers

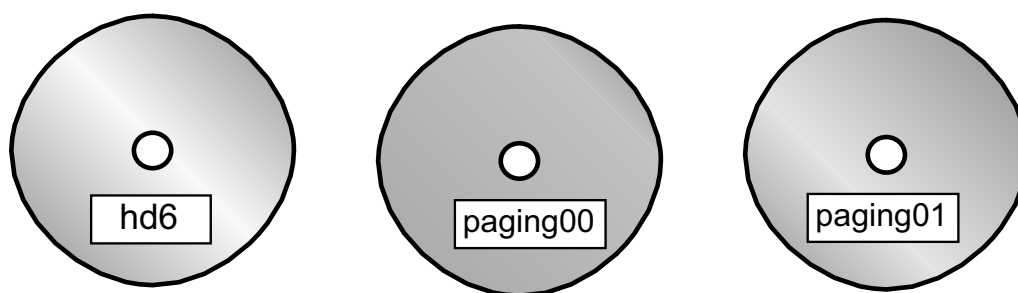


Figure 9-5. Paging Space Placement

AU147.0

Notes:

Placement and size of your paging space will impact its performance. The following are tips for paging space.

Do not have more than one paging space per disk. The paging space is allocated in a round robin manner and will use all paging areas equally. If you have two paging areas on one disk, then you are no longer spreading the activity across several disks.

Paging space will perform best when it is not competing with other activity on the disk. Use disks that do not have much activity.

Paging spaces should be roughly the same size. Because of the round robin technique that is used, if they are not the same size, then the paging space usage will not be balanced. Smaller paging areas will fill faster.

Do not extend a paging space to span multiple physical volumes. Although you can spread a paging area (like a regular logical volume) across several disks, the round robin technique will treat the paging area as one single paging area. Therefore, the activity will not be evenly spread across the disks.

Use disks on different controller. If the disks are attached to different controller you will get better throughput when reading and writing to the disk. That will improve your performance.

Paging Space

List Paging Activity

```
# lsps -a
```

Page Space	Physical Volume	Volume Group	Size	%Used	Active	Auto	Type
hd6	hdisk0	rootvg	64MB	43%	yes	yes	lv
paging00	hdisk2	rootvg	64MB	20%	yes	yes	lv

Total RAM

```
# lsattr -El sys0 -a realmem
```

```
realmem 262144 Amount of usable physical memory in KB False
```

Paging Space Activated at startup

```
# cat /etc/swapspaces
```

```
hd6:
```

```
    dev = /dev/hd6
```

```
paging00:
```

```
    dev = /dev/paging00
```

Figure 9-6. Paging Space

AU147.0

Notes:

The **lsps** command lists details of the paging spaces on the system including whether they are in use at the time and, if so, what percentage of their total space is allocated.

Another useful option available with the **lsps** command is the **-s** option which specifies the summary characteristics of all paging spaces. The information consists of the total size of the paging spaces (in MBs) and the percentage of paging spaces currently used.

Note that the output of the **lsps** command in the example shows two paging spaces: **hd6** and **paging00**. The paging space created during system installation is named **hd6**. Paging spaces created by the system administrator after system installation will be named **paging00**, **paging01**, and so on.

The file **/etc/swapspaces** contains a list of the paging space areas that will be activated at system startup.

Adding Paging Space

smit mkps

Add Another Paging Space

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

Volume group name	[Entry Fields]	
SIZE of paging space (in logical partitions)	rootvg	
PHYSICAL VOLUME name	[4]	#
Start using this paging space NOW?	hdisk2	+
Use this paging space each time system is RESTARTED?	yes	+
	yes	+

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

Figure 9-7. Adding Paging Space

AU147.0

Notes:

To add extra paging space volumes to the system, use SMIT, the **mkps** command, or the Web-based System Manager. If using the **mkps** command, the options are:

mkps [-a] [-n]	-s NumLPs Vgname Pvname
Vgname	The volume group within which to create the paging space
Pvname	Specifies the physical volume of the volume group
-s NumLPs	Sets the size of the new paging space in logical partitions
-a	Activate the paging space at the next restart (add it to /etc/swapspace s)
-n	Activate the paging space immediately

When a paging space is created, the **/etc/swapspace**s file is also updated if needed.

An example of the high-level command is: **# mkps -s 4 -n -a rootvg**

Change Paging Space

smit chps

Change / Show Characteristics of a Paging Space

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

	[Entry Fields]	
Paging space name	paging00	
Volume group name	rootvg	
Physical volume name	hdisk2	
NUMBER of additional logical partitions	[]	#
Or NUMBER of logical partitions to remove	[]	#
Use this paging space each time system is RESTARTED?	yes	+

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

Figure 9-8. Change Paging Space

AU147.0

Notes:

Paging space may have its size increased or decreased and may have its autostart options changed while it is in use (this updates `/etc/swapspaces`). The high-level command to perform this action is **chps**.

Decreasing paging space

The ability to decrease paging space is new with AIX V5.1. The argument **-d** to the **chps** command calls the **shrinkps** shell script to reduce the size of an active paging space. The use of a shell script reduces the possibility of getting into an unbootable state because users will not be allowed to run out of paging space. The script checks paging space actually in use and adds a buffer for paging space warning threshold. The SMIT fastpath is **smit chps**.

The process **chps** goes through to decrease an active paging space is:

Step	Action
1	Create a new, temporary space from the same volume group as the one being reduced
2	Deactivate the original paging space
3	Reduce the original paging space
4	Reactivate the original paging space
5	Deactivate the temporary space

The primary paging space (usually `hd6`) cannot be decreased below 32MB.

When you reduce the primary paging space, a temporary boot image and a temporary `/sbin/rc.boot` pointing to this temporary primary paging space will be created to make sure the system is always in a state where it can be safely rebooted.

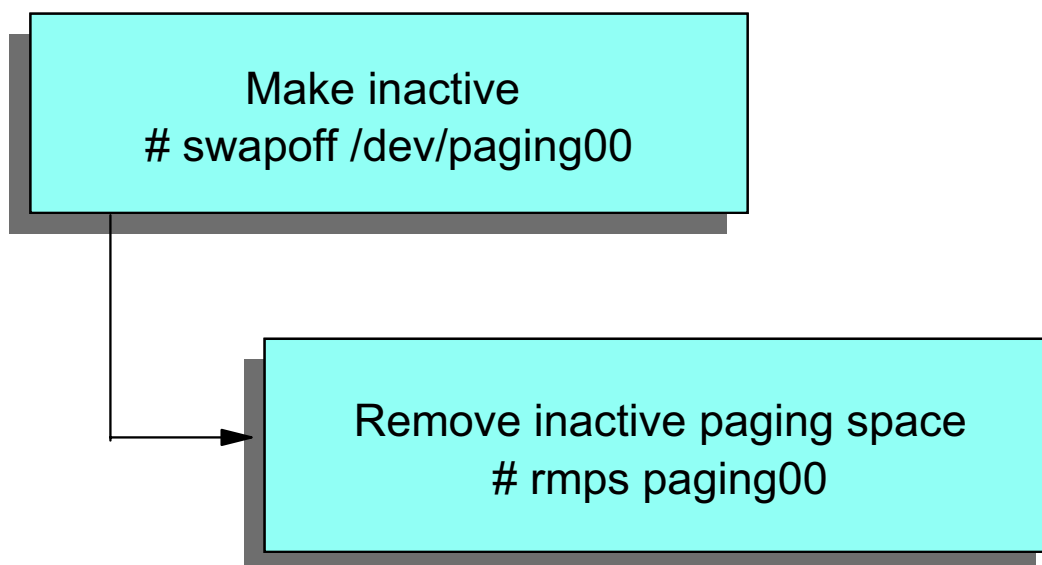
These command enhancements are not available through the Web-based System Manager. The Web-based System Manager allows you, by default, to specify the *increase* in size for a paging space in the Megabytes field.

Activating paging space

Inactive paging spaces may be activated dynamically once they have been defined. To do this enter: **`swapon /dev/pagingnn`** (Note: this operation is supported through SMIT as well, fastpath **`pgsp`**). Alternatively, use: **`swapon -a`** to activate all paging spaces defined in **`/etc/swapspace`**. This command is run in **`/etc/rc`** at system startup.

Remove Paging Space

In order to remove an active paging space:



NOTE: /dev/hd6 cannot be removed using this process

Figure 9-9. Remove Paging Space

AU147.0

Notes:

Paging space can be added to the system, and surplus paging space can be deleted to free up the disk space for other logical volumes.

Inactive paging space can be activated dynamically to meet system demand. In order to delete paging space it must be inactive (that is, not used by the kernel.) Beginning with AIX V5.1, active paging spaces can be deactivated while the system is running using the **swapoff** command or with the SMIT fastpath **swapoff**.

The **swapoff** command may fail due to:

- Paging size constraints. The process to remove an active paging space is to move all the pages of the paging space being removed to another paging space. If there is not enough active paging space to do this, the command will fail.
- I/O errors.

Problems with Paging Space

- Paging space too small:
Dynamically increase the size by allocating more partitions
- Paging space too large (only for user-created paging space):
 - ➔ Decrease paging space

```
# chps -d LogicalPartitions PagingSpace
```

Example:

```
# chps -d 2 paging00
```

Figure 9-10. Problems With Paging Space

AU147.0

Notes:

System defined paging space (for example, **hd6**) cannot be decreased below 32MB.

Documenting Paging Space Setup

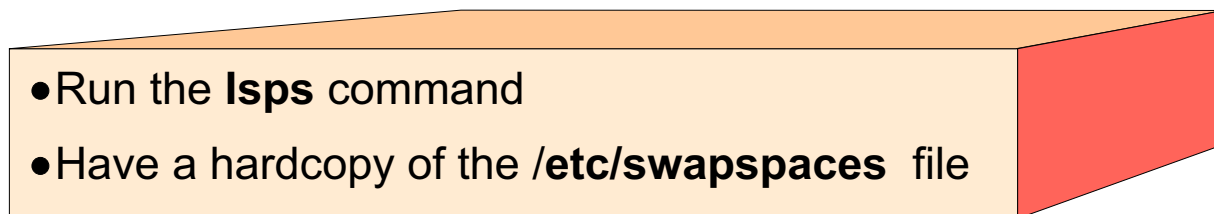


Figure 9-11. Documenting Paging Space Setup

AU147.0

Notes:

Run **lsps** to monitor the activity of the paging space. Keep good documentation so that you know what is "normal" for that system.

Keep a copy of **/etc/swapspaces** so that you know what paging spaces are defined to start at boot.

Exercise: Paging Space



Figure 9-12. Exercise: Paging Space

AU147.0

Notes:

This lab will allow you to add, decrease, monitor, and remove paging space.

This exercise can be found in your Exercise Guide.

Checkpoint

1. What problems can you conclude from the following listing?

Page Space	Physical Volume	Volume Group	Size	%Used	Active	Auto	Type
hd6	hdisk0	rootvg	64 MB	43%	yes	yes	lv
paging00	hdisk1	rootvg	64 MB	7%	yes	yes	lv
paging01	hdisk1	rootvg	16 MB	89%	yes	yes	lv

2. The size of paging00 (in the above example) can be dynamically decreased. True or false? Why?

Figure 9-13. Checkpoint

AU147.0

Notes:

Unit Summary

- Paging space is a requirement in AIX for the system to boot up. The default paging space is /dev/hd6.
- The percent utilization of all the paging spaces should be regularly monitored to ensure that the system has the correct amount of page space defined. The **lsp**s command can be used to do this.
- Paging space can be inactivated and the size can be increased or decreased dynamically.

Notes:

Unit 10. AIX V5.1 Installation

What This Unit Is About

This unit describes the process of installing the AIX V5.1 operating system.

What You Should Be Able to Do

After completing this unit, you should be able to:

- List the different media options available
- List the steps necessary to install the AIX V5.1 base operating system
- Identify the tasks that can be carried out using the Configuration Assistant

How You Will Check Your Progress

Accountability:

- Checkpoint questions
- Activity

References

SC23-2550	<i>AIX Version 4.1 Installation Guide</i>
SC23-1924	<i>AIX Version 4.2 Installation Guide</i>
SC23-4112	<i>AIX Version 4.3 Installation Guide</i>
SC23-4374	<i>AIX 5L Version 5.1 Installation Guide</i>

Objectives

After completing this unit, students should be able to:

- List the different installation and media options available
- List the steps necessary to install the AIX V5.1 base operating system
- Identify the tasks that can be carried out using the Configuration Assistant

Notes:

Installation Methods

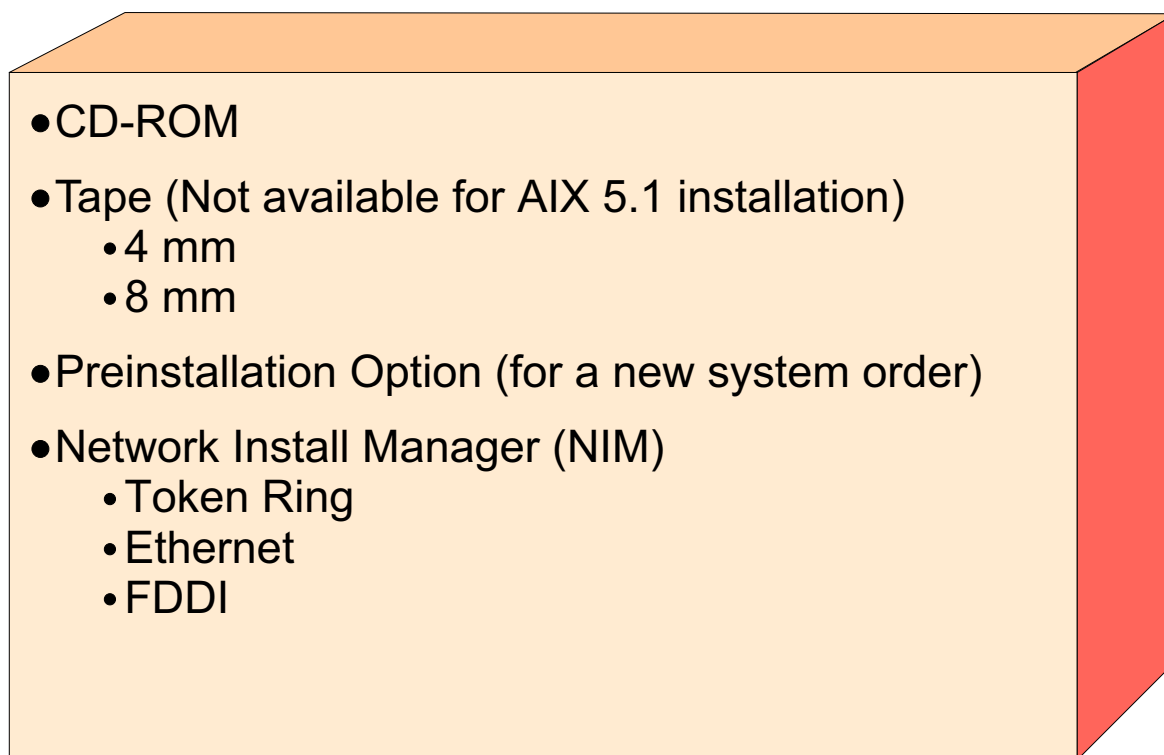


Figure 10-2. Installation Methods

AU147.0

Notes:

In AIX V5.1, 64MB of RAM is required to install the Base Operating System.

The contents of the CD-ROM is packaged in a file system format, thus the installation process from a CD is carried out in a different format than the tape.

The preinstall option is only valid if accompanied by a hardware order that includes the preinstalled AIX Version 5.1.

Network installations are carried out using the AIX Network Install Manager (NIM). This allows the user to manage the installation of the BOS and optional software, on one or more machines in a network environment. The NIM environment is made of client and server machines, where it is the server machine that makes the resources available to the other machines; that is, installation has to be initiated from the server to the client. An existing RISC System/6000 with AIX installed is required to set up a NIM environment. Additional information on how to perform a NIM installation can be found in the *Network Installation Management Guide and Reference*.

Installation Process - from CD

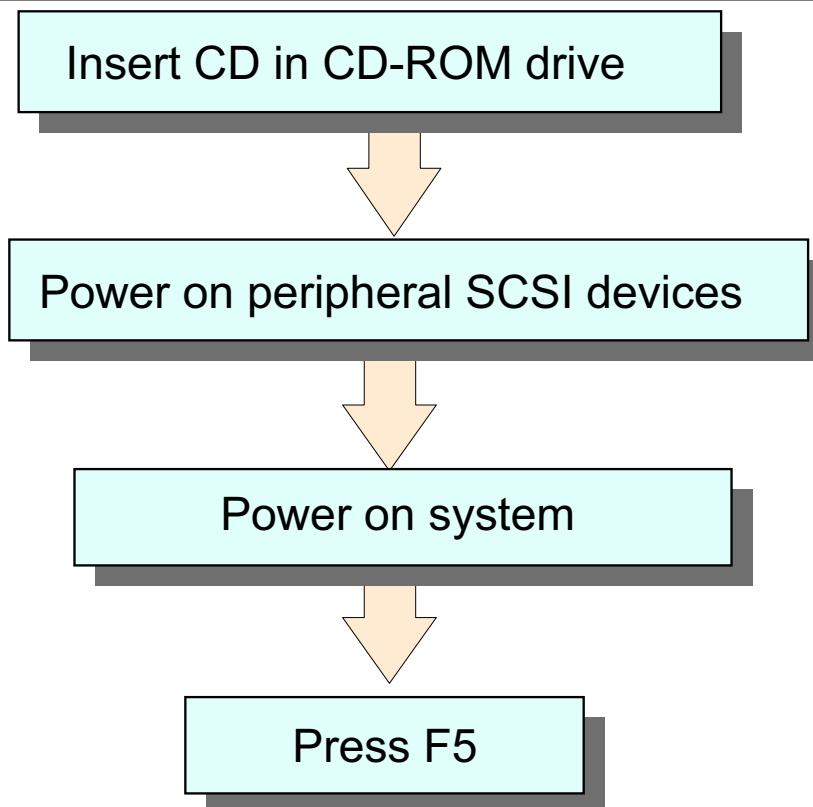


Figure 10-3. Installation Process - from CD

AU147.0

Notes:

The system will need to boot from the installation media. The base operating system (BOS) installation is most commonly performed using a CD.

Insert the installation media into the drive. If it is an external device, you must power it on before powering on the system or the system will not recognize it. It is best to power on all peripheral devices anyway, because during the installation all recognized devices will be configured.

Power on the system to start the boot sequence. The LED's will display numbers indicating the system components are being tested. Also, if you are using a graphical display, you will see icons of the hardware devices appear on the screen. The machine is completing a power on self test (POST).

If the machine doesn't reach the installation menu but instead keeps cycling through the POST, it is because the CD (or whatever installation device you are trying to use) is not in the boot list. If this happens and you are installing by CD, during the POST, depress and release the F5 key on the keyboard. This will invoke the default service boot list. The CD is on that list. If you are attempting to install by tape, you will need to add the tape to the boot

list. This is done via the System Management Services (SMS) program. This will be discussed later.

The CD and tape devices must be powered on to open the door to the device. If they are internal, you will need to power on the system before inserting the installation media. If you insert the media before the POST is done (about 30 seconds), the machine can still boot from that media.

Once the POST is complete, the system will search the boot list for a bootable image. When it finds the bootable image, you will see menus appear on the screen.

Note: Boot procedures for Itanium-based systems are different.

Console and Language Definition

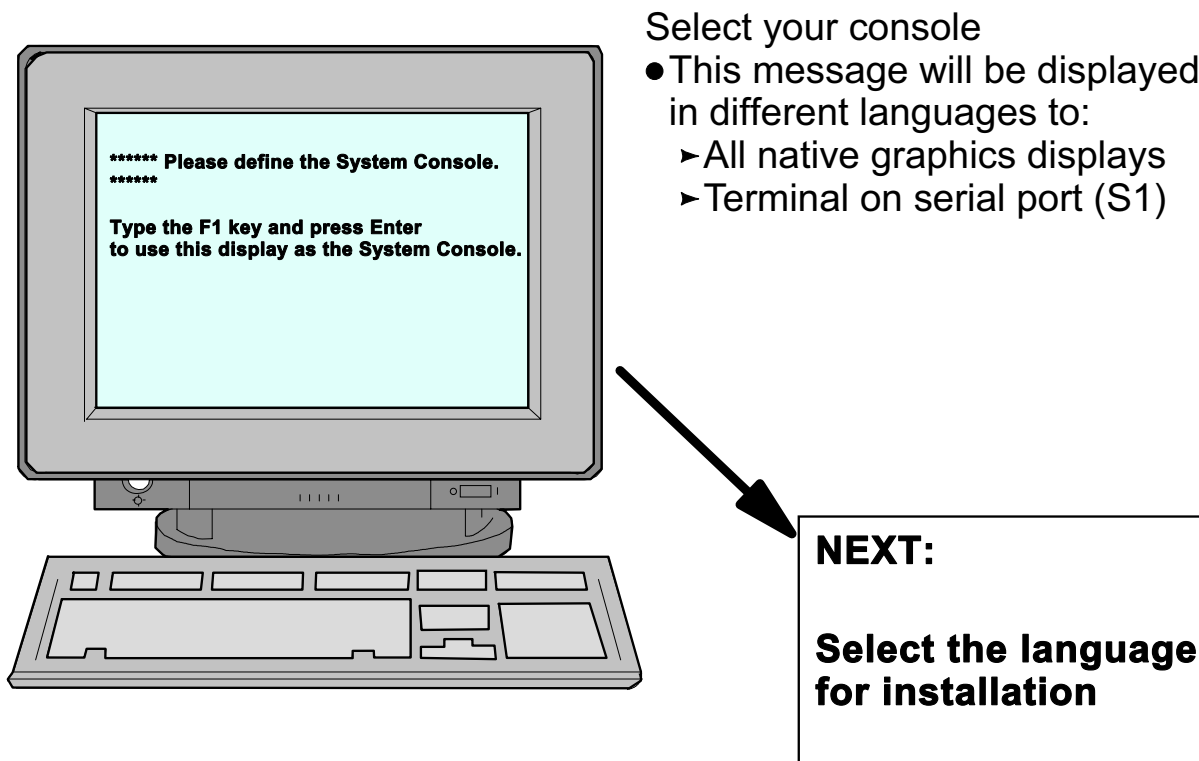


Figure 10-4. Console and Language Definition

AU147.0

Notes:

Each native (graphics) display and the ASCII terminal attached to the first built-in serial port (S1) will display the console messages. Whichever display you respond to will become the console display during the installation. The console display can be changed at a later time if required.

Graphic displays will request that you press the **F1** key and then **Enter** to make it the system console. If you are using an ASCII terminal as the system console, you will need to press **2** and then **Enter**.

If you are using an ASCII terminal as your console, make sure that it is powered on and correctly configured before you begin the installation. AIX will assume these characteristics for the terminal on S1:

```
Terminal type=dumb
Speed=9600
Parity=none
Bits per character=8
Stop bits=1
```


Line Control=IPRTS
Operating mode=echo
Turnaround character=CR

The boot program will not redisplay the message if you missed it the first time. If your terminal was not correctly configured, you can still type **2** and press **Enter** to continue, once you have corrected the problem.

During the installation, you will also be prompted to select the language to be used for the messages and the status information during the installation process. This language need not be the same as the language intended for the primary environment of the system.

Installation and Maintenance Menu

At the Installation and Maintenance menu check all the installation settings.

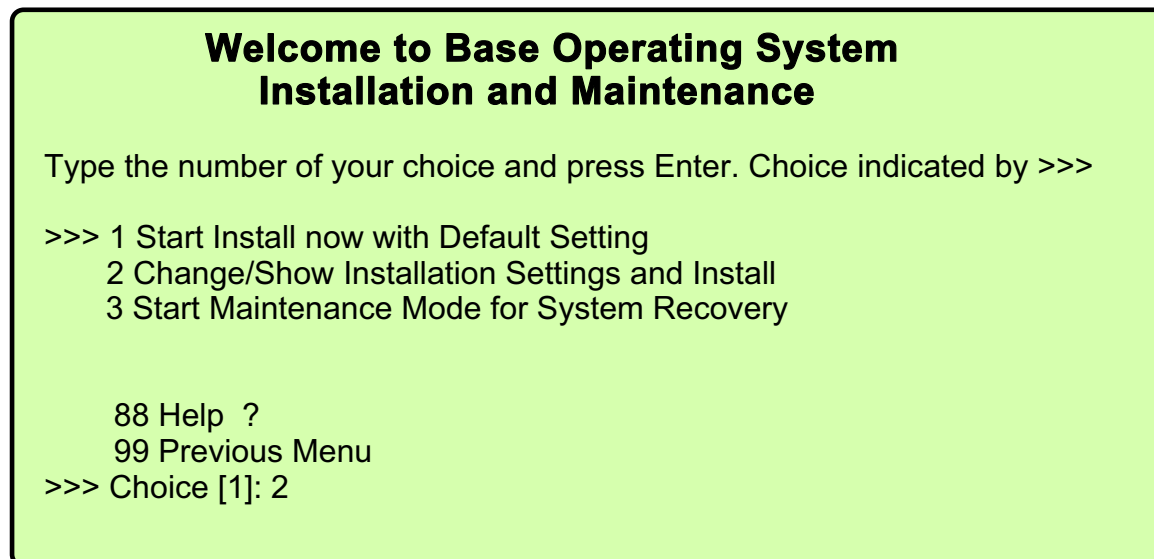


Figure 10-5. Installation and Maintenance Menu

AU147.0

Notes:

To confirm or change the installation and system settings that have been set for this system, type a **2** and press **Enter**. Select **88** to display help on this or any subsequent installation screen.

Installation Settings

Installation Settings

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

1 System Settings:
 Method of installation New and Complete Overwrite
 Disk where you want to Install hdisk0

2 Primary Language Environment Settings (AFTER Install):
 Cultural Convention C (POSIX)
 Language C (POSIX)
 Keyboard C (POSIX)
 Keyboard Type Default

3 Advanced Options

0 Install with the settings listed above
 88 Help ?
 99 Previous Menu

>>> Choice [1]:

Warning: Base Operating System Installation will destroy or impair recovery of SOME data on the destination disk hdisk0

Figure 10-6. Installation Settings

AU147.0

Notes:

The **Installation Settings** screen allows you to:

- Set the type of installation: Migration, Preservation, or New and Complete Overwrite
- Determine the installation disk
- Set the primary language environment
- Set other advanced options

Method of Installation

Option 1 of the Installation and Maintenance menu.

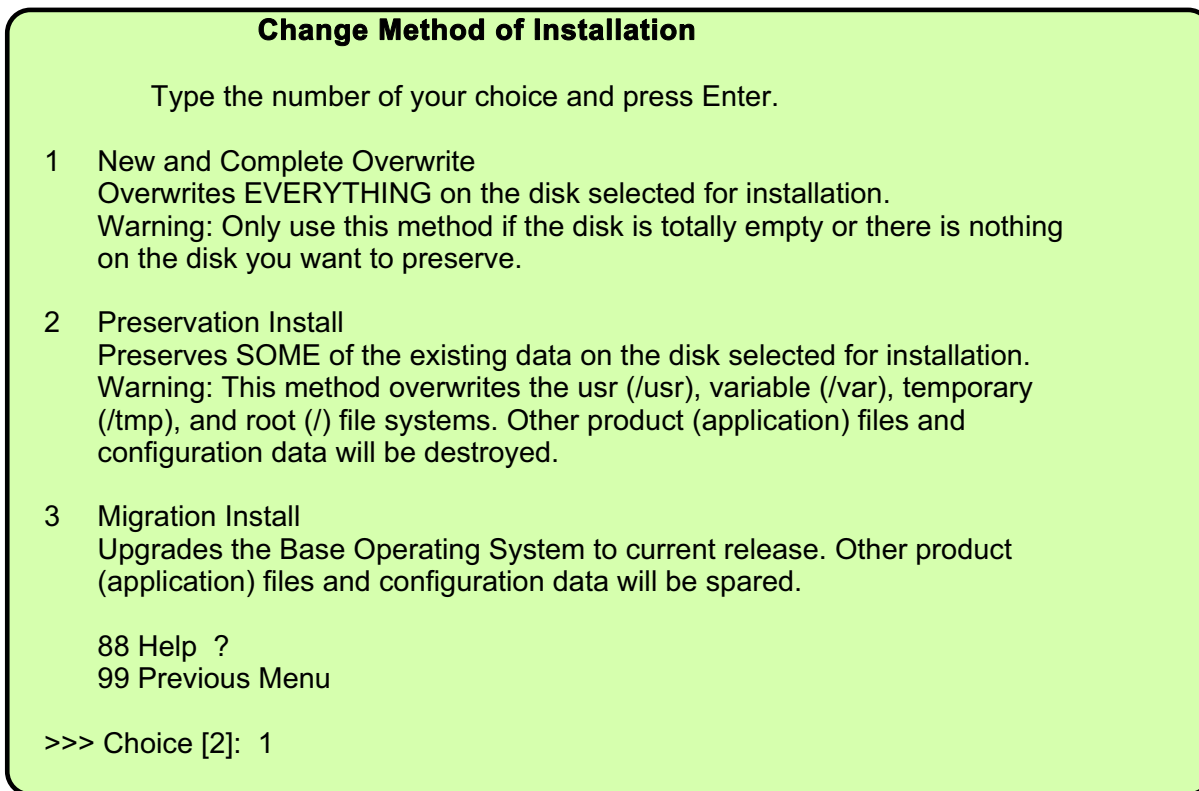


Figure 10-7. Method of Installation

AU147.0

Notes:

When you select option 1 to change the **Method of Installation**, the following submenu will be displayed, the contents of which depends on the current state of the machine.

Complete Overwrite Install

On a new machine, New and Complete Overwrite is the only possible method of installation. On an existing machine, if you want to completely overwrite the existing version of BOS, then you should use this method.

Preservation Install

Use this installation method when a previous version of BOS is installed on your system and you want to preserve the user data in the root volume group. This method will remove only the contents of **/usr**, **/** (root), **/var** and **/tmp**. The Preservation Install option will preserve page and dump devices as well as **/home** and other user-created file systems. System configuration will have to be done after doing a preservation installation.

Migration Install

Use this installation method to upgrade an AIX Version 3.2 or later system to AIX Version 5.1, while preserving the existing root volume group. This method preserves all file systems except **tmp**, as well as the logical volumes and system configuration files. Obsolete or selective fix files are removed. Migration is the default installation method for an AIX system running Version 3.2 or 4.x.

The installation process determines which optional software products will be installed.

Installation Disks

Change Disks Where You Want to Install

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

	Name	Location Code	Size (MB)	VG Status	Bootable
>>> 1	hdisk0	04-C0-00-4,0	2063	rootvg	yes
2	hdisk1	04-C0-00-5,0	2063	rootvg	no

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

>>> Choice [0]:

Figure 10-8. Installation Disks

AU147.0

Notes:

Having selected the type of installation, you must then select the disks that are to be used for the installation. A list of all the available disks will be displayed, similar to the one shown.

This screen also gives you the option to install to an unsupported disk by adding the code for the device first.

When you have finished selecting the disks, type 0 in the Choice field and press Enter. If this is a New and Complete Overwrite of an Itanium-based system, the **Select Disk(s) for Manual Partitioning** screen is displayed next. This screen is not shown.

Primary Language Environment

Option 2 of the Installation and Maintenance menu

Type the number for the Cultural Convention (such as date, time, and money), Language and Keyboard for this system and press Enter, or type 106 and press Enter to create your own combination.

Cultural Convention	Language	Keyboard
>> 1. C (POSIX)	C (POSIX)	C (POSIX)
2. Albanian	English (United States)	Albanian
3. Arabic	Arabic (Bahrain)	Arabic (Bahrain)

... several screens later ...

106. Create your own combination of Cultural Convention, Language and Keyboards.

88 Help ?
99 Previous menu

Choice [1]:

Figure 10-9. Primary Language Environment®

AU147.0

Notes:

At this point in the installation process you can change the language and cultural convention that will be used on the system after installation. This screen may actually display a number of language options, such as French, German, Italian, Byelorussian, Ukrainian, and so forth.

It is recommended that if you are going to change the language, change it at this point rather than after the installation is complete. Whatever language is specified at this point is pulled off the installation media.

Cultural convention determines the way numeric, monetary, and date and time characteristics are displayed.

Language field determines the language used to display text and system messages.

Advanced Options

Option 3 of the Installation and Maintenance menu

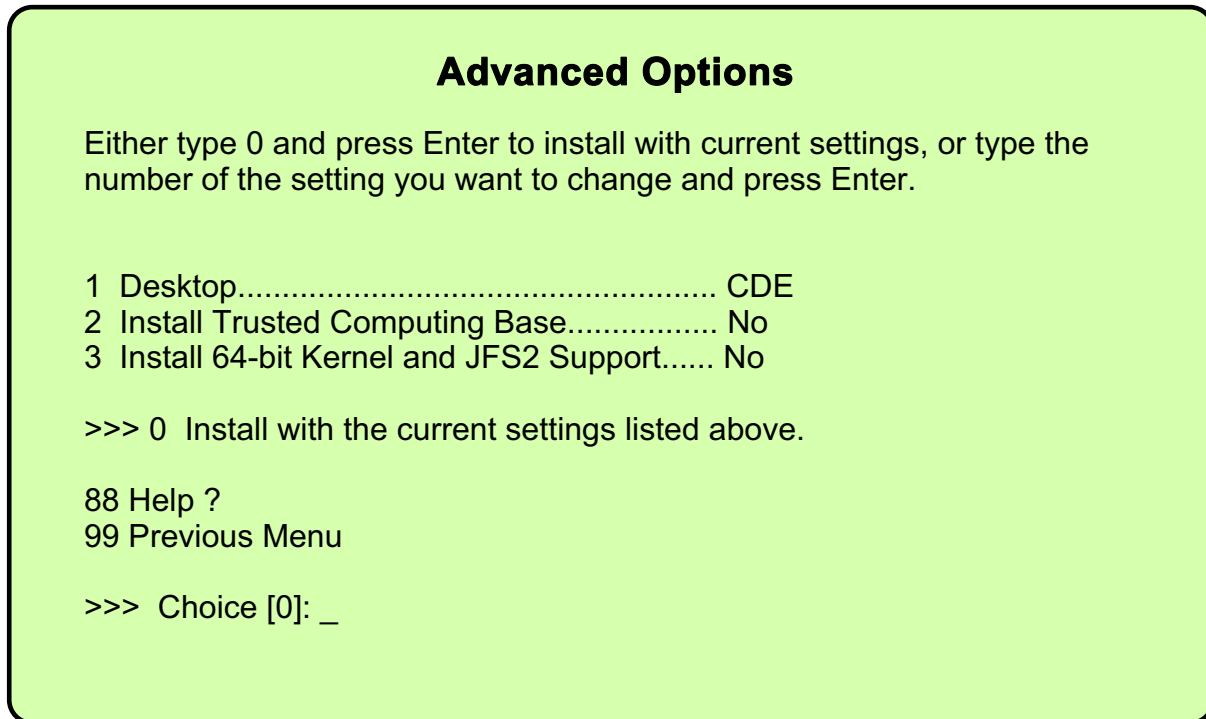


Figure 10-10. Advanced Options

AU147.0

Notes:

Desktop

The first prompt will be either:

- Installation Package Set (for ASCII consoles)
Options are Minimal or Default
- Desktop (for graphical consoles)
Options are CDE, Gnome, KDE, or NONE

The example shown has a graphical console.

For an ASCII console or a system with a graphical console where the desktop selected is NONE, a *minimal* configuration is installed which includes X11, Java, Perl, SMIT, and the Web-based System Manager.

For a system with a graphical console, if you choose CDE, Gnome, or KDE, the desktop and documentation service libraries are also installed. This is considered a *default* installation configuration. If you choose Gnome or KDE, the interface

prompts you for the Toolbox for Linux Applications CD. If this CD is not available, you can type **q** to continue the installation without it.

The default installation configuration may prompt for additional CD volumes during the BOS installation. When prompted, if you decide not to continue with additional volumes or if a volume is not available, you can type **q** and press Enter to continue the installation process. The system will have enough of the BOS loaded to be usable.

Migration installations use the Default configuration and update currently installed filesets to the new level.

Install Trusted Computing Base (TCB)

When you install the Trusted Computing Base (TCB), the trusted path, the trusted shell, and system integrity checking are installed. The trusted path protects your system in case a program is masquerading as the program you want to use. The trusted path tries to ensure that the programs you run are trusted programs.

If you want to install the TCB, you must indicate "yes" now. The TCB cannot be installed later.

Install 64-bit Kernel and JFS2 Support

If you have a 64-bit system and select *Yes* for this option, the 64-bit kernel is linked so that it becomes the running kernel on the system after the installation is complete. If you choose *No*, the 64-bit kernel is still installed on the system, but the running kernel after installation is either the **up** or **mp** kernel, depending on the system. To toggle the choice between *no* (the Default) and *yes*, type 3 and press Enter.

If you choose *Yes* and are installing with the New and Complete Overwrite method, the file systems are created with JFS2 (Journaled File System 2), instead of JFS.

If you want the 64-bit kernel to be the running kernel, but do not want JFS2 file systems, then select *No*. After the install completes, use the following commands to switch to the 64-bit kernel:

- **In -fs /usr/lib/boot/unix_64 /unix**
- **In -fs /usr/lib/boot/unix_64 /usr/lib/boot/unix**
- **bosboot -ad/dev/ipldevice**
- Reboot your system

Begin Installation

Installing Base Operating System

If you need the system key to select SERVICE mode, turn the system key to the NORMAL position anytime before the installation ends

Please wait

Approximate % tasks completed	Elapsed Time (in minutes)
16	1

- Builds AIX directory structure
- Restores BOS, locale and filesets from installation media only
- Installs software for the connected and powered on devices

Figure 10-11. Begin Installation

AU147.0

Notes:

The installation media contains information stored on it to determine the sizes that the standard AIX file systems will have. These will be set large enough for the installation to succeed but will not leave much free space after installation. You can dynamically increase the size of any of the file systems once AIX has been installed. If you are installing from a system image backup tape, the file systems created will be the same sizes and names as those on the system when the tape was created.

The files are restored from the media and then verified. This will take some time but can be left unattended. After the BOS has installed, the appropriate locale optional program will also be installed. At any stage before the installation process completes, if your system has a system key, turn it to the **Normal** position (only found on older micro-channel machines).

Once the installation has completed, the system will automatically reboot from the newly installed operating system on disk.

Installation Flow Chart - All Systems

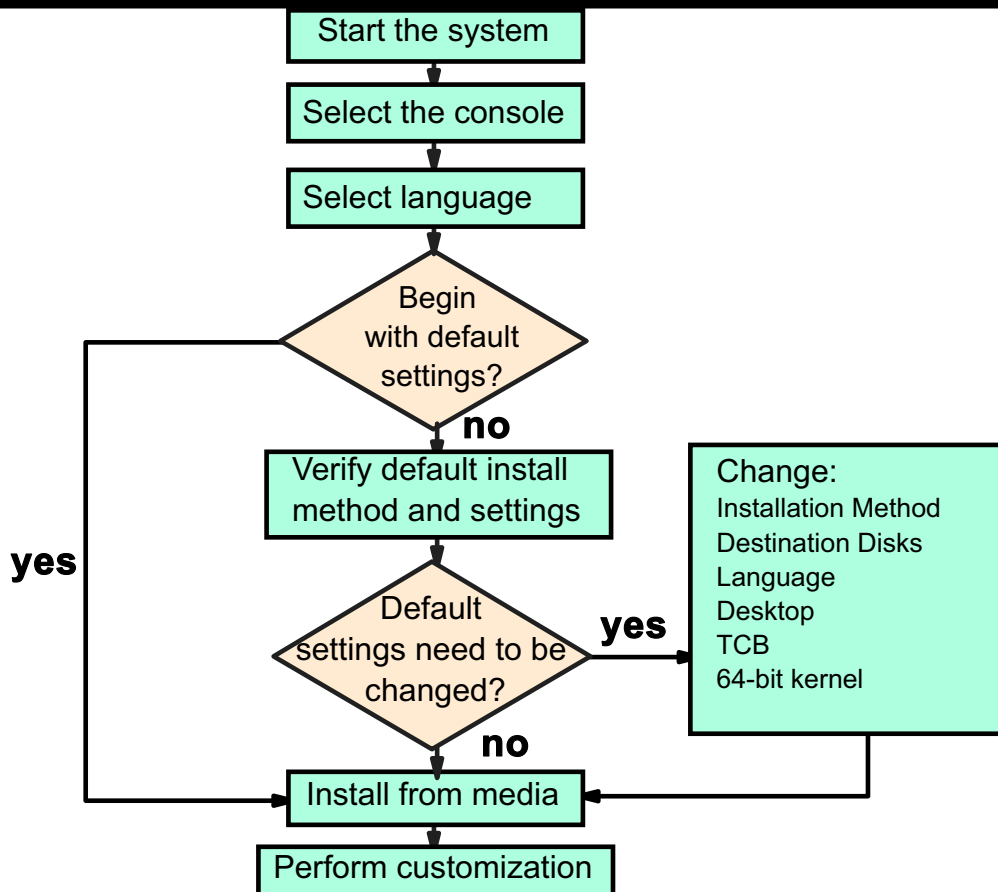


Figure 10-12. Installation Flow Chart - All Systems

AU147.0

Notes:

Configuration Assistant Menu

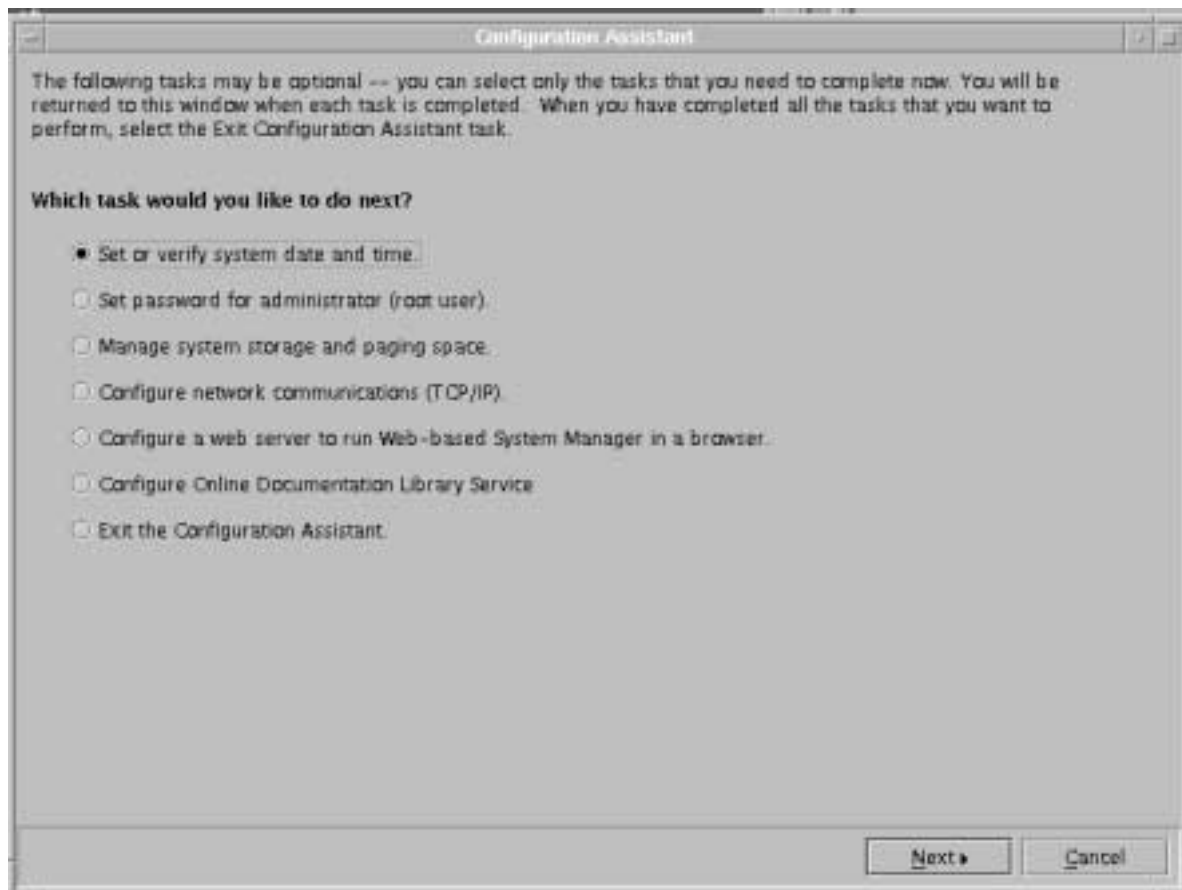


Figure 10-13. Configuration Assistant Menu

AU147.0

Notes:

After installing AIX, the operating system will run with default settings; one user (root), the date and time set for where the system was manufactured, and other very general settings. You probably want to change some or all of these settings. Also, you must provide system and network information if you want to communicate with other systems.

If using a graphics terminal for the installation, the newly installed BOS reboots and starts the **Configuration Assistant**, which guides you through completing customization tasks. When you use the Configuration Assistant immediately after BOS installation, only the tasks that apply to your type of installation will be shown. If an ASCII terminal was used for the installation, an ASCII-based **Installation Assistant** is displayed instead. Both the graphics-based Configuration Assistant and the ASCII-based Installation Assistant provide comparable support.

When you have completed your work using the Configuration Assistant/Installation Assistant, you can indicate that you are done working with the program. This will prevent this program from being displayed at the next reboot.

The Configuration Assistant and Installation Assistant provide step-by-step instructions for completing each customization task. Examples of tasks that can be performed are setting the system date and time, setting root's password and configuring the network.

Complete the tasks in the order that the Configuration Assistant/Installation Assistant lists them. It is helpful to complete all customization tasks before you use your system.

You must have root user authority to use the Configuration/Installation Assistant. From a graphics terminal, type **install_assist** to access the Configuration Assistant. From AIXwindows®, the command **configassist** can also be used to access the Configuration Assistant. From an ASCII terminal, use the **install_assist** command to access the Installation Assistant.

Activity: Configuration Assistant

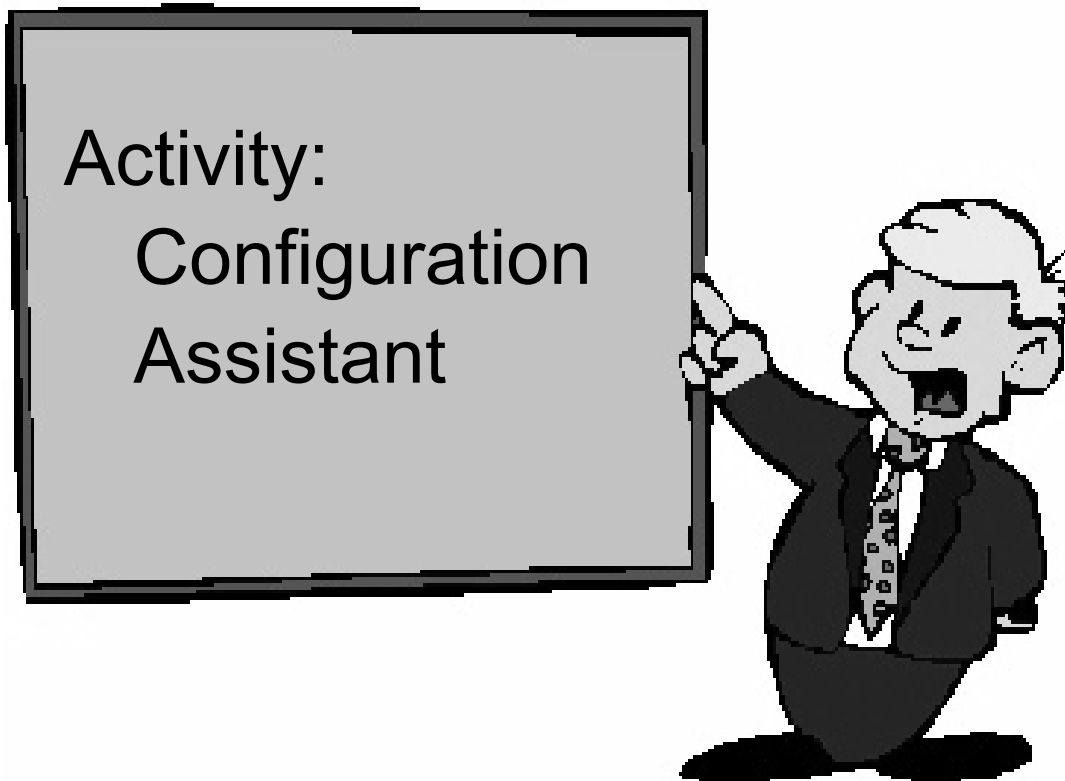


Figure 10-14. Activity: Configuration Assistant

AU147.0

Notes:

Instructions

In this activity, you will have the opportunity to work with the Configuration Assistant. The Configuration Assistant is the first screen that you will be presented with after a successful installation of the operating system.

1. Log into the system and become root.
2. Start Configuration Assistant.
3. Ensure the date and time are set correctly.
4. Familiarize yourself with some of the other options. When you finished, exit from Configuration Assistant and select the option to Finish now, and do not restart Configuration Assistant when restarting AIX.

Instructions with Hints

In this activity, you will have the opportunity to work with the Configuration Assistant. The Configuration Assistant is the first screen that you will be presented with after a successful installation of the operating system.

1. Log into the system and become root.
2. Start Configuration Assistant.

If you are using a graphical interface, type:

install_assist OR # configassist

If you are using an ASCII interface, type:

install_assist

3. Ensure the date and time are set correctly.
 - Click **Next** on the opening screen.
 - Date and time are already selected. Click **Next**.
 - Make any changes if necessary to the date and time.
 - Click **Next** when finished.
4. Familiarize yourself with some of the other options. When you finished, exit from Configuration Assistant and select the option to "Finish now, and do not restart Configuration Assistant when restarting AIX."
 - Click **Exit the Configuration Assistant** (Then Click Next)
 - Click **Finish now, and do not restart Configuration Assistant when restarting AIX** (Then Click Finish)

Checkpoint

1. AIX 5.1 can be installed from which of the following: (select all that are correct)
 - a. 8 mm tape
 - b. CD-ROM
 - c. Diskette
 - d. 4 mm tape
2. True or false? A 'Preservation Install' will preserve all data on the disks.
3. What is the console used for during the installation process?

Notes:

Unit Summary

- AIX 5.1 is only distributed on CD-ROM
- In order to install the base operating system, system specific questions have to be answered before the process can begin
- The Configuration Assistant is used by the system administrator to further customize the system

Figure 10-16. Unit Summary

AU147.0

Notes:

Unit 11. AIX Software Installation and Maintenance

What This Unit Is About

This unit covers the process of installing and maintaining optional software product and updates.

What You Should Be Able to Do

After completing this unit, you should be able to:

- Define the package definitions and naming conventions
- Identify how software products and updates are installed and managed on the system

How You Will Check Your Progress

Accountability:

- Checkpoint questions
- Exercise

References

SC23-2550	<i>AIX Version 4.1 Installation Guide</i>
SC23-1924	<i>AIX Version 4.2 Installation Guide</i>
SC23-4112	<i>AIX Version 4.3 Installation Guide</i>
SC23-4374	<i>AIX 5L Version 5.1 Installation Guide</i>

Objectives

After completing this unit, students should be able to:

- Define the package definitions and naming conventions
- Identify how software products and updates are installed and managed on the system

Figure 11-1. Objectives

AU147.0

Notes:

AIX Product Offerings

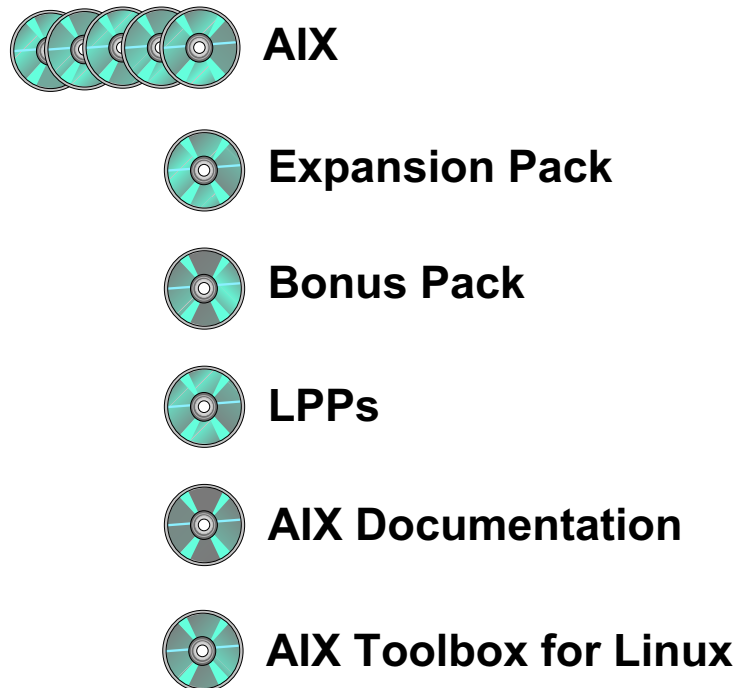


Figure 11-2. AIX Product Offering

AU147.0

Notes:

The AIX Version 5.1 operating system is delivered on multiple CDs. During the ordering process, it is necessary to indicate the system type.

Licensed Program Products are separately orderable products that will run on the AIX operating system.

The contents of the Expansion and Bonus Packs vary over time. Their purpose is to acquaint users with tools and products that may be valuable in their business environment. For example, the AIX V5.1 Expansion and Bonus Packs contains tools to build secure Java applications, Data Encryption Standard (DES) library routines, software security and encryption support, Network Authentication Service, IBM HTTP Server, Netscape Communicator, Adobe Acrobat Reader, AIX Fast Connect Evaluation Software, and SCO Tarantella Evaluation Software

Packaging Definitions

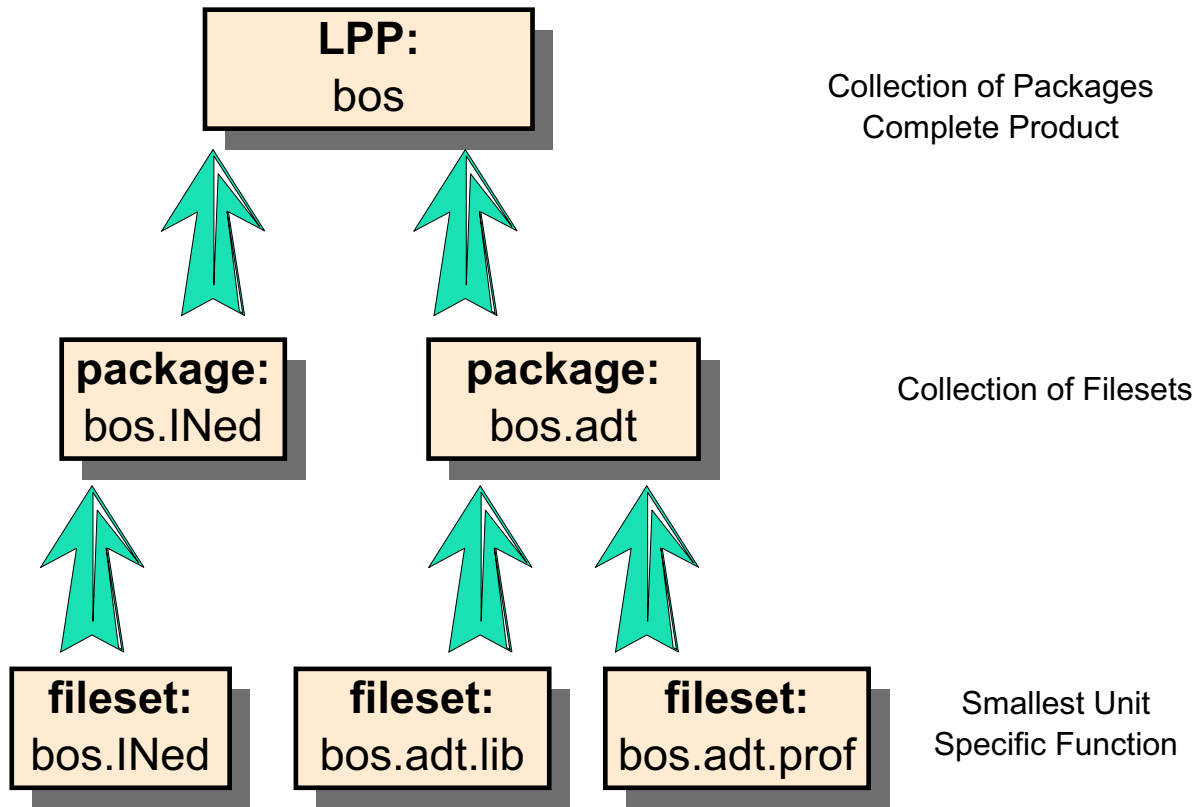


Figure 11-3. Packaging Definitions

AU147.0

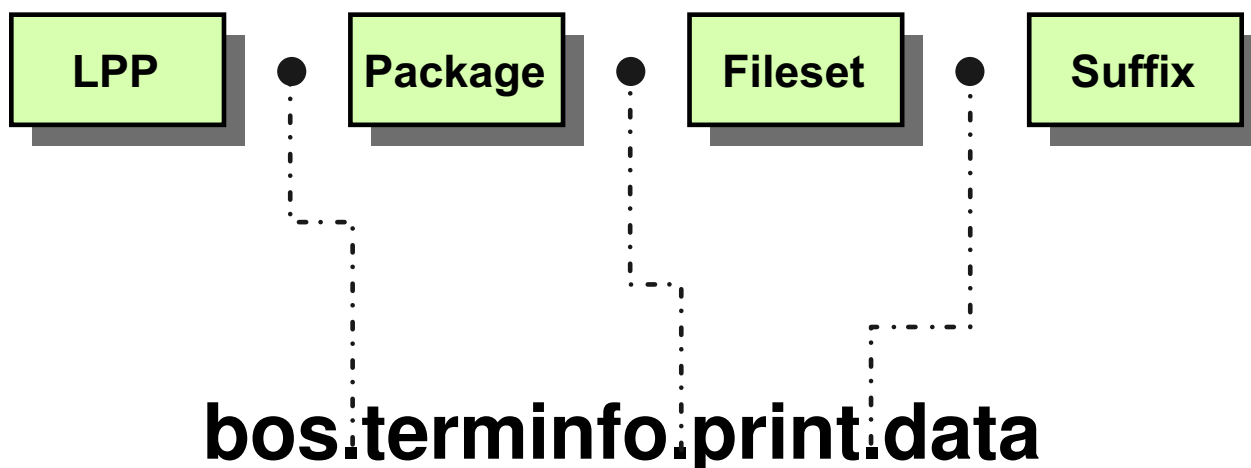
Notes:

Licensed **P**rogram **P**roduct (LPP) is a complete software product collection including all packages and filesets required. For example the Base Operating System (bos) itself is a LPP which in turn is a complete collection of packages and filesets.

A package contains a group of filesets with a common function. It is a single, installable image.

A fileset is the smallest individually installable unit. It is a collection of files that provides a specific function. For example, bos.net.tcp.client is a fileset in the bos.net package.

Fileset Naming



Message Convention:

LPP.msg[.lang].package.fileset

Figure 11-4. Fileset Naming

AU147.0

Notes:

Filesets follow a standard naming convention. The Licensed Program Product name will be the first part of every fileset name. The fileset names are meaningful and describe the contents of the fileset. The following are the standard fileset suffixes:

.adt	Application Development Toolkit for the Licensed Program Product
.com	Common code between two like filesets
.compat	Compatibility code that will be removed in a future release of the Licensed Program Product
.data	/usr/share portion of a fileset
.dev	Device support for that Licensed Program Product
.diag	Diagnostics for a fileset
.fnt	Font portion of a fileset
.info[lang]	InfoExplorer databases for that Licensed Program Product

.help[lang]	Translated help files for that Licensed Program Product
.loc	Locale for that Licensed Program Product
.mp	Multi-processor specific code for a fileset
.msg[lang]	Translated messages
.rte	Run time or minimum set
.smit	SMIT tools and dialogs for a fileset
.ucode	Microcode for a fileset
.up	Uni-processor specific code for a fileset

With message libraries associated with LPPs, the language is also part of the naming convention.

Bundles

- A **Bundle** is a collection of packages and filesets suited for a particular environment
- Predefined system bundles in AIX 5.1 include:
 - App-Dev
 - CDE
 - GNOME
 - KDE
 - Media-Defined
 - Netscape
 - devices
 - wsm_remote

Figure 11-5. Bundles

AU147.0

Notes:

Since there are thousands of filesets, having to determine which individual fileset you want on your machine could be a time-consuming task. AIX has bundles which offer a collection of filesets that suit a particular purpose. For example, if you are developing applications, the **App-Dev** bundle would be the logical choice to install.

Some filesets within a bundle will only be installed if the prerequisite hardware is available (for example, a graphic adapter is needed to run AIXWindow).

In some cases, bundles are equivalent to product offerings. Often, however, they are a subset of a product offering or a separate customized bundle. The bundles available may vary from configuration to configuration.

Software Updates

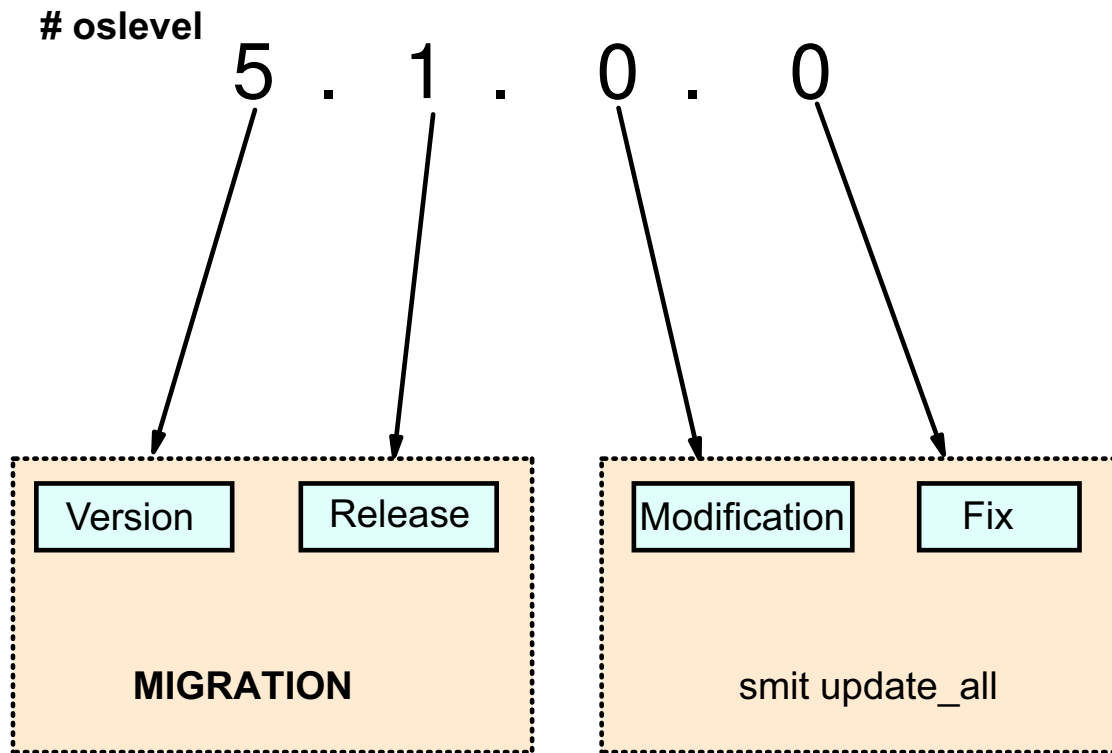


Figure 11-6. Software Updates

AU147.0

Notes:

As new software is created for AIX, you will want to upgrade your system to maintain the latest features and functionality.

The numerical information that shows what level of software you currently have installed is broken into 4 parts: Version, Release, Modification, and Fix. You can see this using the command **oslevel**.

When you want to upgrade the system, how you do it depends on what type of upgrade you are performing. Changes to the Version or Release levels require you to perform a migration installation as discussed in the Installation section. If you want to make a change to the Modification or Fix levels, you will use **smit update_all**. These changes provide fixes to defects or additional functions to the BOS or optional software products.

Version and Release upgrades must be purchased. Modification and fix-level upgrade are available at no charge. They are provided on CD or tape (order via AIX Support Center) or they can be downloaded from the Web. To assist with Web downloads, a tool called FixDist is also available free of charge. FixDist and updates are available at <http://techsupport.services.ibm.com/rs6000/support>.

IBM provides a number of mirrored anonymous ftp servers for FixDist support:

Canada	fixdist.aix.can.ibm.com	204.138.188.126
Germany	mx00d4e0.nyc.ibm.com	204.146.172.212
Japan	fixdist.yamato.ibm.co.jp	203.141.89.41
United Kingdom	www.europe.ibm.com	194.196.0.2
United States	service.boulder.ibm.com	207.25.253.26

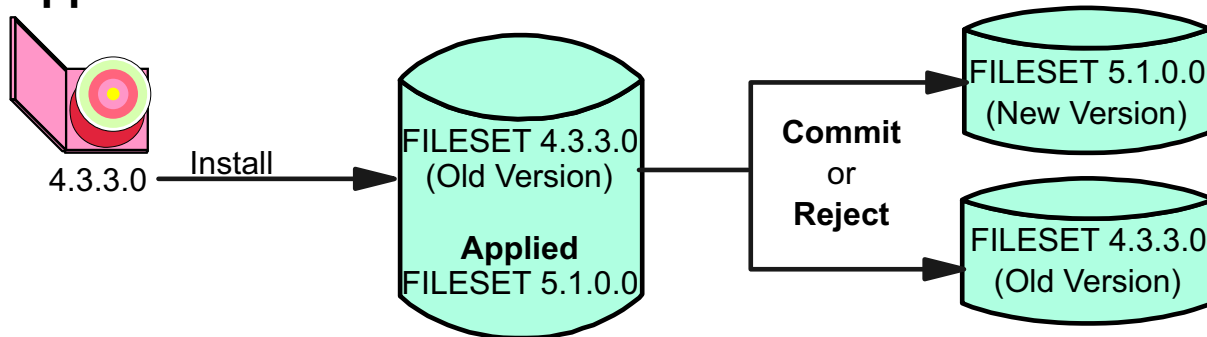
The procedure to download FixDist to a user's system is listed below:

1. **cd /tmp**
2. Get the FixDist tool and Postscript user guide:
ftp to any server site from the list.
login: **anonymous**
password: **your email address**
bin
cd /aix/tools/fixdist
get fd.tar.Z (The FixDist tool in compressed tar format)
get fixdist.ps.Z (User Guide in compressed PostScript)
quit
3. Install the FixDist tool:
cd /
zcat /tmp/fd.tar.Z | tar -xpvf -
4. To start FixDist: **fixdist**

Fixdist also includes a companion tool called TapeGen, which allows you to create a stacked tape containing SMIT installable fixes.

Software States

Applied:



Committed:

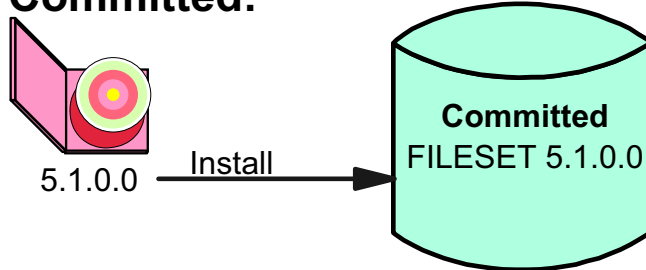


Figure 11-7. Software States

AU147.0

Notes:

AIX has a number of software states. When you are installing software for the first time, the software will automatically install to a committed state. This means there is only one level of that software product installed on your system.

When you are installing a fix- or a maintenance- level upgrade to your system, you will have the option of installing the software either in the committed state or the applied state. The applied state allows you to maintain two levels of the software on your system. When software is installed in the applied state, the older version is saved on the disk and is deactivated while the newer version is installed and becomes the active version.

The applied state gives you the opportunity to test the newer software before committing to its use. If it works as expected, then you can commit the software which will remove the old version from the disk.

If the newer version is causing a problem, you can reject it which will remove the newer version and recommit the old version.

With committed (or applied) software products, you can also remove them. This will cause the product's files to be deleted from the system. Requisite software (software dependent on this product) will also be removed unless it is required by some other software product on your system. If you want to use the software again, you would need to reinstall it.

Software Installation and Maintenance

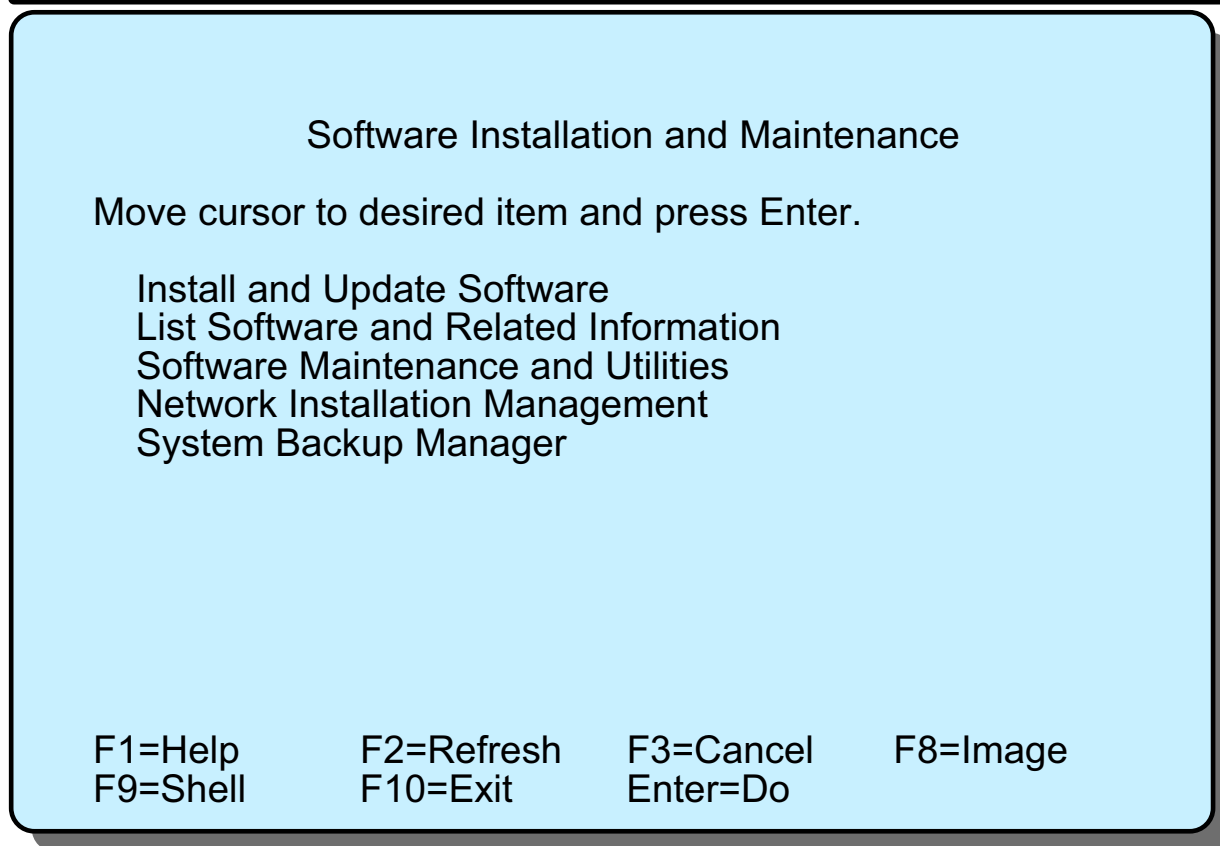


Figure 11-8. Software Installation and Maintenance

AU147.0

Notes:

Use the SMIT fastpath **smit install** to access the **Software Installation and Maintenance** menu.

You can also use the Web-based System manager to install software.

Install and Update Software

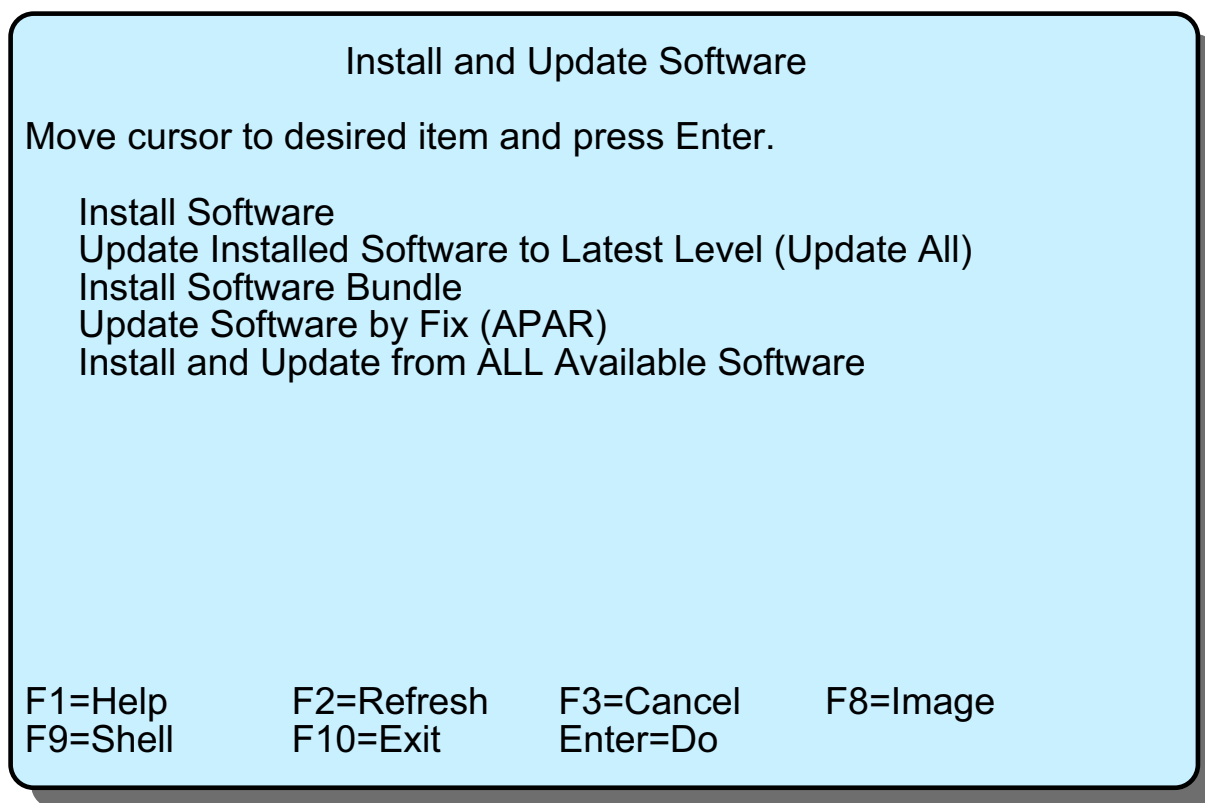


Figure 11-9. Install and Update Software

AU147.0

Notes:

Use the **smit install_update** fastpath to access this menu.

Install Software

This option allows you to install or update to the latest level of software available on the installation media. This allows you to install everything on the installation media if so desired. This is most commonly used to install optional software not currently installed on your system.

Update Installed Software to Latest Level

This option is the **smit update_all** fastpath. It enables you to update all of your currently installed software products. Only the existing installed products are updated; no new optional software will be installed. This is the most commonly used method to install a maintenance level (service) update.

Install Software Bundle

Use this option to install a software grouped into a bundle. For example, if you wish to install the Application Development bundle, choose this option.

Update Software by Fix (APAR)

An APAR is a number used to identify reported problems caused by a suspected defect in a program. A fix to an APAR can be made up of one or more fileset updates. These updates are obtained through the IBM Support Center or by using FixDist. The URL is **www.ibm.com/servers/aix/download**.

Install and Update from ALL Available Software

This option enables you to install or update software from all software available on the installation media. Use this option when none of the other menus fit your needs.

Install Software

Install Software

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

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

commit vs apply

↙

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

Figure 11-10. Install and Update from LATEST Available Software

AU147.0

Notes:

This SMIT dialog screen allows you to install all or selected software from the installation media. If any updates exist for these products, they will also be installed.

If **_all_latest** is left in the **SOFTWARE to install** line, everything on the installation media will be installed (except printers and devices). Usually, this line is used to indicate the new software you want to install. Use "list" (F4) to display all filesets on the media. From there, you can select the fileset, package or LPP that you want to install.

Access this menu using the SMIT fastpath **smit install_latest**.

The input device is usually CD-ROM, tape or diskette. However, it is also possible to install software that has already been loaded to disk. The directory **/usr/sys/inst.images** can be used for this purpose.

The preview option allows you to preview the results of the installation without actually performing the software install. The system will display information on space requirements and a list of software products and updates that will be installed.

If you choose no for **COMMIT software updates?**, then you must choose yes to **SAVE replaced files?**.

This is the line where you decide whether you want to **commit** or **apply** the software product. The default is **commit**. To **apply** the install you must change this line.

To perform and **update_all** the SMIT screen will be identical except in the **SOFTWARE to install** line you will see **[update_all]**.

Beginning with AIX 5.1, software license agreements are shipped and displayed electronically, saving paper and allowing for electronic software distribution in the future. If a product has an electronic license agreement, it must be accepted before software installation can continue.

Software Inventory

smit list_installed

List Installed Software and Related Information

Move cursor to desired item and press Enter.

List Installed Software
 List Applied but Not Committed Software Updates
 Show Software Installation History
 Show Fix (APAR) Installation Status
 List Fileset Requisites
 List Fileset Dependents
 List Files Included in a Fileset
 List Fileset Containing File
 Show Installed License Agreements

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

lspp command

- L lists the installed software
- h shows the history of a software product

Figure 11-11. Software Inventory

AU147.0

Notes:

Use the SMIT fastpath **smit list_installed** to access this menu. This menu provides information about the software and fixes installed on a system.

Most of the SMIT options on this menu actually execute the **lspp** command. The following command options can be used to view specific software information:

- **-l** Displays the name, level, state and description of the fileset
- **-h** Displays the installation and update history for the fileset
- **-p** Displays requisite information for the fileset
- **-d** Displays dependent information for the fileset
- **-f** Displays the names of the files added to the system during installation of the fileset.
- **-w** Lists the fileset that owns a file.

The option **Show Fix (APAR) Installation Status** executes the **instfix** command. This command will be discussed shortly.

List Installed Software

```
# lspp -l bos.*
```

Fileset	Level	State	Description
Path: /usr/lib/objrepos			
bos.adt.base	5.1.0.0	COMMITTED	Base Application Development Toolkit
bos.adt.include	5.1.0.0	COMMITTED	Base Application Development Include Files
bos.adt.lib	5.1.0.0	COMMITTED	Base Application Development Libraries
bos.diag.com	5.1.0.0	COMMITTED	Common Hardware Diagnostics
bos.diag.rte	5.1.0.0	COMMITTED	Hardware Diagnostics
bos.diag.util	5.1.0.0	COMMITTED	Hardware Diagnostics Utilities
bos.docregister.com	5.1.0.0	COMMITTED	Docregister Common
bos.docsearch.client.Dt	5.1.0.0	COMMITTED	DocSearch Client CDE Application Integration
bos.docsearch.client.com	5.1.0.0	COMMITTED	DocSearch Client Common Files
bos.docsearch.rte	5.1.0.0	COMMITTED	DocSearch Runtime
bos.dosutil	5.1.0.0	COMMITTED	DOS Utilities
bos.games	5.1.0.0	COMMITTED	Games

Figure 11-12. List Installed Software

AU147.0

Notes:

The **lspp** command is used to list the installed software on the system. The various options of the **lspp** command allow you to view selected information on the software installed.

The output of the **lspp** command displays the fileset name, the level of the product, its state (applied, committed), and a description of the product.

Other options include:

- -d displays filesets that are dependents on the specified software
- -f displays names of files added to the system during the installation of specified filesets
- -p lists requisite information for a specified fileset

Software Maintenance and Utilities

```
#smit maintain_software
```

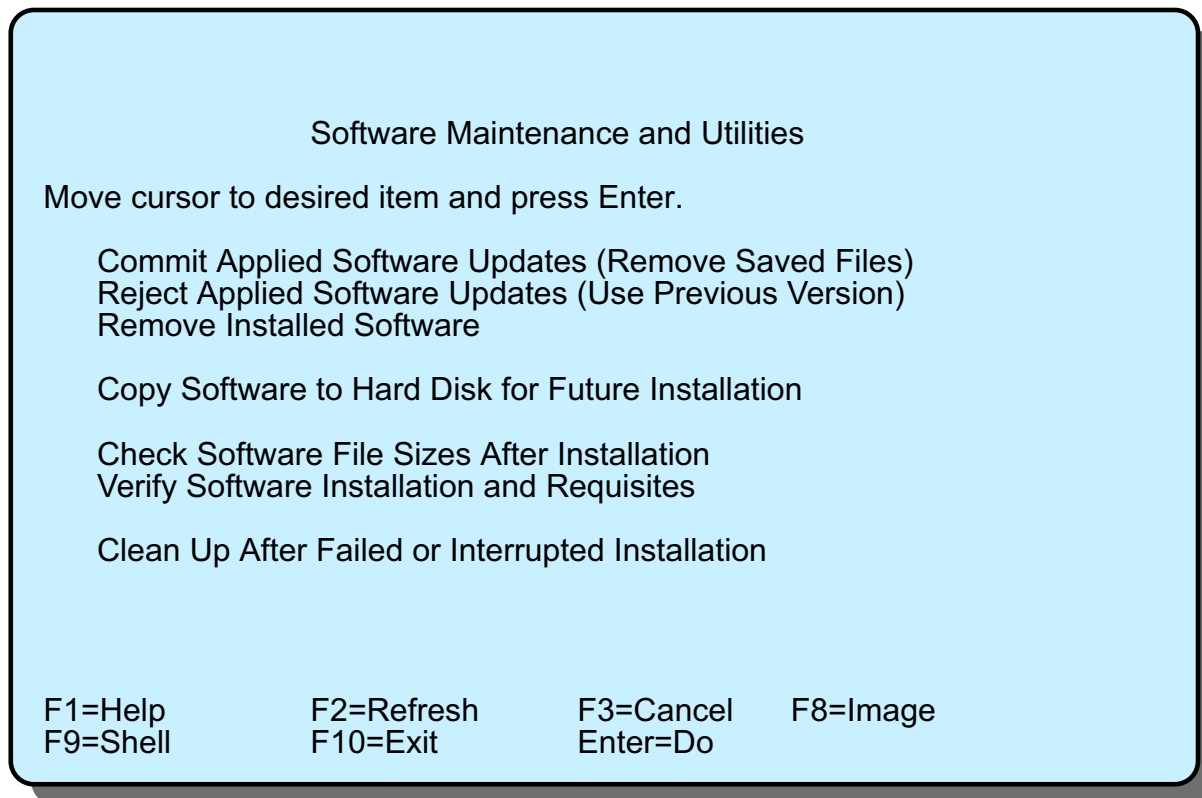


Figure 11-13. Software Maintenance and Utilities

AU147.0

Notes:

The fast path **smit maintain_software** allows you to commit, reject and remove software. You will also find the other menu items useful.

You can copy filesets from the installation media to the hard drive without actually performing an installation. This allows you to install it later without needing the original installation media.

If you are experiencing problems with your software, the check and verify options make the system run an analysis to determine if there is problem. It compares information stored on the disk to the information stored in ODM.

The clean up option "resets" your software installation back to the beginning after a failed install. A failed install is usually due to a power failure or a system shutdown occurring before the installation is complete. You will then need to start your installation/update over.

instfix Command

- Installs a fix

```
# instfix -k IX38794 -d /dev/rmt0.1
```

- Searches for a fix

```
# instfix -ik IX38794  
All filesets for IX38794 were found.
```

- Searches for a fix by keyword

```
# instfix -s SCSI -d /dev/rmt0.1
```

Figure 11-14. instfix Command

AU147.0

Notes:

The **instfix** command allows you to install a fix or a set of fixes without knowing any information other than the Authorized Program Analysis Report (APAR) number (which will be given to you by your Support Center) or other unique keywords identifying the fix.

The **instfix** command can also be used to determine if a fix is installed on your system.

Valid options with the command:

- T** Displays entire table of contents
- s** Search for and display table of contents entries containing the string
- k** Install filesets for a keyword or fix
- f** Install filesets for multiple keywords or fixes using an input file. Note that the output of the **-T** option produces a suitable input file format. **-f** will result in instfix using standard input.

- i** Use with **-k** or **-f** option to display whether fixes or keywords are installed. This option is for information only. Installation is not attempted when this option is used.
- a** Use only with **-i** to optionally display the symptom text associated with a fix
- d** Specify the input device (required for all but **-i**)

The examples on the foil do the following:

1. Install all filesets associated with fix IX38794 from the tape in the /dev/rmt0.1 drive.
2. Inform the user on whether fix IX38794 is installed.
3. List all keyword entries on the tape containing the string "SCSI".

Exercise: AIX Software Installation



Figure 11-15. Exercise: AIX Software Installation

AU147.0

Notes:

This lab will give you the opportunity to install filesets and show software installation history.

This exercise can be found in your Exercise Guide.

Checkpoint

1. Which of the following states can your software be in, in order for you to be able to use it? (select all that apply)
 - a. Applied state
 - b. Removed state
 - c. Install state
 - d. Commit state

2. What command is used to list all installed software on your system?

3. Which of the following can you install as an entity? (select all that apply)
 - a. Fileset
 - b. LPP
 - c. Package
 - d. Bundle

4. What is the difference between the SMIT menus: **Install Software** and **Update Installed Software to Latest Level (Update All)**?

Figure 11-16. Checkpoint

AU147.0

Notes:

Unit Summary

- AIX package naming conventions include the following terms:
 - LPP
 - Package
 - fileset
 - suffix
- Use the **lspp** command, SMIT or the Web-based System Manager to list all software products installed on the system

Notes:

Unit 12. Backup and Restore

What This Unit Is About

This unit describes how a system can be backed up and restored.

What You Should Be Able to Do

After completing this unit, you should be able to:

- Identify issues which have to be considered when deciding which backup policies to implement:
 - media to be used
 - frequency of the backup
 - type of backup
- List the different backup methods supported through SMIT and on the command line
- Create a customized installable system image backup
- Execute other useful commands to manipulate the backed up data on the media

How You Will Check Your Progress

Accountability:

- Checkpoint questions
- Exercises

References

Online	<i>AIX System Management Concepts: Operating System and Devices</i>
Online	<i>AIX System Management Guide: Operating System and Devices</i>
SG24-5765	<i>AIX 5L Differences Guide Version 5.1 Edition</i>

Objectives

After completing this unit, students should be able to:

- Identify issues which have to be considered when deciding which backup policies to implement:
 - Media to be used
 - Frequency of the backup
 - Type of backup
- List the different backup methods supported through SMIT and on the command line
- Create a customized installable system image backup
- Execute other useful commands to manipulate the backed up data on the media

Figure 12-1. Objectives

AU147.0

Notes:

Why Backup?

- Data is very important:
 - Expensive to recreate
 - Can it be recreated?
- Disaster recovery:
 - Hardware failure
 - Damage due to installation/repair
 - Accidental deletion
- Transfer of data between systems
- Reorganizing file systems
- Defragmentation to improve performance
- System image for installation
- Checkpoint (before/after upgrade)
- Long term archive

Figure 12-2. Why Backup?

AU147.0

Notes:

The data on a computer is usually far more important and expensive to replace than the machine itself. Many companies have gone out of business because they did not plan for disaster recovery. Backup to tape is the cheapest alternative but a duplicate disk or complete system would also provide protection and fast recovery from a disaster.

Backups should be taken before installing/maintaining hardware/software, in case a disk or files accidentally get damaged.

Backups are not just used for disaster recovery. One way of transferring a number of files from one machine to another is to back those files up to diskette, tape or a file on disk and then transfer that backup media to another machine.

When reorganizing the file systems on the disk you need to backup file systems so that they can be deleted and moved to another location.

If you are going to install a number of similar machines, or wish to be able to quickly reinstall a machine then a complete system image backup should be used.

Why Restore?

- Recover files
"I've just deleted file abc by mistake"
- Compare versions of files
"I'm sure it was fixed in 6.0.22"
- Recover system after disaster
"well now that the plumber has been in ..."
- Restore archived data
"you remember that project we did in 1997 ..."
- Shrink a file system

Figure 12-3. Why Restore?

AU147.0

Notes:

Types of Backup

Three types of backup:

- **System**
Records image backup of the operating system
- **Full**
Preserves all user data and configuration files
- **Incremental**
Records changes since previous backups
Must be used carefully
Very quick

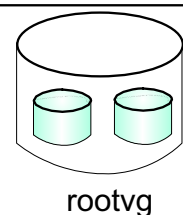


Figure 12-4. Types of Backup

AU147.0

Notes:

There are two types of incremental backups:

The first method is to do a full backup. For example, on Sunday, and then for the rest of the week, only backup the changes from the previous day. This method has the advantage of being quick, but there are a lot of tapes involved. Should one of the tapes be missing, you will have problems restoring using the remaining tapes.

The second method again involves taking a full backup on Sunday. However, the other days of the week backup only the changes made since the full backup; that is, since Sunday. The backups will take slightly longer than the previous method, and towards the end of the week, if most of your system has changed, then the time taken will be similar to a full backup. The restoration procedure does not depend on the tape from the previous day.

Backup Strategy

Backup all data that changes!

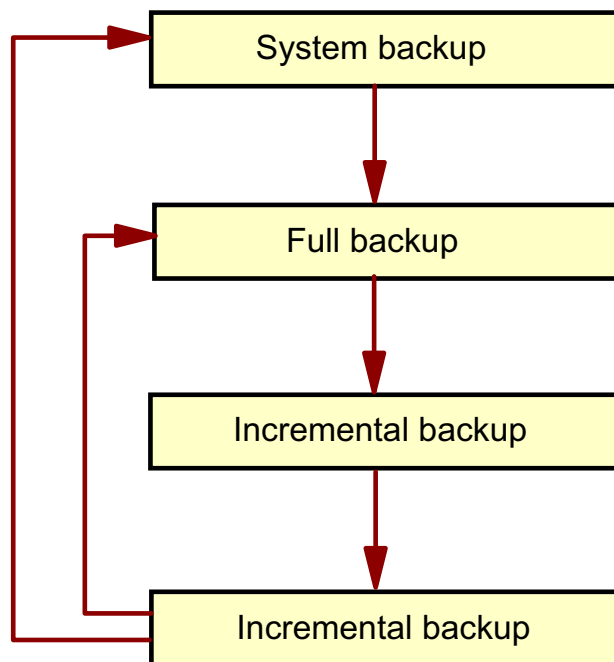


Figure 12-5. Backup Strategy

AU147.0

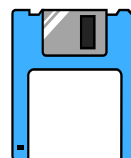
Notes:

Every organization sets its own backup policy, but a suggested strategy could include doing a system backup when the system is installed or upgraded, then a full backup periodically, perhaps weekly. The incremental backups can be run each day to copy files that have changed since the last incremental backup or the last full backup.

The key to any backup strategy is to ensure the data that is changing is saved regularly while trying to avoid interruptions to users' access to the data on your system.

Backup Devices - Diskette

/dev/fd0 Built in 3 1/2" diskette drive
/dev/fd1 Second diskette drive



Drive

	3 1/2"(1.44)	3 1/2"(2.88)
/dev/fd0	720 KB	720 KB
/dev/fd1	1.44 MB	2.88 MB
/dev/fd0.9	720 KB	720 KB
/dev/fd0.18	1.44 MB	1.44 MB
/dev/fd0.36	-	2.88 MB

Figure 12-6. Backup Devices - Diskette

AU147.0

Notes:

Diskettes can be used to backup data. Of course, this media is only practical when backing up small amounts of data.

The logical device name for a diskette drive is **/dev/fdx**. Your system will most likely have one diskette drive - **fd0**. When writing to a diskette, the highest density supported is the default value. The chart shows there are multiple logical names associated with the diskette drive that allow writing at different densities. To read the diskettes on a low-density drive, you must write using the low-density settings.

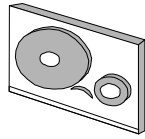
To format a diskette, use the **format** command. There is a **-l** options if you want to format at low density.

The **fcopy** command is used to copy diskettes (similar to the DOS diskcopy command).

Diskettes can also be formatted using DOS formatting with the command **dosformat**. AIX can read from and write to DOS diskettes using **dosread** and **doswrite**. There is also a **dosdir** to view the content of the diskette. To use these tools, the fileset **bos.dosutil** must be installed.

Backup Devices - Tape

- 4 mm DAT
- 8 mm



- 1/2 - inch
- DLT

	Low Capacity	Retention on Open	Rewind on Close
/dev/rmtx	no	no	yes
/dev/rmtx.1	no	no	no
/dev/rmtx.2	no	yes	yes
/dev/rmtx.3	no	yes	no
/dev/rmtx.4	yes	no	yes
/dev/rmtx.5	yes	no	no
/dev/rmtx.6	yes	yes	yes
/dev/rmtx.7	yes	yes	no

Figure 12-7. Backup Devices - Tape

AU147.0

Notes:

The most common device used for backups are tapes. AIX supports a variety of tape devices, tape subsystems and tape libraries. Here are some highlights of some of the tape technologies.

4 mm DAT (Digital Audio Tape) - can hold up to 40 GB of data with a data transfer rate of 6 MB/sec.

8 mm Tape - can hold up to 40 GB of data with a data transfer rate of 6 MB/sec.

1/4 inch Cartridge - can hold up to 4 GB with a data transfer rate of 380 KB/sec.

DLT - Digital Linear Tape - can hold up to 70 GB at a transfer rate of 10 MB/sec.

Magstar® - another tape technology usually used in tape subsystems. It offers up to 420 GB per cartridge with a transfer rate of 15 MB/sec.

For large scale backups, tape subsystems and tape libraries would be the sensible choice. For details on all tape devices supported on the RS/6000, go to:

www.rs6000.ibm.com/hardware/peripherals/storage/index.html

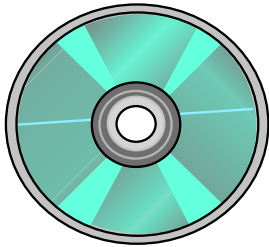
The tape devices use the logical device name of **rmtx** (raw magnetic tape). In the chart, you see the 7 additional logical names assigned to each tape device. These names control tape device characteristics:

- Write at low capacity
- Retension the tape (fast forward and rewind before starting the operation)
- Rewind the tape at the finish of the operation

The most common devices that are used are **rmtx** and **rmtx.1**. For most tape operations, high capacity and no retension are the norm. Whether or not you want to rewind the tape depends on your particular operation.

Tapes are formatted at the factory. Tape movement can be controlled using **tctl** or **mt** commands. And, if there two tape devices, **tcopy** will allow tape to tape transfers. Details on these commands are discussed later.

Backup Device - Read/Write Optical Drive



- Use with CD-ROM file system for read only operations
- Use with journal file systems for read/write operation

Figure 12-8. Backup Device - Read/Write Optical Drive

AU147.0

Notes:

AIX supports read/write optical drives as well as standard CD-ROM. The R/W Optical drives support CD-ROM file systems and JFS file systems. If the optical drive is mounted as a CD-ROM file system, it will be read only.

CD-ROM file system - To use the information on the R/W optical drive like a standard CD-ROM. The steps to access the data is the same as with a regular CD-ROM.

1. Create the file system (**smit crcdrfs -or- crfs -v cdrfs -p ro -d DeviceName**)
2. Mount the file system (**mount mount_point**)

JFS file system - To use the R/W optical drive as a read/write device, you must create a volume group using the same commands that used with a hard drive.

1. Make the VG (**smit mkvg -or- mkvg -f -y VGName -d 1 DeviceName**)
2. Create a file system (**smit crfs -or- crfs -v jfs -g VGName -a size=SizeFileSystem -m MountPoint -A AutomaticMount -p rw**)

3. Mount the file system

The optical drive VG must be wholly contain on the single optical disk. It cannot span beyond one optical drive.

Backup Menu

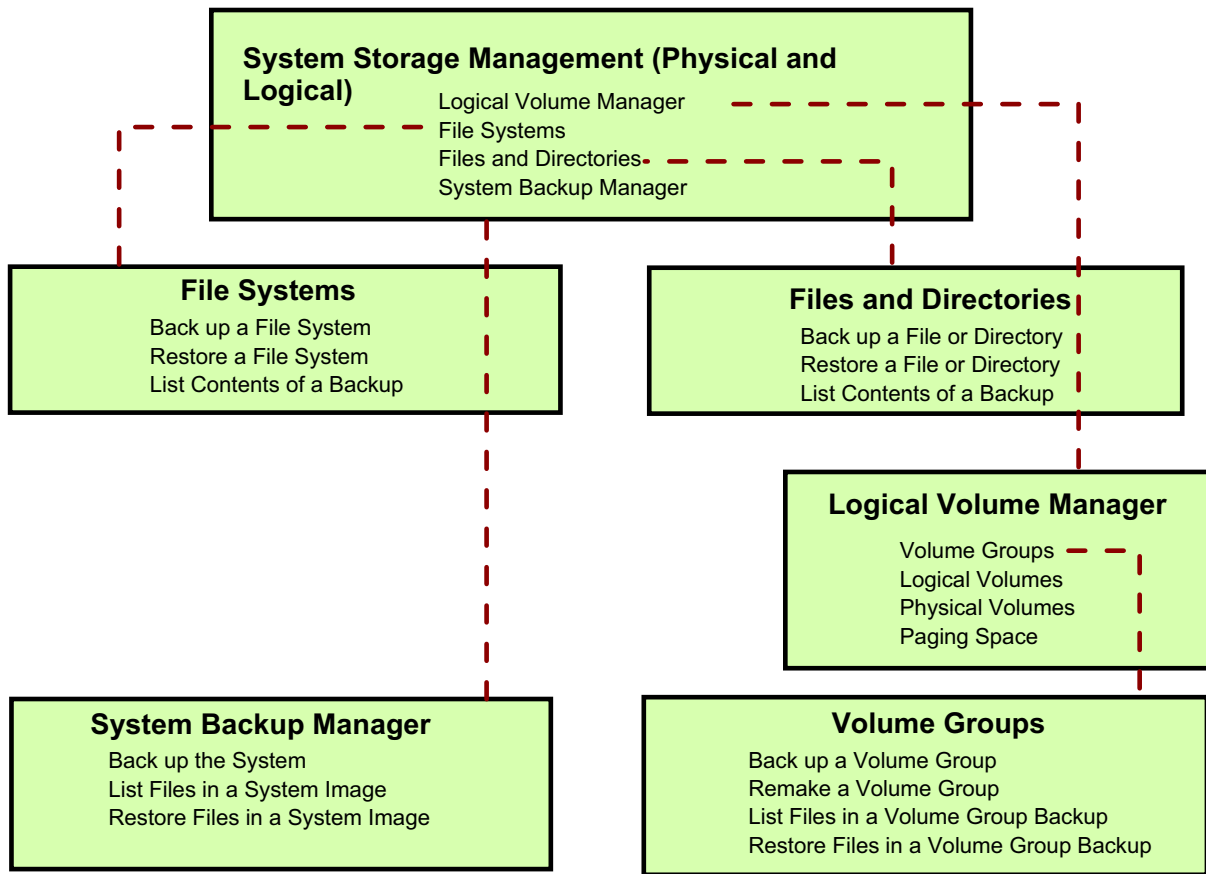


Figure 12-9. Backup Menu

AU147.0

Notes:

Please note that SMIT screens only show the backup options and not all the options. Backups can also be performed using the Web-based System Manager.

rootvg Backup Process - mksysb

- Backs up rootvg only
- Unmounted file systems are not backed up
- Bootable tape is created in backup format
- Provides facilities for a non-interactive installation
- Saves system-created paging space definitions
- Saves logical volume policies
- There should be minimal user and application activity

Figure 12-10. rootvg Backup Process - mksysb

AU147.0

Notes:

mksysb provides the following functions:

- Saves the definition of the paging space
- Provides a non-interactive installation that gives information required at installation time through a data file
- Saves the inter/intra policy for the logical volumes
- Saves map files for logical volumes if requested by the user
- Provides the ability to shrink the file system and logical volume in a volume group at install time
- Saves the file system block size and number of bytes in inodes
- Saves the file system compression characteristics
- Saves striped logical volume attributes in AIX V4.2 and later
- Allows the user to restore single or multiple files from a system image

The volume group image is saved in **backup** format. The **rootvg** is created as an installable image.

If the **mksysb** command is used for a backup of the source system, it is considered a system backup. However, if the intent of the backup is to provide a customized system for use on other machines, the **mksysb** is considered a clone. Cloning means preserving either all or some of a system's customized information for use on a different machine. The **mksysb** files are system specific.

If the **mksysb** tape, by itself, is used to clone a machine that is not a hardware clone, it may not work or may not provide support for hardware devices unique to the new machine. For example, loading a **mksysb** image made from a uniprocessor machines will not install correctly on a multiprocessor machine because they use different AIX filesets. However, this is an easy problem to resolve. In addition to the **mksysb** tape, you will also need an AIX installation CD to provide the filesets needed by the other machine. If the CD is also available, during installation the proper fileset will automatically be selected and loaded from the CD.

If a system backup is being made to install another system or to reinstall the existing system, a customer can predefine install information so questions at installation time are already answered. This keeps user interaction at the target node to a minimum. The system backup and BOS Install interact through several files. **mksysb** saves the data used by the install through taking a snapshot of the current system and its customized state.

The utilities for creating a system backup include messages, SMIT menus, and commands that are packaged in the **bos.sysmgt.sysbr** option of the **bos.sysmgt** package. They are separately installable, although this fileset is automatically installed in beginning with AIX V4.3. If your system does not include the **mksysb** command, install the **bos.sysmgt.sysbr** option to get **mksysb** and the **bos** install routines.

/image.data File for rootvg

```

image data:
    IMAGE_TYPE=bff
    DATE_TIME=Wed May 9 15:07:28 PDT 2001
    UNAME_INFO=AIX sys2 1 5 000930134C00
    PRODUCT_TAPE=no
    USERVG_LIST=
    OSLEVEL=4.3.1.0
logical_volume_policy:
    SHRINK=no
    EXACT_FIT=no
ils_data:
    LANG=C
#Command used for vg_data, /usr/sbin/lsvg
lsvg_data:
    VGNAME=rootvg
    PPSIZE=4
    VARYON=yes
    VG_SOURCE_DISK_LIST=hdisk0 hdisk1
#Command used for source_disk_data: /usr/sbin/bootinfo
source_disk_data: (stanza is repeated for each disk in rootvg)
    LOCATION=(disk location)
    SIZE_MB=(size of disk in megabytes)
    HDISKNAME=(disk name)
#Command used for lv_data; /usr/sbin/lslv
lv_data: (stanza for each logical volume in rootvg)
    .
    .
fs_data: (stanza for each MOUNTED filesystem in rootvg)

```

Figure 12-11. /image.data File for rootvg

AU147.0

Notes:

The **/image.data** file has information used by bos install for creating the target **rootvg**. The **/image.data** file, while being flexible, is not intended for every user. The **mksysb** utility calls **mkszfile** (if **-i** or **-m** options specified) to create an **image.data** file from existing information. If users edit the **image.data** file, then they should call the **mksysb** command without the **-i** or **-m** options to use the existing **image.data** file.

In general, the stanza information found in the **/image.data** file is generated using one of the **lsxx** commands; that is, **lsvg** for the volume group data, **lslv** for the logical volume data, **lsjfs** for the file system data and so forth. Some fields like **LV_MIN_LPS** are created through calculations and are not directly available from commands.

The user can provide additional processing (if required) after bos install by using the **BOSINST_FILE=** field in the **post_install_data** stanza or through their own program. The **BOSINST_FILE** and **SHRINK=** fields must be edited by the user before calling **mksysb** if changes are desired.

logical_volume_policy

Contains information to be used at reinstall time. The **SHRINK=** field when set to YES, causes the system to create logical volumes and file systems in the volume group based on the values set for each with the **LV_MIN_LPs** and **FS_MIN_SIZE** fields. This option is always set to NO when created by **mkszfile**. The **EXACT_FIT=** field when set to YES, causes the system to place the logical volumes on the disk according to the physical partition maps that were generated with the **-m** flag of the **mksysb** or **mkszfile** command.

If the only thing you wish to change is the SHRINK or EXACT_FIT field, there is no need to edit this file. Both of these settings can be controlled by the menus presented during the installation of a mksysb.

vg_data

Contains information about the Volume Group. The **VG_SOURCE_DISK_LIST=** field specifies the disks that bos install will use on a best effort basis to place the volume Group. If the **EXACT_FIT=** field is set to YES, the user will be warned before installation begins.

lv_data

Contains information about logical volumes. This type of data stanza is also used to contain paging space information. Information about striped logical volumes and large file enabled file systems will be placed in this stanza in AIX V4.2 and later.

/bosinst.data File for rootvg

```

control_flow:
    CONSOLE =
    INSTALL_METHOD = overwrite
    PROMPT = yes
    EXISTING_SYSTEM_OVERWRITE = no
    INSTALL_X_IF_ADAPTER = yes
    RUN_STARTUP = yes
    RM_INST_ROOTS = no
    ERROR_EXIT =
    CUSTOMIZATION_FILE =
    TCB = no
    INSTALL_TYPE =
    BUNDLES =
    RECOVER_DEVICES = yes
    BOSINST_DEBUG = no
    ACCEPT_LICENSES =
    INSTALL_64BIT_KERNEL = no
    INSTALL_CONFIGURATION = Default
    DESKTOP = CDE

target_disk_data:
    LOCATION =
    SIZE_MB =
    HDISKNAME =

locale:
    BOSINST_LANG =
    CULTURAL_CONVENTION =
    MESSAGES =
    KEYBOARD =
  
```

Figure 12-12. /bosinst.data File for rootvg

AU147.0

Notes:

This file allows the administrator to specify the requirements at the target system and how the user interacts with the target system. It provides flexibility by allowing different target hardware to use the same backup image. The system backup utilities simply copy the **/bosinst.data** into the second file in the **rootvg** on the **mksysb** tape. If this file is not in the root directory, the **/usr/lpp/bosinst/bosinst.template** is copied to the **/bosinst.data**.

The sample file shown above has been condensed to highlight key areas. The actual file is well documented with comments contained within the file.

The **control_flow** stanza contains variables that control the way the installation program works.

CONSOLE specifies the full path name of the device you want to use as the console. For example, **/dev/lft0**.

INSTALL_METHOD specifies a method of installation: migration, preserve or overwrite.

PROMPT specifies whether the installation program uses menus from which you can make choices. You must fill in values for all variables in the locale and control_flow stanzas if you set the PROMPT variable to no with two exceptions: the ERROR_EXIT and CUSTOMIZATION_FILE variables, which are optional.

EXISTING_SYSTEM_OVERWRITE confirms that the install program will overwrite existing files. This variable is only applicable for non-prompted overwrite installation.

INSTALL_X_IF_ADAPTER installs AIXwindows if the selected console is a graphical terminal.

RUN_STARTUP starts the Installation Assistant on first boot after the BOS installation completes.

RM_INST_ROOTS removes all files and directories in the `/usr/lpp/*/inst_roots` directories.

ERROR_EXIT starts an executable program if an error occurs in the installation program.

CUSTOMIZATION_FILE specifies the path name of a customization file you create. The customization file is a script that starts immediately after the installation program concludes.

TCB specifies whether you want to install the Trusted Computing Base.

INSTALL_TYPE specifies what software to install on the machine. The values are full (full-function configuration), client (client configuration), and personal (personal workstation configuration). The full configuration includes all the software in client and personal. Change full to client or personal if you want to install one of these subsets of the full-function configuration.

BUNDLES specifies what software bundles to install. Type the bundle names separated by a space between each name.

RECOVER_DEVICES specifies whether to reconfigure the devices.

BOSINST_DEBUG specifies whether to show debug output during BOS installation.

ACCEPT_LICENSES specifies whether to accept software license agreements during the BOS installation.

INSTALL_64BIT_KERNEL specifies whether to enable the 64-bit kernel and JFS2 filesystems.

INSTALL_CONFIGURATION specifies Default or Minimal installations.

DESKTOP specifies the desktop to be installed. Choices include CDE (the default), NONE, GNOME, and KDE. If you choose GNOME or KDE, you will install open-source software.

The target_disk_data stanza contains variables for disks in the machine where the program will install BOS. The default **bosinst.data** file has one target_disk_data stanza, but you can add new stanzas to install BOS on multiple disks, one stanza for each disk. The installation program determines a target disk by checking the variables in hierarchical order. For example, if the LOCATION variable specifies a location code, the program installs BOS on

that disk, regardless of the remaining variables. If you accept the default values, which are blank, the installation program chooses a target disk based on the initial hardware query.

LOCATION specifies a location code for the disk where the program will install BOS.

SIZE_MB specifies the formatted size of the disk (in megabytes) where the program will install BOS.

HDISKNAME specifies the path name of the target disk.

The locale stanza contains variables for the primary language the installed machine will use.

BOSINST_LANG specifies the language the installation program uses for prompts, menus and error messages.

CULTURAL_CONVENTION specifies the primary locale to install

MESSAGES specifies the locale for the messages catalogs to install.

KEYBOARD specifies the keyboard map to install.

You must install the Base Operating System (BOS) before you can access and modify the default **bosinst.data** file. Once you have installed BOS, retrieve and edit the file like any other ASCII file. There are basically three different ways that you will use a customized **/bosinst.data** file.

- Customize the **bosinst.data** file, then create a backup image of the system to use in subsequent installations from a backup tape
- Customize a **bosinst.data** file for each client you want to install via the network
- Customize the **bosinst.data** file, then copy the modified file to a diskette that supplements your installation medium, either tape or CD-ROM. Note that if you use this method then you must also have on your diskette a file called signature. The file signature must contain the word data.

With both the **/image.data** and the **/bosinst.data** files created, the reinstallation of AIX Version 4 and later can be made unattended.

The procedure to accomplish this is as follows:

1. Edit the **bosinst.data** file as follows:
 - a. Set **CONSOLE=/dev/lft0** or **CONSOLE=/dev/tty0** according to your system.
 - b. Set **PROMPT=no**
 - c. Set **EXISTING_SYSTEM_OVERWRITE=yes**
 - d. Set **RUN_STARTUP=no**
2. Create the **signature** file:

```
echo "data" > signature
```
3. Create the floppy diskette with

Is ./bosinst.data ./signature | backup -iqv

4. Run the command **mksysb /dev/rmt0.1**

The assumption for 3) is that there is already a preformatted diskette in the drive.

The assumption for 4) is that there is a tape in the first tape drive and that it will be large enough to hold all the data for the root volume group.

Having completed these steps, the diskette is usable with the backup tape.

The diskette is put in the target system's diskette drive prior to starting the installation of the target machine. When the target machine is booted from the install media, the BOS install program will use the diskette file rather than the default **/bosinst.data** file shipped with the install media.

The purpose of the **signature** file is to verify that this really is a **bosinst.data** diskette.

You can break out of an unassisted install by typing **000 <Enter>** when you see the startup symbols `\ | /` on the display.

rootvg Image Backup

smit mkysb

Back Up the System

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

[Entry Fields]

WARNING: Execution of the mkysb 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	+
Generate new /image.data file?	yes	+
EXPAND /tmp if needed?	no	+
Disable software packing of backup?	no	+
Number of BLOCKS to write in a single output (Leave blank to use a system default)	[]	#

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

Figure 12-13. rootvg Image Backup

AU147.0

Notes:

The options on the menu are:

Creation of a MAP File

This option generates a layout mapping of the logical-to-physical partitions for each logical volume in the volume group. This mapping is used to allocate the same logical-to-physical partition mapping when the image is restored.

EXCLUDE Files?

This option will exclude the files and directories listed in the `/etc/exclude.rootvg` file from the system image backup.

List files as they are backed up?

Change the default to see each file listed as it is backed up. Otherwise, you will see a percentage-completed progress message while the backup is created. This option is supported at AIX V4.2 and later.

Generate new /image.data file?

If you have already generated a new /image.data file and don't want a new file created, change the default to no.

EXPAND /tmp if needed?

Choose 'yes' if the /tmp file system can automatically expand if necessary during the backup.

Disable software packing of backup?

The default is no, which means the files are packed before they are archived to tape. Files that cannot be compressed are placed in the archive as is. Restoring the archive automatically unpacks the files packed by this option. If the tape drive you are using provides packing or compression, set this field to yes. This option is supported at AIX V4.2 and later.

Number of BLOCKS to write in a single output

This specifies the number of 512 bytes to write in a single output operation, referred to as the block size. If a number is not specified, the **backup** command uses a default value appropriate for the physical device selected. Larger values result in larger physical transfers to tape devices. The block size must be a multiple of the physical block size of the device being used.

Only mounted file systems in **rootvg** are backed up. Use one of the other **backup** commands to backup other volume groups.

mksysb Image

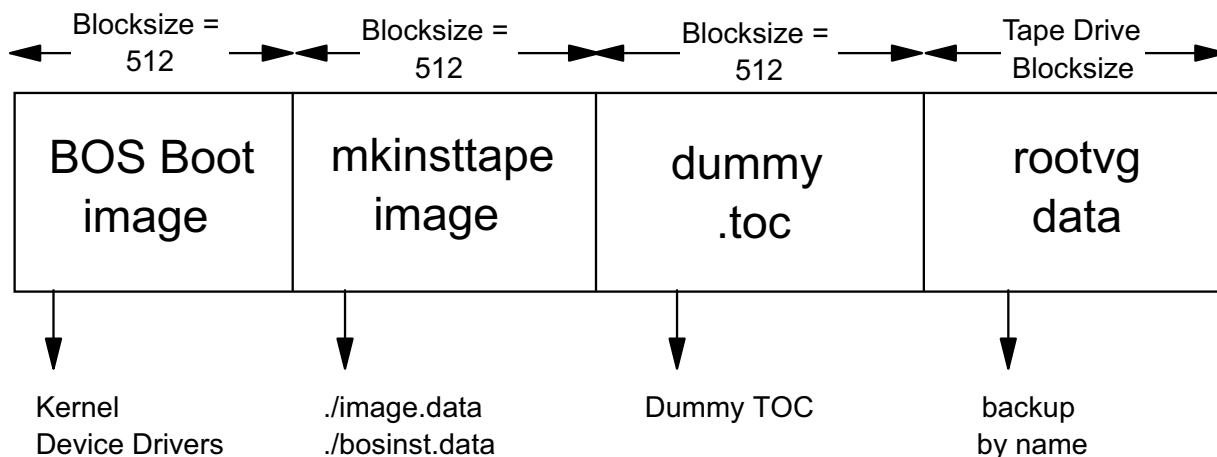


Figure 12-14. mksysb Image

AU147.0

Notes:

This shows the tape layout of a mksysb image.

BOS Boot Image - contains a copy of the system's kernel and device drivers needed to boot from the tape.

mkinsttape Image

`./image.data` - holds the information needed to recreate the root volume group and its logical volumes and file systems.

`./bosinst.data` - contains the customizable install procedures and dictates how the BOS install program will behave. This file allows for the non-interactive installs.

Dummy TOC - used to make mksysb tapes have the same number of files as BOS install tapes.

Rootvg Data - contains all the data from the backup. This data is saved using the backup command which will be discussed shortly.

Non-rootvg Backup

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	[]	+/-
* VOLUME GROUP to back up	[]	+
List files as they are backed up?	no	+
Generate new vg.data file?	yes	+
Create MAP files?	no	+
EXCLUDE files?	no	+
EXPAND /tmp if needed?	no	+
Disable software packing of backup?	no	+
Number of BLOCKS to write in a single output (Leave blank to use a system default)	[]	#

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

Figure 12-15. Non-rootvg Backup

AU147.0

Notes:

The savevg SMIT screen looks very similar to the mksysb SMIT screen. This is because they are both performing a volume group backup except mksysb creates bootable images. The command that SMIT is using is savevg. Listed below are some of the differences between the mksysb screen and this screen.

VOLUME GROUP to back up - name the volume you wish to back up

Generate new **vg.data** file - This file is equivalent to the **image.data** file for rootvg. Unless you have a customized file that you wish to use, let SMIT (**savevg**) create this file for you. The file will be called **/tmp/vgdata/vg_name/vg_name.data**. This file can be created by running **mkvgdata vg_name**.

EXCLUDE files - this allows you exclude file (during the backup) located in mounted file systems within the volume group. Create a file called **/etc/exclude.vg_name** and add the list of filenames that are not wanted.

Restoring a mksysb (1 of 2)

- Boot the system in install/maintenance mode:

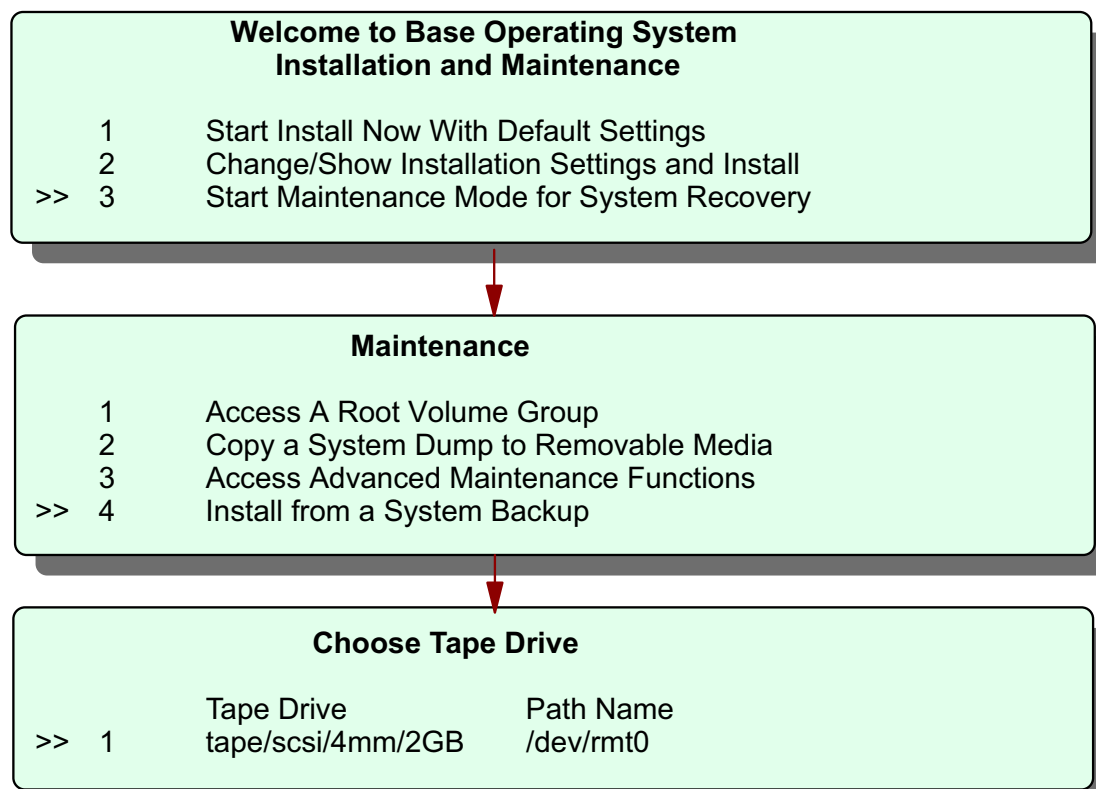


Figure 12-16. Restoring a mksysb (1 of 2)

AU147.0

Notes:

To restore a mksysb image, boot the machine just as if you were performing an installation. Be sure your boot list 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 will boot from the tape and prompt 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.

Also, you can boot from an installation CD. The CD will present the same screens. Just be sure to put the mksysb tape in the tape drive before answering the last question.

Select 3 Start Maintenance Mode for System Recovery, then 4 Install from a System Backup and select the tape drive that contains the mksysb tape.

Restoring a mksysb (2 of 2)

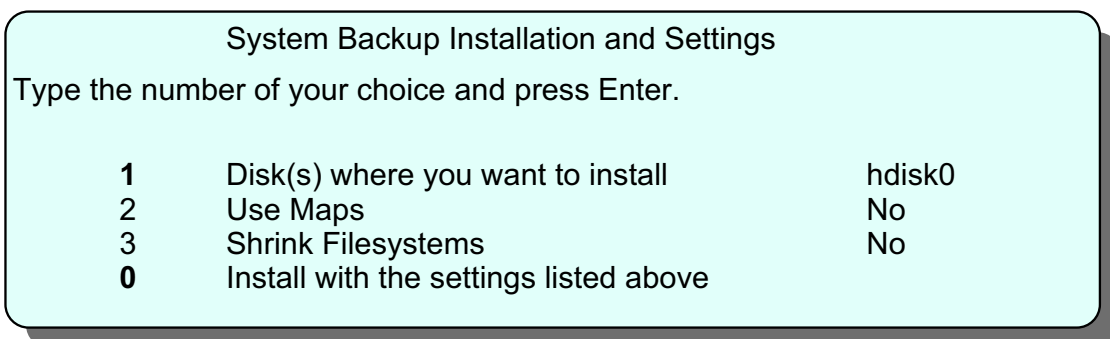
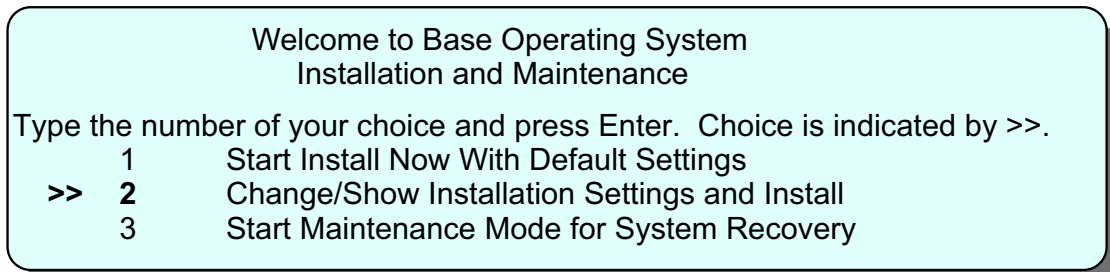


Figure 12-17. Restoring a mksysb (2 of 2)

AU147.0

Notes:

After selecting the tape drive (and a language, which is not shown on the visuals), you will return to the Installation and Maintenance menu. Now select option 2.

From the System Backup and Installation and Settings menu, select 1 and provide the disks where you want to install. Be sure to select all disks where you want to install. If your rootvg was mirrored, you will need to select both disks.

Two other options can be enabled in this menu:

1. The option **Use Maps** lets you use the map file created (if you did create them) during the backup process of the mksysb tape. The default is no.
2. 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.

At the end, select option 0 (Install with the settings above). Your mksysb image will be 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.

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	+	
PHYSICAL VOLUME names (Leave blank to use the PHYSICAL VOLUMES listed in the vname.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)	[]	#	

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

Figure 12-18. Restoring a non-rootvg Volume Group

AU147.0

Notes:

SHRINK the filesystems - When restoring the volume group, like with rootvg, you have the option to shrink the file system contained in the volume group. Always be sure to check the size of the file systems after the restore is complete. You might need to increase them once again.

PHYSICAL VOLUME names - If left blank, the VG will go back to the disks it came from. If you need to change the location, this is the place to do it.

Use existing MAP files - If map files exist, they are used by default during recovery. If you don't want to use them, set this selection to no.

Physical partition SIZE determined based on disk size - AIX will properly size the PPs for the disk it is using. If you prefer to have a larger PP size than the standard, you can set it here. If for example you have a 4.5 GB drive, the partition size will be 8 MB. If you want it to be 16 MB, you can set it here.

This characteristic makes it easy to resize the partitions in a VG. If you want to move the VG to a larger disk, the PP will adjust automatically during the restore.

Activity: savevg



Figure 12-19. Activity: savevg

AU147.0

Backup and Restore - Activity

Instructions

In this activity, you will create a backup of **datavg** and save it to the a file in **rootvg**.

1. Create a file system (called **/home/savevg**) in **rootvg** that will hold the backup image and mount it. Make sure the file system is in rootvg!
2. Back up the **datavg** volume group to **/home/savevg/bkup**.
3. Make sure the backup complete successfully before doing this step.
Remove the datavg volume group. You will need to unmount all mounted file systems in datavg first.
4. Recover the volume group from your backup.
5. Verify that the volume group is back.
6. Examine the contents of the file **/tmp/vgdata/datavg/datavg.data**. What is the file used for?

Instructions with Hints

In this activity, you will create a backup of **datavg** and save it to the a file in **rootvg**.

1. Create a file system (called **/home/savevg**) in **rootvg** that will hold the backup image and mount it. Make sure the file system is in **rootvg**!

```
# smit fs
```

```
Select Add/Change/Show/Delete File Systems
```

```
Select Journalled File Systems
```

```
Select Add a Journalled File System
```

```
Select Add a Standard Journalled File System
```

```
Select rootvg for the volume group name.
```

```
Volume group name rootvg
```

```
* SIZE of file system (in 512-byte blocks) [16384]#
```

```
* MOUNT POINT [/home/savevg]
```

```
OK or ENTER
```

Use <F3> to move back to the File Systems menu.

```
Select Mount a File System
```

```
FILE SYSTEM name [/home/savevg]
```

```
OK or ENTER
```

```
OR
```

```
# crfs -v jfs -g rootvg -a size=16384 -m /home/savevg
```

```
# mount /home/savevg
```

2. Back up the **datavg** volume group to **/home/savevg/bkup**.

```
# smit savevg
```

```
* Backup Device or FILE [/home/savevg/bkup]
```

```
* VOLUME GROUP to back up [datavg]
```

```
OK or ENTER
```

```
OR
```

```
# savevg -f /home/savevg/bkup -i datavg
```

3. Make sure the backup complete successfully before doing this step.
Remove the **datavg** volume group. You will need to unmount all mounted file systems in **datavg** first.

```
# smit vg
```

```
Select Remove a Volume Group
```

```
*VOLUME GROUP name [datavg]
```

```
OK or ENTER
```

```
OR
```

```
# reducevg -df datavg hdiskn
```


4. Recover the volume group from your backup.

```
# smit restvg
```

```
* Restore DEVICE or FILE[/home/savevg/bkup]
SHRINK the filesystems?no
PHYSICAL VOLUME names[hdiskn]
OK or ENTER
```

OR

```
# restvg -q -f /home/savevg/bkup hdisk1
```

5. Verify that the volume group is back.

```
# smit lsvg
```

List Content of a Volume Group

```
VOLUME GROUP name[datavg]+
List OPTIONstatus+
```

OK or ENTER

<F10> to exit SMIT

6. Examine the contents of the file `/tmp/vgdata/datavg/datavg.data`. What is the file used for?

```
# more /tmp/vgdata/datavg/datavg.data
```

END

backup by File Name

```
backup -i [-q] [-p] [-v] [-f device]
```

- q media is ready
- p compress files which are less than 2 GB
- v verbose - display filenames during backup

```
Filenames are read from standard input
```

Figure 12-20. backup by File Name

AU147.0

Notes:

The **backup** command is the preferred command for making backups of AIX files and directories. **Backup** supports two different methods - **backup by name** and **backup by inode** (also call a file system backup). When performing a backup by name, the files must be in a mounted file system to be backed up. Backups by inode backup file systems even when they are unmounted.

The syntax is shown for the backup by name.

The **-i** option is used to indicate a backup by name.

The **-q** option is for quiet. It suppresses the comment, press Enter to continue, that displays when the backup command is executed. This is helpful for automated backups.

The **-p** option compresses files during the backup process. It can only compress files smaller than 2 GB. Also, don't use the **-p** option on active file systems. Modifying a file during the compression may corrupt the file and make it unusable on recovery.

The **-v** option displays the files and pathnames to standard out as they are backed up.

backup by File Name Examples

Read input from a file

```
#cat listfile
/home/roy/file1
/home/roy/file2
/home/roy/file3
# backup -iqvf /dev/rmt0 < listfile
```

Use **find** to generate list

```
# find /home/roy | backup -iqvf /dev/rmt0
# cd /home/roy
# find . | backup -iqvf /dev/rmt0
```

Relative vs Full Filenames will impact
Location of Files on Recovery !

Figure 12-21. backup by File Name Examples

AU147.0

Notes:

The list of files backup uses can be supplied by a file or by commands. The graphics provides a sample of each.

In the first example, the file **listfile** contains the files we want to back up. That is fed into the **backup** command by using a redirection (<).

In the second example, there are two examples that can be used to backup the same data using the **find** command to generate the file list. Both commands will back up the files stored in /home/roy.

Even though both **find** examples will save the same data, the filenames will be stored differently. There are two types of filenames - relative and full (or absolute). The difference is a full pathname shows the location referenced from the root directory. Basically, the name starts with a slash (/). The relative pathname shows the location referenced by the current directory. This distinction will be important when you try to recover the data.

Full pathname backups will restore to the same location in the directory structure since their position is referenced from the root directory. But, a relative pathname file will be

restored based upon the current directory when the restore command is issued. Full pathnames provide certainty of location and relative pathnames provided flexibility.

backup a File or a Directory

smit backfile

Backup a File or Directory

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

[Entry Fields]

This option will perform a backup by name.

* Backup DEVICE	[/dev/fd0]	+ /
* FILE or DIRECTORY to backup	[.]	
Current working DIRECTORY	[]	/
Backup LOCAL files only?	yes	+
VERBOSE output?	no	+
PACK files?	no	+

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

Figure 12-22. backup a File or a Directory

AU147.0

Notes:

You can perform backups through SMIT.

FILE or DIRECTORY to backup - This is a parameter for the **find** command that will run behind the scenes. The dot (.) indicates to start the **find** command from the current directory. This will provide a relative pathname backup. If a full pathname was used here (like /home/roy) then the names would be stored with full pathnames.

Current working DIRECTORY - performs a **cd** to that directory before starting the backup. If you want a backup from the current directory (.), and you want to make sure you are in the right directory, you can put the name of the directory here.

Backup LOCAL files only - ignores any network file systems. Files backed up will be from the local system only.

backup a File System by Inode

Syntax:
backup [-u] [-level] [-f device] filesystem

Levels provide incremental backups:

-0 Full File system back up

-1,-2, ... backup changes since level -1

/etc/dumpdates contains a backup history

-u updates /etc/dumpdates

```
# backup -u -1 -f /dev/rmt0 /home
```

Figure 12-23. backup a File by Inode

AU147.0

Notes:

If you do not specify the **-i** option, the **backup** command will perform a file system backup by inode.

The **-level** option allows you to perform incremental backups. The **-0** level will backup all files in the file system. The **-1** level will backup all files changed since the last **-0** backup, and so on. (If you do not specify a level, **-9** is assumed.)

You should **unmount** the file system before you use backup by inode. This is recommended for user-created logical volumes **/dev/lvnn** and system logical volumes (other than **/**) otherwise errors in mapping on restore may occur. This is not required for **/** (it's difficult to **unmount** it in any case!). If you do not specify a file system, the root **/** will be backed up. The file system parameter can specify either the physical device name or the directory on which the file system is mounted. You must have read access to the file system device in order to perform backups by inode.

The **-u** option causes **backup** to update the **/etc/dumpdates** file to record the date and level of the last inode **backup** for that file system. This file holds file system backup

information for the backup command. The information included in this file is the name of the file system, the level number of the last backup, and the date of the incremental backup.

Incremental backup Example

Sun	Mon	Tue	Wed	Thur	Fri	Sat
					1 level 0	2
3	4 level 6	5 level 6	6 level 6	7 level 6	8 level 3	9
10	11 level 6	12 level 6	13 level 6	14 level 6	15 level 0	16
17	18 level 6	19 level 6	20 level 6	21 level 6	22 level 3	23
24	25 level 6	26 level 6	27 level 6	28 level 6	29 level 0	30
31						

Figure 12-24. Incremental backup Example

AU147.0

Notes:

You can use the **-level** parameter to back up either all files on the system (a full backup) or only the files that have been modified since a specific full backup (an incremental backup). The possible levels are 0 through 9. If you do not supply a level the default level is 9. A level 0 (zero) backup includes all files in the file system. An n level backup includes all files modified since the last n-1 backup or lower. The levels, in conjunction with the **-u** flag, provide a way to maintain a hierarchy or incremental backups for each file system.

The calendar shows how this can be accomplished.

- A level 0 backup is performed on the first Friday, and thereafter every other Friday.
- A level 6 is performed on each day of the week except on the Fridays that a full backup is not carried out, when a level 3 backup is performed.
- During the first full week, the level 6 backup on Monday will backup all files modified since the level 0 backup on the previous Friday. Each level 6 backup Tuesday through Thursday will backup all files that have been modified since the last n-1 backup or lower (in this case level 0).

- The first level 3 backup of the month will backup all files modified since the level 0 backup. The remaining level 3 backups will backup all those files modified since the last level 0 backup.
- During the second full week, the level 6 backups on Monday through Thursday will backup all files that were modified since the last level 3 backup. The level 3 backups on Fridays will backup everything since the last level 0.

Backup a File System by Inode

```
# smit backfilesys
```

Backup a Filesystem

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

[Entry Fields]

This option will perform a backup by inode.

* FILESYSTEM to backup	[]	+/-
* Backup DEVICE	[/dev/fd0]	+/-
Backup LEVEL (0 for a full backup)	[0]	#
RECORD backup in /etc/dumpdates?	no	+

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

Figure 12-25. Backup a File System by Inode

AU147.0

Notes:

SMIT provides a front-end for file system backups as well.

Each line represents the options from the command line.

restore Command

- List files on media (Verify the backup)
`restore -T [-q] [-v] [-f device]`

```
# restore -Tvf /dev/rmt0
```

- Restore individual files.
`restore -x [-q] [-v] [-f device] [file1 file2 ..]`

```
# restore -xvf /dev/rmt0 /home/mike/manual/chap1
```

- Restore complete file system
`restore -r [-q] [-v] [-f device]`
Restore backups in order, that is, -0 then -1 and so forth

```
# restore -rqvf /dev/rmt0
```

Figure 12-26. restore Command

AU147.0

Notes:

The **restore** command is used to restore data backed up with the **backup** command. **restore -T** will show the contents of the media and display the list of files.

restore -x can be used to restore selective files from the backup. The file names and paths on the backup will be preserved on the restore. If the backup was created with relative path names then the files will be restored relative to the current directory when the restore is issued. **restore -x** can be used to restore selected files from a backup by name and a filesystem backup.

restore -r works with backups by inode. It will ensure that the proper order is used to recover incremental backups. During the restore process, a file called **restoresymtable** is created in the root directory (top level directory) of the file system. This file is checked each time **restore -r** is run to ensure that the recovery sequence is correct. The recovery should progress in ascending order by level number. When you have recovered the entire file system, remove the **restoresymtable** file to be ready for future recoveries. Otherwise, the next time you need to restore a level 0, you will be told you are not going in the correct sequence.

Make sure the file system exists and is mounted before recovering a file system backup. The data will be recovered into the existing directory structure using the file names. If the file system is not mounted, the data will go into a different file system. Be careful!

restore -i is another option available when working with an inode backup. This allows for an interactive restore.

Restore a File or a Directory

smit restfile

Restore a File or Directory

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

	[Entry Fields]	
* Restore DEVICE	[/dev/fd0]	+ /
* Target DIRECTORY	[.]	/
FILE or DIRECTORY to restore (Leave blank to restore entire archive.)	[]	
VERBOSE output?	no	+
Number of BLOCKS to read in a single input operation	[]	#

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

Figure 12-27. Restore a File or a Directory

AU147.0

Notes:

There is another SMIT screen (fastpath **smit restfilesys**), which, as the name suggests, can be used to restore an entire file system rather than a file or a directory. The screen contents are identical to this screen (**smit restfile**) apart from the one option FILE or DIRECTORY to restore, which does not appear on the restore a file system screen. All other options are present.

Exercise: Using backup and restore



Figure 12-28. Exercise: Using backup and restore

AU147.0

Notes:

This lab will allow you to perform backups and recoveries using the AIX tools - backup and restore. It will give you an opportunity to perform a backup by name and a backup by inode.

This exercise can be found in your Exercise Guide.

Other UNIX backup Commands

- **tar** (tape archive)
 - Widely available
 - Good for transfer of data between platforms
 - Has no support for extended inode (ACLs)

- **cpio** (copy input to output)
 - Widely available
 - Difficulties can occur with many symbolic links
 - Has no support for extended inode (ACLs)

- **dd** (device to device)
 - Makes backup copies that are an exact image
 - Can also be used for conversions
For example: can convert ASCII to EBCDIC

Figure 12-29. Other UNIX backup Commands

AU147.0

Notes:

The AIX **backup** tool is preferred for an AIX backup intended to be used exclusively on AIX machines. AIX supports access control lists (ACL) and Trusted Computing Base (TCB) which provide additional security-related features for AIX files and directories. Only the **backup** command supports these additional security features. If you are using ACLs or TCB, you need to use **backup** or these elements of security will be lost during the backup.

AIX does support other generic UNIX backup tools. For backups that will be recovered on another UNIX operating system, these tools would need to be used since only AIX supports **backup** and **restore**.

tar is widely used throughout UNIX and is supported on AIX as well. **cpio** is also widely used and is also in AIX. Neither support ACLs or TCB. Also, **cpio** has difficulties following symbolic links. It may not have enough memory to follow the link and the link will be lost in the backup.

dd is used to copy and convert data byte-by-byte.

tar Command

Generate a tar backup

```
# tar -cvf /dev/rmt0.3 /home
```

Restore a file from a tar image

```
# tar -xvf /dev/rmt0 /home/team01/mydir
```

List (verify) content of a tar file

```
# tar -tvf /dev/rmt0
```

Figure 12-30. tar Command

AU147.0

Notes:

The **tar** command only works with mounted file systems.

Here is a list of the commonly use options:

- c create a **tar** backup
- x extract (restore) a file(s) from a **tar** file
- t reads the content of the **tar** file (verify the backup)
- v verbose output - displays files as they are backed up and restored
- f identify the file or device holding the **tar** image

To perform a **tar** backup, use the **-c** option. The **-f** option can specify a device (like rmt0) or a file in a directory. The **tar** command will do recursive backups. In the example, /home is the starting point for the **tar** command. It will backup all of /home and its sub-directories and its sub-directories and so on.

When recovering, use the **-x** to extract a file. If you want just one file, name it on the command line. If you want a directory and all of its subdirectories, name it. The example

shows the recovery of the /home/team01/mydir directory. If no file is named, then the entire **tar** image is restored.

To verify the **tar** image, use **-t**. In the example, the content of rmt0 will be displayed. With **-t**, no files are actually recovered.

cpio Command

Generate a cpio backup

```
# find /home | cpio -ov> /dev/rmt0
```

Restore from a cpio image

```
# cpio -idv </dev/rmt0
```

List (verify) contents of a cpio image

```
# cpio -itv < /dev/rmt0
```

Figure 12-31. cpio Command

AU147.0

Notes:

The **cpio** tool is another generic UNIX tool. **cpio** stands for copy input/output.

Some of the common options that are used with cpio:

- o create a **cpio** image (output)
- i read from a **cpio** image (input)
- t read (verify) the content of a **cpio** image
- v verbose output - list files during backup and restore operations
- d create necessary directories when recovering an image
- m retain the original modification times associated with files contained in a **cpio** image. Without the **-m** option, all files will have modification times associated with the time they were restored

cpio must be fed a list of files much like the **backup** command. The **find** command is frequently used to do this. Instead of using the **-f** option like **tar** and **backup**, **cpio** uses the redirection symbol (>).

Take a look at the examples:

To create the **cpio** image the **find** command recursively lists all files in the **/home** directory. **cpio** then creates its output, **-o**, on **/dev/rmt0**.

To restore from a **cpio** image, the **-i** is used to read in from the image. The **-d** creates directories and **-m** retains the time stamps. If a file is named, then only the file is restored. If no file is named, the entire image is restored.

To verify or read the content of the **cpio** image, use the **-t** option.

dd Command

Converts and copies files

- To copy a file to diskette

```
# dd if=/etc/inittab of=/dev/rfd0
```

- To convert a file from ASCII to EBCDIC

```
# dd if=text.ascii of=text.ebcdic conv=ebcdic
```

- To convert data to uppercase characters

```
# cat lcase.data | dd conv=ucase
```

Figure 12-32. dd Command

AU147.0

Notes:

The **dd** command reads in standard input or the specified input file, converts it, and then writes to standard out or the named output.

The common options are:

if= specifies the input file

of= specifies the output file

conv= designate the conversion to be done

In the first example, the file **/etc/inittab** is copied to the floppy diskette.

In the second example, the file **text.ascii** is converted into ebcdic and is written to a file called **ebcdic.text**.

In the last example, no output or input file is specified so standard out and standard in is used. The file containing lower case characters, **lcase**, is converted into uppercase characters and displayed to standard out.

dd is also useful when you need to copy specific blocks of data. For example, if a file systems superblock (stored in the 1st block of the file system) is corrupt, a copy is kept at

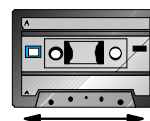
the 31st block. The dd command can copy that 31st block back to the 1st to repair the file system. The command is:

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

dd can span volumes with the **span=yes** parameter on the command line.

Controlling the Tape

tctl
}
 rewind - rewinds a tape
 fsf - fast forwards a tape
 offline - ejects a tape
 rewoffl - rewinds and ejects a tape



```
# tctl -f /dev/rmt0 rewind
# tctl -f /dev/rmt0.1 fsf 3
# tctl -f /dev/rmt0 rewoffl
```

restore -s

```
# restore -s 4 -xvf /dev/rmt0.1 ./etc/inittab
```

Figure 12-33. Controlling the Tape

AU147.0

Notes:

The tape control, **tctl**, command is used to position the tape and eject the tape. All of the backup commands addressed so far assume the tape was positioned correctly. None of those commands reads the entire tape, rather they only look at the tape file where the tape is positioned. To ensure you position it correctly, be sure to document the content and order of the data on the tape.

The **tctl** command has many options. These are ones more commonly used. The **rewind** option is generally the first place to start. This ensures you start from the beginning. The **fsf** moves the tape forward. It counts end-of-file markers. In the example, **fsf 3** would position the tape to the beginning of the fourth file.

The **offline** and **rewoffl** options will eject the tape.

The **restore** command has the capability to position a tape as well. The **-s** option is used to seek the file specified. In the example above, the fourth file on the tape would be read and the **./etc/inittab** file will be restored.

Both the **fsf** example and the **restore -s** example are both positioning the tape to the same location. If they were being used on a mksysb tape, this is how you can restore an individual file from the tape.

There is also a **tcopy** command that can be used to copy a tape to another tape. To do this, you must have two tape devices. The syntax is **tcopy** source target. The **tcopy** command can be given just a source. When this is done, the entire tape is read and a report showing the number of files and blocks sizes is displayed.

Good Practices

- Verify your backups
- Check the tape device
- Keep old backups
- Offsite secure storage
- Label tape
- Test recovery procedures before you have to!



Figure 12-34. Good Practices

AU147.0

Notes:

This shows a list of good practices for your backup strategy.

Always verify your tapes. Use **restore -T** (or **tar -t...**) to view the contents. Even with mksysb tapes, you can position the tape to the correct file and verify it without having to restore the entire contents.

Check your tape device. The **tapechk** command can be used to detect malfunctioning hardware. The command used is **# tapechk 2**. The number specifies how many files on the tape should be read.

Keep old backups in case something goes wrong with the new ones.

Store a set of backups off site in case something happens to your site.

Label your tapes!!! There is no way to know what is on the tape by looking at it. The label should at least list the tape files, the commands used to create the tape, the date created and the block size.

Test your recovery procedure before you have to! Know that you can recover before you have to recover.

Optional Exercise: Using tar and cpio



Figure 12-35. Optional Exercise: Using tar and cpio

AU147.0

Notes:

This is an optional exercise. The instructor will determine if there is time and interest to complete this exercise. It gives an opportunity to try out the generic UNIX tools - tar and cpio.

This exercise can be found in your Exercise Guide.

Checkpoint

1. What is the difference between A and B?
 - a. `find /home/fred | backup -ivf /dev/rmt0`
 - b. `cd /home/fred; find . | backup -ivf /dev/rmt0`

2. On a **mksysb** tape if you entered **tctl rewind** and then **tctl -f/dev/rmt0.1 fsf 3** which element on the tape could you look at?

Which command could you use to restore these files?

3. True or false? SMIT **mksysb** will backup all file systems, provided they are mounted.

Figure 12-36. Checkpoint

AU147.0

Notes:

Unit Summary

- In order to perform successful backups, consideration must be given to the frequency of the backup, the media to be used and the type of backup
- Backups can be initiated on a single file, a file system or an entire volume group, all of which are supported through SMIT
- By modifying the **bosinst.data** and the **image.data** files, a customized system image backup can be created
- There are many other UNIX backup commands which can be used, however their limitations must be fully understood. The commands include: **tar**, **cpio** and **dd**
- Other useful commands also exist to manipulate the data on the backup media such as **tctl**

Figure 12-37. Unit Summary

AU147.0

Notes:

Unit 13. Security and User Administration

What This Unit Is About

This unit introduces the concepts of AIX users and groups, and also the files that contain user account information.

What You Should Be Able to Do

After completing this unit, you should be able to:

- Define the concepts of users and groups, and define how and when these should be allocated on the system
- Define ways of controlling root access on the system
- Define the uses of SUID, SGID, and SVTX permission bits
- Identify the data files associated with users
- Administer user and group accounts

How You Will Check Your Progress

Accountability:

- Checkpoint questions
- Exercise

References

Online *System Management Guide: Operating System and Devices*

<http://www.redbooks.ibm.com> (see security related information)

Objectives

After completing this unit, students should be able to:

- Define the concepts of users and groups, and define how and when these should be allocated on the system
- Define ways of controlling root access on the system
- Define the uses of SUID, SGID and SVTX permission bits
- Add/Change/Delete user and group accounts
- Identify the data files associated with users and security

Figure 13-1. Objectives

AU147.0

Notes:

13.1 Security Concepts

Security Concepts

User Accounts

- Each user has a unique name, numeric ID and password
- File ownership is determined by a numeric user ID
- The owner is usually the user who created the file, but ownership can be transferred by root
- Default users:
 - root super user
 - adm, sys, bin ... IDs that own system files but cannot be used for login

Figure 13-2. Security Concepts

AU147.0

Notes:

The security of the system is based on a user being assigned a unique name, user ID (UID) and password. When the user logs in, the UID is used to validate all requests for file access.

When a file is created, the UID associated with the process that created the file is assigned to the file. Only the owner or **root** can change the access permissions.

There are several user accounts automatically created. **root**, for example, is one. Some user accounts are not made for login but only to own certain files. **adm**, **sys**, and **bin** are examples of that type of account.

Groups

- A group is a set of users, all of whom need access to a given set of files
- Every user is a member of at least one group and can be a member of several groups
- The user has access to files in their groupset. To list the groupset use **groups**
- The user's primary group is used for file ownership on creation. To change the primary group use the **newgrp**
- Default groups:
 - System administrators
 - Staff ordinary users

Figure 13-3. Groups

AU147.0

Notes:

Users that require shared access to a set of files are placed in groups. A user can belong to multiple groups. Each group has a unique name and Group ID (GID). The GID is also assigned to a file when it is created.

There are several groups predefined on an AIX system. For example, the **system** group is root's group and the **staff** group is for all ordinary users.

The creation of groups to organize and differentiate the users of a system or network is part of systems administration. The guidelines for forming groups should be part of the security policy. Defining groups for large systems can be quite complex and once a system is operational, it is very difficult to change the group structure. Investing time and effort in devising group definitions before your system arrives is recommended.

Groups should be defined as broadly as possible and be consistent with your security policy. Do not define too many groups because defining groups for every possible combination of data type and user type can lead to impossible extremes.

A group administrator is a user who is allowed to assign the members and administrators of a group. It does not imply that the user has any "administrative" abilities for the system.

There are three types of groups on the system:

User Groups

User groups should be made for people who need to share files on the system, such as people who work in the same department, or people who are working on the same project.

System Administrator Groups

System administrators are automatically members of the **system** group. Membership of this group allows the administrators to perform some of the system tasks without having to be the **root** user.

System Defined Groups

Several system-defined groups exist. **staff** is the default group for all non-administrative users created in the system. **security** is another system defined group having limited privileges for performing security administration. The system-defined groups are used to control certain subsystems.

Groups

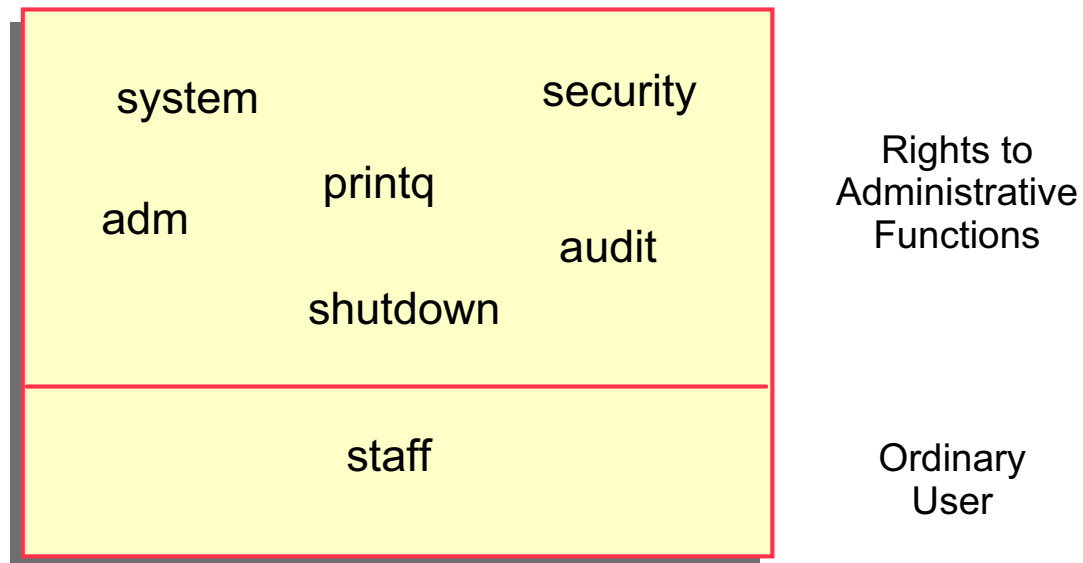


Figure 13-4. Groups

AU147.0

Notes:

Common groups on the system are:

system	For most configuration and standard hardware and software maintenance
printq	For managing queuing. Typical commands which can be run by members of this group are: enable , disable , qadm , qpri , and so forth.
security	To handle most passwords and limits control. Typical commands which can be run by members of this group are: mkuser , rmuser , pwdadm , chuser , chgroup , and so forth.
adm	Most monitoring functions such as performance, cron , accounting
staff	Default group assigned to all new users. You may want to change this in /usr/lib/security/mkuser.defaults
audit	For auditors
shutdown	Allows use of the shutdown command

User Hierarchy

- To protect important users/groups from members of the **security** group AIX has **admin users** and **admin groups**
- Only **root** can add/remove/change an **admin user** or **admin group**
- Any user on the system can be defined as an **admin user** regardless of the group they are in

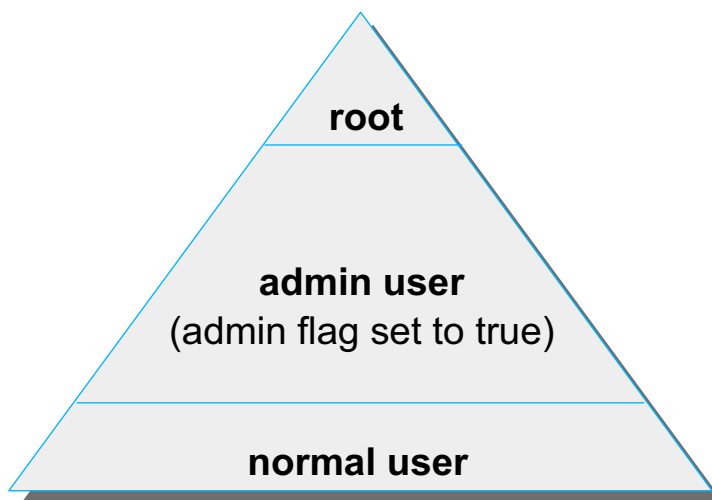


Figure 13-5. User Hierarchy

AU147.0

Notes:

The ability to perform certain system tasks (like creating users) depends upon the standard AIX file permissions. Most system admin tasks can be performed by users other than **root** by assigning those users to groups such as **system**, **security**, **printq**, **cron**, **adm**, **audit** or **shutdown**. In particular, a user in the **security** group can add/remove/change other users and groups.

To protect important users/groups from users in the **security** group, AIX has three levels of user hierarchy: root, admin users/groups and normal users/groups. Only **root** can add/remove/change an admin user or admin group. Therefore, you can define a user with a high level of access, but who is protected from users in the **security** group.

Control root's Access

- Restrict access to privileged logins
- root's passwords should be changed on an unannounced schedule by the system administrator
- Assign different root passwords to different machines
- System administrators should always login as themselves first and then su to root instead of logging in as root. This helps provide an audit trail for root usage
- Do not include unsecured directories in root's PATH

Figure 13-6. Control root's Access

AU147.0

Notes:

If the **root** password is known by too many people, no one can be held accountable. The **root** password should be limited to just two or three administrators. The fewer people who know **root's** password the better.

The system administrator should ensure that distinct **root** passwords are assigned to different machines. You may allow normal users to have the same passwords on different machines, but never do this for **root**.

Attempts to become **root** through **su** can be investigated. Successful and unsuccessful attempts might be logged by the audit system.

root's PATH is used by many implicit system functions, not just by a user logged in as **root**.

Security Logs

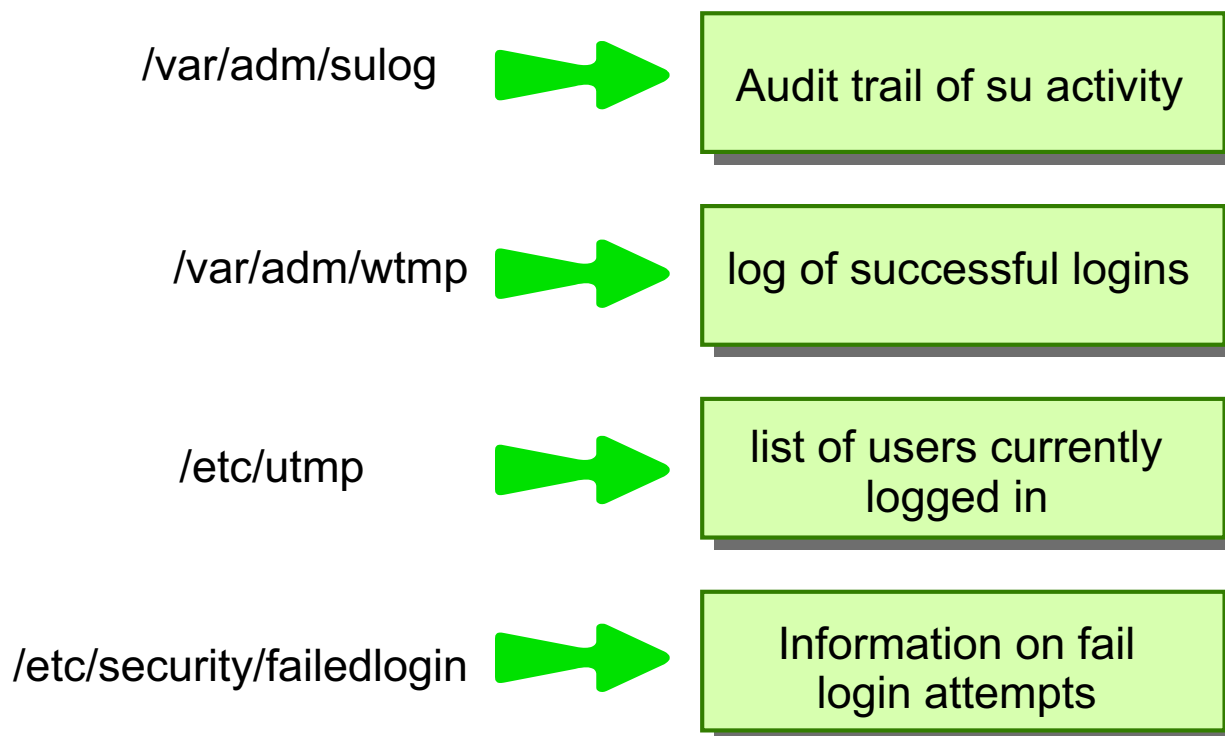


Figure 13-7. Security Logs

AU147.0

Notes:

The **sulog** file is an ASCII text file that can be viewed with **more** or **pg**. In the file, the following information is recorded: date, time, terminal name and login name. The file also records whether or not the login attempt was successful (and indicates a success by a **+** and a failed login by a **-**).

The **/etc/utmp** file contains a record of users logged into the system, and the **/var/adm/wtmp** file contains connect-time accounting records. To obtain information from either file use the **who** command with the file name. The **who** command normally examines the **/etc/utmp** but you can specify either one of the named files as arguments to the command.

The **last** command can also be used to display in reverse chronological order, all previous logins and logoffs still recorded in the **/var/adm/wtmp file**. The **/var/adm/wtmp** file collects login and logout records as these events occur and holds them until the records are processed by the accounting commands.

For example:

last root displays all the recorded logins and logoffs by the user **root**
last reboot displays the time between reboots of the system

File/Directory Permissions

File	Perm. Bit	Directory
read content of file	r	list content of directory
modify content of file	w	create/remove files in directory
use file name to execute as a command	x	gives access to directory
run program with effective UID of owner	SUID	_____
run program with effective GID of group	SGID	files created in directory inherit the same group as the directory
_____	SVTX	must be owner of file to delete files from directory

Figure 13-8. File/Directory Permissions

AU147.0

Notes:

There are a number of permission bits associated with files and directories. The standard **r** (read), **w** (write) and **x** (execute) permissions define three levels of access for the user (owner), group and others. In addition there are three permission bits known as **SUID** (set UID), **SGID** (set GID) and **SVTX** (sticky bit).

SUID on an executable file means that when the file runs, the process runs with an effective UID of the owner of the file. **SUID** is not supported on shell scripts.

SUID has no meaning on a directory.

SGID on an executable file means that when the file runs, the process runs with an effective GID of the group owner of the file.

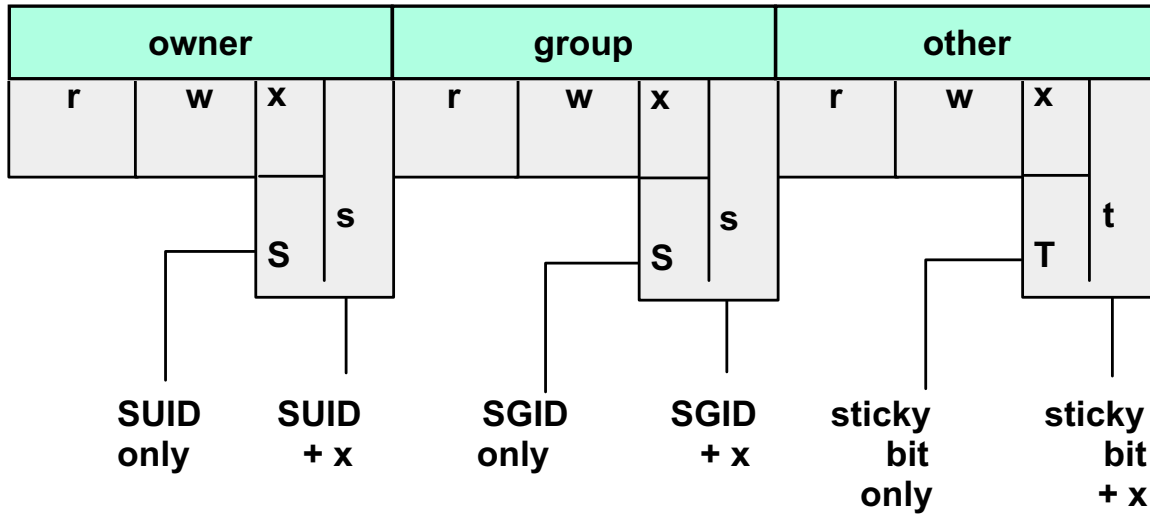
SGID on a directory means that any file/directory created within the directory will have the same group ownership as the directory rather than the primary group of the user.

SVTX on a file has no meaning in AIX (It was used in earlier versions of UNIX.) **SVTX** on a directory means that even if the directory has global write permission (for example, **/tmp**), users cannot delete a file within it unless they either own the file or the directory.

Traditional UNIX used **SVTX** to keep a program in memory after it had completed running, but with memory management routines, this is no longer necessary. **SVTX** is known as the "sticky bit".

The **SGID** permission bits are propagated down through the directory structure, so that any directory created in a directory with the **SGID** bit set also inherits that bit.

Reading Permissions



```
# ls -ld /usr/bin/passwd /usr/bin/crontab /tmp
-r-sr-xr-x    root  security ... /usr/bin/passwd
-r-sr-sr-x    root  cron    ... /usr/bin/crontab
drwxrwxrwt   bin   bin     ... /tmp
```

Figure 13-9. Reading Permissions

Notes:

The SUID bit is indicated by an **S** or **s** in the slot normally reserved for the execute permission for owner (user). The SGID bit is indicated by an **S** or **s** in the slot normally reserved for the execute permission for group. The SVTX bit is indicated by a **T** or **t** in the slot normally reserved for the execute permission for others. Since this slot must show if execute is on/off and whether the additional permission bit on/off, the uppercase **S** or **T** indicates that the execute permission is off. The lower case **s** or **t** indicates the execute permission is on.

There are three examples of files that use these additional permissions. The **passwd** command allows users to change their password even though passwords are stored in a restricted area. **crontab** allows users to create a crontab file even those the directory where these files reside is restricted from ordinary users. And **/tmp** allows everyone to write to the directory, but only the owner of a file can remove a file from the **/tmp** directory.

Changing Permissions

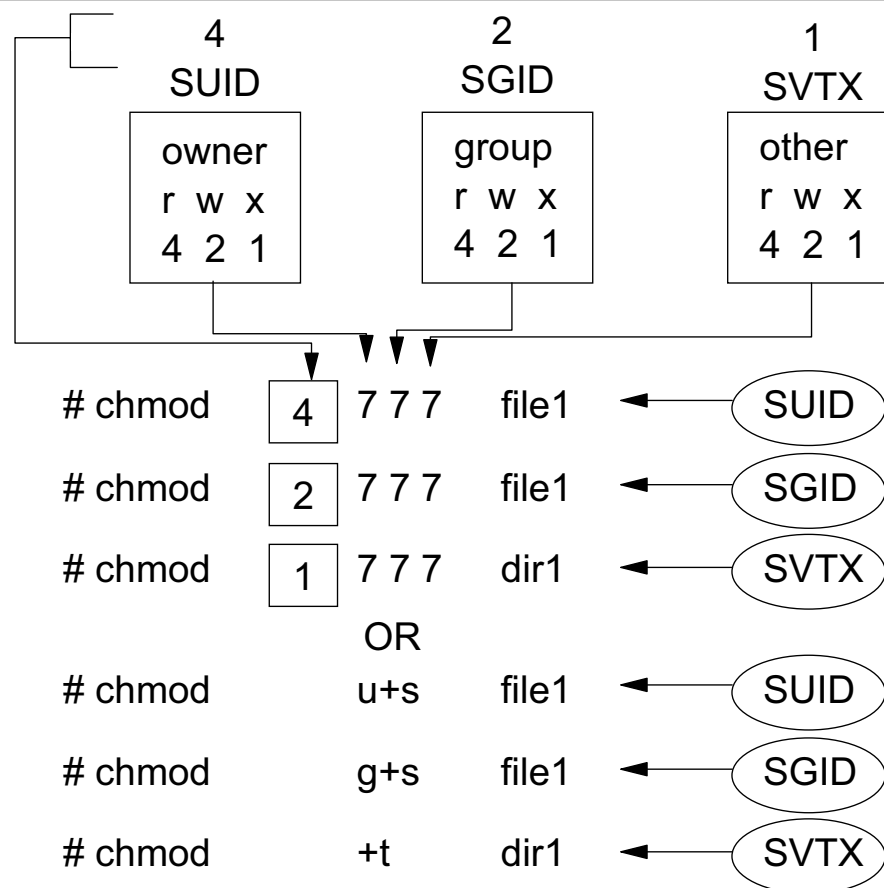


Figure 13-10. Changing Permissions

AU147.0

Notes:

To set the additional permission bits, you use the same command (**chmod**) as you do to set the regular permission bits.

If using the octal notation, you are probably familiar with setting permissions using a command like: **# chmod 777 file1**. When you issue this command, the more complete command would be: **# chmod 0777 file1**. The fourth number, a zero, is implied. This fourth position determines whether the additional bits are turned on.

You normally use the binary values of 4, 2, and 1 to set **r**, **w** and **x**. That remains the same. To set the additional bits, you will be effecting the **x** position in either the user, group or other area. If you assign binary values to user (4), group (2), and other (1), these are the values that you will insert into the fourth position to set the additional bit. SUID is indicated in the user's area; therefore use a **4** in the fourth position. The SGID is indicated in the group area; therefore use a **2** in the fourth position. The SVTX is indicated in the others area; therefore use a **1** in the fourth position.

You can also use the symbolic methods. The graph shows how to set the values using the symbolic method.

umask

- The **umask** governs permissions on new files and directories
- System default umask is 022. A umask of 027 is recommended
- If the umask value is set to 022, then any ordinary files or directories created will inherit the following permissions:
Ordinary file: rw - r -- r --
Directory: rwxr - xr - x
- **/etc/security/user** specifies default and individual user umasks

Figure 13-11. umask

AU147.0

Notes:

The **umask** specifies what permission bits will be set on a new file when it is created. It is an octal number that specifies which of the permission bits will not be set.

If no **umask** was used, then file would be created with permission of 666 and directories would be created with permissions of 777. The system default **umask** is **022** (indicates to remove the 2 bit or **write** from the group and others area). Therefore removing **write** from group and other will result in a initial permission for files of 644 and, for directories, 755. Execute permission is never set initially on a file.

The default setting of the **umask** is 022. For tighter security you should make the umask 027, or even 077.

To view/change the value of the **umask** for the current session use the **umask** command.

The **umask** is specified in **/etc/security/user**. The default stanza in this file specifies the system wide default, but a value can be specified on a per-user basis.

Changing Ownership

chown command

```
# chown fred file1
# chgrp staff file1
# chown fred:staff file1
```

Only root can change file ownership

Figure 13-12. Changing Ownership

AU147.0

Notes:

The **chown** command can be used by root to change the ownership on a file.

The **chgrp** command is used to change the group ownership of a file. Any owner of a file can change the group ownership to any group in their groupset. Root can change the group ownership to any group on the system.

chown can be used by root to set both the ownership and group ownership of a file. It can be done two different ways: **# chown fred:staff file1** or **# chown fred.staff file1**.

Exercise: Security Files



Figure 13-13. Exercise: Security Files

AU147.0

Notes:

This lab gives you a chance to look at some of the security files and allows you an opportunity to work with the SUID, SGID, and SVTX.

This exercise can be found in your Exercises Guide.

13.2 User Administration

Login Sequence

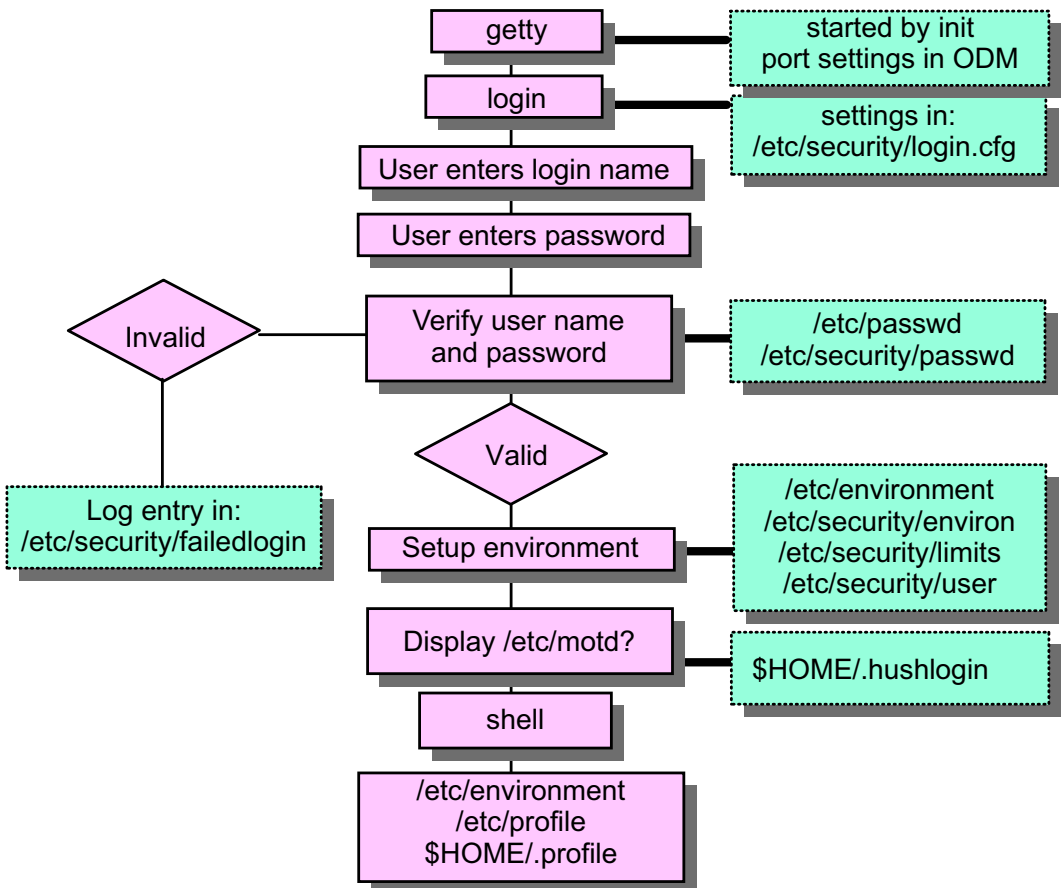


Figure 13-14. Login Sequence

AU147.0

Notes:

When a user attempts to log in, AIX checks a number of files to determine if entry is permitted to the system and, if permitted, what parts of the system the user can access. This provides an overview of the checks performed during the login process.

Ports set up for login are listed in the **/etc/inittab**. When **init** runs, a **getty** process will be started for each port in the list providing a login prompt on the terminal attached to that port. The actual message displayed (also known as the herald) by the **getty** process is defined in **/etc/security/login.cfg**. Once it is displayed, the **getty** process waits for a user to make a login attempt.

When a user is ready to login, they enter their user name at the login prompt. The login program is passed the user name and then checks **/etc/passwd** and **/etc/security/passwd** to see if a password is required. If a password is required or the user name doesn't match a valid name, the "Password:" prompt is displayed and the "invis" terminal attribute is set so that the password is not displayed as it is entered.

When the user enters the password, it is checked. If it is incorrect or an invalid user name was given, then the login fails and an entry is made in the **/etc/security/failedlogin** file.

(To view this file, type: **who /etc/security/failedlogin**.) The number of failed attempts is also tracked (by user account) in **/etc/security/lastlog**. The "Login:" prompt is redisplayed for another attempt.

It is possible to set the characteristics for a user to prevent unlimited attempts on an account. If the number of attempts exceeds the maximum allowable failed attempts, the account is locked.

If a user successfully enters the user name and password, the **usw** stanza in **/etc/security/login.cfg** is checked. This stanza sets the maximum number concurrent logins for a user account. If that number is exceeded, the login is denied and a message is displayed to the user.

If everything is successful to this point, then the user's environment is set using **/etc/environment**, **/etc/security/envIRON**, **/etc/security/limits** and **/etc/security/user**. The login program sets the current directory to the user's HOME directory and displays the content of **/etc/motd** (if no **.hushlogin** file is found in the HOME directory), date of the last successful login, and the number of unsuccessful login attempts since the last successful login.

Finally, control is passed to the login shell (as defined in **/etc/passwd**) which will run **/etc/profile** and **\$HOME/.profile** when using Korn or Bourne shells.

When a user logs out, the shell will terminate and a new **getty** process will be spawned for that port.

User Initialization Process

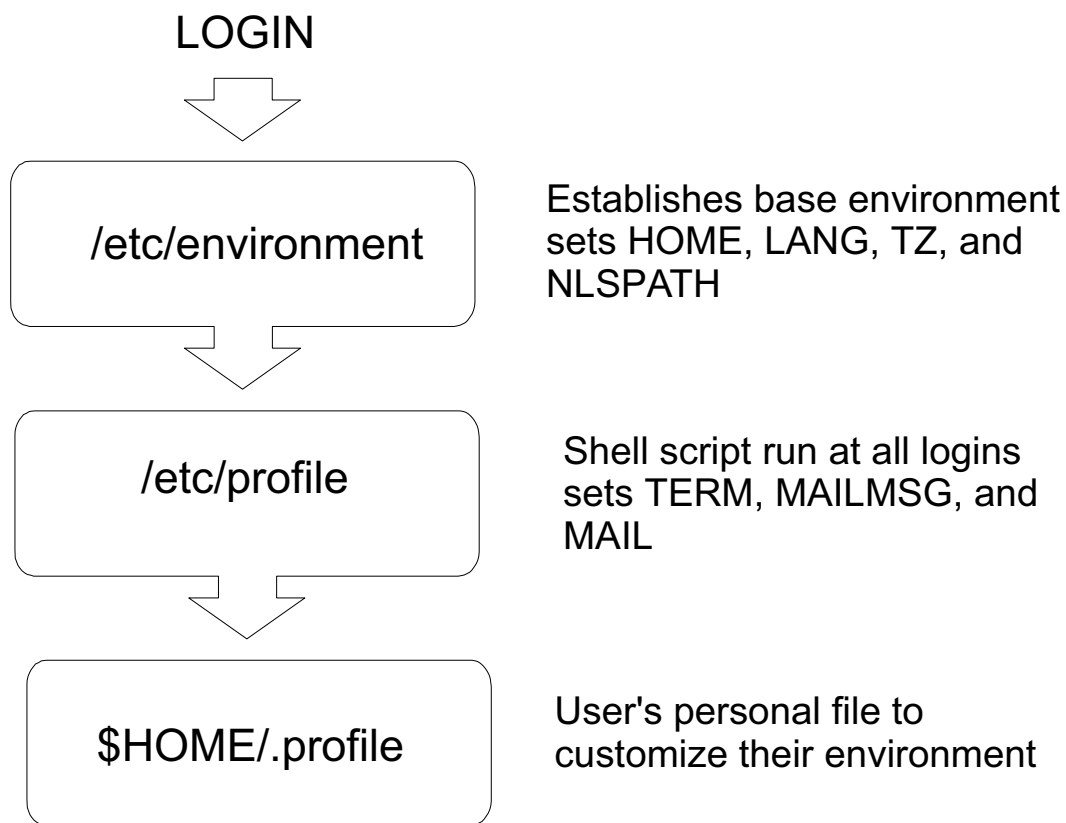


Figure 13-15. User Initialization Process

AU147.0

Notes:

/etc/environment is used to set variable. No commands should be placed in this file.

/etc/profile will be read and executed during every login.

/etc/environment and **/etc/profile** can be changed only by root.

\$HOME/.profile can be customized by the user. The user can overwrite any variable set in **/etc/environment** and **/etc/profile**.

If you are using CDE (Common Desktop Environment), **.profile** is not read by default. In the users HOME directory, the **.dtprofile** file is used to establish the environment when working with CDE. **.dtprofile** replaces the function of **.profile** in the CDE environment. If you want to use both, in the **.dtprofile**, uncomment the line near the end of the file that references the **DTSOURCEPROFILE** variable.

Security and Users

```
# smit security
```

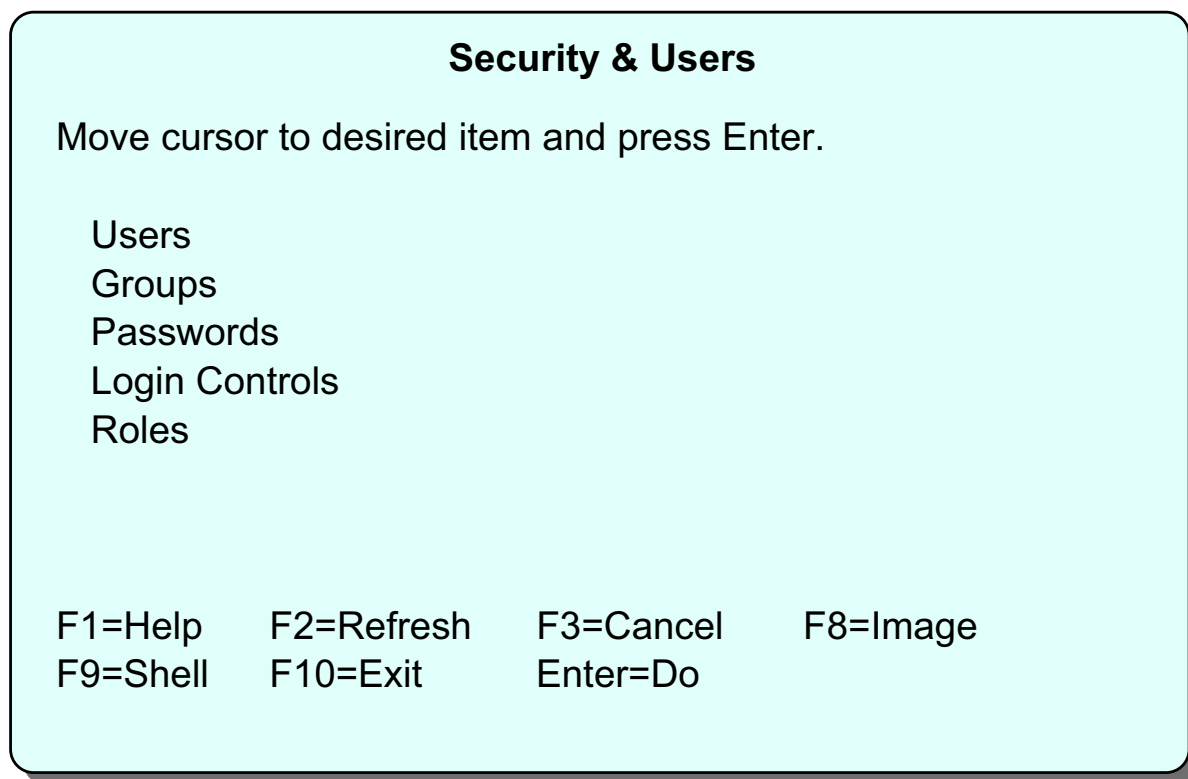


Figure 13-16. Security and Users

AU147.0

Notes:

The **Security and Users** menu is used to manage user and group IDs on the system. The menu consists of five options:

- **Users** - This option is used to add users to the system, delete existing users and change the details of existing users.
- **Groups** - This option is used to add groups to the system, delete groups and change the details of existing groups.
- **Passwords** - This option is used to change the password for a user. It is required when setting up a new user or when a user has forgotten their password.
- **Login Controls** - This option provides functions to restrict access on a user account or on a particular terminal.
- **Roles** - This option sets up user roles. User roles allow root to give authority to an ordinary user to perform a portion of root's functions.

The Web-based System Manager can also be used to manage users and groups.

SMIT Users

smit users

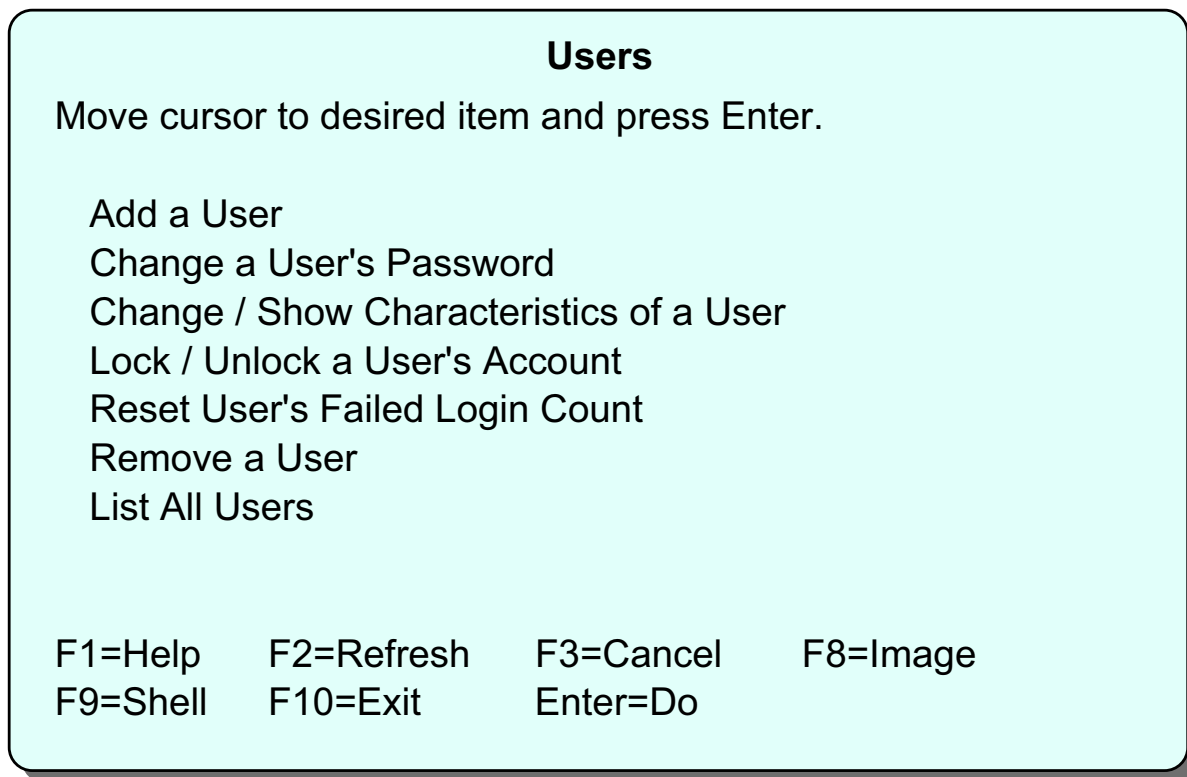


Figure 13-17. SMIT Users

AU147.0

Notes:

Add a User - Add user accounts

Change a User's Password - Password changes

Change/Show Characteristics of a User - Changes the many characteristics that are a part of the user account. The password restrictions are part of this area.

Lock/Unlock a User's Account - This is used to temporarily disable an account. It is a good security practice to disable accounts if they are not expected to be used, like when someone is on an extended leave of absence.

Reset User's Failed Login Count - If the administrator has set a limit to the number of failed attempts that can be made on an account before locking it, this resets that count.

Remove a User - Removes the user account, but not files owned by that user.

List all users - Runs the **lsuser** command.

List All Users

```
lsuser [-c | -f] [-a attribute ..]{ALL | username ..}
```

```
# lsuser -a id home ALL
root id=0 home=/
daemon id=1 home=/etc
bin id=2 home=/bin
john id=200 home=/home/john
```

Figure 13-18. List All Users

AU147.0

Notes:

The **lsuser** command is used to list the attributes of all users (ALL) or individual users on the system.

When the **List All Users** option in SMIT is used, the user name, id and home directory are listed.

When the command is issued directly, the data may be listed in line format, in colon format (**-c**) or in stanza format (**-f**). Individual attributes or all attributes may be selected. The output can also be generated for individual users.

The information is gathered from the various security files: **/etc/passwd**, **/etc/security/limits** and **/etc/security/user**.

Add a User to the System

```
# smit mkuser
```

Add a User

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

	[Entry Fields]	
* User NAME	[]	
User ID	[]	#
ADMINISTRATIVE USER?	false	+
Primary GROUP	[]	+
Group SET	[]	+
ADMINISTRATIVE GROUPS	[]	+
ROLES	[]	+
Another user can SU TO USER?	true	+
SU GROUPS	[ALL]	+
HOME directory	[]	
Initial PROGRAM	[]	
User INFORMATION	[]	
EXPIRATION date (MMDDhhmmyy)	[0]	
Is this user ACCOUNT LOCKED?	false	+

[MORE ...36]

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

Figure 13-19. Add a User to the System

AU147.0

Notes:

Default ID numbers in: `/etc/security/.ids`
 Shell script to set up ID: `/usr/lib/security/mkuser.sys`
 Default characteristics in: `/usr/lib/security/mkuser.default`
`/etc/security/user`
 Default .profile: `/etc/security/.profile`

The **mkuser** command is used to add a user. User attributes can be specified to override the default values.

The defaults for the **mkuser** command are stored in the file `/usr/lib/security/mkuser.default`. This file can only be edited by the **root** user. This file contains the following information:

user:

```
pgrp = staff
groups = staff
shell = /usr/bin/ksh
```

```
home = /home/$USER
```

admin:

```
pgrp = system  
groups = system  
shell = /usr/bin/ksh  
home = /home/$USER
```

The "user" stanza of this file is picked up if an ordinary user is being added and the "admin" stanza is picked up if an administrative user is being added.

If the user ID is not specified, then a default ID number is chosen from the **/etc/security/ids** file. Administrative users are given IDs starting from 6 and normal users starting from 200.

The shell script **/usr/lib/security/mkuser.sys** is run during the user creation process. This creates the user's home directory and creates the **.profile** file. This shell script can be modified to perform any function that is required when setting up the user.

The full list of user characteristics contains some entries which are not often used. Many of these fields may be left empty with no ill-effect. For the complete list refer to SMIT (fastpath **smit mkuser**).

When a new user is created the ID is disabled (an asterisk is placed in the password field of the **/etc/passwd** file). To enable the ID a password must be set with the **Change a User's Password** option or the **passwd** or **pwdadm** commands.

Change / Show Characteristics of a User

```
# smit chuser
```

Change / Show Characteristics of a User

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

	[Entry Fields]	
* User NAME	[george]	
User ID	[206]	#
ADMINISTRATIVE USER?	false	+
Primary GROUP	[staff]	+
Group SET	[staff, security]	+
ADMINISTRATIVE GROUPS	[]	+
ROLES	[]	+
Another user can SU TO USER?	true	+
SU GROUPS	[ALL]	+
HOME directory	[/home/george]	
Initial PROGRAM	[/usr/bin/ksh]	
User INFORMATION	[]	
EXPIRATION date (MMDDhhmmyy)	[0]	
Is this user ACCOUNT LOCKED?	false	+
[MORE ...36]		

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

Figure 13-20. Change / Show Characteristics of a User

AU147.0

Notes:

The **Change / Show Characteristics of a User** option which runs the **chuser** command, allows any of the user characteristics listed previously (except the user name) to be changed. This can only be executed by **root** or a member of the **security** group (only **root** can change an admin user). This SMIT screen holds exactly the same attributes as the "Add a User" screen.

The user information is not required by the system. This is the fifth field in the **/etc/passwd** file which is usually used to hold the user's real name, telephone number etc. Some programs use this information when reporting on user activity for example, the **finger** program in TCP/IP. Users can change their own user information.

The initial program is the shell which the user logs into. It is usually set to one of:

- /usr/bin/bsh** (the Bourne shell)
- /usr/bin/csh** (the C shell)
- /usr/bin/ksh** (the Korn shell) (default)

A user can only change their shell to one of the above whereas root can change a user's shell to any program. Also note that users can change their own full name and login shell.

The following command can be used to change/show characteristics of a user:

chuser attribute=value username

Remove a User from the System

- The **rmuser** command or SMIT can be used to delete a user from the system.

```
# rmuser -p team01
```

- The user's home directory is not deleted, therefore you must manually clean up the user directories (remembering to **backup** important files first !)

```
# rm -r /home/team01
```

Figure 13-21. Remove a User from the System

AU147.0

Notes:

The **Remove a User from the System** option or the **rmuser** command can be used to remove any user from the system. Only the **root** user may remove administrative users.

The **-p** option will remove authentication information from the **/etc/security/*** files. Typically this information is the user password, as well as other login restrictions which have been previously set for the ID.

The user's home directory and associated files are not removed by this option. They must be removed separately by the administrator. To do this you can use the **-r** option on the **rm** command to recursively remove files. Remember to back up any important files before removing the user's home directory.

Passwords

- A new user ID cannot be used until a password is assigned
- There are two commands available for changing the password:

passwd [*username*]

root or *username* only

SMIT uses the **passwd** command

pwdadm *username*

root or user in security group

Figure 13-22. Passwords

AU147.0

Notes:

When a user ID is created with SMIT or with the **mkuser** command, the user ID is disabled (an * is in the password field of **/etc/passwd**). To enable the ID, the **passwd** or **pwdadm** command must be used to set up the initial password for the user.

When passwords are entered, they are not displayed. When changing a password, the new password is requested a second time for verification.

If **root** or a member of the **security** group sets the password for a user, the ADMCHG flag is set in the flags field in **/etc/security/passwd**. The user is then prompted to change the password at the next login.

There is no way to examine an existing password on the system. The only way to recover from a forgotten password is for an administrator or **root** to set a new one for the user.

The option **Passwords** on the **Users** menu of SMIT uses the **passwd** command.

If a user uses **passwd** to change their password, they will first be prompted for the old password and then they will be asked twice for a new password. When root uses **passwd** to set a user's password, root will only be prompted twice for the new password.

Members of the security group can use **pwdadm** to change the password of non-administrative accounts. Security group members will first be prompted to enter their own password, then prompted twice to enter the user's new password. Root will only be prompted twice for the new password.

Only **root** can change the password for a user who has the ADMIN flag set in **/etc/security/passwd**.

Regaining root's Password

Boot from CD-ROM or a bootable tape

- Select option 3 from the Installation and Maintenance menu: Start Maintenance Mode for System Recovery
- Follow the options to activate the root volume group and obtain a shell
- Once a shell is available, execute the **passwd** command to change root's password.
- # sync ; sync

Reboot the system

Figure 13-23. Regaining root's Password

AU147.0

Notes:

To recover the root password, you must boot your machine from media other than its normal hard drive. An installation CD or a mksysb tape works just fine. Remember to invoke the service boot list - usually by pressing **F5** while your machine is booting.

You will need to define your system console and select a language. Then the installation and maintenance menu will display. Be certain to select option **3 - Start Maintenance Mode for System Recovery**. If you select option 1 or 2, you will be reinstalling your operating system.

You need to activate the root volume group and start a shell. This gets you access to **rootvg** without any passwords. Once you get the # prompt, just use the **passwd** command like you normally would.

Run **# sync ; sync**. This ensures the memory buffer is written to disk. In other words, it ensures the new **password** is saved to disk.

Then reboot your system - **shutdown -Fr**.

SMIT Groups

smit groups

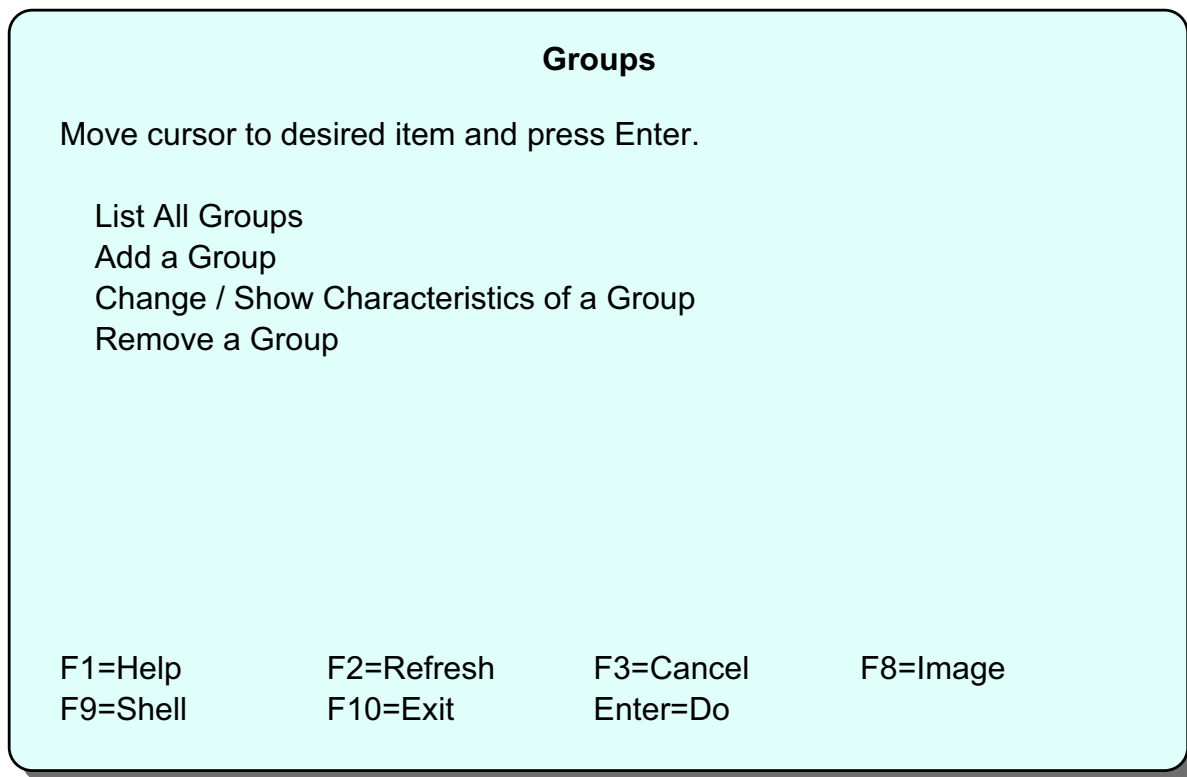


Figure 13-24. SMIT Groups

AU147.0

Notes:

The purpose of groups is to give a common set of users the ability to share files. The access is controlled using the group set of permission bits.

Only root and members of the security group can create groups. Root and security group members can select a member of the group to be the group administrator. This privilege allows the user to add and remove users from the group.

Remember there are a number of predefined groups on AIX systems like the **system** group (which is root's group) and the **staff** group (which contains the ordinary users).

List All Groups

```
lsgroup [ -c | -f ] [ -a attribute ] { ALL | groupname }
```

lsgroup ALL

```
system id=0 admin=true users=root,ray,shutdown,roger
staff id =1 admin=false users=tim,john,mike
bin id=2 admin=true users=root,bin
sys id=3 admin=true users=root,bin,sys
adm id=4 admin=true users=bin,adm
uucp id=5 admin=true users=uucp
mail id=6 admin=true users=
security id=7 admin=true users=root,ray
cron id=8 admin=true users=root
printq id=9 admin=true users=lpd,courses,ray
audit id=10 admin=true users=root
nobody id=-2 admin=false users=nobody,lpd
usr id=100 admin=false users=guest
shutdown id=21 admin=true users=
```

Figure 13-25. List All Groups

AU147.0

Notes:

The **lsgroup** command is used to list selected or all groups on the system. The data is presented in line format by default or in colon format (**-c**) or in stanza format (**-f**).

The **-c** option displays the attribute for each group in colon separated records.

The **-f** option displays the group attributes in stanza format with each stanza identified by a group name.

Add Groups

smit mkgroup

Add a Group

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

	[Entry Fields]	
* Group NAME	[support]	
ADMINISTRATIVE group?	false	+
Group ID	[300]	#
USER list	[fred,barney]	+
ADMINISTRATOR list	[fred]	+

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

Figure 13-26. Add Groups

AU147.0

Notes:

The **mkgroup** command is the command used to create a new group. The group name must be a unique string of eight characters or less. The **(-a)** parameter is used to indicate that the new group is to be an administrative group. Only the **root** user can add administrative groups to the system.

The **(-A)** option makes the invoker of the **mkgroup** command the group administrator.

A user may belong to no more than 32 groups. **ADMINISTRATOR list** is a list of members from the **USER list** that are allowed to change the characteristics of a group and add or remove members.

Change / Remove Groups

smit chgroup

Change Group Attributes

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

	[Entry Fields]	
Group NAME	[support]	
Group ID	[300]	#
ADMINISTRATIVE group?	false	+
USER list	[fred,barney,wilma]	+
ADMINISTRATOR list	[fred]	+

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

Figure 13-27. Change / Remove Groups

AU147.0

Notes:

The **chgroup** command is used to change the characteristics of a group. It can only be run by **root** or a member of the **security** group. The group attributes are:

- Group ID (**id=groupid**). It is not advisable to change the groupID, but it is occasionally done immediately after a group has been created to match the ID of a previously deleted group, or a specific groupID for a particular software package.
- ADMINISTRATIVE group? (**admin=true|false**). Only the **root** user can change a group to be an administrative group or make changes to an existing administrative group.
- USER list (**users=usernames**). This is a comma separated list of the names of all the members of the group. The group may be their primary group or an additional one.
- ADMINISTRATOR list (**adms=adminnames**). This is the list of group administrators.

The **chgrpmem** command can be used by any user to change either the administrators or the members a group for which they are group administrator.

The **rmgroup** command is used to remove a group from the system. This command has no options and the only parameter is the group name. Only the **root** user can delete an administrative group.

Message of the Day

- The file **/etc/motd** contains text that will be displayed every time a user logs in.
- This file should only contain information necessary for the users to see
- If the **\$HOME/.hushlogin** file exists in a user's home directory, then the contents of the **/etc/motd** file will not be displayed to that user

Figure 13-28. Message of the Day

AU147.0

Notes:

This is a convenient way to communicate information to all users, such as installed software version numbers or current system news. The message of the day is contained in the **/etc/motd** file.

To change the message of the day, simply edit this file.

Many other commands exist to communicate with the user community. Several of these commands are covered in the AIX Version 5.1 Basics course such as **write**, **wall**, **mail** and **talk**.

Exercise: User Administration

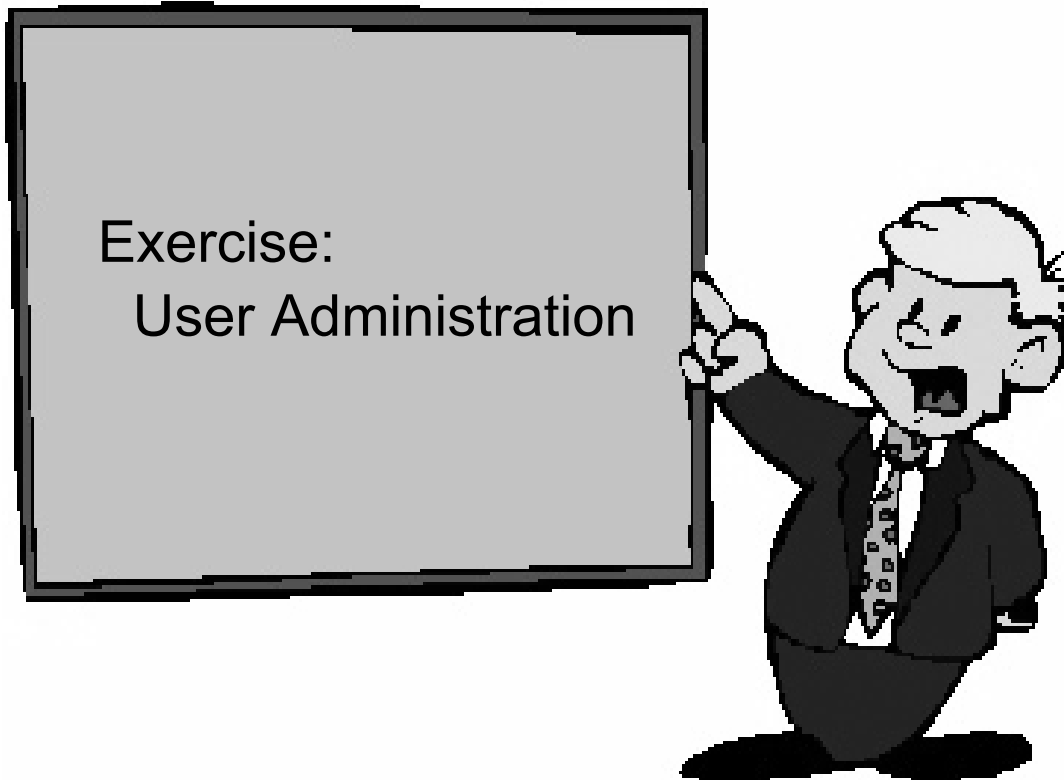


Figure 13-29. Exercise: User Administration

AU147.0

Notes:

This lab gives you an opportunity to expand your knowledge of user administration. You will add users and groups and review many of the user characteristics.

This exercise can be found in your Exercise Guide.

13.3 Security Files

Security Files

- Files used to contain user attributes and control access:

/etc/passwd	valid users (not passwords)
/etc/group	valid groups
/etc/security	directory not accessible to normal users
/etc/security/passwd	user passwords
/etc/security/user	user attributes, password restrictions
/etc/security/group	group attributes
/etc/security/limits	user limits
/etc/security/environ	user environment settings
/etc/security/login.cfg	login settings

Figure 13-30. Security Files

AU147.0

Notes:

The security on the system is controlled by a number of ASCII files.

The **/etc/passwd** file lists the valid users, their user ID, primary group, home directory and default login shell.

The **/etc/group** file lists the valid groups, their group ID and members.

The above files have global read access to all users. A number of other files control the attributes of users. They are in the **/etc/security** directory which can only be accessed by root or the security group.

/etc/security/passwd contains the encrypted password and update information for users.

/etc/security/user contains extended user attributes.

/etc/security/group contains extended group attributes.

/etc/security/limits contains process resource limits for users.

/etc/security/environ contains environment variables for users. This file is not often used.

/etc/security/login.cfg is a configuration file for the login program. This contains security enhancements that limit the logins on a port, for example, the number of login attempts and the valid login programs (shells).

/etc/passwd File

cat /etc/passwd

```
root:!:0:0:::/bin/ksh
daemon:!:1:1::/etc:
bin:!:2:2::/bin:
sys:!:3:3::/usr/sys:
adm:!:4:4::/var/adm:
uucp:!:5:5::/usr/lib/uucp:
guest:!:100:100::/home/guest:
nobody:!:4294967294:4294967294::/
lpd:!:9:4294967294::/
john:!:200:0:X7560 5th floor:/home/john:/usr/bin/ksh
bill:*:201:1::/home/bill:/usr/bin/ksh
```

Figure 13-31. /etc/passwd File

AU147.0

Notes:

The **/etc/passwd** file lists the users on the system and some of their attributes. This file must be readable by all users, because commands such as **ls** access it.

The fields in the **/etc/passwd** file are:

- user name - up to eight alphanumeric characters (not all upper case)
- password - on older UNIX systems this contained the encrypted password. This will still work, but on AIX Version 5.1 can not contain the encrypted password and should contain a ! to refer to the **/etc/security/passwd** file. Other common values are a * which means the id is invalid, and no value means there is no password assigned.
- uid - the user id number for the user
- gid - the id of the primary group to which this user belongs
- information - any descriptive text for the user
- directory - the login directory of the user and the initial value of the \$HOME variable

- login program - Specifies that the initial program or shell that is executed after a user invokes the **login** command or **su** command.

In AIX, additional files can be created to be used as an index for the **/etc/passwd**, **/etc/security/passwd** and **/etc/security/lastlog** files. These index files provide for better performance during the login process. Use the **mkpasswd -f** command to create the indexes. The command **mkpasswd -c** can be used to check the indexes and rebuild any that look suspicious.

/etc/security/passwd File

```
# cat /etc/security/passwd
```

```
root:
    password = 92t.mzJBjfbY
    lastupdate = 885485990
    flags =

daemon:
    password = *

bin:
    password = *

john:
    password = q/gD6q.ss21x.
    lastupdate = 884801337
    flags = ADMCHG,ADMIN,NOCHECK
```

Figure 13-32. /etc/security/passwd File

AU147.0

Notes:

The **/etc/security/passwd** file can only be accessed by root. The **login**, **passwd**, **pwdadm** and **pwdck** commands (which run with **root** authority) update this file. This file is in stanza format with a stanza for each user. The valid entries are:

- **password** Either the encrypted password or * for invalid, or blank for no password
- **lastupdate** The date and time of the last password update in seconds from January 1, 1970
- **flags** ADMCHG - the password was last changed by an administrator or **root**
ADMIN - the user's password can only be changed by **root**
NOCHECK - password restrictions are not in force for this user.
(see **/etc/security/user** for password restrictions)

In AIX, additional files can be created to be used as an index for the **/etc/security/passwd** file. These index files provide for better performance during the login process. These indexes are created using the **mkpasswd** command.

/etc/security/user File

```
# cat /etc/security/user
```

```
default:
    admin = false
    login = true
    su = true
    daemon = true
    rlogin = true
    sugroups = ALL
    admgroups =
    ttys = ALL
    auth1 = SYSTEM
    auth2 = NONE
    tpath = nosak
    umask = 022
    expires = 0
    .
    .
    .
```

Figure 13-33. /etc/security/user File

AU147.0

Notes:

admin

Defines the administrative status of the user. Possible value: true or false.

login

Defines whether a user can login. Possible values: true or false.

su

Defines whether other users can switch to this user account. The **su** command supports this attribute. Possible values: true or false.

daemon

Defines whether the user can execute programs using the system resource controller (SRC). Possible values: true or false.

rlogin

Defines whether the user account can be accessed by remote logins. Commands **rlogin** and **telnet** support this attribute. Possible values: true or false.

sugroups

Defines which groups can switch to this user account. Alternatively you may explicitly deny groups by preceding the group name with a ! character. Possible values: A list of valid groups separated by commas, ALL or *

admgroups

Lists the groups that a user administers. The value is a comma-separated list of valid group names.

ttys

Defines which terminals can access the user account. Alternatively you may explicitly deny terminals by preceding the terminal name with the ! character. Possible values: List of device paths separated by commas, ALL or *

auth1

Defines the primary authentication method for a user, which by default is set to the password program. The commands login, telnet, rlogin and **su** support these authentication methods.

auth2

Defines the secondary authentication methods for a user. It is not a requirement to pass this method to login.

tpath

Defines the user's trusted path characteristics. Possible values: nosak, notsh, always or on. (For more information refer to online documentation).

umask

Defines the default umask for the user. Possible values: 3-digit octal value.

expires

Defines the expiration time for the user account. Possible values: a valid date in the form MMDDHHMMYY or 0. If 0, the account does not expire. The 'YY' supports the last 2 digits of the years 1939 to 2038. If 0101000070 then the account is disabled.

/etc/security/user File Continued

```
default:
    SYSTEM = "compat"
    logintimes =
    pwdwarntime = 0
    account_locked = false
    loginretries = 0
    histexpire = 0
    histsize = 0
    minage = 0
    maxage = 0
    maxexpired = -1
    minalpha = 0
    minother = 0
    minlen = 0
    mindiff = 0
    maxrepeats = 8
    dictionlist=
    pwdchecks =
```

Figure 13-34. /etc/security/user File Continued

AU147.0

Notes:

SYSTEM

This attribute can be used to describe multiple or alternate authentication methods the user must use successfully before gaining access to the system. Possible tokens are:

- | | |
|---------------|---|
| files | which will allow only local users access to the system |
| compat | which is the normal login procedure and therefore will allow local and NIS users access to the system |
| DCE | which is the Distributed Computing Environment authentication. |

logintimes

Defines the times a user can login. The value is a comma separated list of items as follows:

```
[!][MMdd[-MMdd]]:hhmm-hhmm  
or  
[!][MMdd[-MMdd]][:hhmm-hhmm]  
or  
[!][w[-w]]:hhmm-hhmm  
or  
[!][w[-w]][:hhmm-hhmm]
```

where MM is a month number (00=January, 11=December), dd is the day on the month, hh is the hour of the day (00 - 23), mm is the minute of the hour, and w is a weekday (0=Sunday, 6=Saturday).

pwdwarntime

The number of days before a forced password change that a warning will be given to the user informing them of the impending password change. Possible values: a positive integer or 0 to disable this feature.

account_locked

Defines whether the account is locked. Locked accounts cannot be used for login or **su**. Possible values: true or false.

loginretries

The number of invalid login attempts before a user is not allowed to login. Possible values: a positive integer or 0 to disable this feature.

histexpire

Defines the period of time in weeks that a user will not be able to reuse a password. Possible values: an integer value between 0 and 260. 26 (approximately 6 months) is the recommended value.

histsize

Defines the number of previous passwords which cannot be reused. Possible values: an integer between 0 and 50.

minage

Defines the minimum number of weeks between password changes. Default is 0. Range: 0 to 52.

maxage

Defines the maximum number of weeks a password is valid. The default is 0, which is equivalent to unlimited. Range: 0 to 52.

maxexpired

Defines the maximum number of weeks after maxage that an expired password can be changed by a user. The default is -1, which is equivalent to unlimited. Range: -1 to 52. maxage must be greater than 0 for maxexpired to be enforced. (**root** is exempt from maxexpired).

minalpha

Defines the minimum number of alphabetic characters in a password. The default is 0. Range: 0 to 8.

minother

Defines the minimum number of non-alphabetic characters in a password. The default is 0. Range: 0 to 8.

minlen

Defines the minimum length of a password. The default is 0. Range: 0 to 8. Note that the minimum length of a password is determined by minlen and/or "minalpha + minother", whichever is greater. "minalpha + minother" should never be greater than 8. If "minalpha + minother" is greater than 8, then minother is reduced to "8 - minalpha".

mindiff

Defines the minimum number of characters in the new password that were not in the old password. The default is 0. Range: 0 to 8.

maxrepeats

Defines the maximum number of times a given character can appear in a password. The default is 8, which is equivalent to unlimited. Range: 0 to 8.

dictionlist

Defines the password dictionaries used when checking new passwords. The format is a comma separated list of absolute path names to dictionary files. A dictionary file contains one word per line where each word has no leading or trailing white space. Words should only contain 7 bit ASCII characters. All dictionary files and directories should be write protected from everyone except root. The default is valueless which is equivalent to no dictionary checking.

pwdchecks

Defines external password restriction methods used when checking new passwords. The format is a comma separated list of absolute path names to methods or method path names relative to **/usr/lib**. A password restriction method is a program module that is loaded by the password restrictions code at run time. All password restriction methods and directories should be write protected from everyone except root. The default is valueless, which is equivalent to no external password restriction methods.

Group Files

more /etc/group

```
system:!:0:root,john
staff:!:john
bin:!:2:root,bin
sys:!:3:root,bin,sys
adm:!:4:bin,adm
uucp:!:5:uucp
mail:!:6:
security:!:7:root
nobody:!:4294967294:nobody,lpd
usr:!:100:guest
accounts:!:200:john
```

more /etc/security/group

```
system:
        admin=true

staff:
        admin=false

accounts:
        admin=false
        adms=john
```

Figure 13-35. Group Files

AU147.0

Notes:

The fields in the **/etc/group** file are:

- group - up to eight alphanumeric characters (not all upper case)
- password - this field is not used in AIX and should contain a !
- gid - the group ID
- members - a comma-separated list of the users who belong to this group

The **/etc/security/group** file is a stanza file with one stanza for each group. The valid entries are:

admin true or false, whether the group is an administrative group

adms a comma-separated list of the users who are administrators for the group. If **admin=true** this stanza is ignored because only root can change an administrative group.

/etc/security/login.cfg File

```
default:
  herald ="This is the console. Restricted use only.\n\rlogin:"
  logintimes =
  logindisable = 0
  logininterval = 0
  loginreenable = 0
  logindelay = 0
```

Figure 13-36. /etc/security/login.cfg File

AU147.0

Notes:

herald

Specifies the initial message to be printed out when **getty** or **login** prompts for a login name. This value is a string that is written out to the login port. If the herald is not specified, then the default herald is gotten from the message catalog associated with the language set in **/etc/environment**.

logintimes

Defines the times a user can use this port to login.

logindisable

Number of unsuccessful login attempts before this port is locked. Use this in conjunction with logininterval.

logininterval

The number of seconds during which logindisable unsuccessful attempts must occur for a port to be locked.

loginreenable

The number of minutes after a port is locked that it will automatically unlocked.

logindelay

The delay in seconds between unsuccessful login attempts. This delay is multiplied by the number of unsuccessful logins - that is, if the value is two, then the delay between unsuccessful logins will be two seconds, then four seconds, then six seconds and so forth.

Validating the User Environment

- **pwdck** verifies the validity of local authentication information.

`pwdck {-n|-p|-t|-y} {ALL | username}`

Verifies that `/etc/passwd` and `/etc/security/passwd` are consistent with each other and with `/etc/security/login.cfg` and `/etc/security/user`

- **usrck** verifies the validity of a user definition.

`usrck {-n|-p|-t|-y} {ALL | username}`

Checks each user name in `/etc/passwd`, `/etc/security/user`, `/etc/security/limits` and `/etc/security/passwd`. Also, checks are made to ensure that each has an entry in `/etc/group` and `/etc/security/group`.

- **grpck** verifies the validity of a group

`grpck {-n|-p|-t|-y} {ALL | groupname}`

Verifies that the files `/etc/passwd`, `/etc/security/user`, `/etc/group` and `/etc/security/group` are consistent

Figure 13-37. Validating the User Environment

AU147.0

Notes:

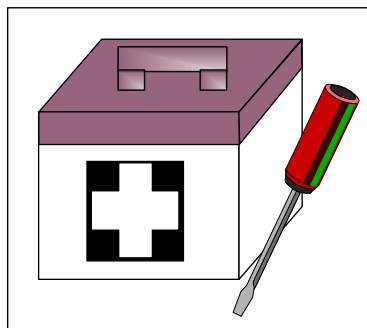
These commands can be executed by **root** or any user in the **security** group to clean up after a change to the user configuration. Because they run with **root** permissions, they give administrative users the ability to make necessary changes to the `/etc/security/passwd` file in a controlled way, without knowing the **root** password.

The **usrck** command verifies the validity of the user definitions in the user database files, by checking the definitions for ALL the users or for the users specified by the user parameter. You must select a flag to indicate whether the system should try to fix erroneous attributes.

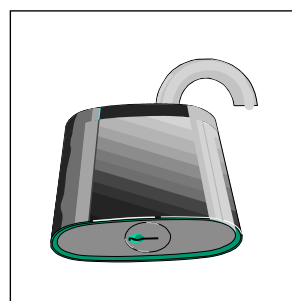
All the options for **pwdck**, **usrck**, and **grpck** are as follows:

- n reports errors but does not fix them
- p fixes errors but does not report them
- t reports errors and asks if they should be fixed
- y fixes errors and reports them

System Management Services



Utilities



PASSWORD

Figure 13-38. System Management Services

AU147.0

Notes:

The power-on and privileged passwords are security features that help protect the information on a your RS/6000.

These passwords can only be enabled or disabled through the Utilities menus in the System Management Services menus.

To get to the correct screen, boot the SMS programs. Select - **Utilities** -> **Password**.

PCI RS/6000 Passwords

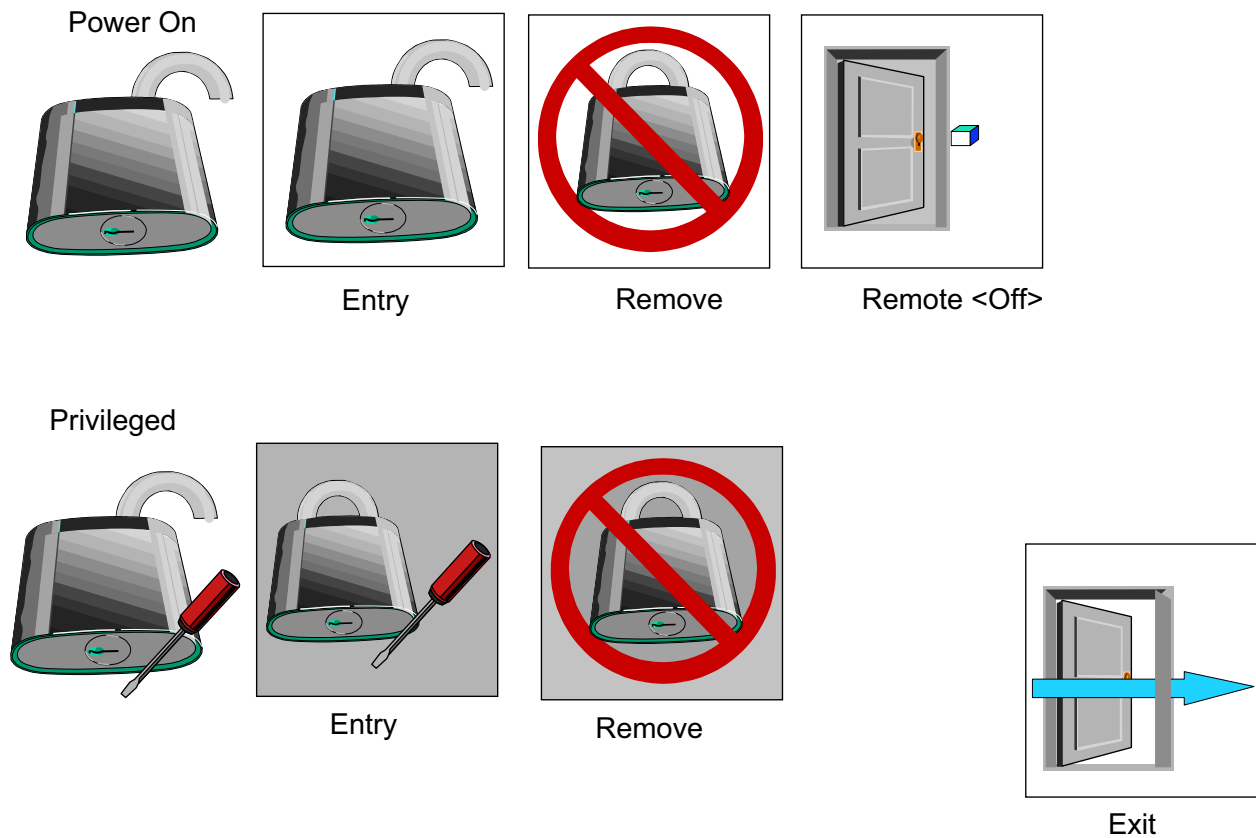


Figure 13-39. PCI RS/6000 Passwords

AU147.0

Notes:

You can use any combination of up to eight characters (A-Z, a-z, and 9-0). After you set a power-on password, you are prompted to enter it each time you power on the system. Before you can use the system, you must type the correct password and press the Enter key.

When you enter the correct password, the system is unlocked and resumes normal operations. If you enter the wrong password, you are prompted to enter the correct one. After three incorrect entries, you must power off the system and start again.

A power-on password can be set only after system power has been turned off and then on again. You cannot set a power-on password after doing a warm system startup.

The supervisory or privileged password protects against the unauthorized use of the System Management Services program. If you forget the supervisory password, there is no way to reset it. Some models of PCI RS/6000 systems allow you to take the covers off the system, remove the ISA/PCI riser, and remove the battery for at least 30 seconds.

However, **be careful** as some models require that they be returned to IBM service if the supervisory/privileged password is lost!

Documenting Security Policy and Setup

- Identify the different types of users and what data they will need to access
- Organize groups around the type of work that is to be done
- Organize ownership of data to fit with the group structure
- Set SVTX on shared directories
- Remember that UNIX/AIX has no concept of application ownership

Figure 13-40. Documenting Security Policy and Setup

AU147.0

Notes:

Plan and organize your user and group administration. Every user does not need their own group. Good planning up front will reduce any reorganizing of user and groups later on.

Always protect your shared directories by setting the sticky bit. Then users won't be removing each others file accidentally (or on purpose).

Checkpoint (1 of 2)

1. What are the benefits of using the su command to switch user to root over logging in as root?

2. Why is a umask of 027 recommended?

3. As a member of the security group, which password command would you use?

4. Which password change command does SMIT use?

5. True or false? When you delete a user from the system, all the user's files and directories are also deleted.

Notes:

Checkpointing (2 of 2)

6. If an ordinary user forgets their password, can the system administrator find out by querying the system as to what the user's password was set to? Why? _____
7. Password restrictions are set in which of the following files?
- /etc/passwd
 - /etc/security/passwd
 - /etc/security/restrictions
 - /etc/security/user
8. Which of the following statements are true?
- A user can only belong to one group.
 - A member of the security group can administer user accounts.
 - An admin user is a user whose account cannot be administered by any member of the security group.
 - The `chmod g + s` command sets the SUID permission of a file.
 - The root user, commonly known as the superuser has UID=0 and GID=0.

Notes:

Activity: Examine the Security Files

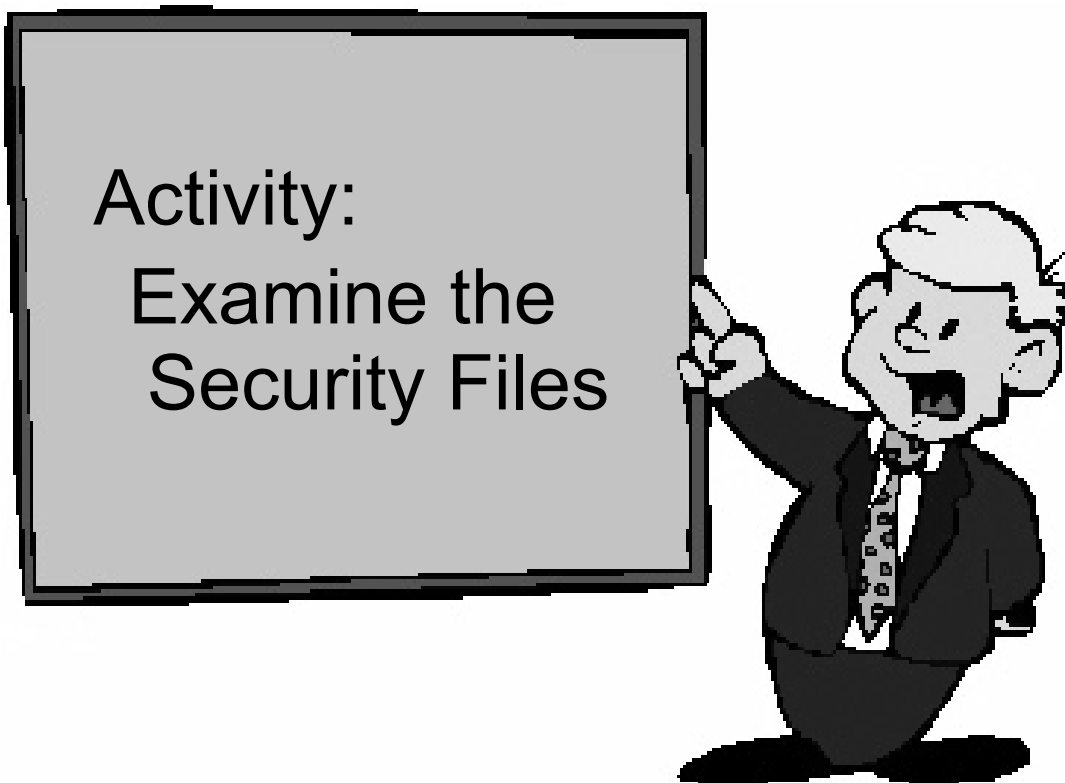


Figure 13-43. Activity: Examine the Security Files

AU147.0

Activity

In this activity, you will examine the security files covered in this unit and change your login prompt. If you are sharing your machine with other students, you will need to work together to configure the new login prompt.

Examine the Security Set Up

1. As **root**, examine any of the security files you have not yet seen.

```
#more /etc/password
```

```
# more /etc/security/passwd
```

```
# more /etc/security/user
```

```
# more /etc/group
```

```
# more /etc/security/group
```

```
# more /etc/profile
```

```
# more /etc/environment
```

```
# more /etc/security/envIRON
```

```
# who /etc/security/failedlogin
```

```
# who /etc/security/login.cfg
```

Customizing the Login Herald

2. ONLY ONE PERSON PER MACHINE CAN PERFORM THIS STEP.

You can customize the login prompt that appears on an individual terminal or customize it so that all terminals use the same login herald depending on what stanza you change in `/etc/security/login.cfg`. In this step, you will set up a global default herald.

The herald should read:

```
. Your Company Name
. Unauthorized users will be prosecuted.
```

First, make a backup of the `/etc/security/login.cfg` file. If you corrupt this file, you will not be able to log in again.

```
# cp /etc/security/login.cfg /etc/security/login.cfg.bk
```

3. Edit the **default** stanza in `/etc/security/login.cfg` and add the herald line.

```
# vi /etc/security/login.cfg
```

default:

```
sak_enabled = false
```

```
logintimes =
```

```
logindisable = 0
```

```
logininterval = 0
```

```
loginreenable = 0
```

```
logindelay = 0
```

```
herald = "\n\n YOUR COMPANY NAME\r\nUnauthorized users will be
prosecuted..\r\nlogin: "
```

Tip! - Do not use <ENTER> at what looks like the end of the line. Let `vi` wrap the line. The `\r` and `\n` to the <RETURN> and <NEW LINE> for you.

When finished, save the file and log out.

If you are using an ASCII screen, your new herald should be displayed. If you are using CDE, you will not see your changes unless you select the **Option** button and selection "**Command Line**". If you don't see your new herald, then check your entry in `/etc/security/login.cfg`.

END OF ACTIVITY

Unit Summary

- User and groups can be added and deleted from the system SMIT or by high level commands
- Passwords must be set for all users either using **pwdadm** or **passwd**
- Administrative users and groups can only be administered by root
- Every user must be in at least one group
- Certain groups give users additional privileges
- Security files are located in ASCII text files in **/etc** and **/etc/security**

Notes:

Unit 14. Scheduling

What This Unit Is About

This unit describes how jobs can be scheduled on the system.

What You Should Be Able to Do

After completing this unit, you should be able to:

- Use **crontab** files to schedule jobs on a periodic basis
- Use the **at** command to schedule a job or series of jobs at some time in the future
- Use the **batch** command to schedule jobs in a queue, to alleviate immediate system demand

How You Will Check Your Progress

Accountability:

- Checkpoint questions
- Exercise

Objectives

After completing this unit, students should be able to:

- Use **crontab** files to schedule jobs on a periodic basis
- Use the **at** command to schedule a job or series of jobs at some time in the future
- Use the **batch** command to schedule jobs in a queue, to alleviate immediate system demand

Notes:

cron Daemon

- Responsible for running scheduled jobs
- Starts:
 - **crontab** command events
(regularly scheduled jobs)
 - **at** command events
(one time only execution at specified time)
 - **batch** command events
(run when CPU load is low)

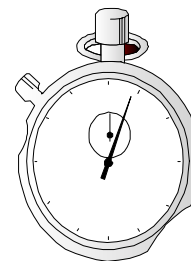


Figure 14-2. cron Daemon

AU147.0

Notes:

The system process that allows batch jobs to be executed on a timed basis is the **cron** daemon. Jobs are submitted to the **cron** daemon a number of different ways.

The **at** and **batch** facilities are used to submit a job for one-time execution. **crontab** files are used to execute jobs periodically - hourly, daily, weekly ...

cron is usually started at system startup by **/etc/inittab**. It runs constantly as a daemon. If killed, it is automatically restarted.

All **cron** events can be configured.

For example, **crontab** events, by default will be inspected every 60 seconds and run at a nice value of two less than the default and there may be up to 100 executing simultaneously.

This may be changed by modifying the **/var/adm/cron/queuedefs** file.

For example, if jobs were to run at a nice value of 10 less than the default with files inspected every two minutes and allow up to 200 jobs allowed, then the following entry should be made to the file:

```
c.200j10n120w
| | | |
| | | wait period (in seconds)
| | |
| | nice value
| |
| jobs
|
cron
```


crontab Files

- Used to start regularly occurring jobs
- Schedule is defined in
/var/spool/cron/crontabs/ \$USER
- Files to control users' crontab privileges

/var/adm/cron/cron.deny list user who cannot use crontab

/var/adm/cron/cron.allow list user who can use crontab

- An empty **cron.deny** exists by default

Figure 14-3. crontab Files

AU147.0

Notes:

The **cron** daemon starts processes at specified times. It can be used to control regularly scheduled jobs using files in the **/var/spool/cron/crontabs** directory or it can be used to schedule a command for one-time-only execution using the **at** command.

All users by default have the privilege to set up scheduled jobs to be monitored by **cron**. This is because the file **/var/adm/cron/cron.deny**, which denies privileges to users, is empty. As the administrator, you can restrict access to **cron** by adding user names to this text file.

There is another file that also restricts users privileges - **/var/adm/cron/cron.allow**. To use this file, you should remove **cron.deny** and create the **cron.allow** to list the users that are allow to use cron. If **cron.allow** exists and is empty, NO user will be able to use cron - that includes root. If both **cron.allow** and **cron.deny** exist, then **cron.allow** is the file that is used. If neither **cron.allow** or **cron.deny** exist, then only root can use cron.

crontab File

To view current crontab

```
# crontab -l
#
#COMPONENT_NAME: (CMDCTL) commands needed for
#basic system needs
#
#0 3 * * * /usr/sbin/skulker
#45 2 * * 0 /usr/lib/spell/compress
#45 23 * * * ulimit 5000;
/usr/lib/smdemon.cleau > /dev/null
0 11 * * * /usr/bin/errclear -d S,O 30
0 12 * * * /usr/bin/errclear -d H 90
```

Format:

```
minute hour date month day-of-week command
```

Figure 14-4. crontab File

AU147.0

Notes:

The crontab file can be view by using **crontab -l**. In this file is the schedule of jobs to run. Each user will have their own crontab file located in **/var/spool/cron/crontab/\$USER**.

The syntax for the lines in this file is:

minute (0-59)

hour (0-23)

date of the month (1-31)

month of the year (1-12)

day of the week (0-6, where 0=Sunday, 1=Monday, etc.)

Each field is separated by a space. To indicate a field is always true use an asterisk (*). To indicate multiple values in a field, use a comma (.). A range can also be specified by using a dash (-). Here are some examples:

To start the **backup** command at midnight, Monday through Friday:

```
0 0 * * 1-5 backup -0 -u -f /dev/rmt0
```

To execute a command called **script1** every 15 minutes between 8 AM and 5 PM, Monday through Friday:

```
0,15,30,45 8-17 * * 1-5 /home/team01/script1
```

Editing crontab

To edit crontab file:

```
# crontab -e
```

Safer method:

```
# crontab -l > /tmp/crontmp  
  
# vi /tmp/crontmp  
  
# crontab /tmp/crontmp
```

Figure 14-5. Editing crontab

AU147.0

Notes:

To schedule a job, you must create a **crontab** file. The cron daemon keeps the crontab files in memory so you cannot update the **cron** entries by just modifying the file on disk.

To edit the file, one method is to use **crontab -e**. This will open up your crontab file with the editor set with the EDITOR variable. Edit the file as you normally would any file. When the file is saved, the **cron** daemon is automatically refreshed.

The **crontab -l** command will always show your file that cron is using. Another method to update the file is to use **crontab -l > mycronfile**. This creates a copy of the current crontab file and allows you to safely edit the **mycronfile** without effecting the current crontab file. To submit your changes, use the command:

crontab mycronfile. The content of the **mycronfile** will replace the content of your file in the **crontab** directory and refresh the cron daemon all at once. Now, you also have a backup of the **crontab** file in **mycronfile**.

Use **crontab -r** if you would like to remove your current crontab file.

The at and batch Commands

- The **at** command submits a uniquely occurring job for cron

```
# at now +2 mins  
banner hello > /dev/tty3  
<ctrl-d>  
job user.time.a will be run at date
```

```
# batch  
banner hello > /dev/tty3  
<ctrl-d>
```

Figure 14-6. The at and batch Commands

AU147.0

Notes:

The **at** command runs a unique occurring job for **cron** to process. It reads the commands to execute from standard input.

at can only be used by **root** unless one of the following file exists:

- **/var/adm/cron/at.deny**: if this file exists, anybody can use **at** except those listed in it. An empty **at.deny** file exists by default. Therefore, all users can use **at** by default.
- **/var/adm/cron/at.allow**: if this file exists, only users listed can use **at** (root included)

The Bourne shell is used by default to process the commands. If **-c** is specified the C shell is run, and if **-k** is specified the Korn shell is run. If you specify the **-m** option, **at** will mail you to say that the job is complete.

The time can be specified as an absolute time or date (for example, **5 pm Friday**), or relative to now (for example, **now + 1 minute**).

The **batch** command submits a job when the processor load is sufficiently low.

Example keywords or parameters that can be used with **at** are: **noon, midnight, am, pm, A for am, P for pm, N for noon, M for midnight, today, tomorrow.**

Controlling at Jobs

- To list **at** jobs:

at -l [user]
atq [user]

```
# at -l  
  
root.962649853.a   Mon Jul 3 14:44:13 EDT 2000  
root.962649886.a   Mon Jul 3 14:44:46 EDT 2000  
adm.962649912.a   Mon Jul 3 14:45:12 EDT 2000
```

- To cancel a job:

at -r job
atrm [job| user]

```
# at -r adm.962649912.a  
The adm.962649912.a at file is deleted
```

Figure 14-7. Controlling at Jobs

AU147.0

Notes:

To list **at** jobs use the **at -l** command or **atq**. **root** can look at other user's jobs with **atq user**.

To cancel a job use **at -r** or **atrm** followed by the job number. Use **atrm -** to cancel all of your jobs. **root** can cancel all jobs for another user using **atrm user**.

Documenting Scheduling

- Have a copy of user's crontab files
- Have a copy of /etc/inittab

Figure 14-8. Documenting Scheduling

AU147.0

Notes:

Exercise: Scheduling



Figure 14-9. Exercise: Scheduling

AU147.0

Notes:

This lab gives you the opportunity to schedule jobs using both `at` and `crontab`.
This exercise can be found in your Exercise Guide.

Checkpoint

1. True or false? The **at.allow** and **at.deny** files must be used to specify which users are allowed and denied use of the **at** command.

2. Using **cron**, how would you specify a job to run every Thursday at 10 past and 30 minutes past every hour?

3. How would you schedule the script "myscript" to run 10 minutes from now?

Notes:

Unit Summary

- **cron** daemon is responsible for running scheduled jobs
- **crontab** file holds schedule for recurring jobs
- The **at** command is used to schedule a command for one time only execution

Notes:

Unit 15. Printers and Queues

What This Unit Is About

This unit describes the concepts behind the spooling mechanisms in AIX Version 5.1.

What You Should Be Able to Do

After completing this unit, you should be able to:

- Describe the purpose and the benefits of a queuing system
- Identify the major components that are responsible for processing a print request
- Add a printer queue and device under different circumstances
- Submit jobs for printing
- View the status of the print queues

How You Will Check Your Progress

Accountability:

- Checkpoint questions
- Exercise

References

GG24-3570	<i>Printing for Fun and Profit Under AIX V5L</i>
Online	<i>AIX System Management Guide: Operating System and Devices</i>
Online	<i>AIX Guide to Printers and Printing</i>

Objectives

After completing this unit, students should be able to:

- Describe the purpose and the benefits of a queuing system
- Identify the major components that are responsible for processing a print request
- Add a printer queue and device
- Submit jobs for printing
- Manage jobs in the queue

Figure 15-1. Objectives

AU147.0

Notes:

AIX 5.1 Printing Environments

- Print subsystems
 - AIX print subsystem
 - System V print subsystem
- Print directly to local printer device
- Print directly to a remote printer via a socket program
- Infoprint Manager (or similar advanced print management system)

Figure 15-2. AIX 5.1 Printing Environments

AU147.0

Notes:

Introduction

The slide gives an overview of the different approaches that can be taken to printing under AIX 5.1. In the next two slides, we'll compare System V printing to the traditional AIX print subsystem. The remainder of this unit will focus on using the AIX print subsystem.

Note: You can use either the AIX print subsystem or the System V print subsystem. They will not run concurrently.

Print directly to a local printer device

This is the simplest form of printing. If your printer is directly attached to a serial or parallel port on the local machine, it is possible to print by just sending a file directly to the device. For example:

```
# cat /home/karlmi/myfile > /dev/lp02
```

In this approach, you lose the ability to serialize (spool) print requests. Only one user may print at a time. On the other hand, if a printer is being dedicated to one use, this may be a good solution. Examples might be logging to a printer or printing checks.

Print directly to a remote printer via a socket program

This is similar to printing to a device driver, except that in this case, you are sending the output to a program which makes a connection to the printer over the network.

Print using the System V print subsystem

In this environment, files to be printed are sent to the System V print service daemon (`lp sched`) using the `lp` or `lpr` commands. The print service daemon serializes the jobs so they will be printed in the order in which they were submitted. The print service may filter the file to format the data so that it matches the types of data acceptable to the printer. The print service then sends files, one at a time, to the interface program, which may do additional filtering before sending the file to the local printer driver or network printing application.

Print using the AIX print subsystem

In this environment, files to be printed are sent to the AIX print spooler daemon (`qdaemon`) using any of the AIX print commands (`enq`, `qprt`, `lp`, or `lpr`). The spooler daemon serializes the jobs. The spooler sends jobs, one at a time, to back-end programs that may filter the data and before sending it to the local printer driver or network printing application.

Print using IBM's Infoprint Manager (or similar advanced print management system)

Infoprint Manager provides serialization and filtering similar to the System V or AIX print subsystems. In addition, it adds extra capabilities of security, customization, and control not provided by either System V printing or AIX printing. For additional information, refer to the Infoprint Manager web site:

<http://www.printers.ibm.com/R5PSC.NSF/Web/ipmgraixhome>

AIX Print Subsystem: Advantages

- Powerful and flexible printer drivers
- System management tools
 - Limits fields and options validation
 - Easy printer customization
 - Single step print device and queue creation
 - Support for dial-in administration
- Customizable spooling subsystem

Figure 15-3. AIX Print Subsystem: Advantages

AU147.0

Notes:

Powerful and flexible printer drivers

AIX printer drivers provide many printing options that can be easily controlled using command line options to the `qprt` command. Printer defaults can be easily managed using SMIT or the command line.

System management tools

The AIX print subsystem includes mature and powerful system management using either WebSM or SMIT, as well as the command line. System management tools for the System V print subsystem are less mature in this initial release. Some specific system management advantages using the AIX print subsystem are:

- Limits fields and options validation
Limits fields give the user or administrator a range of valid values for print options and will prevent the user from using an invalid value.
- Easy printer customization
Printers can be customized using menu selections or command line options. Under System V printing, customizing printers often requires a knowledge of shell programming.
- Single step print device and queue creation
Under System V printing, you must first add a print device and then create the print queue.
- Support for dial-in administration
Under AIX printing, administrators can dial-in and use the text-based SMIT interface to administer the print subsystem. The Web-based System Manager (which is currently the only non-command line interface for the System V print subsystem) would be hard to use over a dial-in connection.

Customizable spooling subsystem

The AIX print subsystem is specifically designed so that it can be used to serialize other types of jobs beyond just printing.

System V Print Subsystem: Advantages

- Compatibility
- Availability of interface programs
- Security
- Support for forms
- Standard PostScript filters
- Long term strategy

Figure 15-4. System V Print Subsystem: Advantages

AU147.0

Notes:

Compatibility

System administrators with experience in other UNIX variants that use System V printing will find it easy to manage printing under AIX's System V print subsystem.

Availability of interface programs

Many printer manufacturers provide interface shell scripts to support using their products under System V printing. Usually only minor modifications are required for individual UNIX variations. Because the AIX print subsystem is proprietary, an interface program written for another operating system cannot be used in the AIX print subsystem. It must be completely re-written. This has led to a limited number of printers supported under AIX. With the support of System V printing in AIX 5.1, it will be easier for manufacturers to include support for AIX printing.

Security

Controlling user access to printers can be an important issue. For example, you might need to limit access to the printer used to print checks. System V printing includes built-in capabilities for restricting user access to certain printers. Using the AIX print subsystem, the back-end program must be customized to restrict user access.

Support for forms

If you are printing to pre-printed forms, it's important that other users not be able to print while the expensive forms are loaded on the printer. The System V print subsystem provides a mechanism for mounting forms on printers and allowing or denying user access based on the form which is mounted. To provide this capability under AIX printing, you must create multiple queues and manage which queues are enabled while a form is mounted.

Standard PostScript filters

The System V print subsystem includes a number of filters for converting a number of different file formats to PostScript. Some formatting and page selection capabilities are also included.

Long term strategy

IBM's long term printing strategy for AIX is to maintain compatibility with other UNIX systems. This means that new features and functions will be added to the System V print subsystem in later releases, while the AIX print subsystem will be supported, but not enhanced in future releases.

Concepts of Queues

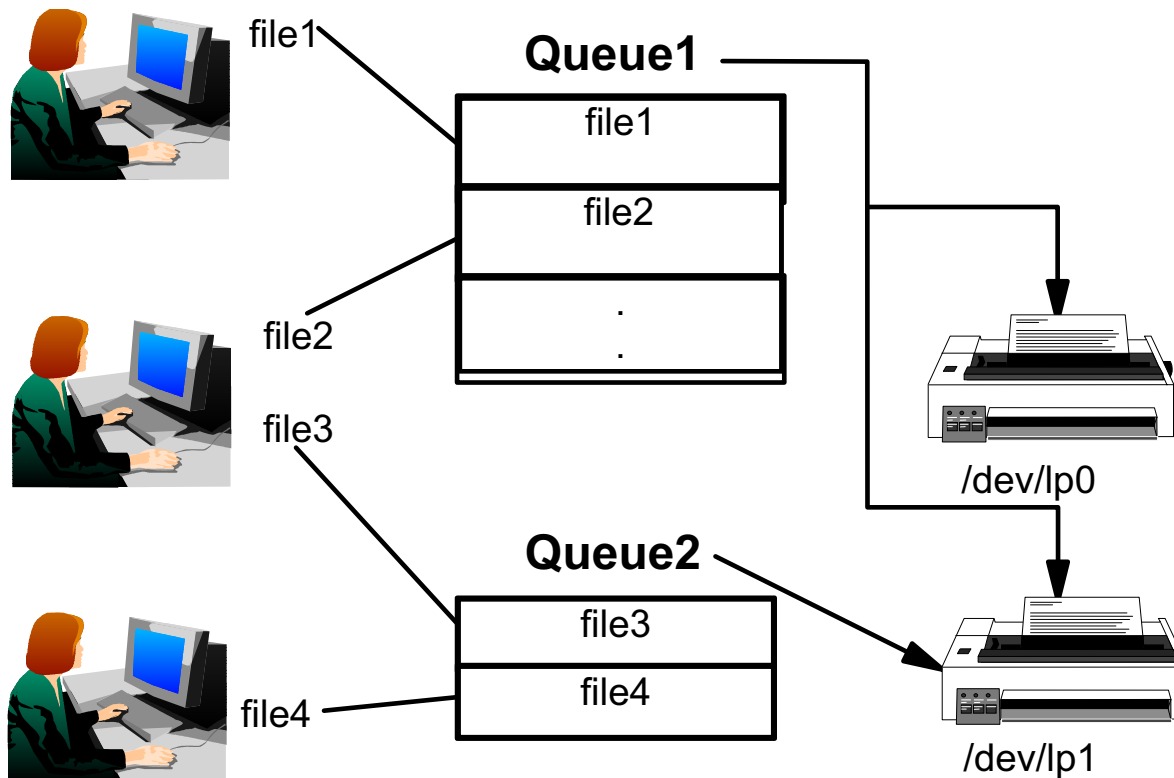


Figure 15-5. Concepts of Queues

AU147.0

Notes:

The purpose of the queuing system is to maintain a queue of jobs that are waiting for their turn to run (that is, use some system resource, like a printer or the CPU). The AIX Version 5.1 queuing system performs this function.

The queues also give control to the system administrator over the queuing mechanism. Therefore, the system administrator can perform tasks like cancelling jobs on queues, changing priorities of jobs, and so forth.

A queue enables the sharing of resources in an ordered fashion.

The diagram above illustrates three important issues:

- One print queue can point to a number of printers (and it is the job of the **qdaemon** to determine the next available printer to print on), for example, Queue1.
- Users may submit their jobs to a number of different queues.
- A printer can have a number of different queues pointing to it, for example, the printer **/dev/lp1** is accessed by both Queue1 and Queue2.

Printer Data Flow

```
# qprt -Pps [-c] file
```

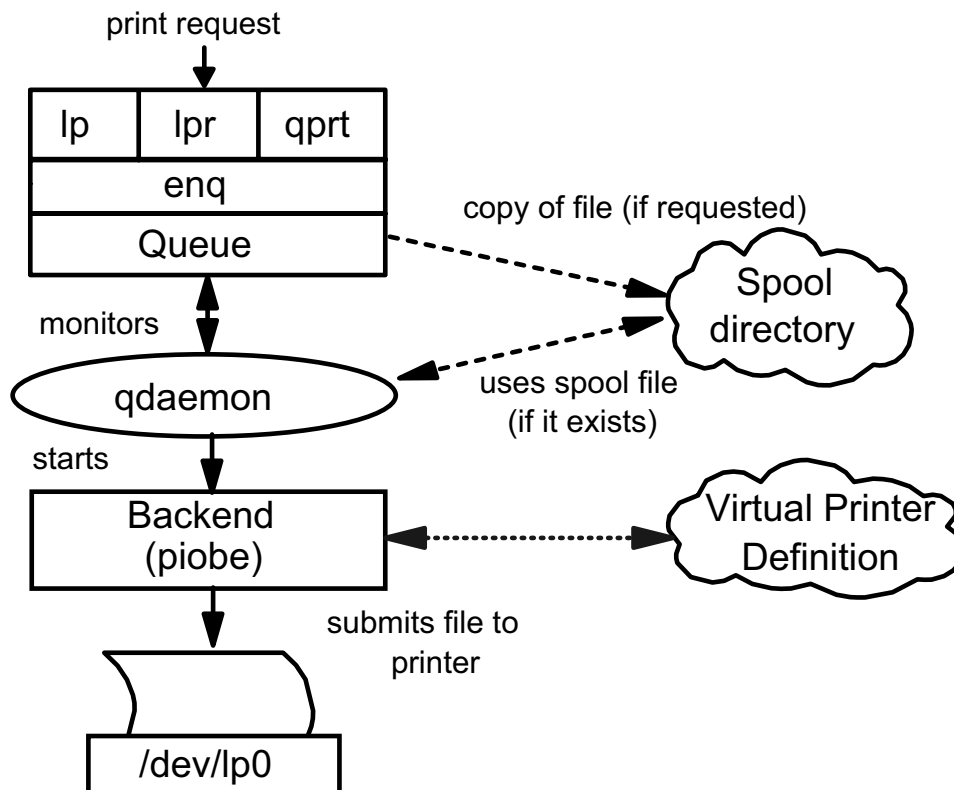


Figure 15-6. Printer Data Flow

AU147.0

Notes:

Local printing is implemented through a queuing mechanism. The user can issue one of the printer commands **qprt**, **lp**, **lpr**, or **enq** to submit a print job. Although a user can use any one of these four commands, the true entry point to the spooler is the **enq** command which is responsible for processing the job request, creating a job description file (JDF) and notifying the **qdaemon** of the new job.

The **qdaemon** process is running all of the time. The **qdaemon** maintains a list of all of the defined queues and monitors the queues for newly submitted jobs. **qdaemon** will try to process the job if the destination device is available, otherwise the job will remain in the queue and **qdaemon** will try again later.

The flow of the queuing system shown:

- The printing command calls **enq**. **enq** checks to see if the queue name desired is a valid queue and all of the parameters are correct. If so, it continues, if not, an error message is returned to the user.

- An entry is made in the **/var/spool/lpd/qdir** directory identifying the job to be run. If the printer command uses an option to indicate that a copy of the file is to be made, the copy is placed in the spool directory **/var/spool/qdaemon**.
- The **qdaemon** is notified of a new job in its **qdir** directory.
- When the queue is ready for the job, the **qdaemon** reads information from the **/etc/qconfig** file describing the queue.
- The **qdaemon** updates the **/var/spool/lpd/stat** file for the appropriate queue to show that the queue is now working on a new job.
- The **qdaemon** starts the backend program passing the file names and appropriate options on the command line.
- The backend determines the correct data stream characteristics and merges these with the actual file. The data stream characteristics are stored as **virtual printer definitions** in the **/var/spool/lpd/pio/@local** directory.
- The backend program sends its data stream to the device driver for the appropriate printer.

When a file is spooled, a copy of that file is sent to the print spool directory, **/var/spool/qdaemon**. The copy will remain in that directory until it is printed. This means that if you spool a file to the printer, a user could continue to make revisions to the original since the copy in the print spool directory will not be altered. This ensures that the file that is sent to the printer gets printed in its original form, even if a user edits the original file that is on disk. Spooled files will take up disk space in **/var** until they are printed.

When a file is queued, one line of information is sent to the **/var/spool/lpd/qdir** directory which points back to the original file on disk. If revisions are made to the file on disk before it is pulled from the queue to print, the revised file will get printed.

System Files Associated with Printing

/etc/qconfig	queue configuration files
/var/spool/*	spooling directories
/var/spool/lpd/qdir/*	queue requests
/var/spool/qdaemon/*	temporary enqueued files
/var/spool/lpd/stat/*	line printer status information
/var/spool/lpd/pio/@local	virtual printer directories

Figure 15-7. System Files Associated with Printing

AU147.0

Notes:

The **/etc/qconfig** file describes the queues and devices available for use by the printing commands.

The **/var/spool** directory contains files and directories used by the printing programs and daemons.

The **/var/spool/lpd/qdir** directory contains information about files queued to print.

The **/var/spool/qdaemon** directory contains copies of the files that are spooled to print.

The **/var/spool/lpd/stat** directory is where the information on the status of jobs is stored. It is used by the **qdaemon** and backend programs.

The **/var/spool/lpd/pio/@local** directory holds virtual printer definitions. This is where the attributes of printers are paired with the attributes of corresponding data stream types.

It is recommended that SMIT be used to update these device-related files. In most cases, updating standard system files is not recommended.

qdaemon

- **qdaemon** manages queues
- Started in the **/etc/inittab** file
- Invokes the backend programs
- Optionally records accounting data

Figure 15-8. qdaemon

AU147.0

Notes:

The **qdaemon** program schedules jobs that have been enqueued.

It is a background process that is usually started at system IPL via the **startsrc** command run from **/etc/inittab**.

qdaemon is controlled by the **/etc/qconfig** file. **/etc/qconfig** contains a stanza for each queue. The stanza identifies any queue management options and points to a queue device stanza which identifies the destination printer, the formatting options, and the backend program. The backend program is called by **qdaemon** to actually process each request. The backend program is determined by how the printer is connected to the RS/6000. For local printing the backend program is **/usr/lib/lpd/piobe**. For a remote printer, it is **/usr/lib/lpd/rembak**.

The **backend** program uses printer attribute information to prepare the printer and format the data for output. It will also print header and trailer pages if they are enabled.

The /etc/qconfig File

```

lp0:          device = lp0dev          * 1 queue pointing to 1 device
              up = TRUE
              discipline = fcfs

lp0dev:      file = /dev/lp0
              backend = /usr/lib/lpd/piobe
              header = group
              trailer = never
              feed = never

lpq:         device = lpqdev1, lpqdev2 * 1 queue pointing to 2 devices
lpqdev1:    file = /dev/lp1
              backend = /usr/lib/lpd/piobe
lpqdev2:    file = /dev/lp2
              backend = /usr/lib/lpd/piobe

ps:         device = psdev            * 2 queues pointing to 1 device
psdev:      file = /dev/lp3
              backend = /usr/lib/lpd/piobe

asc:        device = ascdev
ascdev:     file = /dev/lp3
              backend = /usr/lib/lpd/piobe

```

Figure 15-9. The /etc/qconfig File

AU147.0

Notes:

The **/etc/qconfig** file is the key to customizing the queues. Although the file can be edited directly, it is recommended it be changed through high-level commands or via SMIT.

The **/etc/qconfig** file contains two types of stanzas:

- **Queue stanza**

This starts with the queue name, which can be up to 20 characters, followed by a colon. The queue name is used by the person submitting a job to indicate the desired queue. The first queue in the **/etc/qconfig** file is the default queue, which receives any job requests submitted without a specific queue name. A queue stanza points to a device stanza via the *device=* attribute in the queue stanza.

Attributes that can be found in the queue stanza include:

	Default:	Other:
discipline	fcfs	sjn
acctfile	false	file name
up	TRUE	FALSE

- **Queue Device stanza**

This starts with the device stanza name, which can be up to 20 characters, followed by a colon.

Attributes that can be found in the device stanza include:

	Default:	Other:
access	write	both (used for modems or backends needing read capability)
header	never	always group
trailer	never	always group
feed	never	integer
align	FALSE	TRUE

The device stanza must contain an attribute that designates the **backend** program. It may also have an attribute that specifies the pathname of the logical device that serves the queue with which this device stanza is associated.

The function of the **backend** is to manage the printing of the actual job. It also produces the final data stream that goes to the printer. The most common **backend** program for local printing is **piobe**.

If different users desire different default printers, then the **PRINTER** variable can be setup on a per user basis. The **PRINTER** variable should be set to the queue that the user wishes to be their own default queue for example:

```
# PRINTER=ps ; export PRINTER
```

Printer Menu

smit spooler

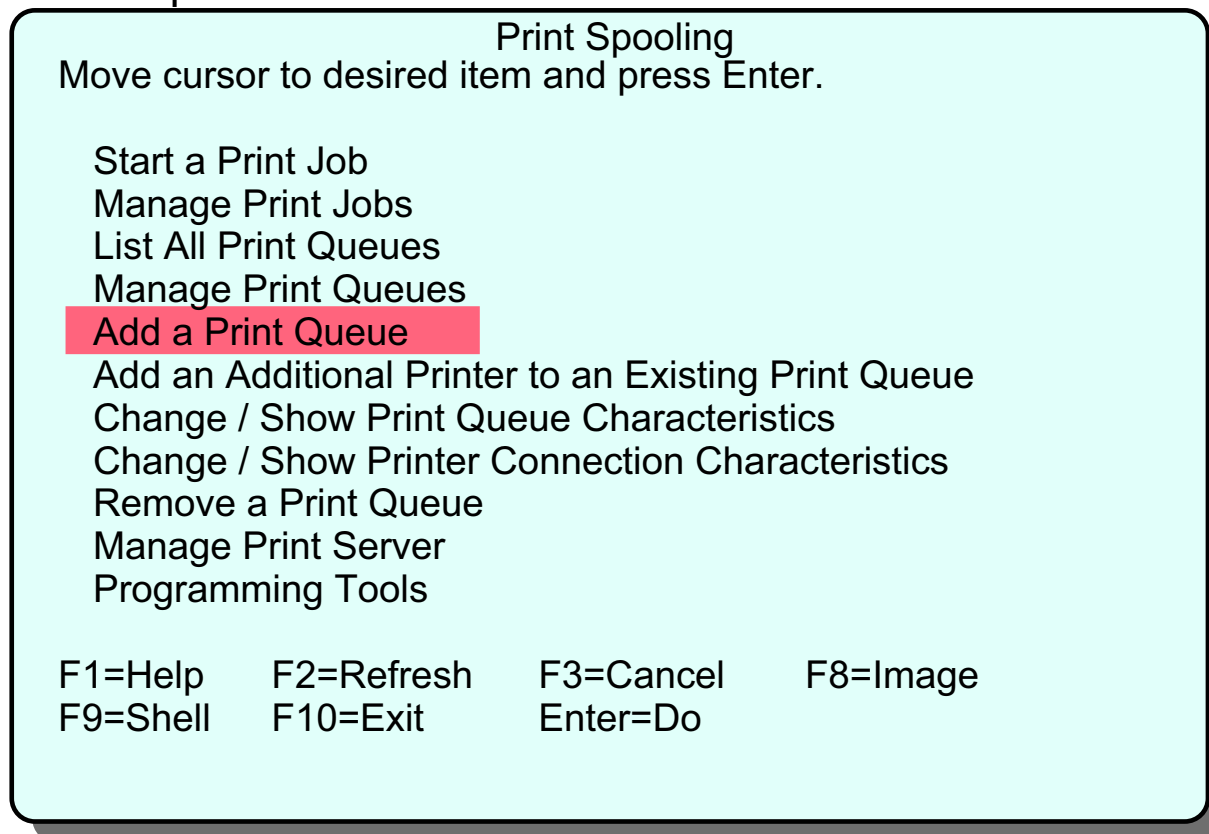


Figure 15-10. Printer Menu

AU147.0

Notes:

Start a Print Job

This option will start a print job by submitting the job to a print queue.

Manage Print Jobs

This option will put you into a submenu which will allow you to cancel jobs, show the status of jobs, prioritize jobs, hold and release jobs, and move jobs between print queues.

List All Print Queues

This displays a list of all the print queues and their associated printers.

Manage Print Queues

You can start and stop print queues, show the status of print queues and change the system's default print queue.

Add a Print Queue

This option adds a print queue to the system configuration and creates the associated queue device and printer device definition, if needed.

Add an Additional Printer to an Existing Print Queue

Adds another printer to an existing queue.

Change/Show Print Queue Characteristics

This option will provide access to screens that will allow you to change the printer setup, default print job attributes, accounting file setup, and queuing discipline.

Change/Show Printer Connection Characteristics

Changes or shows printer communication and startup characteristics.

Remove a Print Queue

Removes a Print Queue from the system configuration. It also removes the associated spooler queue device and printer device definition. If a print queue has more than one printer associated with it, then all the printers will be removed from the print queue.

Manage Print Server

Configures this machine as a print server. Allows you to control which clients have print access to this machine, list clients with print access, add and remove clients, and stop and start the server subsystem.

Programming Tools

Low-level utilities for manipulating databases and filters.

Printers and print queues can also be managed using the Web-based System Manager. To do so, use the fastpath **wsm printers**.

Configuring a Printer with a Queue

Add A Print Queue

Move cursor to desired item and press Enter. Use arrow keys to scroll.

#ATTACHMENT TYPE	DESCRIPTION
local	Printer Attached to Local Host
remote	Printer Attached to Remote Host
xstation	Printer Attached to Xstation
ascii	Printer Attached to ASCII Terminal
hpJetDirect	Network Printer (HP JetDirect)
file	File (in /dev directory)
ibmNetPrinter	IBM Network Printer
ibmNetColor	IBM Network Color Printer
other	User Defined Backend

F1=Help F2=Refresh F3=Cancel
 F8=Image F10=Exit Enter=Do
 /=Find n=Find Next

Figure 15-11. Configuring a Printer with a Queue

AU147.0

Notes:

In our example, assume that the printer is directly attached to our RS/6000 system. To configure a printer attached in this way, choose **local**.

Some applications contain their own print control mechanisms and thus require that a printer be configured without a queue. Use the SMIT fastpath **smit pdp** to define a printer without a queue.

Selecting a Printer Type

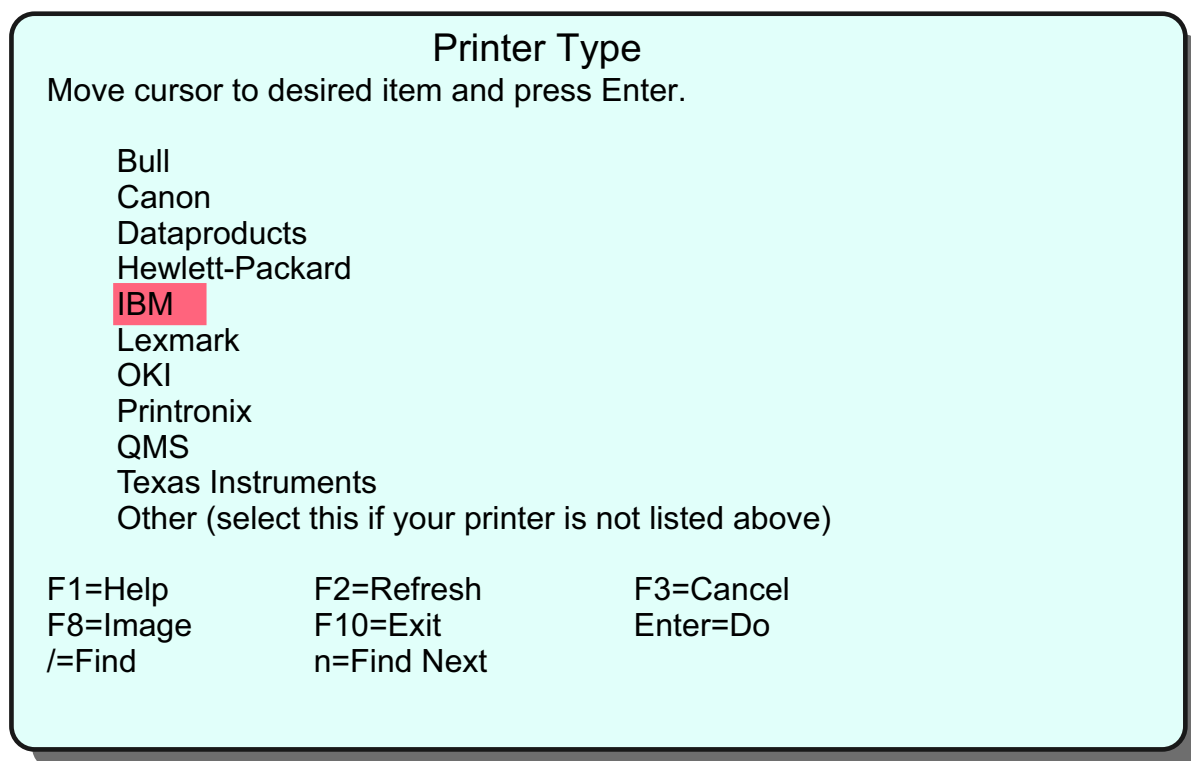


Figure 15-12. Selecting a Printer Type

AU147.0

Notes:

Selecting a Printer Type

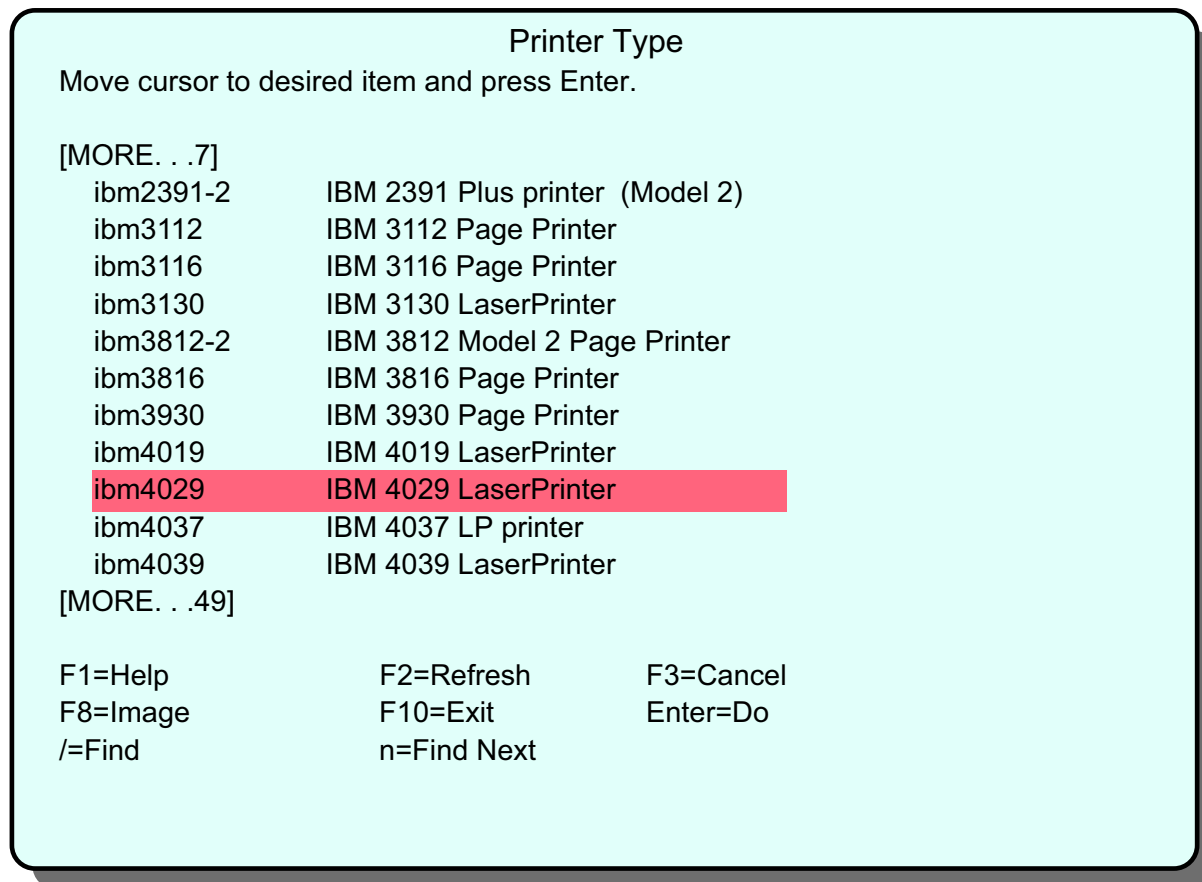


Figure 15-13. Selecting a Printer Type

AU147.0

Notes:

If you do not have the software installed for your printer, you will be prompted to insert the media to install the software first before configuring the device and the queue.

The choice of printer will determine the queue (or the virtual printer) setup. For example, an IBM 4029 Laser Printer is capable of handling PostScript, ASCII, GL Emulation and PCL Emulation. The SMIT print spooling menus will guide you through the creation of up to four separate queues which will submit to the same printer.

Printer Attachment

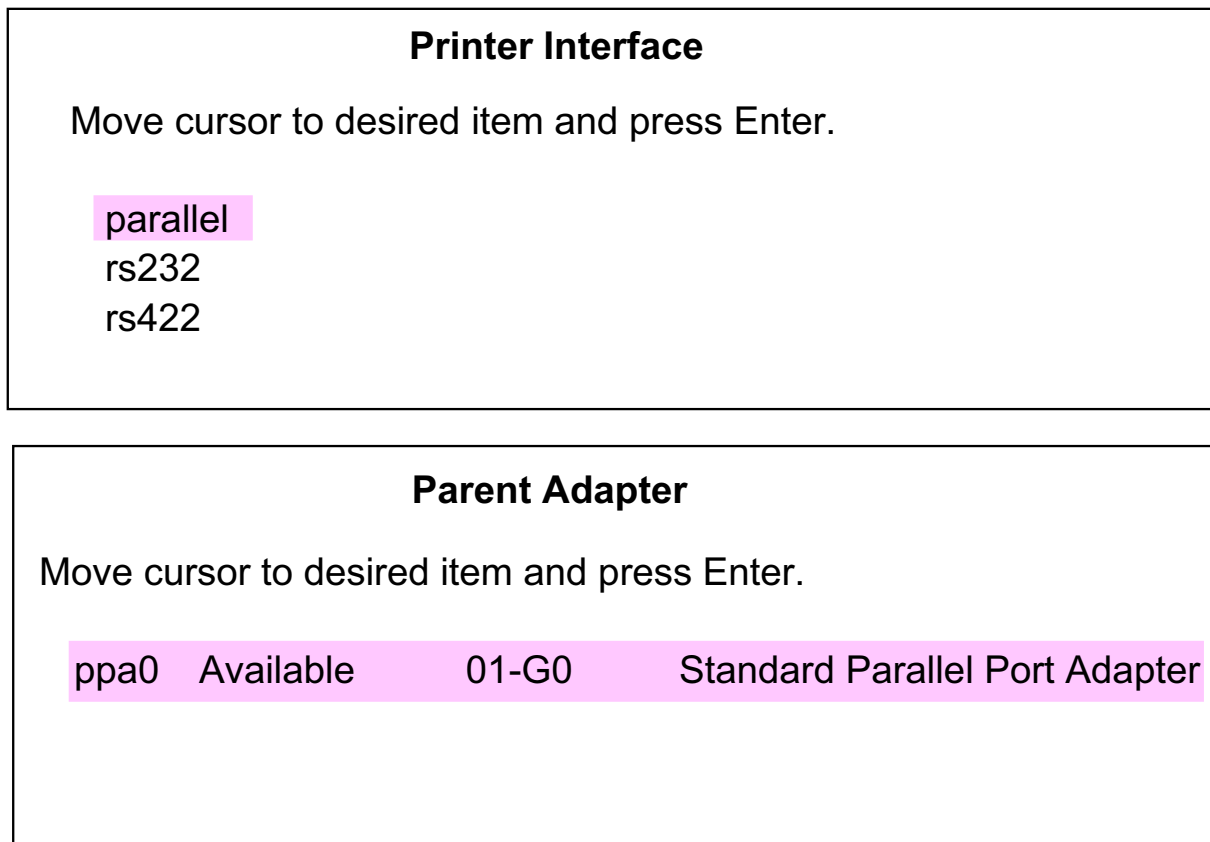


Figure 15-14. Printer Attachment

AU147.0

Notes:

If the device driver is not already installed, SMIT will prompt you to install the driver first before carrying on with the configuration.

After selecting a printer type, a pop-up window will be displayed where the printer interface must be chosen. Possible values are parallel, RS232 and RS422. Some printers support multiple attachment methods.

Then, a list of installed adapters that support that method of attachment will be presented.

Add the Print Queues

Add a Print Queue

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

[Entry Fields]
IBM 4029 LaserPrinter

Description

Names of NEW print queues to add

ASCII	[asc]	
GL Emulation	[]	
PCL Emulation	[]	
PostScript	[ps]	

Printer connection characteristics

* PORT number	[p]	+
Type of PARALLEL INTERFACE	[standard]	+
Printer TIME OUT period (seconds)	[600]	+#
STATE to be configured at boot time	available	+

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

Figure 15-15. Add the Print Queues

AU147.0

Notes:

This menu will vary depending on the characteristics of the physical printer. If the printer is capable of two or three different modes or emulations the system will prompt you for a separate queue name for each emulation. Once these queues are created they are sometimes referred to as "virtual" print devices.

Additional queues can be added to this printer after the initial queues are created.

Remote Printing

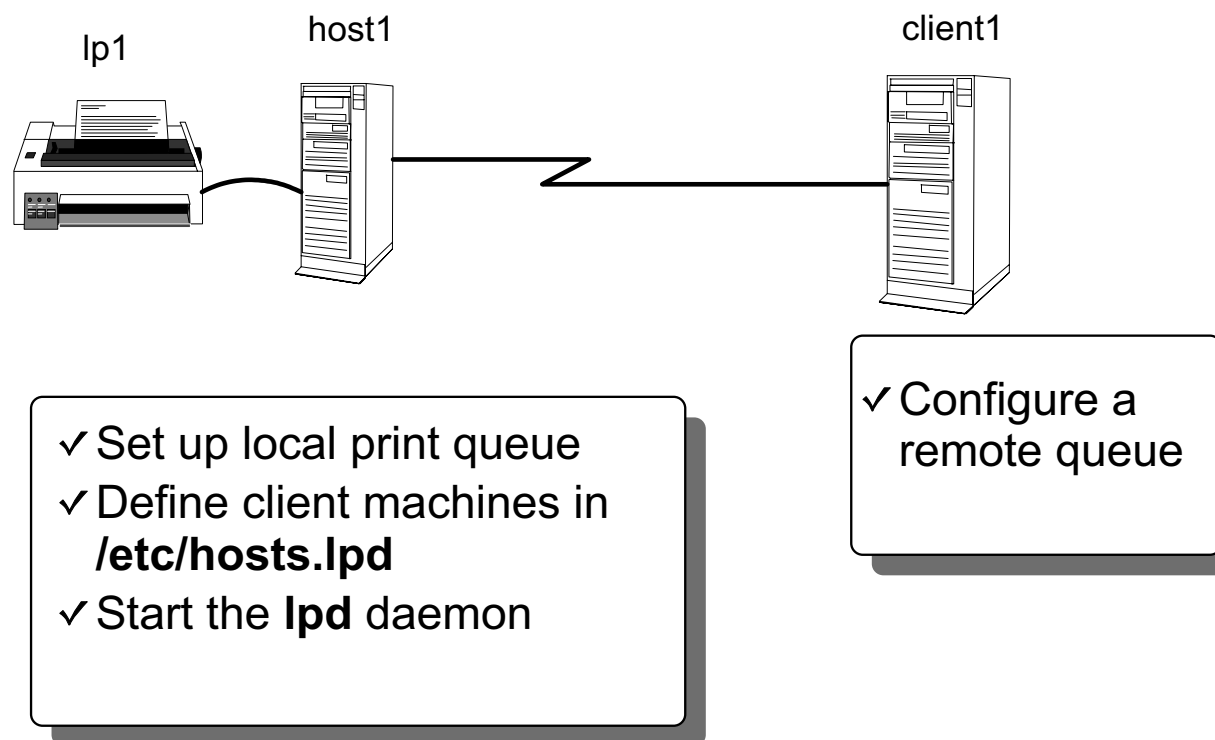


Figure 15-16. Remote Printing

AU147.0

Notes:

Once your system has the local queue set up, any user on that system can print. If the machine is networked, it can also provide printing for client machines by becoming a print server.

To set up a print server, you need to define the client machine names or IP addresses in **/etc/hosts.lpd** and then start the **lpd** daemon. Both of these tasks can be done through SMIT. To use SMIT, the fastpath to identify the client system is **smit mkhostsldpd**.

The **lpd** daemon is controlled by SRC. You should use SMIT to start it however, because SMIT will also add entries to **/etc/inittab** to ensure that it is started on reboot. The fastpath for this screen is **smit mkitab_lpd**.

Client Authorization

```
# smit mkhostsldap
```

```

                Add Print Access for a Remote Client

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

                                     [Entry Fields]
* Name of REMOTE CLIENT                [client1]
  (Hostname or dotted decimal address)

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

Figure 15-17. Client Authorization

AU147.0

Notes:

This step is done on the print server. On this screen, enter the client machine's name or IP address. A plus sign (+) is also valid. It indicates that this RS/6000 will be a print server to all machines.

Start lpd

smit mkitab_lpd

```

                                Start the Print Server Subsystem

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

                                [Entry Fields]
Start subsystem now, on system restart, or both      [both]      +
TRACE lpd daemon activity to syslog?                [no]        +
EXPORT directory containing print attributes?        [no]        +

Note:
Exporting this print server's directory containing its
print attributes will allow print clients to mount the
directory. The clients can use this server's print attributes
to display and validate print job attributes when starting
print jobs destined for this print server. Note that the
Network File System (NFS) program product must be installed
and running

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

```

Figure 15-18. Start lpd

AU147.0

Notes:

This step is done on the print server. The **lpd** daemon is controlled by the system resource controller (SRC). The commands **startsrc** and **stopsrc** can be used to control **lpd**. By using SMIT, an entry is placed in the **/etc/inittab** file to ensure that **lpd** is started each time the machine is booted.

Add a Remote Print Queue

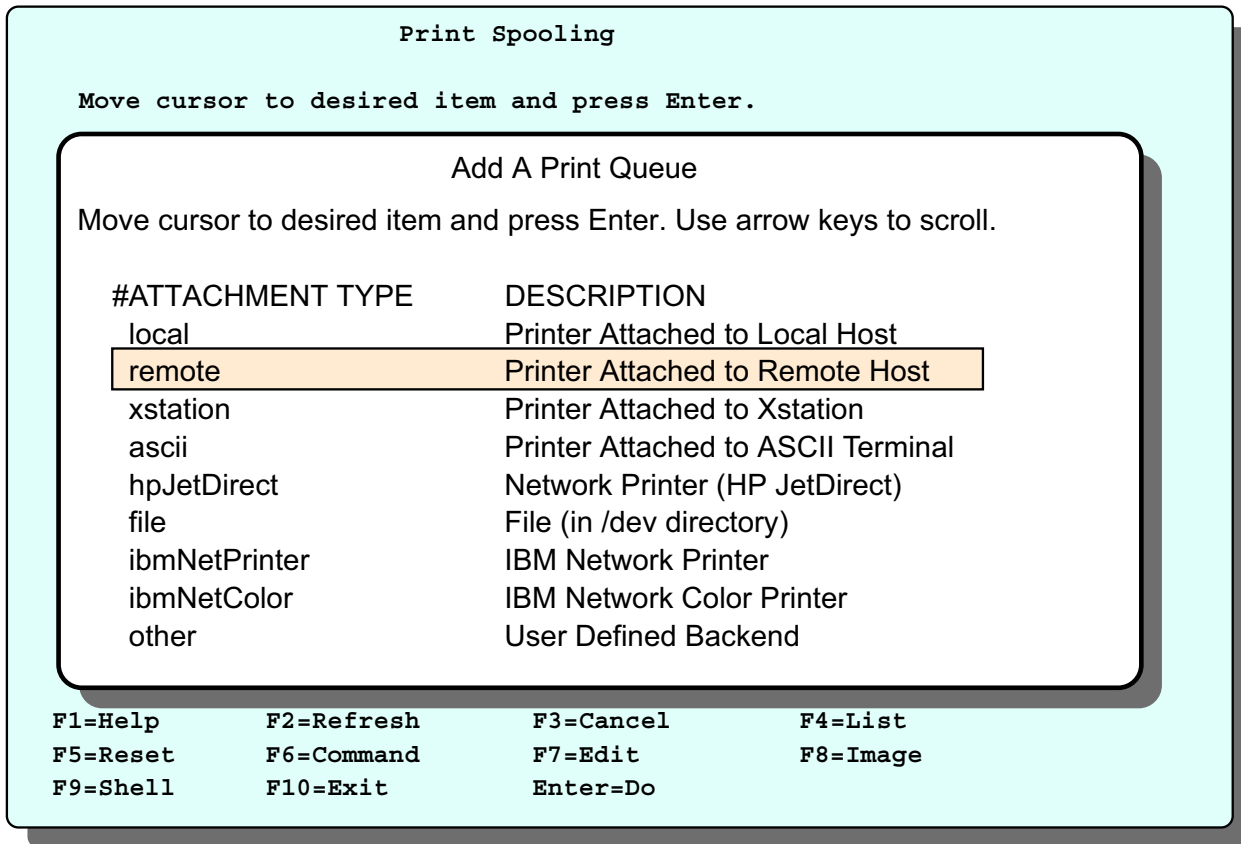


Figure 15-19. Add a Remote Print Queue

AU147.0

Notes:

This step is done on the client machine. The procedure to add remote queue starts the same way as a local queue: **smit spooler -> Add a Print Queue**. This time select **remote** as the attachment type.

You will be prompted to determine if you want to perform any type of filtering or pre-processing to the print job before it is sent. Normally, **Standard Processing** is selected. This just sends the job to the printer server and the print server is responsible for processing the job.

Define the Print Server on the Client

```

Add a Standard Remote Print Queue

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

                                     [Entry Fields]
*Name of QUEUE to add                [rq1]
*HOSTNAME of remote server           [lp1]
*Name of QUEUE on remote server      [host1]
Type of print spooler on remote server AIX Version 3 or 4 +
Backend TIME OUT period (minutes)    [] #
Send control file first?              no +
TO turn on debugging, specify output []
file pathname
DESCRIPTION of printer on remote server []

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

```

Figure 15-20. Define the Print Server on the Client

AU147.0

Notes:

Only three lines are required to complete the queue set up. You must name your local (to the client) queue name. Then, provide the name of the printer server. Lastly, name the queue on the print server.

Let's Review



Figure 15-21. Let's Review

AU147.0

Notes:

1. T or F The qdaemon is responsible for printing jobs.
2. To set up remote printing, what daemons are needed and do they run on the server, the client or both?
3. What does the **up = TRUE** indicate in the **/etc/qconfig** file?
4. What does **discipline** mean in reference to the **/etc/qconfig** file? What are its possible values?

Submitting Print Jobs

- AIX print systems offer compatibility to System V print commands
- To submit a job to a queue:

System V	BSD	AIX
lp	lpr	qprt

```
$ lp -d queueName filename
```

- OR -

```
$ qprt -P queueName filename
```

Figure 15-22. Submitting Print Jobs

AU147.0

Notes:

There are three sets of commands for submitting, listing and cancelling print jobs. They come from either System V, BSD or IBM versions of UNIX and are all available in AIX. The commands have slightly different options.

To submit a print job to a queue, use either **lp**, **lpr**, or **qprt**. All jobs will go to the system default queue unless the **PRINTER** or **LPDEST** variables are set. You can also specify, on the command line, which queue to use. Use **-d** with **lp** or use **-P** with **qprt** and **lpr**.

The commands **lp** and **qprt** both queue without spooling by default. Specify the **-c** option if spooling is desired. The command **lpr** spools and queues by default. The **-c** option will turn off spooling with **lpr**.

To print multiple copies, with **qprt** use the **-N #** option, with **lp** use **-n #** option, and with **lpr** use just a dash followed by the number of copies (**- #**).

The **lp**, **lpr** and **qprt** commands create a queue entry in **/var/spool/lpd/qdir** and (depending upon the options specified) copy the file to be printed to the **/var/spool/qdaemon** directory.

All the print commands, **lp**, **lpr**, and **qprt**, actually call the **enq** command which places the print request in a queue. **enq** can be used instead of the other commands to submit jobs, view job status, and so forth. To submit a job using **enq**:

\$ enq -Pqueuename filename

Ordinarily your request is serviced by the first device on the queue that becomes available. However, if more than one printer services a queue, you can request a specific printer by using the name of the queue followed by a colon (:) and then the name of the printer. For example, if a system with one queue (ps) is serviced by two printers (lp0 and lp1) and a print job needs to be printed on the lp1 printer, use the command:

\$ qprt -Pps:lp1 /home/team01/myfile

Listing Jobs in a Queue

- To list jobs in a queue:

SYSTEM V

BSD

AIX

lpstat

lpq

qchk

For example:

\$ qchk

Queue	Dev	Status	Job	Files	User	PP	%	Blks	Cp	Rnk
ps	lp0	DOWN								
		QUEUE	569	/etc/motd	root			1	1	1

Figure 15-23. Listing Jobs in a Queue

AU147.0

Notes:

Many of the print job control tasks require the user to supply a job number. The job number, along with other queue status information is available by checking the status of print jobs.

The fields from the **qchk** command are as follows:

Queue	Queue name
Dev	Logical device name for the queue
Status	Status of the queue (READY, DOWN, WAITING, RUNNING etc.)
Job	The job number assigned by the qdaemon
Files	Files sent to the queue
User	User who sent the print request
PP	Number of pages printed
%	Percent completed

- Blks** The number of 512-byte blocks the print job has been split into
- Cp** Copies of each job to be printed
- Rnk** Order on that queue

Other commands that can be used to view printer status include:

- lpstat** Shows status of all queues
- lpq** Shows status of the default queue
- qchk -A** Shows status of all queues
- enq -A** Shows status of all queues
- qchk -W** Shows status in wide-form mode. This is helpful if using long queue and device names, and 6-digit job numbers. This option is available with AIX V4.2.1 or later.

Change Characteristics of a Queue

smit chpq

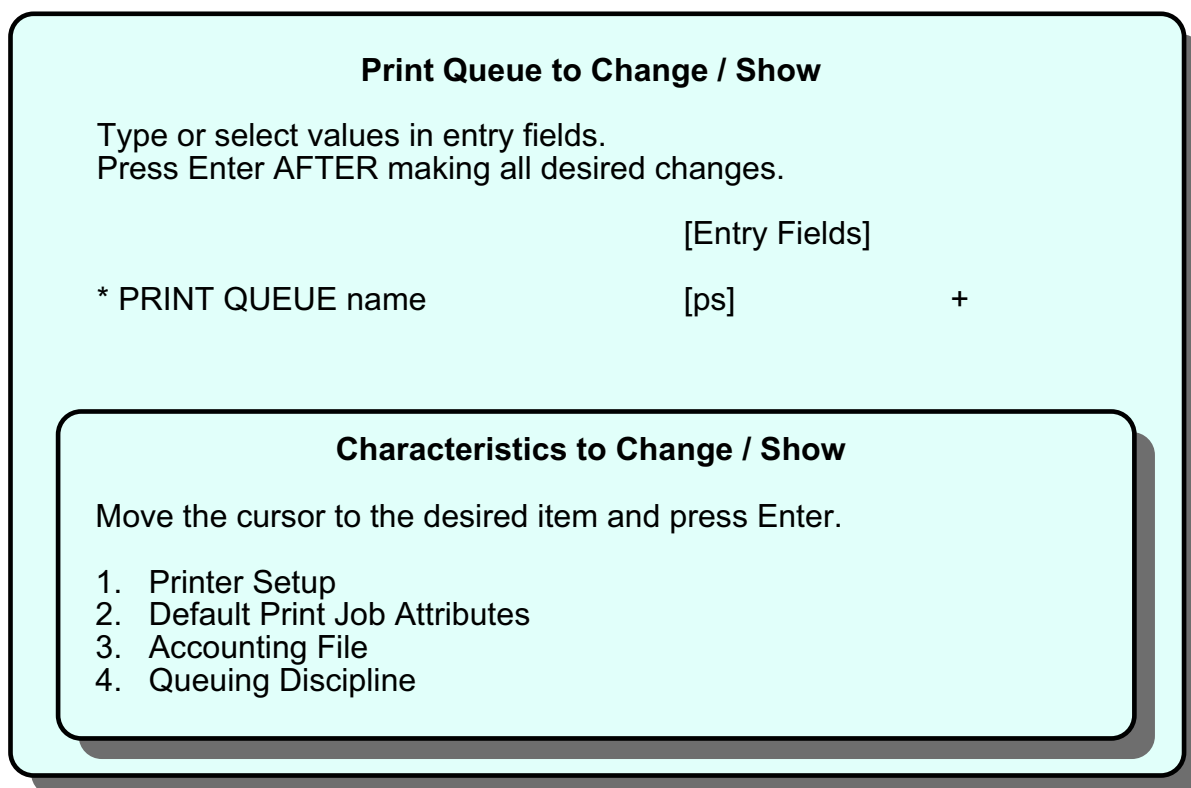


Figure 15-24. Change Characteristics of a Queue

AU147.0

Notes:

The four options contain attributes such as:

1. Printer Setup

- Automatic mode switching to postscript
- Paper size in trays and the manual feeder
- Envelope size
- ID of the font cards
- Paper trays for header and trailer pages
- Formatting flags for the header and trailer pages
- Users to get the intervention messages
- Flags prohibited for all print files
- Mode in which to leave the printer at the end of the job
- Width of printable area on header page

2. Default Print Job Attributes

- Text print options such as emphasized print

Job processing options such as page number where printing should begin

Text formatting options such as top Margin and lines per page

Paper/Page Options such as page orientation

Header/Trailer Page such as separator pages

Messages/Diagnostics

3. Accounting File

Accounting file name

4. Queuing Discipline

Queuing discipline

Removing a Queue

smit rmpq

Remove a Print Queue

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

	[Entry Fields]
Print queue to remove	ps:lp0
Local printer device	/dev/lp0
KEEP the local printer device?	no +

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

Figure 15-25. Removing a Queue

AU147.0

Notes:

It is not possible to remove a queue containing jobs. The jobs would have to be removed first.

Managing Queues

smit pqmanage

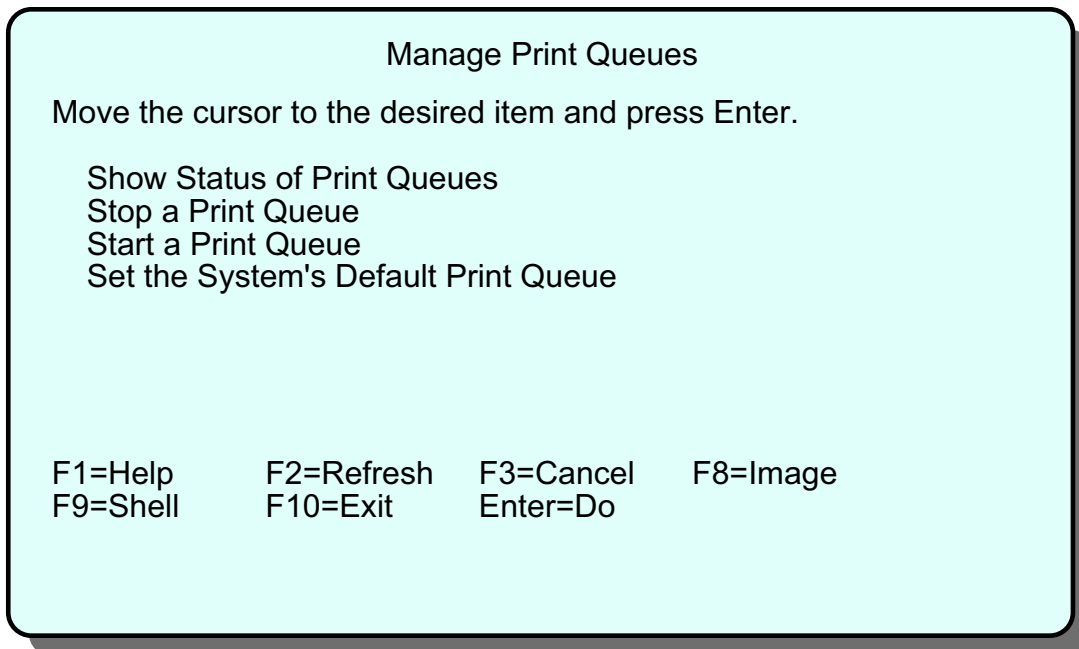


Figure 15-26. Managing Queues

AU147.0

Notes:

Show Status of Print Queue - give output similar to **qchk** and **lpstat**

Stop a Print Queue - Runs the **disable** command

Start a Print Queue - Runs the **enable** command

Set the System's Default Print Queue - Reorders the **/etc/qconfig** file to ensure the default queue is the first queue in the file.

Understanding Queue Status

Queue	Dev	Status	Job	Files	User	PP %	Bks	Cp	Rnk
ps	lp0	DOWN QUEUED	1569	/etc/motd	root		1	1	1

State	Description
DEV_BUSY	Printer is busy servicing other print requests
DEV_WAIT	Queue is waiting for the printer
DOWN	Queue is down and no jobs will be serviced from this queue until it is brought up
OPR_WAIT	The queue is waiting for operator intervention
QUEUED	Job is queued and waiting
READY	Everything is ready to receive a print request
RUNNING	Print file is printing
UNKNOWN	Problem with the queue - need to investigate further to determine cause

Figure 15-27. Understanding Queue Status

AU147.0

Notes:

The status of the queues and jobs can be displayed with **qchk**, **lpstat** or **lpq**. There are a number of different statuses that may be seen.

DEV_BUSY

This status can occur when more than one queue is defined to a print device and another queue is currently using the print device. It could result when the qdaemon attempts to use the printer port device and another application is currently using that print device. Normal recovery: You have to wait until the queue or application has released the print device, or kill the job or process that is using the printer port.

DEV_WAIT

This status means that the queue is waiting on the printer because the printer is offline, out of paper, jammed, or the cable is loose, bad or wired incorrectly. Normal recovery: Check to see if the printer is offline, out of paper, jammed or loosely cabled. Sometimes the jobs have to be removed from the queue before the problem can be corrected.

DOWN

This status is set when the device driver cannot communicate with the printer after TIME OUT seconds (which can be set through SMIT). This variable indicates the amount of time, in seconds, that the queuing system waits for a printer operation. If the printer is off, the queue will go down. Also, the operator can bring down the queue intentionally, which might be necessary for system maintenance. Normal recovery: Correct the problem that has brought the queue down and then bring the queue up again.

OPR_WAIT

This status is set when the backend program is waiting on the operator to change the paper, change forms and so on. This is usually software related. Normal recovery: Respond appropriately to the request that is made by the queuing system.

QUEUED

This status is set when a print file is queued and is waiting in line to be printed.

READY

This is the status of a queue when everything involved with the queue is ready to queue and print a job.

RUNNING

This status occurs when a print file is printing.

UNKNOWN

This status occurs when a user creates a queue on a device file that another queue is using and its status is DEV_WAIT. The queue cannot get a status from the printer device when it is on hold. Normal recovery: Bring down the other queue or fix the problem with the printer (paper out, jammed, offline and so on). Bring the new queue down and then back up so that the queue will register as READY.

Bringing Queues Up and Down

lpstat

Queue	Dev	Status	Job	Files	User	PP %	Bks	Cp	Rnk
draft	lp0	DOWN							
		QUEUED	132	/etc/motd	team01	1	1	1	
quality	lp0	READY							

- To enable a queue whose status is DOWN:

```
# enable draft
```

- To disable a queue whose status is READY:

```
# disable quality
```

You must be a member of the **printq** group or **root**

Figure 15-28. Bringing Queues Up and Down

AU147.0

Notes:

Occasionally, problems with printers can bring a queue down. Once the problem has been fixed it can be brought back up with:

```
# enable <queuename>
```

-OR-

Sometimes, you may wish to bring a queue down. This is recommended if any maintenance is going to be performed on the printer.

```
# disable <queuename>
```

-OR-

```
# enq -D -P <queuename>
```

Managing Print Jobs

smit jobs

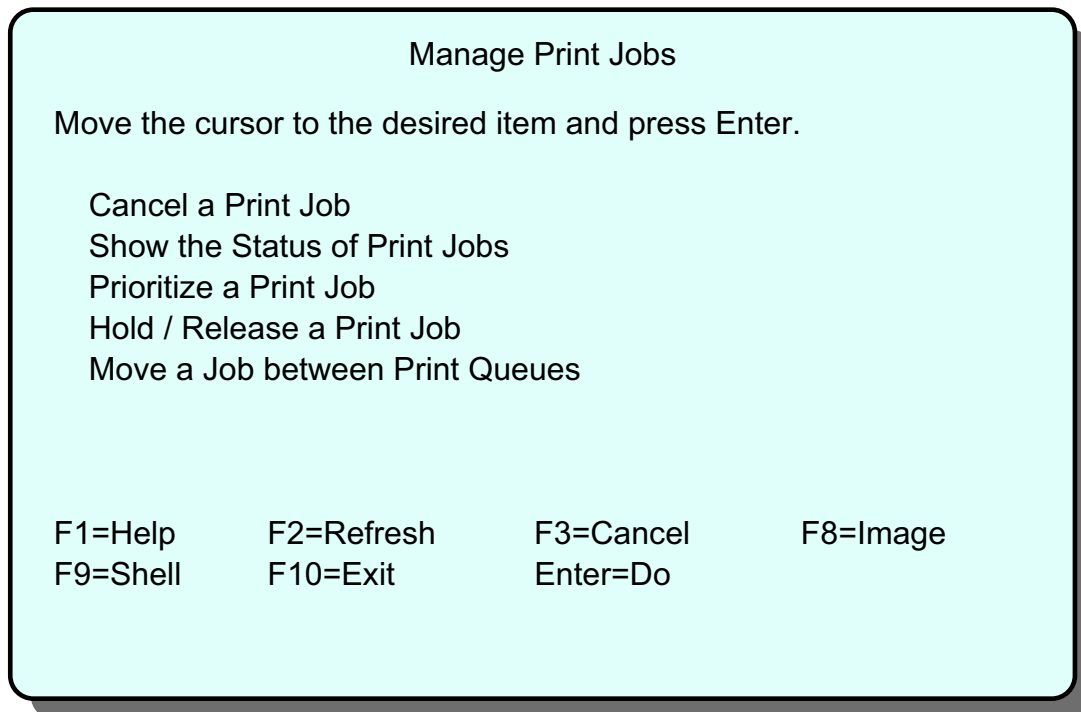


Figure 15-29. Managing Print Jobs

AU147.0

Notes:

The root user or a member of the print group can work with any print request. Normal users can only work with their own print jobs.

Cancelling Print Jobs

smit qcan

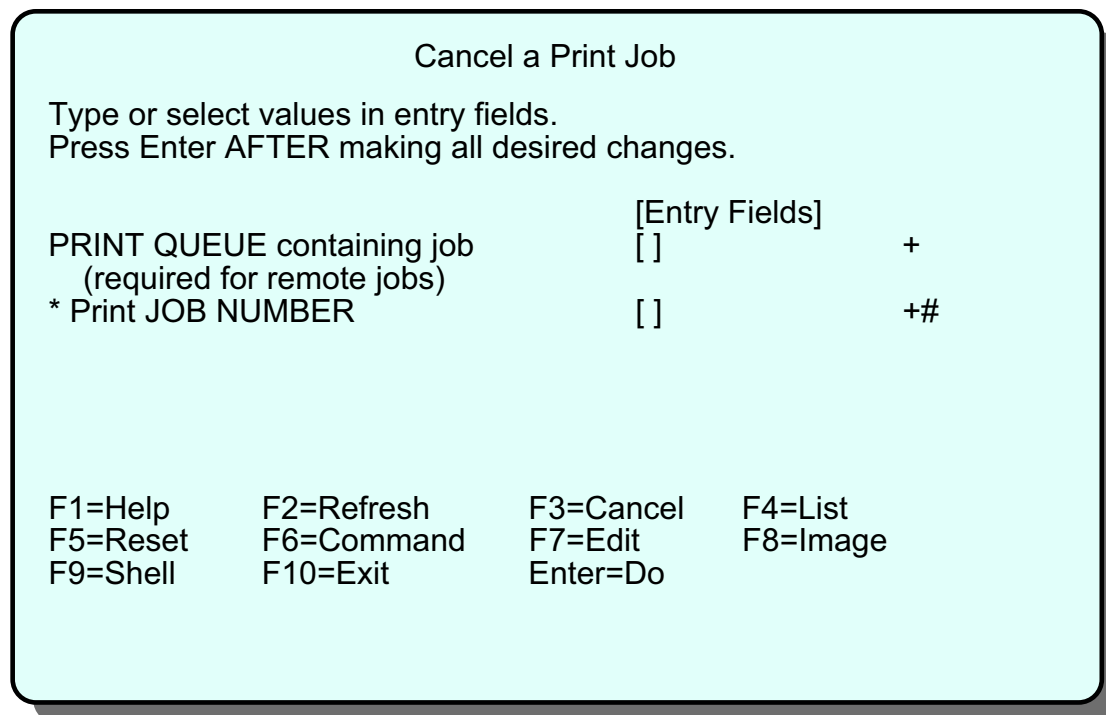


Figure 15-30. Cancelling Print Jobs

AU147.0

Notes:

The **qcan** command cancels either a particular job number or all jobs in a print queue.

Normal users can only cancel their own jobs, whereas root can cancel any job.

To cancel a job you can either use the **smit qcan** fastpath, or use one of the following commands:

	System V	BSD	AIX
To cancel a job:	cancel	lprm	qcan

To cancel Job Number 127 on whatever queue the job is on:

qcan -x 127 or **# cancel 127**

To cancel all jobs queued on printer lp0, enter:

qcan -X -Plp0 or **# cancel 1p0**

Job Priority Example

```
# qchk -L
Queue Dev      Status Job              Name      From      To
----- --      -
ps      lp0      DOWN
        Submitted
        Rnk      Pri      Blks      Cp      PP %
1/07/99 09:39:25 1      15      2      1
        /etc/qconfig

        QUEUED 570 /etc/motd    root      root
1/07/99 09:40:15 2      15      1      1
        /etc/motd

# qpri -#570 -a 25
# qchk -L
Queue Dev      Status Job              Name      From      To
----- --      -
ps      lp0      DOWN
        Submitted
        Rnk      Pri      Blks      Cp      PP %
1/07/99 09:40:15 1      25      1      1
        /etc/motd

        QUEUED 569 /etc/qconfig  root      root
1/07/99 09:39:25 2      15      2      1
        /etc/qconfig
```

Figure 15-31. Job Priority Example

AU147.0

Notes:

The discipline line in the `/etc/qconfig` file determines the order in which the printer serves the requests in the queue. In the queue stanza the discipline field can either be set to **fcfs** (first-come-first-serve) or **sjn** (shortest-job-next). If there is no discipline in the queue stanza, requests are serviced in **fcfs** order.

Each print job also has a priority that can be changed via SMIT or with the **qpri** command. Print jobs with higher-priority numbers are handled before requests with lower-priority numbers. Only a user who has root authority or who belongs to the `printq` group can change the priority of a local print request. Note: you can only set priorities on local print jobs. Remote print jobs are not supported.

The **qpri -R** command can also be used to set job priority.

The example shows that when print jobs are submitted they receive the default priority of 15. The example shows how the **qpri** command can be used to change the priority of job number 570 to 25. Use the **qchk -L** command to show the new job priorities.

SMIT can also be used to change print job priorities (**smit qpri**).

Holding a Job in a Queue

```
# qchk
Queue Dev      Status      Job      Files      User      PP %      Blks      Cp Rnk
ps      lp0      DEV_BUSY
        QUEUED 1493      /etc/qconfig      root              1      1      1

# qhld -#1493
# qchk
Queue Dev      Status      Job      Files      User      PP %      Blks      Cp Rnk
ps      lp0      DEV_BUSY
        HELD   1493      /etc/qconfig      root              1      1      1

# qhld -r -#1493
# qchk
Queue Dev      Status      Job      Files      User      PP %      Blks      Cp Rnk
ps      lp0      DEV_BUSY
        QUEUED 1493      /etc/qconfig      root              1      1      1
```

Figure 15-32. Holding a Job in a Queue

AU147.0

Notes:

The **qhld** command is used to put a temporary hold on a job that is waiting in the queue. The **qhld** command is also the command that is used to release job back in the queue.

The graphics provides a example of using the **qhld** command to hold and then release job # 1493.

This task can also be accomplished through smit - **smit qhld**.

Moving a Job between Queues

```
# qchk -A
```

<u>Queue</u>	<u>Dev</u>	<u>Status</u>	<u>Job</u>	<u>Files</u>	<u>User</u>	<u>PP %</u>	<u>Blks</u>	<u>Cp</u>	<u>Rnk</u>
asc	lp0	DOWN							
		QUEUE	11	/etc/qconfig	root	2		1	1
ps	lp0	READY							

```
# qmov -mps -#11
```

```
# qchk -A
```

<u>Queue</u>	<u>Dev</u>	<u>Status</u>	<u>Job</u>	<u>Files</u>	<u>User</u>	<u>PP %</u>	<u>Blks</u>	<u>Cp</u>	<u>Rnk</u>
asc	lp0	DOWN							
ps	lp0	RUNNING	11	/etc/qconfig	root	2		1	1

Figure 15-33. Moving a Job between Queues

AU147.0

Notes:

You can move jobs between queues in AIX. The command **qmov** is used. The **-m** option specifies what queue to move the job to and the **-#** option specifies the job number.

This can be done through smit using **smit qmov**.

Printing-related Directories to Monitor

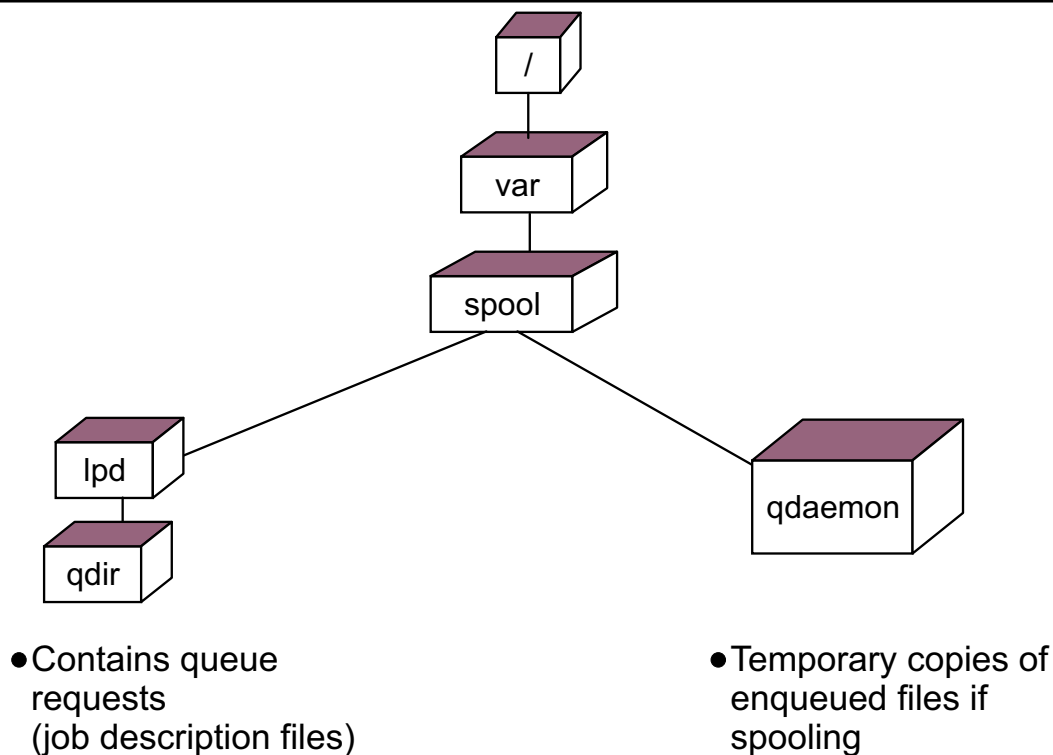


Figure 15-34. Printing-related Directories to Monitor

AU147.0

Notes:

The above directories will fill up very quickly if the spooling mechanism encounters a problem. For example, if the queue goes down, or if there are many users submitting jobs there may not be enough room to handle the requests.

Remember, when print jobs are submitted to spooling rather than just queuing, a copy of that file is created and stored in `/var/spool/qdaemon` directory until that job has printed. At that time, the temporary file is removed. If the queue or multiple queues quit working, jobs don't get through the system. This could cause a full condition in this directory structure.

Printing Problem Checklist

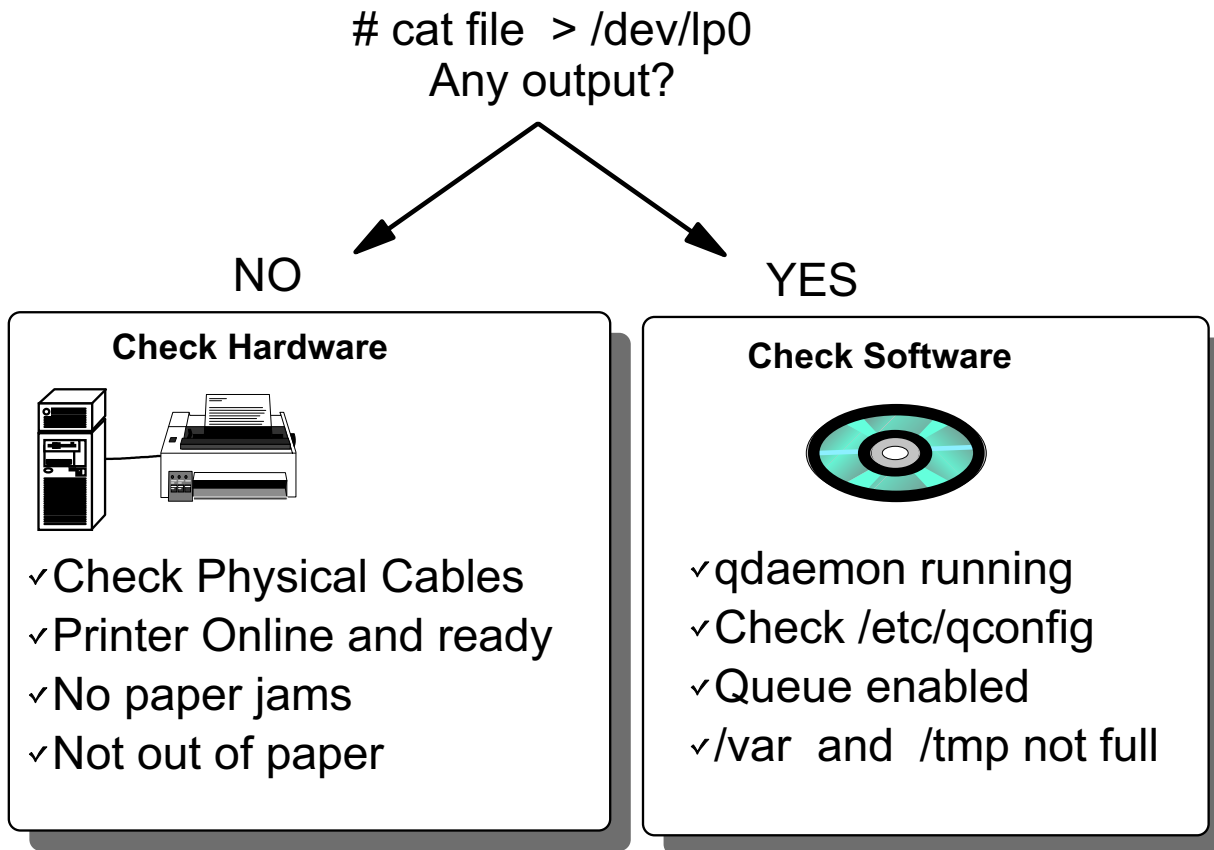


Figure 15-35. Printing Problem Checklist

AU147.0

Notes:

If you experience problems trying to print, start by checking the simple things first.

The easiest test to perform is to **cat** a file and redirect standard output to the printer device file. This by-passes the queuing system and helps to narrow the problem.

After redirecting a file to the print device, if it does not print, the problem is usually hardware-related. Check to make sure the cables are attached securely. Make sure the printer is ready to print (online). Make sure there is paper in the printer and there are no paper jams.

If something does print out using **cat** but not print out when using **lp**, **qprt**, or **lpr**, the problem is most likely software-related.

Check to make sure the **qdaemon** is running. If not, start it.

```
# lssrc -s qdaemon
# startsrc -s qdaemon
```

Look at the contents of **/etc/qconfig** to make sure it is not corrupt.

Ensure the queue are enabled. If not, enable them.

```
# lpstator# qprt -A  
# enable queuename
```

Check to make **/tmp** and **/var** are not full.

```
# df
```

Exercise: Printers and Queues



Figure 15-36. Exercise: Printers and Queues

AU147.0

Notes:

This exercise will give you an opportunity to work with the AIX queuing system. If your classroom does not have locally attached printers, your instructor will need to supply you with local modification for this lab.

This exercise can be found in your Exercise Guide.

Checkpoint (1 of 2)

1. True or false? One of the advantages of queues is that each user can have a different default queue set up for them.
2. True or false? The `/etc/qconfig` file is read by the **backend** program to determine what the queue discipline is.
3. True or false? All printer software is automatically installed when you install the base operating system.
4. What is the difference between these two commands?
`qprt -Pasc file1`
`qprt -c -Pasc file1`

Notes:

Checkpoint (2 of 2)

5. What methods can be used to find out what the system default queue is?

6. Can any user bring the print queues down? Name a few people who can.

7. True or false? Once the queue is down, no more jobs can be submitted to the printer.

8. Can users hold all their print jobs in a specific queue? If so, how?

Notes:

Unit Summary

- Queues can be added for local or remote printing
- Queue characteristics can be changed either through SMIT or via high-level commands
- Queues can be brought up and down by the system administrator
- The following tasks were considered:
 - Submit and cancel print jobs
 - List the jobs in a queue
 - Hold and release jobs in a queue
 - Move a job from one queue to another
 - Change priorities of a print job

Figure 15-39. Unit Summary

AU147.0

Notes:

Unit 16. Networking Overview

What This Unit Is About

This unit gives an overview of TCP/IP and networking concepts.

What You Should Be Able to Do

After completing this unit, you should be able to:

- Define the basic TCP/IP terminology
- Configure TCP/IP for an Ethernet or Token-Ring connection
- Use some of the standard TCP/IP facilities to:
 - Log in to another system
 - Transfer files
 - Run commands

How You Will Check Your Progress

Accountability:

- Checkpoint question
- Exercise

References

Online *System Management Guide: Communications and Networks*

Objectives

After completing this unit, students should be able to:

- Define the basic TCP/IP terminology
- Configure TCP/IP for an Ethernet or Token-Ring connection
- Use some of the standard TCP/IP facilities to:
 - Log in to another system
 - Transfer files
 - Run commands

Figure 16-1. Objectives

AU147.0

Notes:

What Is TCP/IP?

- **T**ransmission **C**ontrol **P**rotocol/**I**nternet **P**rotocol
- Software to enable different systems to exchange data over a variety of types of network
- The way in which systems are connected and how data is passed between them is transparent to the user
- TCP/IP is vendor-independent. Development is overseen by the "Internet Architecture Board"

Figure 16-2. What Is TCP/IP?

AU147.0

Notes:

TCP/IP is a networking architecture which defines a mechanism for cooperating computers connected by some sort of network to exchange data. TCP/IP software has been implemented across many platforms from mainframes to personal computers, although it is most commonly associated with UNIX environments.

TCP/IP is a set of protocols which define various aspects of how two computers on a network may communicate with each other. A protocol is a set of rules which describe the mechanisms and data structures involved. Using these definitions, vendors can write software to implement the protocols for particular systems.

TCP/IP stands for "Transmission Control Protocol/Internet Protocol". These are the names of the two most important protocols. There are many others. Where possible, the protocols are defined independently of any operating system, network hardware or machine architecture. In order to implement TCP/IP on a system, interface software must be written to allow the protocols to use the available communications hardware.

This means that heterogeneous environments can be created where machines from different manufacturers can be connected together, and different types of networks can be interconnected.

TCP/IP is the result of work commissioned in 1968 by DARPA the US Department of Defense, Advanced Research Projects Agency. Many other research and vendor organizations have contributed to the development of TCP/IP.

DARPA implemented a point-to-point network using leased lines called ARPANET using protocols which eventually evolved into TCP/IP. In 1980, ARPANET became the backbone to the Internet which links many US government, military, research, educational and commercial organizations.

The main popularity of TCP/IP has been due to its association with UNIX systems. In particular DARPA funded University of California, Berkeley to integrate TCP/IP into their versions of UNIX (BSD 4.2, 4.3)

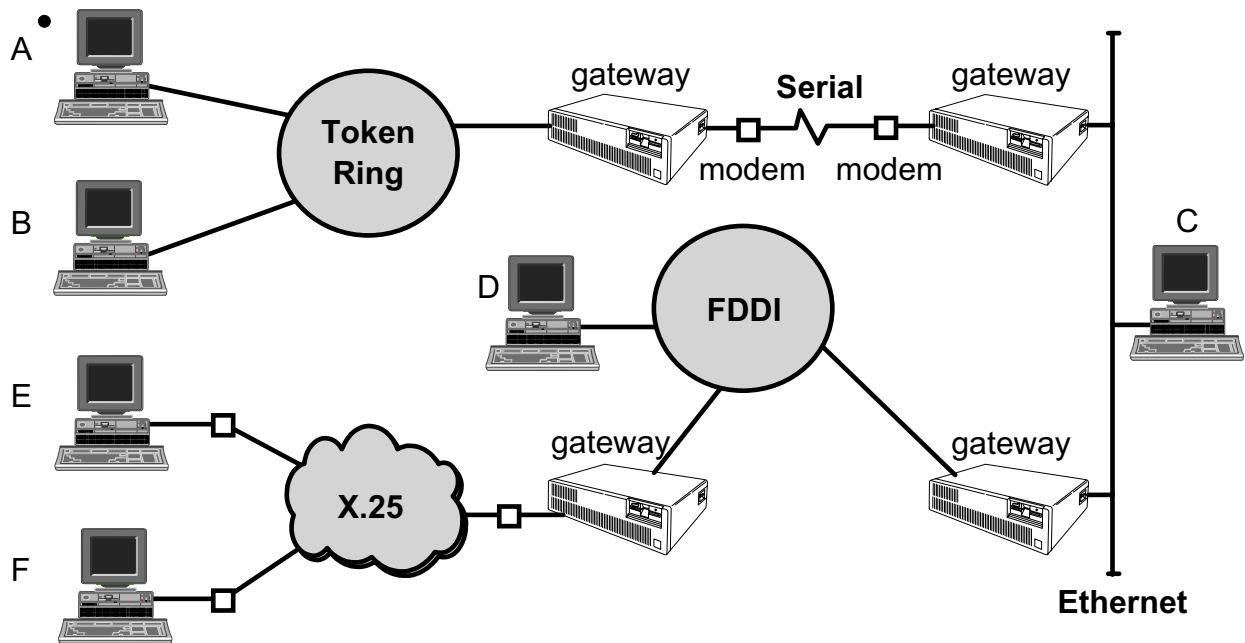
Most TCP/IP development is initiated by an organization called the Internet Architecture Board (IAB) which oversees development of the Internet network and the TCP/IP software it uses. Other TCP/IP development is performed by vendor organizations who write protocols which may become Internet standards.

The IAB distributes documents called Request For Comments (RFC) which describe TCP/IP protocols and other relevant information. RFCs are the primary source of TCP/IP and Internet information and are freely available in the Internet.

There are two subseries of the RFCs of interest. The STD (standards) describe all of the official TCP/IP standard protocols. The FYI (for your information) documents provide useful information about TCP/IP, the Internet and running a TCP/IP network.

An Internet

- A TCP/IP network is often called an **internet**



- Individual machines are called **hosts**. Hosts may vary in size and functionality but have equal standing as far as TCP/IP is concerned
- Hosts which link two or more physical network segments to each other are called **gateways**

Figure 16-3. An Internet

AU147.0

Notes:

TCP/IP works with many different types of networks from slow-speed serial type connections to fast local area networks like Token-Ring or Ethernet or even faster networks like FDDI (using fiber optics) and the SP Switch.

- **Local Area Network (LAN)**

Networks in a close geographical area. Often high-speeds over short distances

Computers must connect directly to network media (via a transceiver or tap)

Token-Ring (4 or 16 megabits per second)

Ethernet (10 or 100 megabits per second)

FDDI (100 megabits per second)

SP Switch (150 megabytes per second)

- **Wide Area Network (WAN)**

Can be far apart

Computers often connected indirectly (modems, public telephone networks)
Generally slower speeds than LANs

An Internet is a term given to a number of TCP/IP networks connected together. An Internet can be a combination of similar networks or heterogeneous networks. In an Internet, data can be transferred transparently from one host to another without the sending host needing to know the route taken or the type or number of connections involved.

There are a number of public Internets worldwide, the largest of which is called "The Internet" (or the "connected Internet"). The Internet consists of millions of connected systems.

A host is any computer attached to the network which has a TCP/IP address. This includes machines of any size or functionality. For example, an X-Terminal is a host as far as TCP/IP is concerned. Each host is given a unique name (for users) and address (for software) so that it can be uniquely identified in the interconnection of networks.

A host which has multiple network adapter cards is called a gateway. This can either be a dedicated machine to provide the function of routing data between networks or can be a machine providing applications as well (often called a multihomed host). This is not often recommended because of the extra load that the routing function will add.

Names and Addresses

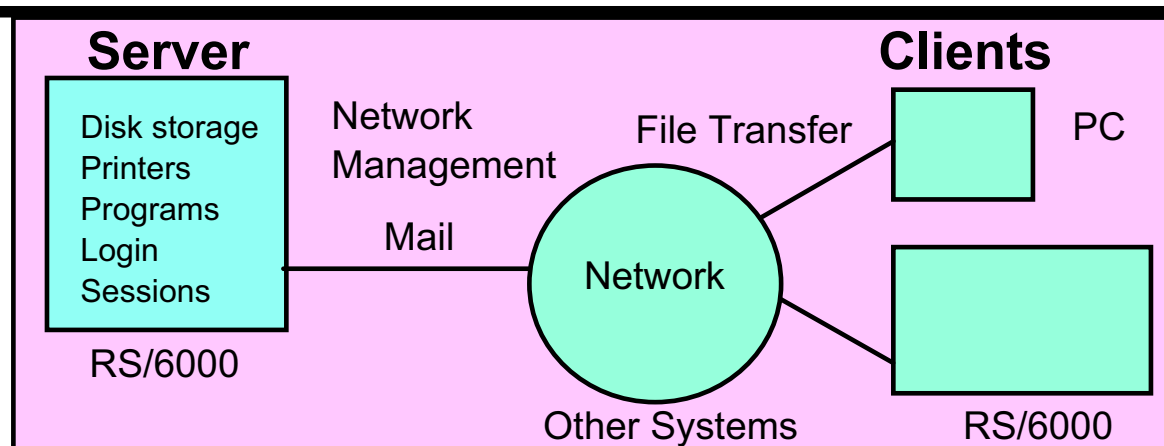
- Each system in a TCP/IP network is given a name
For example: **sys3**
- When contacting another system you only need to know the name
For example: \$ **telnet sys3**
- When contacting another user you need to know the system and user name
For example: \$ **mail fred@sys3**
- Each system will have one or more TCP/IP addresses
For example: **128.27.9.13**
If you know the address but not the name, you can use some TCP/IP facilities with the address

Figure 16-4. Names and Addresses

AU147.0

Notes:

TCP/IP Network Facilities



Standard TCP/IP facilities include: Mail, File Transfer, Remote Login, Remote Execution, Remote Printing

A number of AIX Applications use TCP/IP:

- Network File System (NFS)
- Network Information Services (NIS)
- Domain Name Service (DNS)
- Dynamic Host Configuration Protocol (DHCP)
- Network Computing System (NCS)
- Distributed Computing Environment (DCE)
- X Windows and AIXWindows
- TME10 Netview for AIX

Figure 16-5. TCP/IP Network Facilities

AU147.0

Notes:

There are many applications that require or can take advantage of TCP/IP. The ones listed are available from IBM either as standard or as licensed program products. There are many third-party applications (for example databases) that can also use TCP/IP for distributed work.

- Network File System (NFS) allows access of remote files as if they were local.
- Network Information Services (NIS) provides a distributed database of system information.
- Domain Name Service (DNS) provides server support to keep track of host names and addresses in the network.
- Dynamic Host Configuration Protocol (DHCP) allows a host to dynamically obtain a TCP/IP address from a server in the network.
- Network Computing System (NCS) allows applications to be written to run procedures on other systems in a network.

- Distributed Computing Environment (DCE) provides a rich set of facilities for developing and running distributed applications. It is based on NCS with many other services including Security Service, Directory Service, Time Service and management tools.
- X Windows / AIXWindows provide a distributed graphical user interface.
- Tivoli Management Environment Netview for AIX provides a sophisticated set of management tools for TCP/IP networks. It uses the AIXWindows environment to provide a graphical user interface for the network manager and uses Simple Network Management Protocol (SNMP) to pass management information around the network.
- AIX V4 also includes the Xstation Manager®, which allows attachment of an IBM Xstation (requires AIXWindows).

Information Needed to Configure TCP/IP

- Addresses
 - Each adapter is given a unique **TCP/IP** address and often a **subnet mask**. These will usually be assigned by your network administrator
- Names
 - Each machine has a unique **hostname**
 - Each machine must have access to a table of name to address translations. This can be either:
 - /etc/hosts** file
 - Domain Name Server** - You must know:
 - Domain Name
 - Address of the Name Server
- Routes

In order to communicate with systems in other networks, you may need to find the **address of the default gateway**

Figure 16-6. Information Needed to Configure TCP/IP

AU147.0

Notes:

Each system in a TCP/IP network must have a unique **TCP/IP address** and **hostname**. Your network administrator will centrally manage tables of names and addresses and will assign these for your system. On some networks a **subnet mask** is also required which is used to determine which network your machine belongs to for routing purposes.

AIX V4.3 provides support for both IPV4 and IPV6 addresses. The IPV6 addresses are 128 bits in length, represented as eight 16-bit fields separated by colons. A technique called "tunneling" is used to allow systems with IPV4 and IPV6 to coexist. SMIT and the Web-based System Manager provide separate support for IPV6.

Each host in a network is allocated a name which the users find easier to remember. However, the TCP/IP protocols can only use TCP/IP addresses when sending data. Therefore a portion of TCP/IP is responsible for translating the symbolic host names into TCP/IP addresses. This process is called name resolution.

There are two separate mechanisms defined for name resolution:

Flat Network

- Each host in the network has a record of the name and address of every other host it will communicate with. This is in a text file called **/etc/hosts**. This is quick but becomes difficult to administer if there are a large number of hosts.

Domain Network (Domain Name System)

- Hosts are grouped together into domains which form a hierarchy (similar to the file directory structure). One (or more) hosts in a domain (called nameservers) have a record of the name and address of all hosts. Client hosts request name to address translations from a nameserver. Use the **/etc/resolv.conf** file.

There may be more than one name server in a domain network for backup, but only one will have the primary copy of the database on its local disk. Clients only need to know the **domain name** and the **address of the nameservers**. This mechanism is much more suitable for large networks because administration is centralized on a few machines.

If your network is just part of a larger network then you will need to know about the gateway machines which link your network to others. Most network designs will only have one gateway called the default gateway. You will need to know the **address of the default gateway** to allow your system to communicate with other systems through the gateway.

Configuring TCP/IP

smit mktcpip

Minimum Configuration & Startup

To Delete existing configuration data, please use Further Configuration menus

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

	[Entry Fields]	
*HOSTNAME	[sys1]	
*Internet ADDRESS (dotted decimal)	[192.9.200.1]	
Network MASK (dotted decimal)	[255.255.255.0]	
*Network INTERFACE	en0	
NAMESERVER		
Internet ADDRESS (dotted decimal)	[]	
DOMAIN Name	[]	
Default GATEWAY		
Address (dotted decimal or symbolic name)	[192.9.200.192]	
Cost	[0]	#
Do Active Dead Gateway Detection?	no	+
Your CABLE Type	N/A	+
START TCP/IP Daemons Now	no	+
F1=Help	F2=Refresh	F3=Cancel
F4=List	F5=Reset	F6=Command
F7>Edit	F8=Image	F9=Shell
F10=Exit	Enter=Do	

Figure 16-7. Configuring TCP/IP

AU147.0

Notes:

The Web-based System Manager can also be used to configure the network. Use the fastpath **wsm network**.

When the TCP/IP software is installed, a new menu called TCP/IP (fastpath: tcpip) is added to SMIT in Communications Applications and Services and other places.

The SMIT Minimum Configuration & Startup option (fastpath: mktcpip) or the **mktcpip** command can be used to quickly configure TCP/IP on the RS/6000. This initializes TCP/IP (for client services) but further customization will be required.

The minimum information that is required to start TCP/IP is the hostname, and one interface and its Internet address. If subnetting is used then the subnet mask should be specified. A static route can be specified to a default gateway. Also the domain name and name server for a client in a domain network can be specified.

You can decide whether to start the TCP/IP daemons when initializing TCP/IP through this option.

The “Further Configuration” menu (fastpath: **configtcp**) contains a series of menus for customizing TCP/IP options. For example, hostnames, routes, interfaces, name resolution, server and client services.

Dead gateway detection is a mechanism for hosts to detect a dead gateway or a gateway that is not responding.

The cost is used with dead gateway detection to prioritize routes.

Flat Name Resolution

more /etc/hosts

```
#The format of this file is:
#Internet Address Hostname #Comments
#Items are separated by any number of blanks or tabs. A"#
#indicates the beginning of a comment; characters up to the end of the
#line are not interpreted by routines which search this file. Blank lines are
#allowed in this file.

#Internet Address      Hostname      #Comments
127.0.0.1              loopback     localhost
192.9.200.1            sys1         timeserver
192.9.200.2            sys2
192.9.200.3            sys3
192.9.200.4            sys4
```

Figure 16-8. Flat Name Resolution

AU147.0

Notes:

Host names and their Internet addresses are mapped by entries in the **/etc/hosts** file. In a flat network it should have entries for the local machine name, local host and all other hosts known to the system. Typically, **/etc/hosts** is kept consistent between all machines.

In a domain network the **/etc/hosts** file can be empty. Although, usually some hosts can be added for access to other hosts if the name server is down.

On AIX you can use SMIT or the vi command to add entries to **/etc/hosts**. The Hosts Table menu (fastpath: hosts) contains options to list/add/change/delete hosts in the **/etc/hosts** file. To get to this menu from the TCP/IP menu select Further Configuration then Name Resolution. The **/etc/hosts** file can be edited directly if desired.

Identifying the Hostname

hostname Command

```
# hostname  
sys3
```

host Command

```
# host sys3  
sys3 is 9.25.35.10, Aliases: sys3.washington.ibm.com  
  
#host 9.25.35.10  
sys3 is 9.25.35.10, Aliases: sys3.washington.ibm.com
```

Figure 16-9. Identifying the Hostname

AU147.0

Notes:

Two useful commands are **hostname** and **host**.

hostname - used to determine the name of the machine

host - when used with the hostname, determines the IP address. When used with the IP address, determines the hostname.

Basic TCP/IP User Functions

The following commands work with any TCP/IP system (not just UNIX/AIX):

- Test connectivity **ping**
- File Transfer **ftp**
- Remote Login **telnet**
- Remote Execution **rexec**

Figure 16-10. Basic TCP/IP User Functions

AU147.0

Notes:

The **ARPA** commands for File Transfer, Remote Login and Remote Execution are **ftp**, **telnet**, **rexec** respectively.

These commands can be used between any TCP/IP system that supports the appropriate protocols, not just UNIX/AIX systems.

In order to ensure security across the network, these commands will always require a user name and password to be supplied when you establish a connection.

ping - Tests connectivity with another system

\$ ping sys2

```

PING sys2: (192.9.200.2): 56 data bytes
64 bytes from 192.9.200.2: icmp_seq=0 ttl=255 time=15 ms
64 bytes from 192.9.200.2: icmp_seq=1 ttl=255 time=3 ms
64 bytes from 192.9.200.2: icmp_seq=2 ttl=255 time=2 ms
64 bytes from 192.9.200.2: icmp_seq=3 ttl=255 time=2 ms
64 bytes from 192.9.200.2: icmp_seq=4 ttl=255 time=2 ms
^C

```

```

----sys2 PING Statistics----

```

```

5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 2/4/15 ms

```

```

$

```

rexec - Run a command on another system

- Cannot run interactive commands
- Cannot run commands that run full screen

```

sys1 rexec sys2 uname -x
Name (sales2:tom): tom
Password: tom's password
AIX sys2 526332 2 3 000003F41C00

```

```

sys1$ _

```

The **rexec** command executes a command on another host. The command format is:

rexec host command

If the command contains metacharacters for the remote system, they must be enclosed in quotes.

The **ftp** command is used to transfer files from one system to another. It is normally an interactive environment and it provides a number of commands for transferring files. It can also be used for batch operation.

ftp will require you to specify a user and password to establish a connection to the remote system. This userid and password can be stored in a file in your home directory called **.netrc**. You can also specify automatic login procedures in this file.

Typical tasks that can be carried out are:

- List/Transfer/Delete Local/Remote Files
- Change Current Local/Remote Directory
- Create/Remove Directories

For example:

```
sys1$ ftp sys2
connected to sys2
220 sys2 FTP Server ready.
Name (sys2:smith): user1
331 Password required for user1
Password (sys2:user1): user1's password
230 User user1 logged in
ftp> binary
200 Type set to I
ftp> put file1 /tmp/f1
200 PORT Command successful.
150 Opening data connection for /tmp/f1 (192.9.200.1,1016)
226 Transfer Complete.
308310 bytes sent in 3.58 seconds (85.71 Kbytes/s)
ftp> quit
221 Goodbye.
sys1$ _
```

There are many **ftp** subcommands. To obtain a list use **?** or **help**. To get help on an individual subcommand use **? subcommand** or **help subcommand**. For example:

```
ftp> help open
open  connect to remote ftp
ftp>
```

To find out what commands are supported on the remote host use **rhelp** or **remotehelp**.

telnet - Remote Login (telnet)

```
sys1$ telnet sys2
Trying . . .
Connected to sys2.
Escape character is '^]'.

AIX telnet (sys2)
```

```
login: tom
password: tom's password
sys2$ ^]
telnet> ?
```

The **telnet** command implements the client end of the TELNET protocol for remote login.

If you are running **telnet** from one AIX/UNIX system to another, your terminal type will be passed correctly. Otherwise, you will have to set the TERM variable after you login.

The Escape character (usually ^]) allows you to escape into a telnet menu from which you can execute local commands or manage connections.

You can use the ^] key to escape to a **telnet** menu from which you can enter subcommands. Use **?** or **help** to list the available subcommands.

You can open or close connections and manipulate connection settings.

Use **z** to execute a shell on the local host.

Exercise: Networking

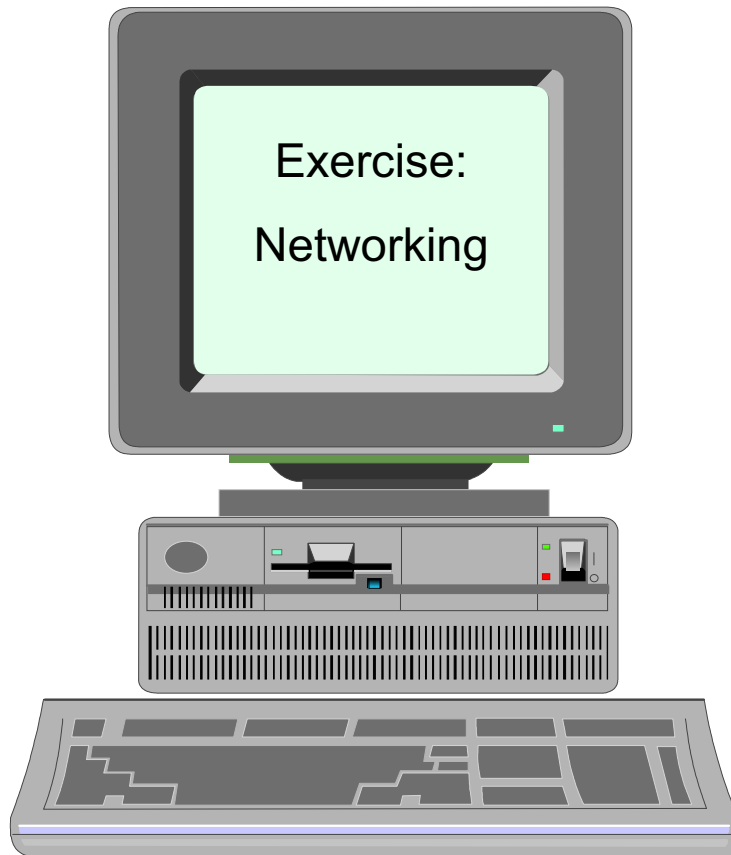


Figure 16-11. Exercise: Networking

AU147.0

Notes:

This lab will give you an opportunity to configure an RS/6000 on a TCP/IP network. This will give you practical application of the concepts presented in this unit.

This exercise can be found in your Exercise Guide.

Checkpoint

1. What are the following commands used for?

ftp _____

rexec _____

telnet _____

2. What is the difference (if any) between a **host** and a **gateway** .

3. True or false? Each machine in a TCP/IP network must have a unique hostname and TCP/IP address.

4. Which file holds the name and the TCP/IP address of each host in a flat network?

Notes:

Unit Summary

- TCP/IP is a networking architecture which defines a set of rules. These rules describe how computers can communicate with one another over a network.
- A flat TCP/IP network can be configured through SMIT by supplying the following information: addresses, subnet mask and hostnames
- There are many useful utilities which are provided by TCP/IP, such as **telnet** to login to another system, **ftp** to transfer files and **rexec** to execute a command on a remote system
- Use the **ping** command to check for connectivity to remote hosts

Notes:

Appendix A. Command Summary

Startup, Logoff and Shutdown

<Ctrl>d (exit)	log off the system (or the current shell).
shutdown	shuts down the system by disabling all processes. If in single-user mode, may want to use -F option for fast shutdown. -r option will reboot system. Requires user to be root or member of shutdown group.

Directories

mkdir	make directory
cd	change directory. Default is \$HOME directory.
rmdir	remove a directory (beware of files starting with ".")
rm	remove file; -r option removes directory and all files and subdirectories recursively.
pwd	print working directory: shows name of current directory
ls	list files <ul style="list-style-type: none">-a (all)-l (long)-d (directory information)-r (reverse alphabetic)-t (time changed)-C (multi column format)-R (recursively)-F (places / after each directory name & * after each exec file)

Files - Basic

cat	list files contents (concatenate). Can open a new file with redirection, for example, cat > newfile. Use <Ctrl>d to end input.
chmod	change permission mode for files or directories. <ul style="list-style-type: none">•chmod =+- files or directories•(r,w,x = permissions and u, g, o, a = who)•can use + or - to grant or revoke specific permissions.•can also use numerics, 4 = read, 2 = write, 1 = execute.•can sum them, first is user, next is group, last is other.

- For example, "chmod 746 file1" is user = rwx, group = r, other = rw.

chown	change owner of a files, for example, chown owner file
chgrp	change group of files
cp	copy file
mv	move or rename file
pg	list files content by screen (page) <ul style="list-style-type: none">•h (help) q (quit)•<cr> (next pg) f (skip 1 page),•l (next line) d (next 1/2 page)•\$ (last page) p (previous file),•n (next file) . (redisplay current page)•/string (find string forward) ?string (find string backward)•-# (move backward # pages) +# (move forward # pages)
.	Current Directory
..	Parent Directory
rm	remove (delete) files (-r option removes directory and all files and subdirectories)
head	print first several lines of a file
tail	print last several lines of a file
wc	report the number of lines (-l), words (-w), characters (-c) in files. No options gives lines, words, and characters.
su	switch user
id	displays your user id environment, user name and id, group names and ids.
tty	displays the device that is currently active. Very useful for Xwindows where there are several pts devices that can be created. It's nice to know which one you have active. who am i will do the same.

Files - Advanced

awk	programmable text editor / report write
banner	display banner (can redirect to another terminal 'nn' with '> /dev/ttynn')
cal	calendar (cal month year)
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"> -name fl (file names matching fl criteria) -user ul (files owned by user ul) -size +n (or -n) (files larger (or smaller) than n blocks) -mtime +x (-x) (files modified more (less) than x days ago) -perm num (files whose access permissions match num) -exec (execute a command with results of find command) -ok (execute a cmd interact'ly with results of find command) -o (logical or) -print (display results. Usually included) <p>find syntax: find path expression action</p> <ul style="list-style-type: none"> •For example, find / -name "*.txt" -print •or find / -name "*.txt" -exec li -l {} \; <p>(executes li -l where names found are substituted for {}) ; indicates end of command to be executed and \ removes usual interpretation as command continuation character)</p>
grep	<p>search for pattern, for example, grep pattern files. Pattern can include regular expressions.</p> <ul style="list-style-type: none"> -c (count lines with matches, but don't list) -l (list files with matches, but don't list) -n (list line numbers with lines) -v (find files without pattern) <p>expression metacharacters</p> <ul style="list-style-type: none"> •[] matches any one character inside. •with a - in [] will match a range of characters. •^ matches BOL when ^ begins the pattern. •\$ matches EOL when \$ ends the pattern. •. matches any single character. (same as ? 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	<p>sort and merge files</p> <ul style="list-style-type: none"> -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: <ul style="list-style-type: none">•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" will pass 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 & 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 RISC System/6000 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
migratelp	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
umount	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

=	set a variable (for example, d="day" sets the value of d to "day"). Can also set the variable to the results of a command by the ` character, for example, now=`date` sets the value of now to the current result of the date command.
HOME	home directory
PATH	path to be checked
SHELL	shell to be used
TERM	terminal being used
PS1	primary prompt characters, usually \$ or #
PS2	secondary prompt characters, usually >
\$?	return code of the last command executed
set	displays current local variable settings
export	exports variable so that they are inherited by child processes
env	displays inherited variables

echo echo a message (for example, "echo HI" or "echo \$d").
 Can turn off carriage returns with \c at the end of the message.
 Can print a blank line with \n at the end of the message.

Tapes and Diskettes

dd reads a file in, converts the data (if required), and copies the file out

fdformat formats diskettes or read/write optical media disks

flcopy copies information to and from diskettes

format AIX command to format a diskette

backup backs up individual files.
 -i reads file names from standard input
 -v list files as backed up;
 For example, "backup -iv -f/dev/rmto file1, file2"
 -u backup file system at specified level;
 For example, "backup -level -u filesystem"

Can pipe list of files to be backed up into command. For example,
 "find . -print | backup -ivf/dev/rmt0 " where you are in directory to be
 backed up.

mksysb creates an installable image of the root volume group

restore restores commands from backup
 -x restores files created with "backup -i"
 -v list files as restore
 -T list files stored of tape or diskette
 -r restores filesystem 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).
 -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"
 "file1"
 "file2"
 "<Ctrl-d>"

	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 backup 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 : 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)
uuto/uupick	send and retrieve files to public directoryon)
uux	execute on remote system (UNIX to UNIX execute)

System Administration

df	display filesystem usage
installp	install program
kill (pid)	kill batch process with id or (pid) (find using ps); kill -9 (PID) will absolutely kill process
mount	associate logical volume to a directory; for example, "mount device directory"
ps -ef	shows process status (ps -ef)
umount	disassociate filesystem from directory
smit	system management interface tool

Miscellaneous

banner	displays banner
date	displays current date and time

newgrp	change active groups
nice	assigns lower priority to following command (for example, "nice ps -f")
passwd	modifies current password
sleep n	sleep for n seconds
stty	show and or set terminal settings
touch	create a zero length files
xinit	initiate X-Windows
wall	sends message to all logged in users.
who	list users currently logged in ("who am i" identifies this user)
man,info	displays manual pages

System Files

/etc/group	list of groups
/etc/motd	message of the day, displayed at login.
/etc/passwd	list of users and signon information. Password shown as !. Can prevent password checking by editing to remove !.
/etc/profile	system wide user profile executed at login. Can override variables by resetting in the user's .profile file.
/etc/security	directory not accessible to normal users
/etc/security/environ	user environment settings
/etc/security/group	group attributes
/etc/security/limits	user limits
/etc/security/login.cfg	login settings
/etc/security/passwd	user passwords
/etc/security/user	user attributes, password restrictions

Shell Programming Summary

Variables

var=string	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 (), redirection (<, >, >>), and & symbols are not interpreted.
\$var	gives value of var in a compound
echo	displays value of var, for example, "echo \$var"
HOME	= home directory of user
MAIL	= mail file name
PS1	= primary prompt characters, usually "\$" or "#"
PS2	= secondary prompt characters, usually ">"
PATH	= search path
TERM	= terminal type being used
export	exports variables to the environment
env	displays environment variables settings
\${var:-string}	gives value of var in a command. If var is null, uses string instead.
\$1 \$2 \$3...	positional parameters for variable passed into the shell script
\$*	used for all arguments passed into shell script
\$#	number of arguments passed into shell script
\$0	name of shell script
\$\$	process id (pid)
\$?	last return code from a command

Commands

#	comment designator
&&	logical-and. Run command following && only if command preceding && succeeds (return code = 0).
	logical-or. Run command following only if command preceding fails (return code < > 0).
exit n	used to pass return code n from shell script. Passed as variable \$? to parent shell
expr	arithmetic expressions Syntax: "expr expression1 operator expression2" operators: + - * (multiply) / (divide) % (remainder)
for loop	for n (or: for variable in \$*); for example,: do

	command done
if-then-else	if test expression then command elif test expression then command else then command fi
read	read from standard input
shift	shifts arguments 1-9 one position to the left and decrements number of arguments
test	used for conditional test, has two formats. if test expression (for example, "if test \$# -eq 2") if [expression] (for example, "if [\$# -eq 2]") (spaces req'd) 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
:set wm=n	set wrap margin to n
:set showmode	sets display of "INPUT" when in input mode

Read, Write, Exit

:w	write buffer contents
:w file2	write buffer contents to file2
:w >> file2	write buffer contents to end of file2
:q	quit editing session
:q!	quit editing session and discard any changes
:r file2	read file2 contents into buffer following current cursor
:r! com	read results of shell command "com" following current cursor
:!	exit shell command (filter through command)
:wq or ZZ	write and quit edit session

Units of Measure

h, l	character left, character right
k or <Ctrl>p	move cursor to character above cursor

j or <Ctrl>n	move cursor to character below cursor
w, b	word right, word left
^, \$	beginning, end of current line
<CR> or +	beginning of next line
-	beginning of previous line
G	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.

0	move to first character in line
\$	move to last character in line
^	move to first nonblank character in line
fx	move right to character "x"
Fx	move left to character "x"
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 "n"
<Ctrl>g	display current line number

Search and Replace

/pattern	search forward for "pattern"
?pattern	search backward for "pattern"
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 char 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 w/next line

Appendix B. Sample Shell Scripts Used in Class

The information in this appendix has not been submitted to any formal IBM test and is distributed AS IS. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Shell scripts have been provided to support optional exercises in Exercises 5 and 6, and scripts that support concepts discussed during the class to document the system configuration and backup of VGDA information.

The concept of automatically documenting your system configuration was discussed in class. Creation of the following shell script lists customized devices, vital product data, and attributes for all the devices on your system.

```
for DEV in $(lsdev -CF name)
do
echo $(lsdev -C | $DEV -F "name location") >> /tmp/d.log
lsattr -EH | $DEV >> /tmp/d.log
done
lscfg -v >> /tmp/d.log
```

The following script will save logical volume maps for possible data recovery if a volume group descriptor area (VGDA) is lost:

```
# ! /bin/ksh
# save.map = a simple script to save logical volume maps
# for possible data recovery if a volume group descriptor
# area (VGDA) is lost
# usage: save.map VOLUME_GROUP_NAME
# maps are saved in /tmp/LOGICAL_VOLUME_NAME.map
if ( ( $# < 1 ) )
then
print "Usage: save.map VG_NAME"
exit 1
fi
VG=$1
lsvg -l $VG | tail +3 | cut -f1 -d" " | while read LV
do
lslv -m $LV > /tmp/$LV.map
done
```


The following shell script, **lab 6**, was used in the Optional Exercises section in the Printers Exercise, instruction 13:

```
#!/usr/bin/ksh
echo "Working, please wait .\c"
stopsrc -s qdaemon 2> /dev/null 1>/dev/null
echo ".\c"
echo "\n\n: \n" >> /etc/qconfig 2>/dev/null
echo ".\c"
```

The following shell script, **lockvi**, is first used in instruction 4 of the Managing File Systems Exercise:

```
while true # always perform loop unless see a break statement
do
filename='basename $1' # retrieve just the filename, not the directory
if [ -f /tmp/lock${filename} ]
then echo "Someone else is editing $1. Please wait in the queue."
sleep 2
continue # to top of while loop until lock is removed
else
trap "rm /tmp/lock${filename}" 1 2 3 15 # If they try to cut out
# early, clean up the lock
touch /tmp/lock${filename}
echo "now editing $1"
sleep 1
/usr/bin/vi $1
rm /tmp/lock${filename}
break # only when you're done can you break out of while loop
fi
done
```

The following shell script, **mkfile**, is used in instruction 13 of the Managing File Systems Exercise:

```
#!/usr/bin/ksh
# mkfile filesize
usage()
{
clear
echo " "
echo " "
echo " "
echo " "
echo "Usage: mkfile filesize"
echo "    filesize should be in multiples of 512 bytes"
echo " "
echo " "
echo " "
echo " "
exit
}
# Main...
if [ $# != 1 ]
then
usage
fi
filesize=$1
filename="$1"bytefile
integer mod='expr $filesize % 512'
integer div='expr $filesize / 512'
if [ $mod != 0 ]
then
usage
fi
integer i=0;
integer j='expr $div \* 128'
> $filename
echo " "
echo "Creating file \"$filename\". Please wait ... "
while true
do
echo "yes" >> $filename
i=i+1
if ' $i = $j
then
break
```

fi
done

Appendix C. MCA Appendix

The original RS/6000's used IBM's Microchannel Architecture (MCA). These machines are also referred to as Classical RS/6000's. System administration for MCA and PCI machines is basically the same. There are a few differences that mostly effect the startup procedures for the machine.

Key Positions

The classical machine has a physical key switch that has three positions - normal, service, and secure. The various positions effect the machine's boot sequence.

Secure - The machine will not boot while in this position. If the machine is up and running and the key is turned to the secure position, the reset button is disabled.

Normal - The machine will use the "normal" boot list during its start up procedure. The boot list can be seen with the command **bootlist -om normal**.

Service - The machine will use the "service" boot list during its start up procedure. The service boot list can be seen with the command **bootlist -om service**. This is used to boot from alternate bootable media like a CD or tape. If the machine boots from the hard drive with the key in the service position, the diagnostic controller will load instead of the normal AIX Operating system.

Setting/Changing Boot Lists

The boot list can be viewed using the bootlist command. The boot list can also be set using the **bootlist** command. Specify the mode and the devices to set the boot list. For example:

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

The boot list can also be viewed and changed using **diag**. From the command line, type in the command **diag**. This will present a menu. Select **Tasks Selection**. Then select **Display or Change Bootlist**.

Modes

The classical machines have two modes - normal and service.

Normal mode with MCA is the same as normal mode with PCI. In this mode, all programs and processes are running, terminals are enabled, volume groups are varied on, file systems are mounted, communication is started and the machine is ready for multi-user mode.

When using service mode, it will depend on what boot device is used for boot up as to what the machine will let you do. A service boot that uses the hard drive will load the diagnostics routines. If the key is in the service position, generally the machine is looking for a boot

device other than the hard drive. If it finds a bootable CD (like an installation CD) or a boot tape (like a mksysb tape), then it will boot from those devices. This will present the Maintenance and Installation menu.

You need to use the service boot from alternate media if the machine fails to boot from the hard drive.

The Startup Sequence and LEDs

The classical machines have a slightly different boot sequence. Classical machines can only run the AIX operating system.

Once the machine is powered on, two different hardware checks are performed - Built-in Self Test (BIST), and the Power-on Self Test (POST).

The BIST initializes the very basic components of the system like the processor, memory and system planar. While the BIST is running, the LEDs will display numbers in the range of 100-199.

When the BIST is done, the POST is started. The purpose of this phase is to identify the device to boot from (locate the Boot Logical Volume) and load the boot program from that device into memory. While the POST is running, the LEDs will show numbers in the range of 201-299. 299 is the actual loading of the kernel. When the POST is complete, it turns control over to the kernel to continue the boot process.

You may have notice that we skipped the LED number 200. 200 will display when a boot is attempted while the key is in the secure position. Turning the key to the normal or service position will allow the machine to continue into the POST phase.

Once the kernel is loaded, the boot process proceeds in the same fashion as the PCI machines. The LEDs that are displayed from this point on will be the same as with PCI-based machines.

Boot Problems

If your machine fails to boot, the LED display will indicate what the problem is. This chart provides some common problem that may be encountered.

Microchannel: Boot Problem Map

LED	Explanation	User Action
100 - 195	Built-in self-test encountered a hardware error	<ol style="list-style-type: none"> 1. Check boards and cables 2. Record SRN 101-xxx in Problem Summary Form
200	Key mode switch is in the secure position	Turn the key switch to the correct position (Normal, Service)
201	Checkstop occurred during system start	<ol style="list-style-type: none"> 1. If LED 299 passed, recreate the boot volume (bosboot) 2. If LED 299 has not passed, record SRN 101-201 in Problem Summary Form
201 (solid) 721 (solid) 221-229 (cycle) 223-229 (cycle) 225-229 (cycle) 233-235 (cycle)	Boot list in NVRAM is incorrect. Boot list device has no boot record, or the device is unavailable	<ol style="list-style-type: none"> 1. Check cables and external devices (Power, SCSI-addresses) 2. Boot in maintenance mode and check the boot list 3. Boot in maintenance mode and recreate the boot logical volume (bosboot)

xxx = Replace with your LED code

SRN = Service Request Number

A complete list of LEDs codes unique to the MCA machines are located at the end of this section.

Location Codes

Location codes on classical machines are different than those found on PCI machines but the concept is still the same. The location code shows a path from the system board to the device using a numerical address.

Below is some information concerning the location codes for classical machines.

Microchannel Location Codes

Device Name	Location Code
Processor	00-00
Motherboard	00-00
Microchannel bus	00-00
Diskette adapter	00-00-0D
Diskette drive	00-00-0D-00
Serial port 1	00-00-S1
Terminal	00-00-S1-00
Standard SCSI controller	00-00-0S
Disk drive	00-00-0S-0,0
CD-ROM drive	00-00-0S-2,0
SCSI controller, not on-board	00-01
Disk drive attached to external SCSI controller	00-01-00-3,0
Memory Card	00-0A
Another memory card	00-0H
SSA adapter	00-02
SSA physical disk drive	00-02-P
SSA logical disk drive	00-02-L

The general format of a microchannel location code is

AB-CD-EF-GH

1. AB

•00

Device is in the system unit or CPU drawer (larger systems)

•0n

Device is in an I/O expansion drawer

2. CD

•00

Device is integrated on the standard I/O board

•0n

n = Slot number on the standard I/O board:

1-8 for adapters

A-H for memory

1n: Adapter in slot n on an optional I/O bus

3. EF

- 00 for SCSI devices
- Parallel Port Adapter, Serial Adapter Port 1, Serial Adapter Port 2, Diskette Adapter, Keyboard Adapter, Mouse Adapter, Tablet Adapter, Ethernet Adapter, SCSI Adapter: for built-in adapters

4. GH or G,H

- GH = Port / Location on the adapter
- G,H = SCSI address of the device

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 rest 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 Checktop 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.
- h12.200 - 299
- 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
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 device 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 device 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 def device list.
- 234 Progress indicator. Attempting a Normal-mode system restart from the 9333 High-Performance Disk Drive Subsystem specified in the IPL ROM def
- 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 boot list.
- 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 boot list reached.
- 270 Progress indicator. Ethernet/FDX 10 Mbps MC adapter POST is running.
- 271 Progress indicator. Mouse and mouse port POST is 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.

Appendix D. AIX Control Book Creation

AIX Control Book Creation

List the licensed program products	lspp -L
List the defined devices	lsdev -C -H
List the disk drives on the system	lsdev -Cc disk
List the memory on the system	lsdev -Cc memory (MCA)
List the memory on the system	lsattr -El sys0 -a realmem (PCI)
	lsattr -El mem0
	lsattr -EHI sys0
List system resources	lscfg -v
List the VPD (Vital Product Data)	lscfg or smit screen capture F8
Document the tty setup	qchk -A
\Document the print queues	lspv
Document disk Physical Volumes (PV's)	lslv
Document Logical Volumes (LV's)	lsvg -l vgname
Document Volume Groups (long list)	lspv -l pvname
Document Physical Volumes (long list)	lsfs fsname
Document File Systems	/etc/filesystems
	df
Document disk allocation	mount
Document mounted file systems	lsps -a
Document paging space (70 - 30 rule)	/etc/swapspaces
Document paging space activation	/etc/passwd
Document users on the system	lsuser -a id home ALL
	/etc/security/user
Document users attributes	/etc/security/limits
Document users limits	/etc/security/environ
Document users environments	/etc/security/login.cfg
Document login settings (login herald)	/etc/group
Document valid group attributes	lsgroup ALL
	/etc/profile
Document system wide profile	/etc/environment
Document system wide environment	/var/spool/cron/crontabs/*
Document cron jobs	/usr/sbin/skulker
Document skulker changes if used	/etc/inittab
Document system startup file	/etc/hosts
Document the hostmanes	/etc/hosts.lpd
Document network printing	/etc/hosts/equiv
Document remote login host authority	

Directories to monitor

<code>/var/adm/sulog</code>	Switch user log file (ASCII file). Use cat , pg or more to view it and rm to clean it out.
<code>/etc/security/failedlogin</code>	Failed logins from users. Use the who command to view the information. Use " cat /dev/null > /etc/failedlogin " to empty it,
<code>/var/adm/wtmp</code>	All login accounting activity. Use the who command to view it use " cat /dev/null > /var/adm/wtmp " to empty it.
<code>/etc/utmp</code>	Who has logged in to the system. Use the who command to view it. Use " cat /dev/null > /etc/utmp " to empty it.
<code>/var/spool/lpd/qdir/*</code>	Left over queue requests
<code>/var/spool/qdaemon/*</code>	temp copy of spooled files
<code>/var/spool/*</code>	spooling directory
<code>smit.log</code>	smit log file of activity
<code>smit.script</code>	smit log of commands and scripts

Appendix E. Licensing Overview

What This Unit Is About

This unit is an introduction to the features of LUM and NCS. It explains the advantages and issues in using a licensing subsystem. The LUM and NCS daemons, commands and files needed to implement and administer LUM are described.

What You Should Be Able to Do

After completing this unit, you should be able to:

- Discuss the licensing options available with AIX V4.1, V4.2 and V4.3
- List the types of licenses available with LUM
- Explain how LUM manages licenses in the AIX Environment

References

	<i>AIX Version 4.2 Differences Guide</i>
	<i>LUM Documentation located in /USR/OPT/IFOR/DOC</i>
SC23-2665	<i>FOR/LS System Management Guide</i>
SC23-2666	<i>FOR/LS Tips and Techniques</i>

Objectives

- Discuss the licensing options available with AIX V4.1, V4.2 and V4.3
- List the types of licenses available with LUM
- Explain how LUM manages licenses in the AIX environment

Notes:

All these topics will be covered during this unit.

Note: There were no changes to licensing in the AIX 5.1 release.

LUM Policies and License Types

LUM Policies

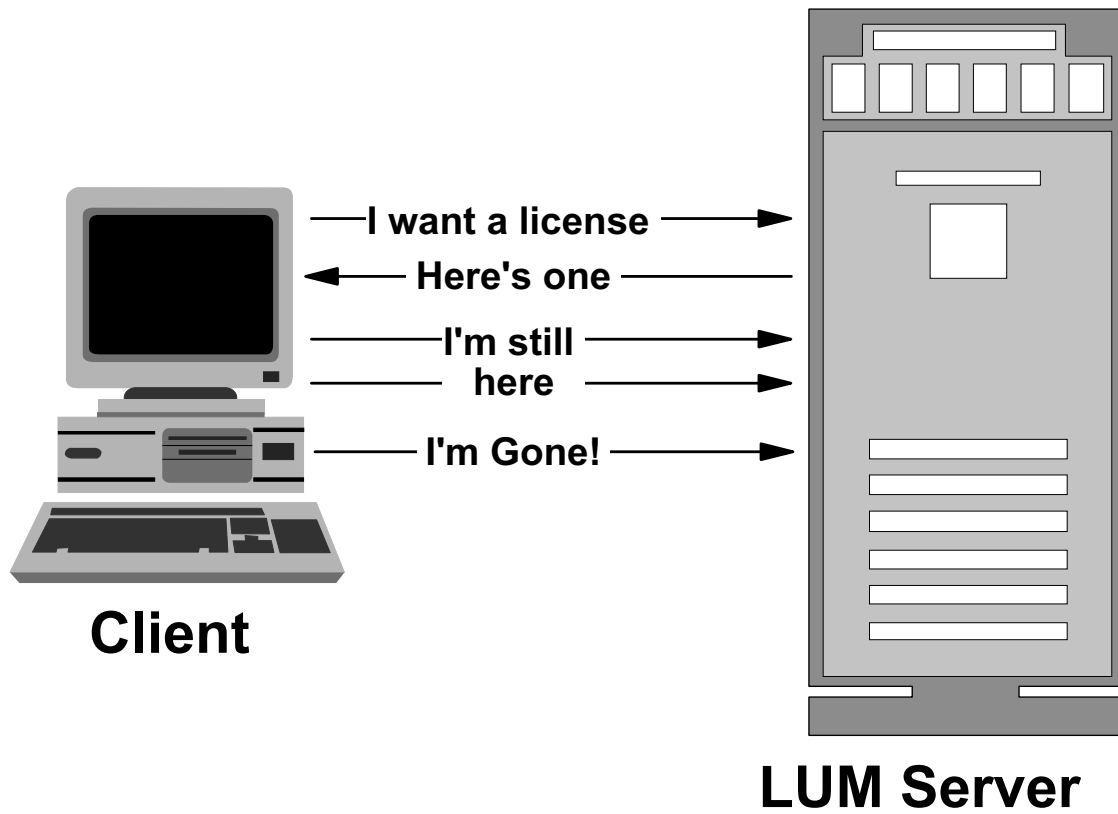


Figure E-2. LUM Policies

AU147.0

Notes:

Software developers implement their policies in their applications.

LUM incorporates very few policies within the license record. The advantage is that the license record is fairly small (fewer than 40 bytes) and does not incur much network overhead. Software vendors must consider the following common policies:

- What happens when a license request from an application is not granted?
- Where and how should errors be treated?
- How should disaster recovery (such as a server failure) be handled?

The most common policy for software developers to consider is what to do when an application requests a license and cannot get one. There may be several reasons for this:

- The license has not been entered into the system.
- A network failure has occurred and the license server cannot be reached.
- The LUM server daemon has failed.
- All available licenses are being used by other users.

License Request Failure

- What should the application do if it cannot successfully obtain a license?
- **Hardstop** - The application does not start
- **Softstop** - The application starts without a license

Figure E-3. License Request Failure

AU147.0

Notes:

What should the application do if it cannot successfully obtain a license?

- **Hardstop:** - application fails
- **Softstop:** - application continues without a license

Hardstop policy:

- Allows CD-ROM distribution mechanism for multiple software products.
- An example is the "AIX Software Development Solutions" CD-ROM distributed by the IBM Toronto Lab. This CD-ROM contains all the XL compilers, CSet++, CMVC, Workbench and DB2. The customer can install the products on their system, but cannot execute any of the applications without first installing the application's license key.

Softstop policy:

- Benefit of doubt given to the application user

Check Period Failure (Heartbeat)

- What should the application do if the heartbeat fails?
- "Heartbeat" hardstop - The application stops
- "Heartbeat" softstop - The application continues running

Figure E-4. Check Period Failure (Heartbeat)

AU147.0

Notes:

What happens when an application already has an LUM license and it fails to contact the server for the "heartbeat", or check period?

- **"Heartbeat" Hardstop** - the application terminates (gracefully).
- **"Heartbeat" Softstop** - the application continues.

Softstop Policy:

- Benefit of the doubt given to application user.
- Unlicensed use has a lower priority than customer satisfaction.

AIX Version 4 BOS - "Heartbeat" Softstop

Check Period Interval

- Specified by the application
- Passed to the LUM server daemon at application start-up
- What time period should be used for the check period?
 - Long interval periods minimize network traffic
 - Short interval periods speed up license downtime recovery
 - Recommended: 5-10 minutes
 - Default: 15 minutes

Figure E-5. Check Period Interval

AU147.0

Notes:

The check period is used to maintain communication with the server. If communication is "broken", the server retrieves licenses after the check period expires.

The check period interval is specified by the application and can be hardcoded or configurable.

At application startup time, the application communicates with the LUM server daemon to define which heartbeat time-out period to use.

What time period should be used for the check period?

- Short intervals result in heavier network traffic.
- Long intervals force longer recovery periods.
- Recommended: 5-10 minutes.
- Default: 15 minutes.
- Configurable in one-minute intervals.

License Request Queuing

- What should the application do if all licenses are in use?
- Applications may choose to:
 - Wait for next license
 - Quit application
 - List machines currently using a license
 - Display position in the license queue
- Queuing should be used for interactive applications

Figure E-6. License Request Queuing

AU147.0

Notes:

If the application is requesting a license, should it allow the user to wait in a queue if all licenses are currently in use?

Applications could implement the following:

- Wait for next license
- Quit application
- Obtain a list of other machines with licenses
- Display position in the license queue

Queuing provides flexibility and should be used for interactive applications.

Types of LUM Licenses

- Nodelock
 - Tied to one machine
- Concurrent Nodelock
 - Fixed number of users on one machine
- Concurrent-use
 - "Floating" network license
- Use-once
 - "Floating" consumable network license
- Compound
 - "Bag" of licenses

Figure E-7. iFOR/LS and AIX4.1 BOS

AU147.0

Notes:

LUM can use a number of different types of licenses. Are listed below the most common types that we will see used with AIX.

Nodelock

Nodelock is a licensing mechanism that requires each workstation on which the licensed product operates to have its own unique key. When the client is invoked, the licensing mechanism runs in the local workstation's memory and thus checks the local workstation's target ID. Note that this implies nothing about where the software is installed. For example, in LUM, client that is nodelocked can be run through an NFS mount. When the client is invoked, it will run in the local workstation's memory and thus check the local workstation's planar ID. If the NFS server has a nodelocked key, it cannot be used to satisfy the license request made by the NFS client, because the local planar ID is checked.

The nodelock mechanism does not require that an LUM server be active in the network. Nodelocked keys are stored in a file in the local file system, `/var/ifor/nodelock`. If you had a previous nodelock file from iFOR/LS, located in `/usr/lib/netls/conf/nodelock`, it has been copied in the LUM directory during installation or migration. A common nodelock file can

also be shared Through a distributed file system mount provided that the appropriate nodelock keys are included for each workstation using this facility. **p.Note: If you are using a nodelock scheme for licenses**, then neither NCS nor the license server daemon is needed.

Concurrent Nodelock

This is a new type added by LUM to the types available within iFOR/LS. The nodelock type, we have seen before just tell if you have or do not have the right to execute the application on a machine. The concurrent nodelock type authorizes a fixed number of concurrent users for this software at the same time.

The concurrent-use licensing mechanism allows the software usage right to "float" in the network. One or more license servers contains keys that satisfy license requests from software applications.

In LUM, license servers have partitioned databases. The total number of licenses available is the sum of licenses found in all individual license servers. License servers are autonomous and do not communicate with each other. Therefore, when a license server fails, its licenses are temporarily unavailable in the network. Similar problems arise when a network is partitioned.

Note: LUM introduces the notion of per-server license and per-seat licenses. This apply to client-server applications. In the LUM scheme, the client starts, contact the LUM server for a license, obtains it and then contact the application server to start working.

In LUM, this is called per-seat license, since the LUM code resides in each client on the network. With the per-server license model, the application client directly contacts the application server, and it is the application server that will request a license on behalf of the client.

Use-Once

The use-once mechanism is a consumable form of license. Unlike concurrent-use, where a license is returned to the pool, a use-once license cannot be reused. In LUM, a use-once license for a particular product contains a count that is decremented each time a license is used. When this value reaches zero, the license cannot be used for any additional requests. Use-once licenses cannot be deleted from the license database.

AIX Version 4 BOS login does not utilize use-once licenses.

Compound

A compound license contains a password that allows the creation of more licenses. The primary advantage of a compound license is its flexibility for redistributing licenses at a later date.

LUM Licensing of AIX 4 BOS

iFOR/LS and AIX4.1 BOS

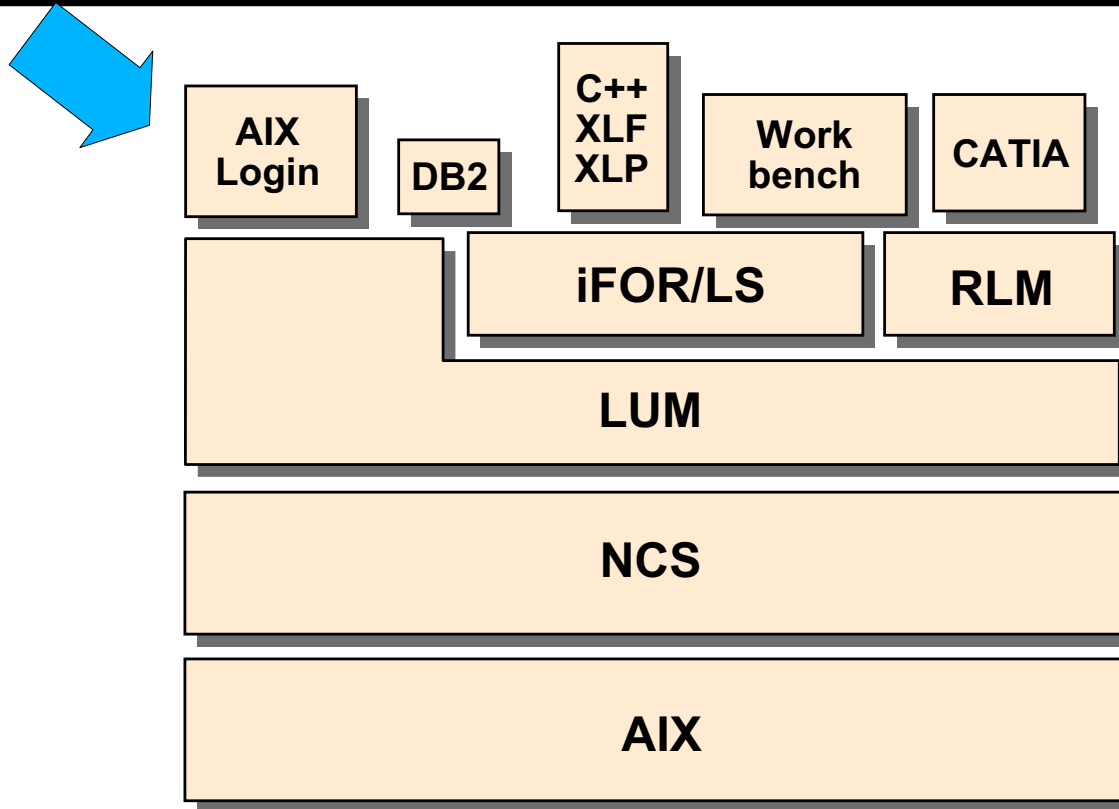


Figure E-8. iFOR/LS and AIX4.1 BOS

AU147.0

Notes:

AIX Version 4.3 is just one application among others that use the service of the license server. In Fact, AIX Version 4.3 is still calling the iFOR/LS commands to operate. But as we've already said, these commands are now implemented using LUM binaries and scripts.

chlicense Command

•chlicense -u licenses -f on/off

- u Changes the number of fixed licenses
- f Changes the status of the floating licensing
 - chlicense -f on
 - chlicense -f off
 - chlicense -u5 -f on

Figure E-9. chlicense Command

AU147.0

Notes:

chlicense -u Licenses -f on|off

- -u changes the number of fixed licenses
- -f changes the status of floating licensing

"-f on"

- Creates /etc/security/.netls file. No length, no contents, rw for root only
- Adds a line in /etc/inittab to start the *monitord* daemon
monitord:2:once:/usr/sbin/monitord >/dev/console 2>&1
- Automatically starts *monitord* daemon using "/usr/sbin/monitord"

"-f off"

- Removes /etc/security/.netls file if present
- Removes the /etc/inittab line for *monitord* daemon

For example, to set the fixed licenses to "5" and enable floating license mode

```
# chlicense -u5 -fon
```

monitord Activities

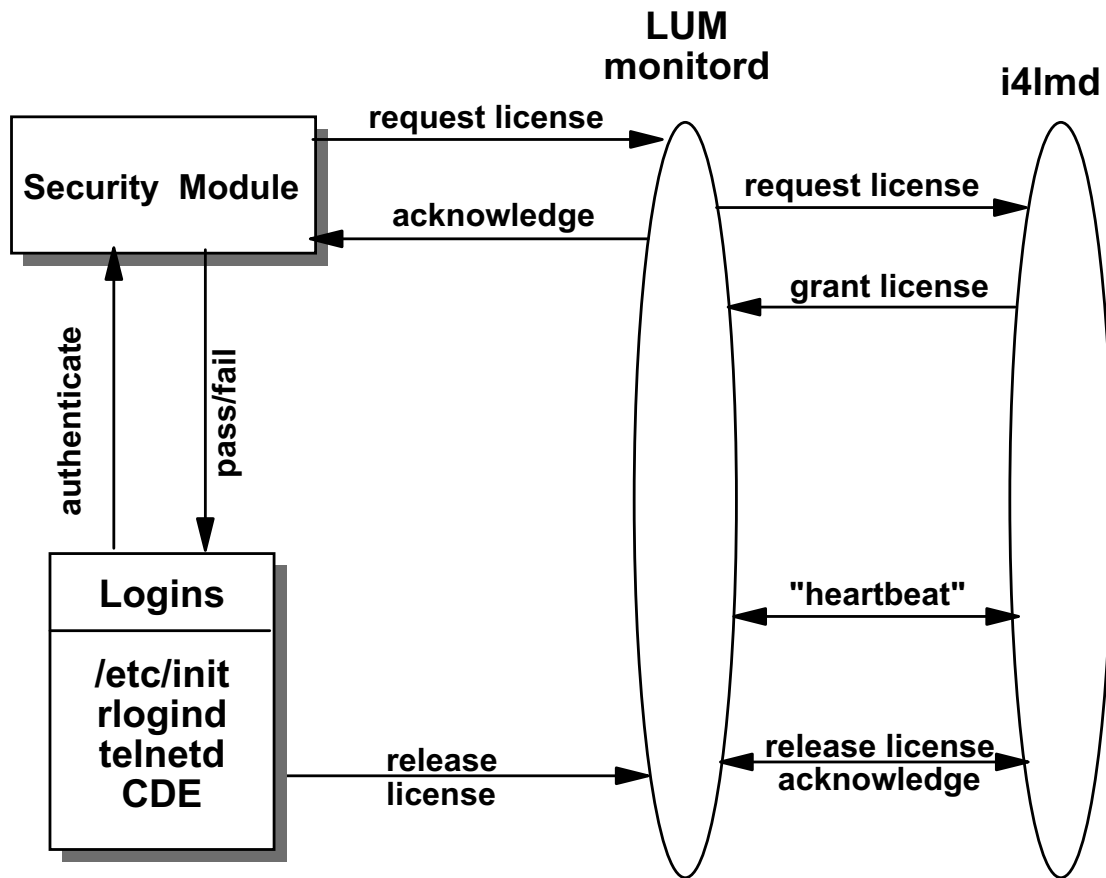


Figure E-10. monitord Activities

AU147.0

Notes:

monitord will be executed as a daemon at system startup through an entry in `/etc/inittab` (using a "boot" keyword) or by issuing the `chlicense -fon` command. It will run continuously, stopping only for a shutdown or single-user mode operation (maintenance mode). The **monitord** daemon will communicate with the NetLS security load module and the login subsystems via signals and a named pipe, set to appropriate privilege.

The monitord daemon will appear to the LUM environment as an enabled application. It will be linked with the LUM library and perform all the communications between the running BOS and the **netlsd**, performing the actual token request, release and heartbeat operations.

A named pipe will be used to make license-related requests to **monitord**. The information written into the pipe will consist of an integer representing a command and a process ID associated with the command.

This named pipe will be opened for reading by the daemon. It will also be opened for writing by the release routines and the authentication routine.

When a login process exits, it will need to notify **monitord** so that the license that was acquired can be returned to the license pool. This should be done by a release routine.

The release routine will write two things into the named pipe to tell **monitord** to release a license. The first thing that will be written is the integer value -1, which means "release license" to **monitord**. The next thing that will be written into the pipe is the process ID of the licensed login, so **monitord** will know which license to return.

There is a new option for **monitord**. The **-v**, followed by an AIX Version number, will configure the **monitord** daemon to request licenses for this version of AIX (On a 4.3 system, you should use `/usr/sbin/monitord -v 4.3`).

AIX 4.3 BOS Fixed License Scenario

AIX 4.3 BOS Floating License Server

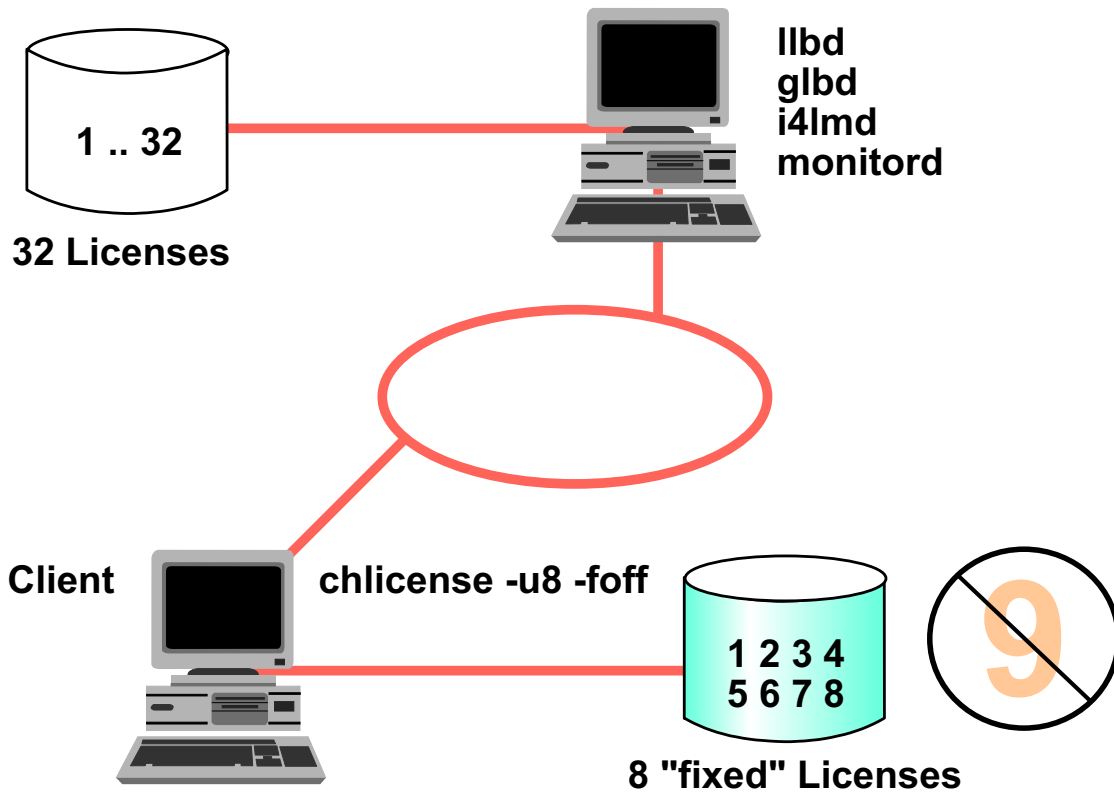


Figure E-11. AIX 4.3 BOS Fixed License Scenario

AU147.0

Notes:

In this example, both machines are on the same network. The client system is configured for eight fixed licenses and floating license mode is disabled (`chlicense -u8 -foff`). The LUM server system has been configured with 32 floating licenses to grant AIX 4.3 BOS license requests from other machines in its network.

When the ninth user attempts to log in to the client system, the user receives a message that all login sessions are in use and the login attempt fails.

AIX 4.3 BOS Floating License Scenario

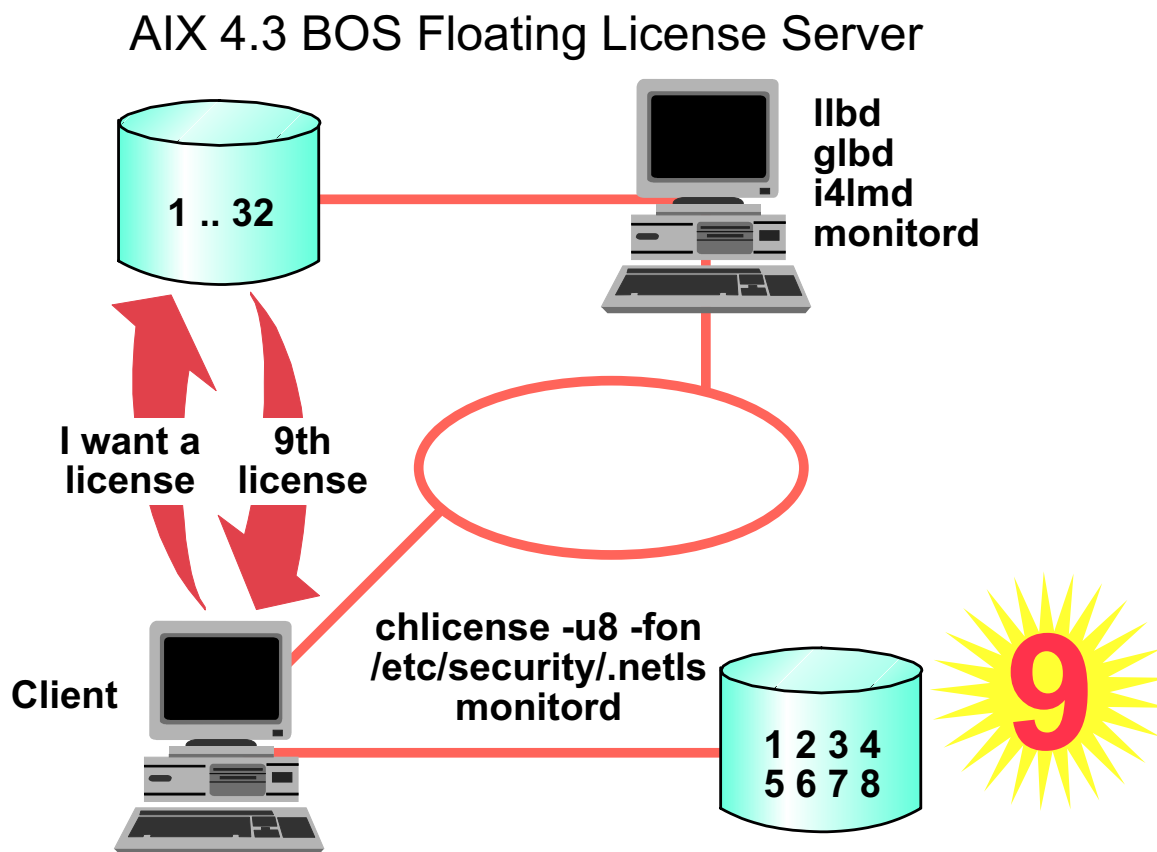


Figure E-12. AIX 4.3 BOS Floating License Scenario

AU147.0

Notes:

In this example, both machines are on the same network. The client system is configured for eight fixed licenses and floating license mode is enabled (`chlicense -u8 -fon`). The LUM server system has been configured with 32 floating licenses to grant AIX 4.3 BOS license requests from other machines in its network.

When the ninth user attempts to login to the client system, the client system checks the number of logins and recognizes it has exhausted its number of fixed licenses. Since floating license mode is enabled, it sends a broadcast out to the network requesting the location of an LUM server that has AIX 4.3 BOS licenses available. The LUM server system hears the request, checks the number of available AIX 4.1 BOS floating licenses, and grants the license key request to the client system.

Implementing LUM

Setting Up the Environment

- Installing the software
- Configuring the License Server
- Configuring the Daemons
- Getting the licenses
- Administering the licenses
- Getting statistics on the usage of the licenses

Figure E-13. Setting Up the Environment

AU147.0

Notes:

These are the steps you have to complete to set up an environment to in which operate a licence server.

LUM Packaging

- The basic filesets
 - ifor_ls.base.cli
 - ifor_ls.msg.en_US.base.cli
- The optional filesets
 - ifor_ls.base.gui
 - ifor_ls.compat.cli
 - ifor_ls.compat.gui

Notes:

When you install AIX Version 4.3, you automatically install NCS 1.5.1 and two filesets for LUM:

- ifor_ls.base — Includes the scripts, binary and documentation for LUM as well as the essential links to ensure compatibility with iFOR/LS and NetLS.
- ifor_ls.msg.\$LANG.base.cli — Includes the messages for the LUM tools.

You can add several other filesets to those defaults:

- ifor_ls.base.gui — Includes the graphical interfaces version of the command line tools found in ifor_ls.base.
- ifor_ls.compat.cli — Includes compatibility commands with LUM Version 1.0 that is included with AIX Version 4.2.
- ifor_ls.compat.gui — Includes the graphic interface version of the compatibility tools with LUM Version 1.0.

Configuring the License Server

- Use of the new LUM tool
 - i4cfg
 - i4cfg -script
- Use of NetLS compatibility tools
 - Run the /usr/lib/netls/conf/auto_config script
 - Run the /usr/lib/netls/conf/netls_config script

Figure E-15. Configuring the License Server

AU147.0

Notes:

As mentioned previously, you have two ways to configure the License server. Either use the new tools including with LUM or use the scripts you used to use with iFOR/LS (don't forget they are just interfaces to LUM technology).

The configuration tool in LUM is called **i4cfg** (located in /var/ifor).

If you have CDE started, just type **i4cfg** and this will start the graphic interface, this is shown in the next visual.

If you are on an ASCII terminal, or if you prefer to be prompted just type **i4cfg -script** to have the configuration script started. This is discussed later in this topic.

If you are used to iFOR/LS, you have now two scripts to configure the server :

- /usr/lib/net/conf/auto_config will configure the server, assuming it is a standalone license server.
- /usr/lib/netls/conf/netls_config will configure the server as a full network license server, you will be prompted with questions that allow you to decide whether or not to use direct binding, whether or not you use the default cell and so on.

Daemons

- NCS daemons
 - llbd (Local location broker)
 - glbd (Global location broker)
 - nrglbd (Non-replicable global location broker)
- LUM Daemons
 - i4lmd (License server subsystem)
 - i4llmd (Nodelocked license server subsystem)
 - i4gdb (Central registry subsystem)
 - i4glbcd (Global location broker data cleaner subsystem)
- NetLS daemon
 - Netlsd (link to i4lmd)

Figure E-16. Daemons

AU147.0

Notes:

Since we have already seen all these NCS daemons, the following is just for review.

Let's have a quick look at the LUM daemons and their functions.

- i4lmd (License Server Subsystem) is the daemon responsible for the network licensing, it will grant requests coming from the network
- i4llmd (Nodelocked License Server subsystem) is the daemon responsible for the nodelock license. If you are using concurrent nodelock you need a daemon to control the number of license used. if you are using simple nodelock, you do not need this daemon.
- i4gdb (Central Registry Subsystem) is the main daemon. It provides a mechanism for storing data pertaining to the licensing information. There must be one and only one central registry license server running per cell. This ensures that the data are accurate and complete. If you try to start a second occurrence of the i4gdb it will shut down immediately.

- i4glbcd (Global location broker data cleaner) is the daemon responsible for cleaning the database used by the global location broker. There may be some corrupted data in this database due to applications exiting without unlocking their records in the database. i4glbcd deals with those entries. Only one is allowed per cell.

netltd is here only for compatibility reasons, and as we've seen before, it is just a link to i4lmd.

Getting Licenses

- Obtain the target_id for the server
 - i4target
 - ls_targetid
- Request the license from the vendor with
 - target_id for the server
 - Software feature code
 - Desired amount and type of licenses
- Obtain these keys

You do not need key for AIX Version 4.3 bos login

Notes:

The target_id of the machine is a unique string that identifies your system. It may contain letters. So do not consider it as a number. The nodelock licenses are based on this identifier. Once more you can request the target_id of your machine either using the new LUM tool, i4target or by using the iFOR/LS compatibility tool ls_targetid which is a link to i4target.

Remember that if you want a nodelock license, give the identifier corresponding to the machine that will execute the application. If you want concurrent licenses give the identifier of the license server.

Once you have this identifier and the feature code of the product for which you want to get licenses, call the vendor and request the number of licenses you want.

The answer may come in a fax and you will have to retype the password for the product. So prefer an electronic distribution that would save some time.

You do not need a key for AIX Version 4.3 since AIX comes with a compound key that allows you to generate any number of keys.

Managing Licenses

- Using the LUM tools
 - i4blt
 - For nodelocked licenses
 - For floating licenses
 - i4nat
 - For nodelocked licenses (compatibility)
- Using the iFOR/LS compatibility tool
 - ls_admin
 - For floating licenses
 - /usr/lib/netls/conf/nodelock (file)
 - For nodelocked licenses

Figure E-18. Managing Licenses

AU147.0

Notes:

The tool in LUM that is used to manage licenses is called i4blt. It has a graphic interface and a command line interface. We will see in the next visual what its graphic interface looks like. Its command-line interfaces has many options, and covering each of these options is beyond the scope of this class.

i4blt can be used to administer floating licenses as well as nodelock licenses. It is the universal tool of LUM since, as we will see later, it is also used to get statistics.

The NAT tool is a tool that is included in AIX Version 4.3 only for compatibility with the previous version of LUM. This tool was used to administer nodelock licenses. We'll look at a visual to see its graphic interface.

As always, you can still use the iFOR/LS tools for compatibility reasons.

ls_admin uses a graphic interface and command line to administer floating licenses.

You can always enter the license in the /usr/lib/netls/conf/nodelock file using your favorite editor.

Testing the Environment - Getting Statistics

- Testing the environment
 - LUM
 - i4tv
 - iFOR/LS
 - ls_tv

- Getting statistics
 - LUM
 - i4blt
 - iFOR/LS
 - ls_rpt

Figure E-19. Testing the Environment - Getting Statistics

AU147.0

Notes:

LUM comes with a tool that allows you to check on the configuration of your server, providing everything works properly. This tool is the i4tv.

The tool that came with iFOR/LS and that was called ls_tv is still there but it is a link to i4tv.

The following is a sample of the output of i4tv:

```
i4tv Version 4.0 AIX -- LUM Test and Verification Tool
(c) Copyright 1995-1997, IBM Corporation, All Rights Reserved
US Government Users Restricted Rights - Use, duplication or disclosure
restricted by GSA ADP Schedule Contract with IBM Corp.
(c) Copyright 1991-1997, Gradient Technologies Inc., All Rights Reserved
(c) Copyright 1991,1992,1993, Hewlett-Packard Company, All Rights Reserved
```

```
Completed license transaction on node 44091c running LUM 4.0 AIX
Active License Servers:
ip:aix4xdev.itsc.austin.ibm.com (IBM/AIX) running LUM 4.0 AIX
Active Central Registry License Server:
ip:aix4xdev.itsc.austin.ibm.com (IBM/AIX) running LUM 4.0 AIX
```

Within LUM, the statistics are given by i4bt, either by its graphic interface or its command line. You can find how many licenses are being used at that moment or over a period of time.

Once more, if you were used to iFOR/LS, the tool that had the same role, ls_rpt, is still present on the disk. It is a real binary and not a link to a command from LUM.

Summary

- In AIX V4.1 use iFOR/LS for license management
- In AIX V4.2 use iFOR/LS for LUM for license management
- In AIX V4.3 use LUM for license management

Notes:

Appendix F. Serial Devices

What This Unit Is About

This unit introduces the concepts and configuration of serial devices.

What You Should Be Able to Do

After completing this unit, you should be able to:

- Describe a serial device to the system
- Set terminal characteristics
- Describe the purpose of the terminfo database
- Diagnose and solve common problems with terminals

How You Will Check Your Progress

Accountability:

- Checkpoint questions
- Optional Exercise, Appendix B, Student Exercise Guide

Objectives

After completing this unit, students should be able to:

- Define a serial device to the system
- Set terminal characteristics
- Describe the purpose of the terminfo database
- Diagnose and solve common problems with terminals

Notes:

Non-Self Configuring Devices

Devices not configured automatically at boot up by the configuration manager (cfgmgr):

- ASCII (dumb) terminals
- Printers
- Modems

Figure F-2. Non-Self Configuring Devices

AU147.0

Notes:

During the boot-up of a RS/6000 running AIX, the **cfgmgr** command is run to bring certain devices up and available on the system.

Only devices which have a defined industry standard that describes the way in which they can identify themselves to the system are configured by **cfgmgr**. For example, Micro Channel Architecture (MCA) adapters and the SCSI adapter for CD-ROM disks, tape drives, etc., will be made available.

Some devices do not have the mechanism for identifying themselves. These non-self-configuring devices include ASCII terminals and printers. These devices must be manually defined to the operating system.

Adding a Terminal

Questions to be answered before adding TTYs:

- Server Configuration
 - TTY interface
 - Adapter
 - Port number
- ASCII Terminal Configuration
 - Line characteristics
 - Terminal type
 - Keyboard attributes

Notes:

To add a terminal to the system, you must add a TTY logical device using **Add a TTY** on the TTY menu in SMIT or the **mkdev** high-level command. You can use the SMIT fastpath **maktty** or **mktty** to access this menu.

When adding a TTY, you must know the port where the terminal is plugged into the system, the terminal type (for the TERM variable) and the line characteristics for the port.

Enable/Disable

- Enable Login Attribute:
 - login=disable available for dial-out line
 - login=enable login prompt on terminal
 - login=delay user must press key first
 - login=share bi-directional port
- Use SMIT or **chdev** for permanent change

Figure F-4. Enable/Disable

AU147.0

Notes:

Appropriate values for the login attribute are:

- | | |
|----------------|---|
| disable | The port is still defined, but it will only be available as a dial-out port for an asynchronous connection to another machine |
| enable | The port is enabled for login, a getty process will run on the port when not in use |
| delay | The port is enabled for login, but the login prompt is not displayed until the user presses a key |
| share | The port can be used in either direction upon demand. |

Port Attributes

Various attributes play an important role during communication between the computer and the serial device.

These include:

- **bps/Baud Rate**
The speed of the line in bits per second.
- **The number of stop bits**
A signal to a receiving mechanism to wait for the next signal.
- **bits per character**
The number of bits per character to be transmitted.
- **Parity**
A simple error detection mechanism.

Notes:

Baud Rate:

The speed of an asynchronous communications line is usually expressed in bits per second (bps). Sometimes the term Baud Rate is used to mean the same thing although the Baud Rate actually means the number of possible voltage changes on the line per second.

Stop bit:

During communication, the voltage on one of the lines (the receive/transmit line) is normally set to high. When a system starts to send a byte, the voltage is set to low for 1.5 clock pulses. This is called a start bit. Similarly, at the end of the transmission of the byte, (i.e., after the last bit), the voltage is set high for a further clock pulse. This is called a stop bit.

Using two stop bits on low-speed lines or poor quality lines will improve communications.

Bits per character:

Serial communications standards allow for the transmission of different lengths of characters, or words. When communications software asks you to select word length, it is

asking whether you want to send seven-bit characters or eight-bit characters. Other lengths such as 5 or 6 can be used, but this is rare.

Parity:

Parity is a method of detecting transmission errors. If enabled, a parity bit is appended to each character transmitted. Types of parity checking are:

- | | |
|--------------|---|
| EVEN | If there are an odd number of ones in the binary representation of the character sent, the parity bit is set to one so that an EVEN number of ones is always transmitted. |
| ODD | The parity bit ensures that the number of ones transmitted is always odd. |
| MARK | The parity bit is always set to 1. |
| SPACE | The parity bit is always set to 0. |
| NONE | No parity. |

SMIT TTY Menu

smit tty

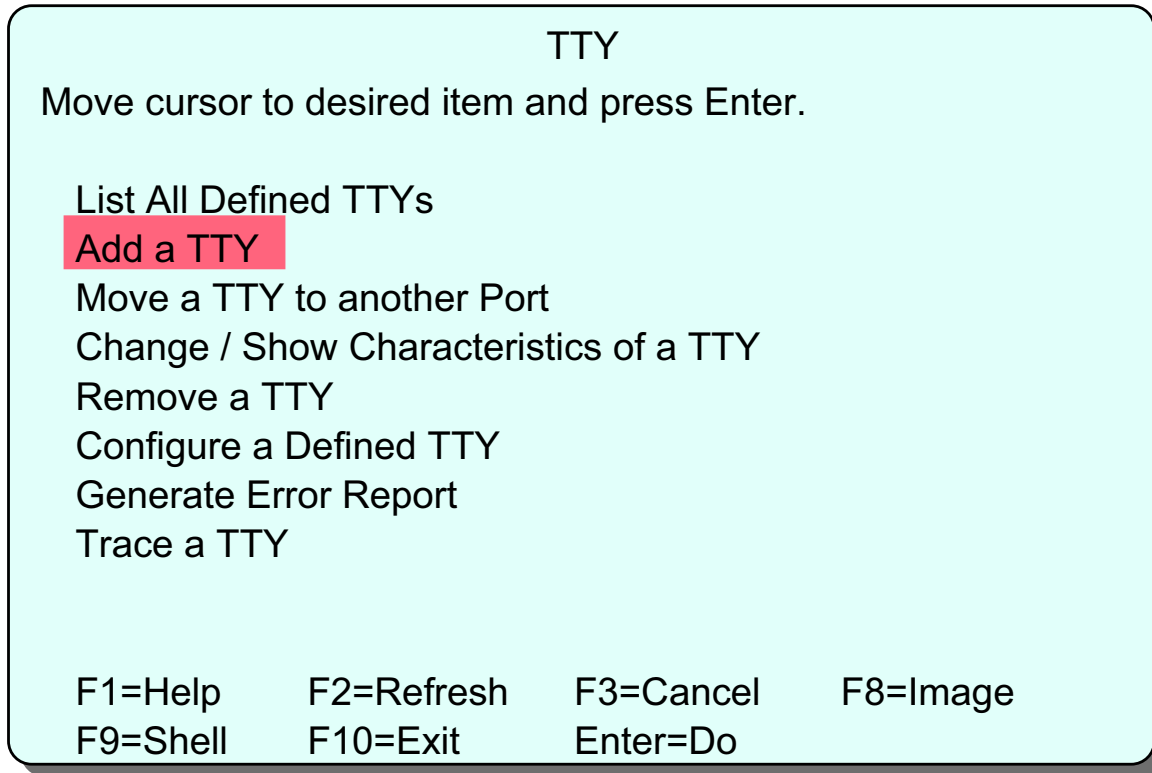


Figure F-6. SMIT TTY Menu

AU147.0

Notes:

The SMIT TTY menu is used to manage the configuration of asynchronous terminals and other TTY devices in the system. These are typically TTY devices attached directly to either RS232 or RS422 communication adapters. TTY devices attached to network terminal servers or serial printers are not generally configured using this method for performance reasons.

TTY devices can be listed, added to the system, made unavailable/available, removed and have their characteristics changed using these menus.

Attachment

TTY Type

Move cursor to desired item and press Enter.

tty	rs232	Asynchronous Terminal
tty	rs422	Asynchronous Terminal

Parent Adapter

Move cursor to desired item and press Enter.

sa0	Available	00-00-S1 Standard I/O Serial Port 1
sa1	Available	00-00-S2 Standard I/O Serial port 2
sa2	Available	00-03-11 16-Port RAN EIA-232 for 128-Port adapter
sa3	Available	00-03-12 16-Port RAN EIA-232 for 128-Port adapter
sa4	Available	00-03-13 16-Port RAN EIA-232 for 128 Port adapter

Figure F-7. Attachment

AU147.0

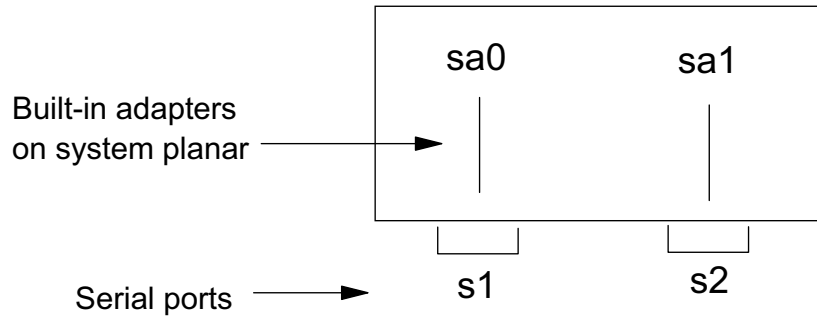
Notes:

When you select **Add a TTY** from the TTY menu you will be presented with two pop-ups to select the TTY type and adapter.

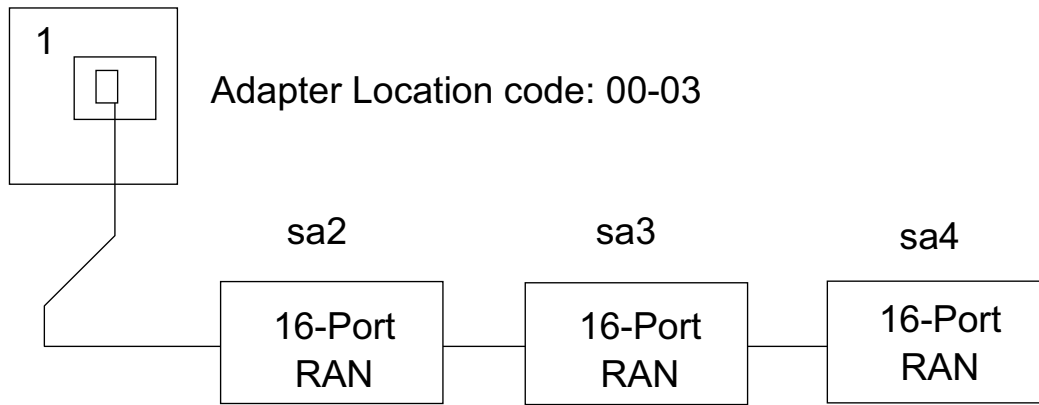
TTYs can either be connected to an RS232 or RS422 adapter.

Once a type has been selected, you will be presented with a list of installed adapters that support that method of attachment.

For the built-in serial connections the nomenclature looks like this:



For the 128-port adapter the nomenclature looks like this:



Add a TTY

```
# smit mktty
```

Add a TTY

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

<pre>[TOP] TTY type TTY interface Description Parent Adapter * PORT number Enable LOGON BAUD rate PARITY BITS per character Number of STOP BITS TIME before advancing to next port setting TERMINAL type FLOW CONTROL to be used [MORE ...29]</pre>	<pre>[Entry Fields] tty rs232 Asynchronous Terminal sa2 [] + disable + [9600] + [none] + [8] + [1] + [0] +# [dumb] [xon] +</pre>
---	--

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

Figure F-8. Add a TTY

AU147.0

Notes:

There is only one mandatory field on this screen and that is the PORT number. The F4 key will provide a list of possible port numbers. For the first built-in serial port it is s1, for the second it is s2. On a 16-port RAN, the choices are 0-15. Select the one to which the terminal is connected. The combination of the appropriate RAN selected on the Parent Adapter selector screen and the port number shown here provides the system with the correct location code.

You must supply the port number to uniquely locate the device. The value required depends upon the adapter specified. For example:

Built-in serial port S1	s1
Built-in serial port S2	s2
8-Port Adapter	0-7
16-Port Adapter	0-15
Each 16-PORTRAN	0-15

The 'Enable LOGIN' attribute will be set to 'disable' by default. If you are adding a terminal that should have a login prompt, you should change this to 'enable'.

The asynchronous line characteristics must be specified: baud rate, parity, bits per character, stop bits. In a national language environment you must use 8 bits with no parity (the default). Set the speed appropriately for the terminal device or modem you are using, up to 38400.

The 'TERMINAL type' attribute is used to assign the TERM environment variable when a user logs in on the device. There is no list available for this entry. The easiest way to find out the required values is to refer to the terminfo database, which will be discussed shortly.

terminfo

- Database of terminal capabilities
- Required by full screen programs
TERM variable
`/usr/share/lib/terminfo/?/$TERM`
- IBM, DEC and Wyse terminals supported
- Sample files for many other terminal types
`/usr/share/lib/terminfo/*.ti`

Figure F-9. terminfo

AU147.0

Notes:

When a function key is pressed on the keyboard a sequence of characters (escape sequence) is sent to the system. When the system needs to display a special terminal feature such as reverse video or clear screen, the system must send a sequence of characters to the terminal.

Because there are a large number of ASCII terminals on the market which all offer a variety of functions, there is no standard for how these functions are implemented. The solution has been to build a terminal-independent set of programming interfaces which get the terminal information from a database of known terminals.

The terminfo database is this kind of facility. Another example is the termcap facility on Berkeley systems. (This is also available in AIX Version 5 through the file `/etc/termcap`.)

The way in which programs know what your terminal type is, and what characters to send, is controlled by the TERM environment variable. This is set to a default value when a terminal is added ('TERMINAL type'). The TERM variable points to a file `/usr/share/lib/terminfo/?/$TERM` where the `?` is the first letter of the TERM value. This is

a binary file containing the definitions for that terminal type. For example: TERM=ibm3151 means use **/usr/share/lib/terminfo/i/ibm3151**.

There are a number of terminal types supported by default on AIX Version 5, including IBM, DEC and Wyse terminals. There are also a number of sample definition files (**/usr/share/lib/terminfo/*.ti**) for many of the common ASCII terminals available. These can be used to create the binary definition files.

Note: Not all applications use the terminfo database for their terminal support. Some provide their own termcap/terminfo facility which may restrict the number of supported terminals (for example, INed).

Ensure that the package bos.terminfo.* is installed. This package contains the terminal descriptions for various terminals. These descriptions are used by libraries such as curses to obtain information about the terminal's capabilities.

Before making changes to source terminfo files to support application requirements, it is a good idea to see if your terminal will emulate another terminal whose functions are supported by the application. Many ASCII terminals have this ability.

Change the Characteristics of a TTY

Change/Show Characteristics of a TTY

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

[TOP]	[Entry Fields]		
TTY	tty3		
TTY type	tty		
TTY interface	rs232		
Description	Asynchronous Terminal		
Status	Available		
Location	01-G0-00-00		
Parent Adapter	sa0		
PORT number	[s1]	+	
Enable LOGIN	enable	+	
BAUD rate	[19200]	+	+
PARITY	[none]	+	
BITS per character	[8]	+	
Number of STOP BITS	[1]	+	
[MORE. . .33]			
F1=Help	F2=Refresh	F3=Cancel	F4=List
F5=Reset	F6=Command	F7=Exit	F8=Image
F9=Shell	F10=Exit	Enter=Do	

Figure F-10. Change the Characteristics of a TTY

AU147.0

Notes:

TTY characteristics cannot be adjusted or changed while the port or the device is busy. The device has to be temporarily disabled (for example, **pdisable** command) before proceeding and subsequently enabled again (using the **penable** command) before use.

IBM 3151 Setup Menus

General	Communication	Keyboard/Printer	Function
Code Page	CP 850	Forcing Insert	Both
Screen	NORMAL	Tab	Field
Row & Column	24 x 80	Characters	
Scroll	Jump		
Auto LF	Off		
CRT Saver	Off	Term.id	
Line Wrap	On	Alarm Volumn Level	7
Message Type	NON-DISPLAY	Curosr	Steady-block

General	Communication	Keyboard/Printer	Function
Operation Mode	ECHO	Interface	RS-232C
Lind Speed (bps)	19200	Line Control	IPRTS
Word length (bits)	8	Break Signal	500ms
Parity	NO	Send Null Suppress	ON
Stop Bit	1	Pacing	XON/XOFF
Turnaround Character	CR		

Figure F-11. IBM 3151 Setup Menus

AU147.0

Notes:

The example shows the settings for the UK-English AIX Multiuser Enhancement Cartridge to work with AIX Version 5. The menus will appear different depending on the cartridge. A cartridge is not necessary to operate in US-English mode.

To access the setup menus on an IBM 3151 press **<Ctrl+Setup>**. Use the cursor keys to move between fields and the space bar to toggle values. To go to the next menu press the **<Send>** key.

IBM 3151 Setup Menus (cont.)

General	Communication	Keyboard/Printer	Function
KEYBOARD		PRINTER	
General Code set	ASCII	Line Speed (bps)	19200
Enter	RETURN	Word Length (bits)	8
Return	NEW LINE	Parity	NO
New Line	CR	Stop Bit	1
Send	PAGE	DTR Pacing	OFF
Insert Character	SPACE		
NUM Message	ON		

General	Communication	Keyboard/Printer	Function
Recall	Save	Default	
Reset Terminal			

[EMBKB]

- This information is stored in the terminal in NVRAM
- On many terminals the menus are dependant on the options cartridge

Figure F-12. IBM 3151 Setup Menus (cont.)

AU147.0

Notes:

Most modern ASCII terminals store their characteristics in non-volatile memory and provide some setup menus to modify these characteristics.

The IBM 3151 ASCII terminal has different capabilities depending upon a cartridge which is plugged in the back. This will provide different emulation modes and national language support.

The FUNCTION menu provides options to “Recall” the previous values, “Save” the current values, reset to the “Default” values, or “Reset Terminal”. To exit without updating the values press **<Ctrl+Setup>** again.

Deleting TTYs

smit rmv tty

Remove a TTY

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

	[Entry Fields]	
TTY	tty0	
KEEP definition in database	yes	+

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

Figure F-13. Deleting TTYs

AU147.0

Notes:

You cannot remove a TTY if it is in use, either with a user logged in or a getty process running. So, if a user is using the TTY and you wish to remove it, the user will need to log out. Then, disable the TTY either by changing its attributes (through SMIT or with the **chdev** command directly) or using the **pdisable** command.

If a TTY has been disabled, a user may still be able to use it if they were already logged in to that TTY. The user will need to log out before you attempt to delete the TTY, otherwise problems may occur. When they log out a new getty process will NOT run on the terminal because it is disabled.

Now you can delete the TTY using either SMIT or the **rmdev** command.

penable/pdisable

- To enable terminals, run the **penable** command.
penable [-a] [device_name]
- To disable terminals, run the **pdisable** command.
pdisable [-a] [device_name]
- **penable** and **pdisable** by themselves, list all the terminals enabled or disabled respectively
- Only the console cannot be disabled in this manner.

Figure F-14. penable/pdisable

AU147.0

Notes:

The **penable** command enables asynchronous ports and allows users to log in. The system enables the port by updating the getty entry in the **/etc/inittab** file, and then sending a signal to the init process. This process then starts the getty placing the logon herald (logon prompt) on the terminal allowing user access.

The **pdisable** command works in a similar fashion to the **penable** command, by again updating the **/etc/inittab** file and informing the init process.

Use the **-a** option with the commands to enable or disable all ports excluding the console.

When a fault exists in the cabling between a TTY and the system, quite often the **/etc/sbin/getty** program, which displays the login prompt, gets killed and is restarted by init. If this respawning takes place too often, the message “tty respawning too rapidly” is displayed on the console. Temporarily disable the TTY while you check and repair the cabling.

TTY Problems

- Incorrect terminal type/settings
 - Change attributes (SMIT)
 - Terminal setup menu
 - TERM variable/terminfo database
- Hung terminal (crashed program or **cat** binary file)

From terminal:

Try start key **<ctrl-q>**
Reset terminal from setup menu
Try interrupt, quit keys
<ctrl-j> stty sane <ctrl-j> then log off/on again

From another terminal:

stty -a < /dev/tty
then **stty sane </dev/tty**

or **kill -9 pid_of_login_shell**

Figure F-15. TTY Problems

AU147.0

Notes:

When approaching a terminal problem, there are several issues to investigate:

1. Can the system communicate with the terminal? Try the command **echo hello > /dev/tty** and check if any output is sent to the tty.
2. Are cabling, power, brightness, contrast correct?
3. Are there any processes running on the terminal? Verify this using the **fuser -u /dev/tty** command

These questions will normally produce a resolution to the problem.

Other things to try:

- **<Ctrl+q>** (release screen)
- **<Ctrl+c>** (kill current process)
- Power-off, then power-on the terminal
- Check the NVRAM setup
- Is there a getty process running on the device? If so, **pdisable** the tty, then **penable** it.

If the backspace key does not work correctly, it will need to be remapped. Use the **stty** command to do this:

stty erase (press backspace key)

Documenting TTY Setup

- Always have a map of the concentrator boxes to the physical terminals, so that port numbers can be easily identified
- Physical labels on the cables help to identify location codes and tty numbers
- Document the settings for the setup menus
- Run **lscfg** (if you have not already done so previously) and keep a hardcopy of the output

Notes:

Checkpoint

1. True or false? If a device, like a TTY, is left for cfgmgr to configure automatically, it will pick up the default values which might not be desirable.
2. True or false? If TTYs are connected via concentrator boxes, they must all be connected in sequence on the concentrator box otherwise they will not be configured.
3. True or false? /dev/tty0 indicates that the TTY is connected to port 0, /dev/tty1 to port 1 and so on.
4. What environment variable holds the terminal type for a terminal?

Notes:

Unit Summary

- Serial devices, such as TTYs and modems must be configured manually, either through SMIT or by the high-level command
- To ensure the correct operation of devices such as TTYs, certain characteristics, such as the terminal type and baud rate must be set
- The terminfo database stores all the terminal characteristics
- Enable and disable TTYs using the **penable** and **pdisable** commands

Notes:

Appendix G. Itanium-based System Startup

What This Unit is About

This unit describes the changes made to the system initialization process in AIX 5.1. Primarily, this is support for the Intel Itanium architecture.

What You Should Be Able to Do

After completing this unit, students should be able to:

- Describe the system initialization process with AIX 5.1
- List four differences in system initialization between POWER systems and Itanium-based systems
- Startup an Itanium-based system running AIX 5.1

How You Will Check Your Progress

Accountability:

- In-line activities
- Machine exercises

References

Online *System Management Concepts: Operating System and Devices, Chapter 2. Starting and Stopping the System*

SG24-5765 *AIX 5L Differences Guide Version 5.1 Edition*

<http://redbook.ibm.com>

Redbook web page

<http://developer.intel.com/technology/efi/index.htm>

Extensible Firmware Interface (EFI) Home

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

RS/6000 Service Manuals

Unit Objectives

After completing this unit, students should be able to:

- Describe the *system initialization process* with AIX 5.1
- List *four differences in system initialization* between POWER systems and Itanium-based systems
- *Startup* an Itanium system running AIX 5.1

Figure G-1. Unit Objectives

AU147.0

Notes:

Introduction

AIX 5.1 adds support for the Intel Itanium processor. Changes have been made to the startup process to support this new hardware environment.

In addition several changes have been made to the boot process to support new AIX features.

In this unit, we will:

- Review the AIX boot process
- Identify the changes that have been made to support the Intel Itanium architecture
- Discuss the new features

AIX 5.1 Startup Process Overview

IBM POWER processor

Intel Itanium processor

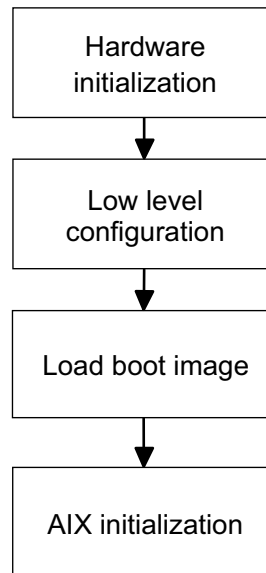


Figure G-2. AIX 5.1 Startup Process Overview

AU147.0

Notes:

Boot process overview

For the purposes of our discussion, we will divide the boot process into the four stages shown in the visual. In the next few visuals, we will review the AIX boot process for POWER systems and contrast the differences for Itanium-based systems. Then we will provide details of the boot process for Itanium-based systems.

Architecture types

With the addition of support for the Intel Itanium architecture, AIX 5.1 now supports four architecture types:

Architecture	Processor	Description
rs6k	POWER	This is the original or "classic" RS/6000 workstation based on the microchannel bus
rspc	POWER	POWER Reference Platform, based on the PCI bus
chrp	POWER	Common Hardware Reference Platform, based on the PCI bus
ia64	Intel Itanium	Intel Itanium architecture

The `bootinfo -p` command returns the system architecture type.

Overview: Hardware Initialization

IBM POWER processor

Intel Itanium processor

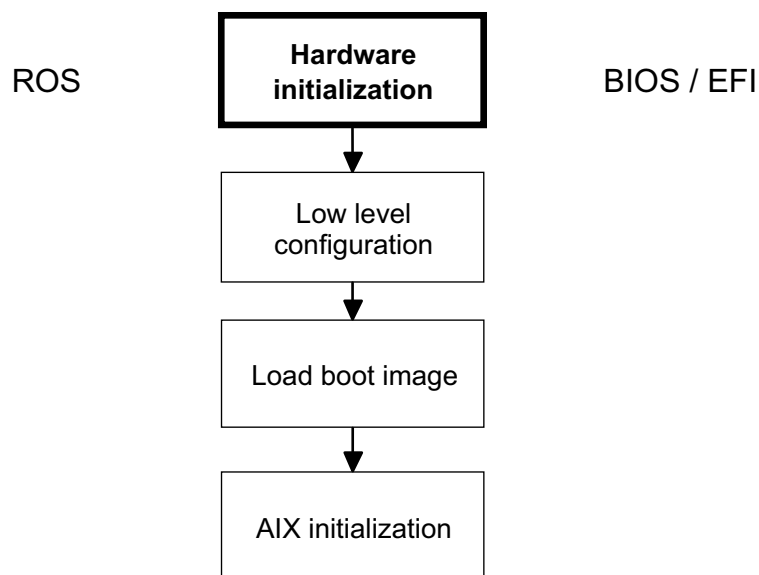


Figure G-3. Overview: Hardware Initialization

AU147.0

Notes:

Hardware initialization

When the power is turned on, system firmware performs hardware initialization. Hardware initialization includes: detecting what hardware is present, performing basic functionality tests, and initializing hardware settings.

POWER systems

On RS/6000 machines, this firmware is called ROS (Read Only Storage).

Itanium-based systems

On Itanium-based systems, this firmware is called BIOS (Basic Input Output System) and EFI (Extensible Firmware Interface).

Overview: Low Level Configuration

IBM POWER processor

Intel Itanium processor

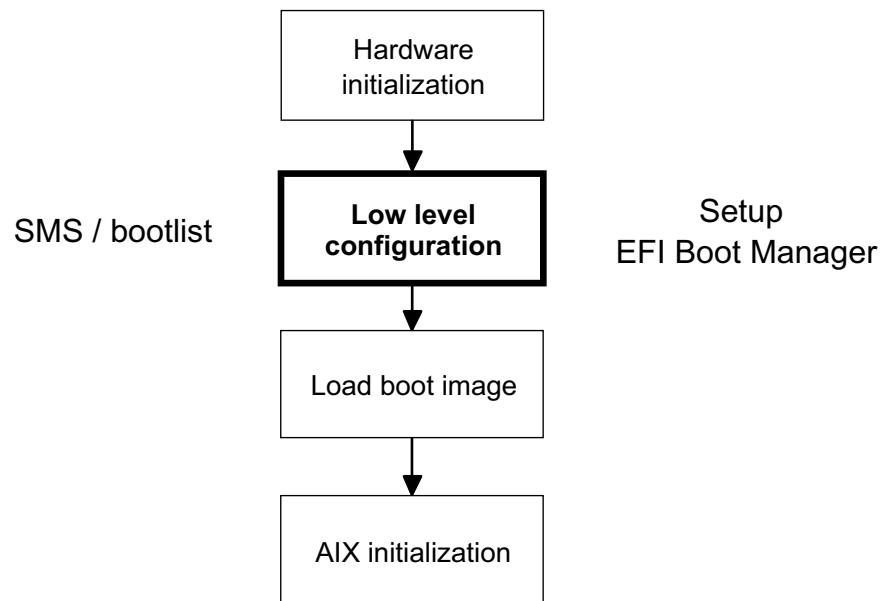


Figure G-4. Overview: Low Level Configuration

AU147.0

Notes:

Low level configuration

After the hardware is initialized, the system provides an opportunity to enter a low level configuration tool. This tool allows you to display system information and perform various configuration tasks, including configuring the boot list. The boot list is an ordered list of possible devices.

POWER

On RS/6000 machines, this function is provided by SMS (System Management Services). In addition, you can modify the boot list from AIX using the `bootlist` command.

Itanium-based systems

On Itanium-based systems, the **Setup** menu provides the ability to display and set hardware configuration. You can manage the boot list using the **EFI Boot Manager**. There is no facility to modify the boot list from AIX.

Overview: Load Boot Image

IBM POWER processor

Intel Itanium processor

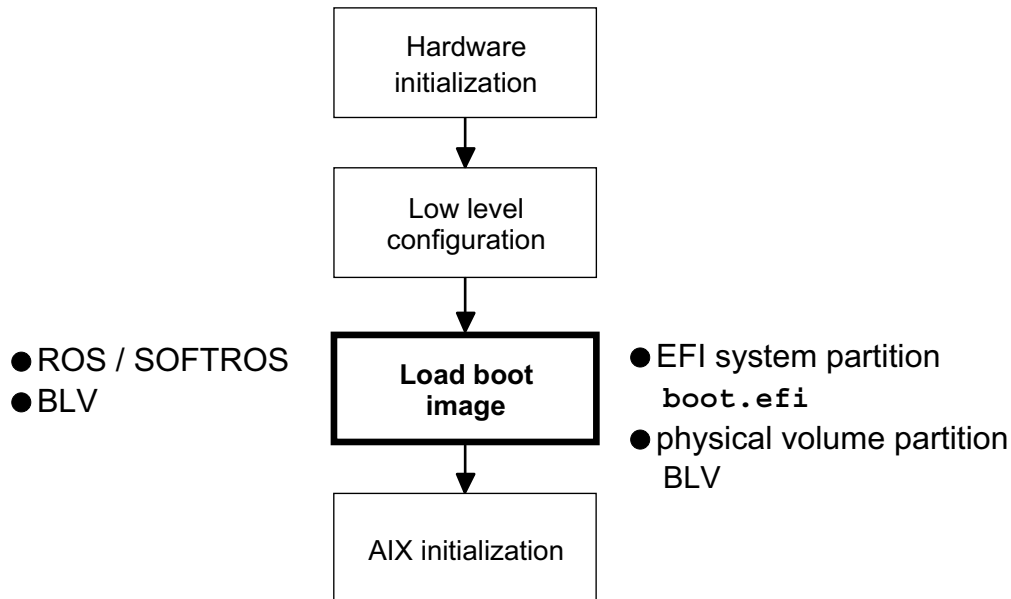


Figure G-5. Overview: Load Boot Image

AU147.0

Notes:

Load the boot image

Using the information in the boot list, the system locates the boot device and attempts to load the boot image into system memory. The boot image includes the programs and files required to start AIX, including the AIX kernel, the RAM file system (RAMFS), and a copy of the base ODM.

POWER

On RS/6000 machines, depending on the model, either ROS or SOFTROS loads the boot image into memory. The boot image resides in the boot logical volume (BLV). SOFTROS is a program that resides in the BLV.

Itanium-based systems

On Itanium-based systems, the boot disk is divided into two partitions:

- EFI system partition
- Physical volume partition

The EFI system partition is a small section of the disk required by the EFI firmware. It contains the boot loader (a program called `boot.efi`).

The physical volume partition contains rootvg, which contains the boot logical volume.

`boot.efi` loads the boot image from the boot logical volume into memory.

Overview: AIX Initialization

IBM POWER processor

Intel Itanium processor

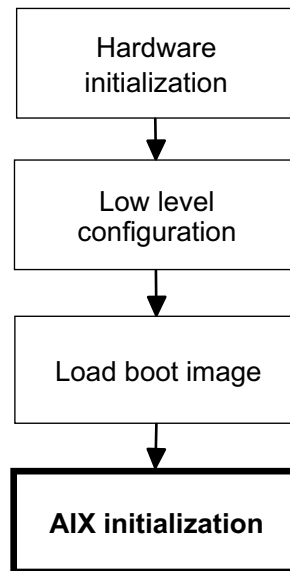


Figure G-6. Overview: AIX Initialization

AU147.0

Notes:

AIX initialization

AIX initialization begins when the boot image has been loaded into memory. AIX initialization is essentially the same for both POWER and Itanium-based systems.

There are several new features added to AIX initialization in AIX 5.1. These will be discussed later in this unit.

Startup Progress and Error Messages

- Display codes
 - LED
 - Console
- Documentation
 - *Service Manual*
 - *AIX Messages Guide and Reference*

Figure G-7. Startup Progress and Error Messages

AU147.0

Notes:

Introduction

Like earlier versions of AIX, AIX 5.1 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, these codes are displayed on the console or on an LED display on the front panel.

On some platforms, system firmware also provides display codes during hardware initialization.

PCI POWER systems with LEDs

Newer, PCI-based RS/6000 systems use both an LED display and the console to show boot status:

- System firmware sends boot status codes (called firmware checkpoints) to the LED display.
- Once the console is initialized, the firmware sends 8-digit firmware error codes to the console.
- During AIX initialization, the `rc.boot` script and the device configuration methods send progress and error codes to the LEDs.
- The LEDs are also used if `cfgmgr` is run manually from AIX.

PCI POWER systems without LEDs

Early PCI-based RS/6000 systems do not have an LED display. Status and error messages are sent using the system speaker (beeps), the hard disk status light, or the console.

Microchannel POWER systems

The LED display is the progress and error indicator for RS6K RS/6000 systems. Both system firmware and AIX send display codes to these LEDs during system boot.

Itanium-based systems

At the time of this writing, Itanium-based systems do not have an LED display. During AIX initialization, the `rc.boot` script and the device configuration methods send progress and error indicators to the system console.

Hardware initialization status messages (if any) depend on the manufacturer.

Documentation

AIX display codes are documented in the *AIX Messages Guide and Reference*. This documentation applies to both POWER and Itanium-based systems running AIX.

Firmware checkpoints and error codes are documented in the *RS/6000 Service Manual* for your model. This documentation applies to POWER systems only. See: http://www.rs6000.ibm.com/resource/hardware_docs

Itanium-based Systems: Hardware Initialization

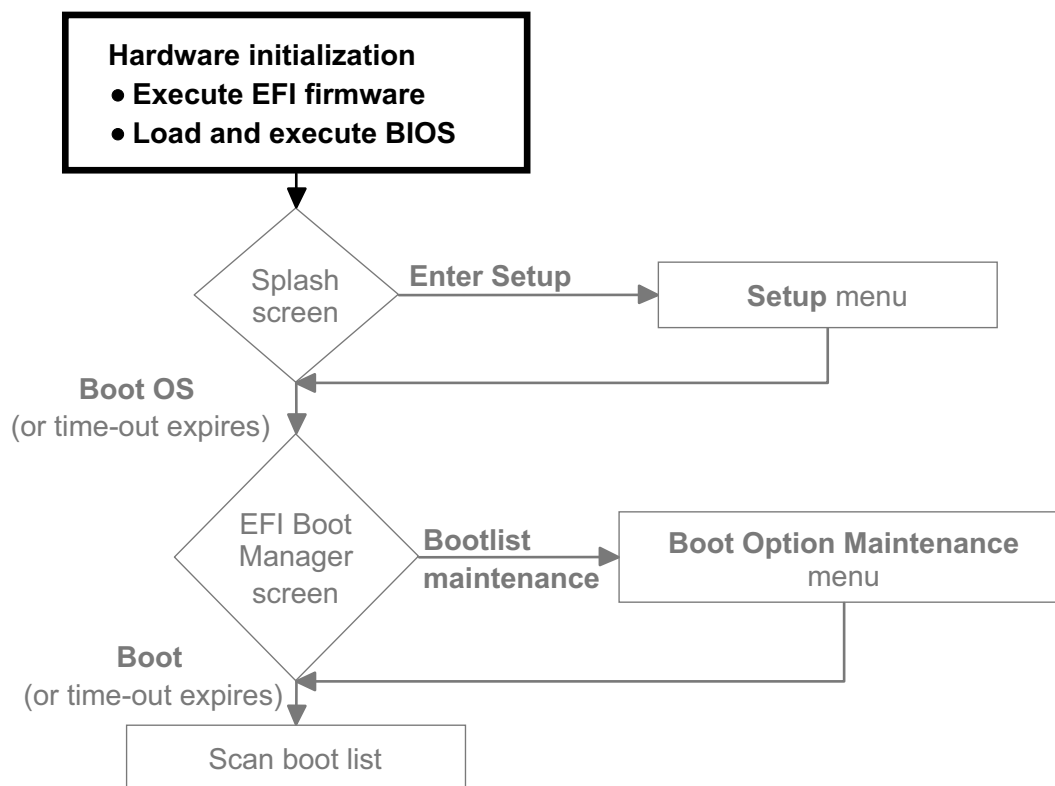


Figure G-8. Itanium-based Systems: Hardware Initialization

AU147.0

Notes:

Hardware initialization

Hardware initialization is performed by EFI, system BIOS, and any device specific BIOS.

EFI

The Extensible Firmware Interface (EFI) is a new standard created by Intel. EFI provides a standard interface between the operating system and the firmware. EFI runs on all Itanium-based systems regardless of the hardware vendor, or the operating system.

EFI benefits

EFI provides a standard interface between the operating system and the firmware. In this way, it will be easier for Intel to support multiple operating systems on

Itanium-based systems. In addition, it should be easier for OS vendors to move an operating system from one Itanium-based system to another. The platform-specific firmware is hidden from the operating system by the EFI layer.

Intel has explicitly made EFI extensible, so that hardware manufacturers can add on their own features if desired.

BIOS

Firmware in the Intel architecture world has traditionally been called BIOS (Basic Input Output System). There are two types of BIOS:

- System BIOS resides on the system board.
- Device specific BIOS resides on the controller board for that device. Examples of device specific BIOS would be firmware for a PCI SCSI controller or a graphics adapter card.

When a BIOS executes, you may see output to the console screen. For example, a SCSI controller's BIOS may probe the SCSI bus and report on the drives that it found.

Itanium-based Systems: Low Level Configuration

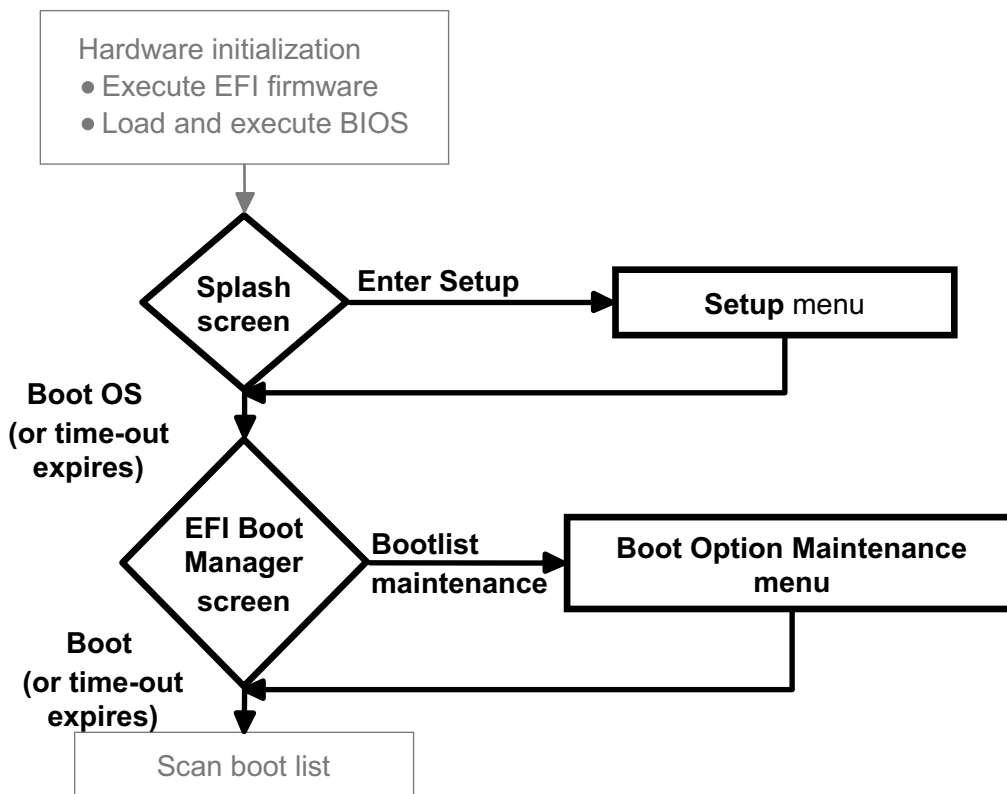


Figure G-9. Itanium-based Systems: Low Level Configuration

AU147.0

Notes:

Splash screen

Following hardware initialization, the system displays the splash screen. Typically this screen will display the hardware vendor's logo, and so forth. The screen is displayed for a delay period. The delay period may also be called a time-out.

From this screen, you can:

- Take no action
The system will attempt to boot the operating system at the end of the delay period.
- Choose **Boot the OS**
The system will boot immediately, skipping the delay:
- Choose **Enter Setup**
The system will enter the **Setup** menu.

Setup

The **Setup** menu allows you to display and configure the hardware. You can also set the splash screen delay. Specifics of what can be done from the **Setup** menu are manufacturer dependent. Typically, you can configure the IDE bus controller, set system date and time, view hardware status (processor type, firmware level, memory), and so forth.

EFI Boot Manager

Next, the EFI Boot Manager screen is displayed for a delay period. A list of *boot options* (possible boot devices) is displayed. For example, boot options might include the AIX boot disk and the CD-ROM drive.

From this screen you can:

- Take no action. After the delay the system will attempt to boot from the first boot option.
- Boot from the selected boot option immediately, skipping the delay.
- Select another boot option from the list and attempt to boot from it.
- Select **Bootlist maintenance** to enter the **Boot option maintenance** menu.

Note: EFI provides a default boot manager. Vendors can also create their own, so the details of how the boot manager operates may be platform dependent.

Boot Option Maintenance menu

This menu allows you to manage how the system will boot. You can choose to:

- Boot from a file
- Add or delete boot options
- Change the order of the boot options
- Set the EFI Boot Manager delay time

Boot from a file allows you to locate and boot from a file (usually an OS boot loader) on an EFI partition.

Installation

When you perform a **New and Complete Overwrite** installation of AIX, the installation process automatically creates a new entry in the EFI Boot Manager and places it at the top of the boot list. On the next boot, the system will automatically boot from the newly installed disk.

Itanium-based Systems: Loading the Boot Image

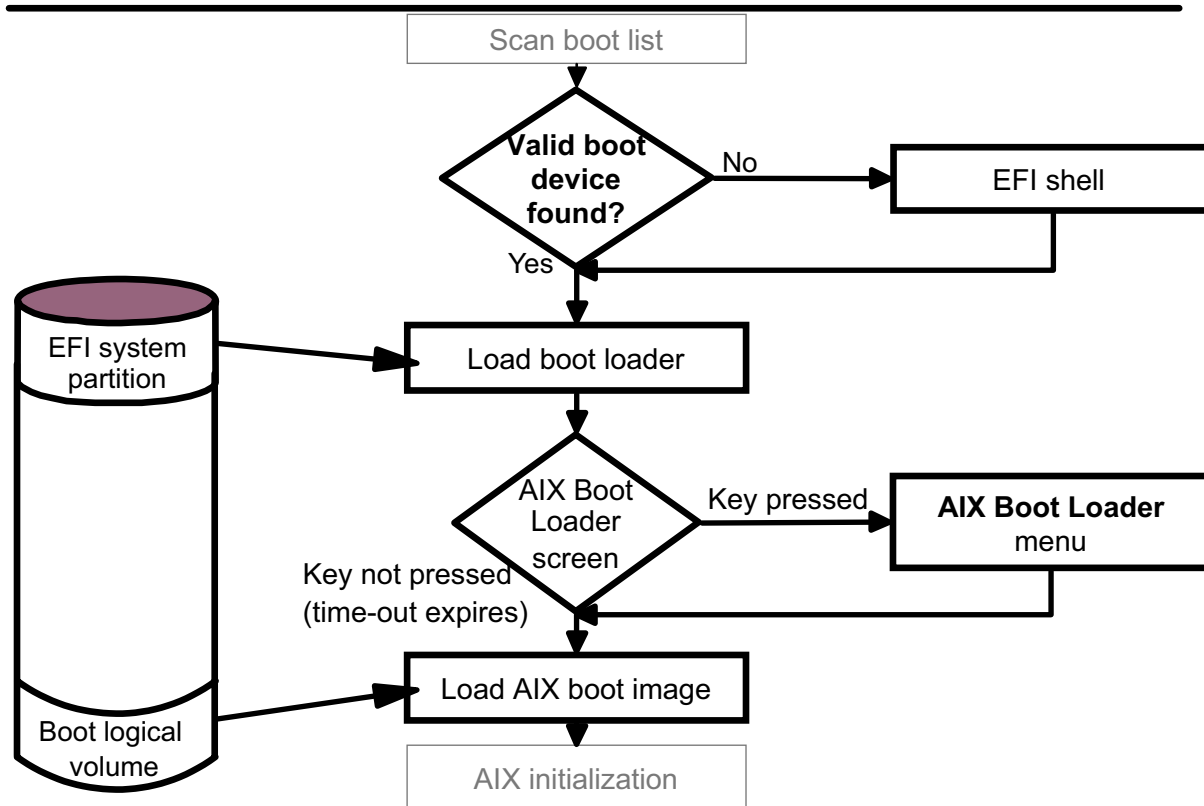


Figure G-10. Itanium-based Systems: Loading the Boot Image

AU147.0

Notes:

EFI shell

If a valid boot device is not found, the **EFI Boot Manager** opens the EFI shell. The shell provides a DOS-like interface. The primary purpose of the shell is to locate a boot device. A number of other functions are also available.

Although not shown in the drawing, you can also enter the EFI shell directly from the **EFI Boot Manager**.

Note: On some early Itanium-based systems, the system enters the EFI shell directly following the splash screen. Type `exit` to go to the **EFI Boot Manager**.

AIX boot loader

If the **EFI Boot Manager** locates a valid boot device, it loads and passes control to the boot loader on that device. The AIX boot loader displays a choice screen for five seconds.

From this screen, you can:

- Take no action
The AIX boot loader will load the boot image after the delay.
- Press any key
The system will enter the **AIX Boot Loader** menu.

AIX Boot Loader menu

From this menu you can set various boot loader and kernel options.

Boot logical volume

The boot loader loads the boot image into memory. The Boot Logical Volume (BLV) is a logical volume on the boot disk which contains the boot image. The boot logical volume is part of rootvg and has a logical volume type attribute of `boot`.

AIX initialization

When the boot image is loaded into memory, control is transferred to the AIX kernel. This is the start of AIX initialization.

Itanium-based Systems: Layout of the Boot Disk

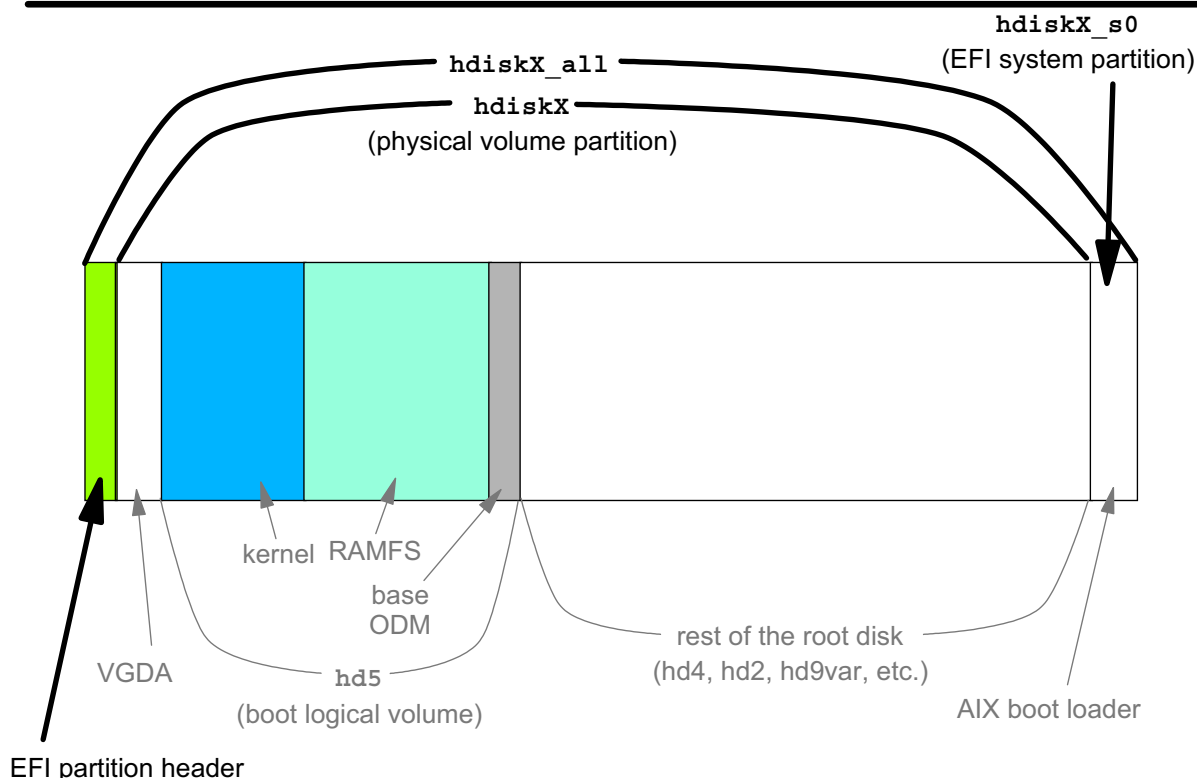


Figure G-11. Itanium-based Systems: Layout of the Boot Disk

AU147.0

Notes:

EFI partitioning

EFI has been designed to support a variety of operating systems on Itanium-based systems. In order to do this, EFI defines a new disk partitioning scheme to replace the legacy DOS partitioning support. EFI recognizes two types of partitions:

Partition type	Description
EFI system partition	When booting from a disk device, the EFI firmware utilizes one or more EFI system partitions containing an EFI file system (FAT32). An EFI system partition contains extensions to EFI, the most important is the OS boot loader.
Physical volume partition	The content of the physical volume partition is defined by the operating system. EFI does not access physical volume partitions. The physical volume partition contains the logical volumes of rootvg.

EFI partitions used with AIX 5.1

AIX 5.1 for Itanium-based systems supports up to four partitions (and the EFI header). Normally there are only two partitions: one EFI system partition and one physical volume partition (and the EFI header).

The following table and the drawing on the preceding page show the default boot disk partitions and how they are seen by EFI and AIX.

Item	EFI view	AIX view
Entire physical disk	Entire physical disk	/dev/hdiskX_all
EFI system partition	This is a FAT32 file system containing the AIX boot loader (and potentially other EFI applications).	/dev/hdiskX_s0 Normally not accessed by AIX. Utility programs can use this device to update the AIX boot loader.
Physical volume partition	Contents are not accessible to EFI.	/dev/hdiskX To AIX, this looks like an entire physical disk. It contains the rootvg logical volumes (hd5, hd4, and so forth.)

EFI partition header

The EFI partition header contains information defining the size and location of the disk partitions.

Partitioning the boot disk

The boot disk is partitioned at installation time. The installation scripts will automatically partition the boot disk for you, or you can specify the partitions manually. The installation scripts use the following programs:

- `efdisk` to partition the disk
- `eformat` to create the FAT32 file system in the EFI partition

From AIX, use `efdisk` to display the partitioning information for the boot disk:

```
# efdisk -p /dev/hdiskX_all
```

POWER differences

AIX 5.1 for the POWER architecture does not support partitioning physical disks. Disks are divided into logical volumes using LVM. In AIX 5.1 for Itanium-based systems, the boot disk is first partitioned into the EFI system partition and the physical volume partition, then the physical volume partition is divided into logical volumes using LVM.

Itanium-based Systems: Boot Loader and Boot Logical Volume

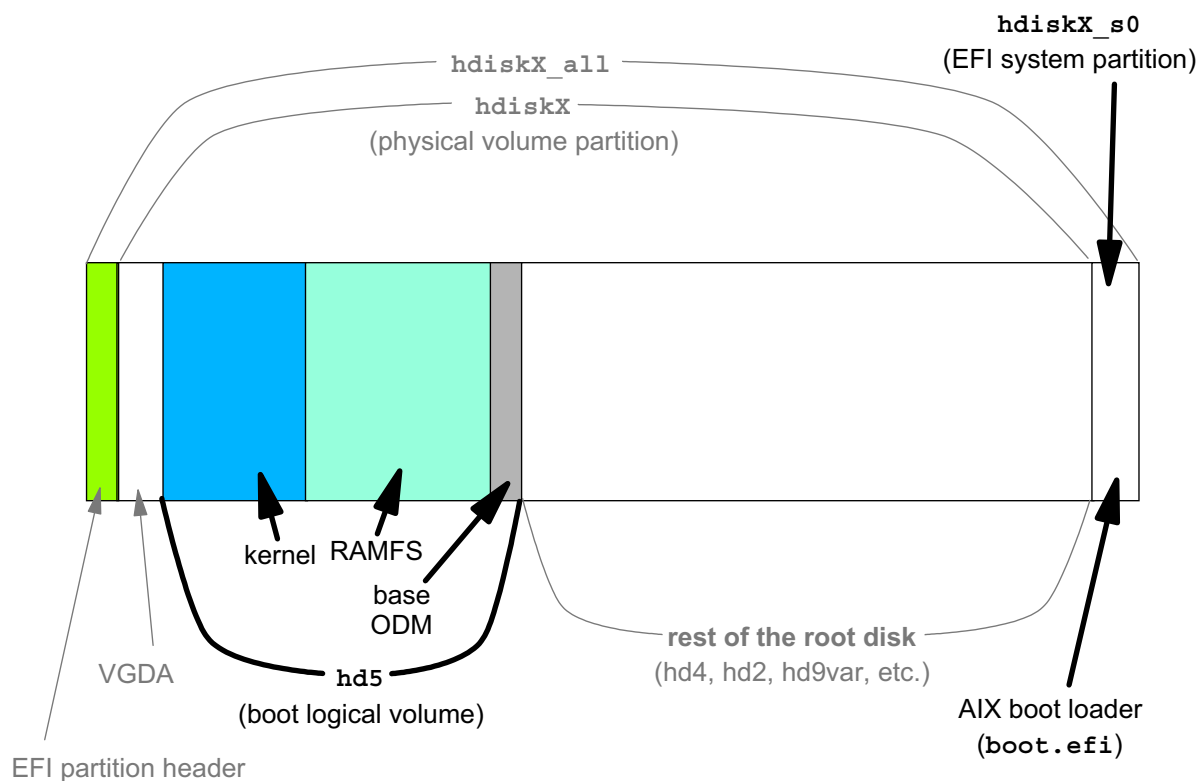


Figure G-12. Itanium-based Systems: Boot Loader and Boot Logical Volume

AU147.0

Notes:

AIX boot loader

Once the boot device is selected, the **EFI Boot Manager** loads and runs `boot.efi`.

`boot.efi`

`boot.efi` initializes various data structures for the operating system, locates the boot logical volume, and loads the kernel and the RAMFS (RAM File System). After loading the boot image, `boot.efi` passes control to the kernel.

Boot logical volume

The boot logical volume (normally named hd5) contains the AIX boot image. The boot image consists of the kernel, the RAM file system, and the base ODM (a reduced copy of the ODM database). The AIX 5.1 boot logical volume components are essentially the same as AIX 4.3. They are described briefly here as a review.

AIX kernel

The kernel is that part of the operating system which remains resident in memory and which controls access to the system hardware and provides basic services to application programs.

When started by the boot loader, the kernel initializes itself and then executes `/etc/init` in the RAM file system.

Note: The kernel that is loaded from the boot logical volume is never replaced during the boot process. The same kernel is used in multiuser mode. A copy of the kernel also exists on the hard disk (`/unix`), but this copy is only used for reference by the `ps` command. If you need to use a new kernel, you must re-create the boot logical volume with the new kernel.

RAM file system (RAMFS)

This is a reduced or miniature root file system which is loaded into memory and used as if it were a disk based file system. RAMFS is used during phases 1 and 2 of AIX initialization.

Base ODM

This is a small version of the ODM database. It contains only information for the base devices. Base devices are those devices needed to access the root disk. Once the disk-based root file system is mounted, the full disk-based ODM is used.

The base ODM is copied into the RAM file system during phase 1.

POWER differences: boot logical volume

The BLV for POWER systems differs from the BLV for Itanium-based systems:

- In AIX 5.1, the contents of the BLV are not compressed for Itanium-based systems. For POWER machines, the kernel and RAMFS are compressed in the BLV. The POWER BLV includes the `bootexpand` program to uncompress them as they are loaded into memory.
- The BLV for the RSPC and CHRP POWER architectures also includes SOFTROS, which is the boot loader. On RS6K architectures, ROS acts as the boot loader.

Let's Review (1)

1. On Itanium-based systems, hardware initialization is performed by _____ and _____.
2. Early in the boot process, you can use the _____ menu to view hardware status and configure the IDE bus controller.
3. You can manage the boot list using the _____.
4. The _____ contains the AIX boot loader.
5. The physical volume partition contains _____.
6. What is the AIX device name for the EFI system partition?

7. What is the AIX device name for the physical volume partition?

8. List the three components of the boot logical volume.
_____, _____, _____

Figure G-13. Let's Review (1)

AU147.0

Notes:

EFI Shell

Command	Description
<code>map</code>	Displays the mapping of boot devices to short volume names.
<code>short_name:</code>	Navigate to a particular boot device (volume) by typing the short name, followed by a colon (:). The prompt changes to the short name of the volume.
<code>ls</code> or <code>dir</code>	Lists the files in a directory in an EFI system partition (FAT32 file system).
<code>cd</code>	Changes directories.
<code>set</code>	Creates or displays EFI variables.
<code>boot</code>	This is the default name for operating system boot loaders. When <code>boot</code> is typed on the command line, EFI searches for a file named <code>boot.efi</code> and executes it, if found.
<code>help</code> or <code>?</code>	Lists the available EFI commands and a short description of each.
<code>exit</code>	Exits the EFI shell, returning to the EFI Boot Manager.

Figure G-14. EFI Shell

AU147.0

Notes:

Introduction

It's useful to know how to use the EFI shell because if EFI cannot find a valid boot device, you will be given the EFI shell prompt. Also, we'll use the shell to learn how EFI names devices. In order to specify boot devices in the **EFI Boot Manager**, you need to understand how EFI names devices.

EFI commands

The EFI shell provides a number of commands. The visual summarizes the commands we will be using in this class. Most of these commands work like AIX or DOS commands that you already know. More details on using these commands follow in the next few visuals.

Types of commands

There are two types of EFI shell commands:

- commands that are built into the shell
- file-based commands

set

The shell uses the `path` variable to locate file-based commands. The `set` command can be used to display or change the value of `path`.

boot

The `boot` command is an example of a file-based command. The `boot` command will only work if the file `boot.efi` exists on an EFI system partition and the `path` variable includes the partition and directory which contains the file.

EFI Device Naming

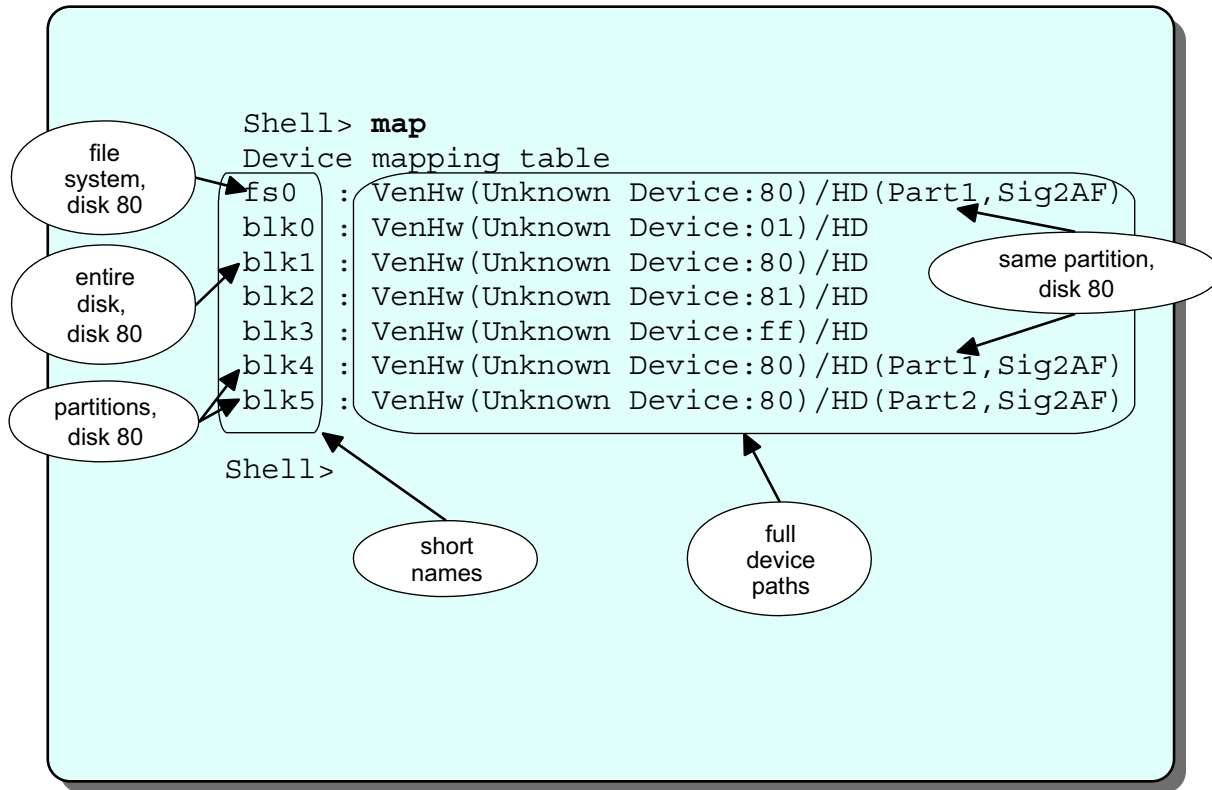


Figure G-15. EFI Device Naming

AU147.0

Notes:

Introduction

EFI maintains two types of names for boot devices: the **full device path** and the **short name**. A short name is an alias for the full device path. A single device can have several short names. The `map` command displays the mapping of full paths to short names. `map` is run whenever you enter the EFI shell.

Example

The example shows output from the `map` command. The first column is the short name. Everything after the first colon is the full device path. Notice that some devices have several short names.

Short names

EFI assigns short names to the boot devices to make it easier to refer to them. The user can create additional short names if desired. EFI assigns short names in the following manner:

- Every device is given a `blkX` (*block device*) short name.
- If a device has several partitions, the entire disk gets a block device name and each partition will get a separate block device name.
- If EFI detects a FAT32 file system on a partition, that partition will also get a `fsX` (*file system*) short name. This indicates an EFI system partition.

Each time the system boots, EFI recreates the short names based on the devices it finds.

Full device path

The full device path can include a:

- Device number (assigned by EFI during hardware initialization)
- Partition number
- Hardware signature, uniquely identifying the disk

Partition names

If the disk is partitioned, each partition will have at least one entry in the mapping table.

Itanium-based Systems: Boot Devices

- **IDE buses**
 - Primary bus
 - Master: UHD (Ultra High Density) floppy
 - Slave: CD-ROM
 - Secondary
 - Master
 - Slave
- **SCSI buses**
 - AIX boot disk (rootvg)
 - Other disk or tape drives
- **Device numbers** in full device path
 - 01 floppy drive on IDE bus
 - ff CD-ROM on IDE bus
 - 80, 81, ... hard disks. IDE disks are numbered first

Figure G-16. Itanium-based Systems: Boot Devices

AU147.0

Notes:

Itanium-based system boot devices

Boot devices for Itanium-based system can be found on two types of buses:

Bus	Description
IDE bus	The IDE controller provides two buses. The CD-ROM drive and a UHD floppy drive are on the first, or primary, IDE bus. Two additional hard drives could be connected to the secondary bus.
SCSI bus	The SCSI bus controller is a PCI card. Hard disk drives or tape drives can be connected to this bus.

Other types of boot devices

Booting from the network is not yet supported on Itanium-based systems. Booting from tape is not supported on Itanium-based systems. You can boot from CD-ROM and then restore from a `mksysb` tape. You can also create a bootable `mksysb` CD-ROM.

Device numbers

EFI numbers devices as follows:

- Device 01 = floppy drive
- Device ff = CD-ROM drive
- Device 80, 81, etc. = hard disk drives
IDE hard drives (if any) are numbered first, then the SCSI drives. For example, if you had one IDE drive and two SCSI drives, the IDE drive would be device 80 and the SCSI drives would be 81 and 82.

Each time the system boots, EFI rennumbers the drives based on the devices found.

Signatures

The disk signature is a string of numbers and letters uniquely identifying each disk. This number does not change across system reboots, even if a disk's bus location changes. The disk signature is also referred to as the GUID (Globally Unique Identifier).

Note: Signatures in the visuals are truncated to save space.

EFI Hard Disk Example

```
Shell> map
Device mapping table
fs0 : VenHw(Unknown Device:80)/HD(Part1,Sig2AF02DE2)
blk0 : VenHw(Unknown Device:01)/HD
blk1 : VenHw(Unknown Device:80)/HD
blk2 : VenHw(Unknown Device:81)/HD
blk3 : VenHw(Unknown Device:ff)/HD
blk4 : VenHw(Unknown Device:80)/HD(Part1,Sig2AF02DE2)
blk5 : VenHw(Unknown Device:80)/HD(Part2,Sig2AF02DE2)

Shell> fs0:
fs0:\> cd \efi\aix
fs0:\> ls
Directory of fs0:\EFI\AIX\
 10/18/00 11:25p      2,636,000  BOOT.EFI

    1 Files 2,636,000 bytes
    0 Dir   0 bytes
```

Figure G-17. EFI Hard Disk Example

AU147.0

Notes:

Introduction

In this example, the system was powered-on without a CD-ROM in the drive, thus device `ff` has only a block short name.

There are two SCSI disks attached to this machine (devices 80 and 81).

Device 81 is not partitioned and has no EFI system partition, hence there is no file system short name associated with it.

EFI system partition

Since partition 1 of disk 80 has a file system short name, we know that it is an EFI system partition. By typing `fs0:`, we navigate to that partition. The AIX boot loader file (`BOOT.EFI`) resides in `\EFI\AIX`.

EFI CD-ROM Example

```

Shell> map
Device mapping table
fs0 : VenHw(Unknown Device:80)/HD(Part1,Sig2AF02DE2)
fs1:  : VenHw(Unknown Device:ff)/CDROM(Entry1)
blk0  : VenHw(Unknown Device:01)/HD
blk1  : VenHw(Unknown Device:80)/HD
blk2  : VenHw(Unknown Device:81)/HD
blk3  : VenHw(Unknown Device:80)/HD(Part1,Sig2AF02DE2)
blk4  : VenHw(Unknown Device:80)/HD(Part2,Sig2AF02DE2)
blk5  : VenHw(Unknown Device:ff)/HD
Shell> fs1:
fs1:\> ls
Directory of fs1:\
09/07/00 12:24p <DIR>          2,048  AIX
09/07/00 12:24p <DIR>          2,048  EFI
09/27/00 04:33p                2,636,000  boot.efi

      1 File    2,636,000  bytes
      2 Dirs     4,096    bytes

```

Figure G-18. EFI CD-ROM Example

AU147.0

Notes:

Introduction

In this example, the same machine was powered-on with the Software Installation CD-ROM in the drive. Now there is a file system short name (`fs1:`) listed for device `ff` as well as the block short name (`blk5:`).

Note: The EFI shell only reads the CD-ROM once during power up. If you insert a CD-ROM when the system is at the **EFI Boot Manager** menu or EFI shell, EFI still acts like there is no CD-ROM in the drive. You must cycle the power to the system to have EFI recognize the CD-ROM.

CD-ROM vs. disk

The AIX installation CD-ROM contains only one partition. The boot loader files and the boot image are both in the one EFI system partition.

AIX Boot Image on CD-ROM

Kernel and RAMFS are files in the AIX directory

```
fs1:\> cd AIX
fs1:\AIX> ls
Directory of fs1:\AIX
09/07/00 12:24p <DIR>      2,048 .
09/07/00 12:24p <DIR>      0 ..
09/27/00 04:32p      7,505,688 aix64
09/27/00 04:32p      5,229,119 ramfs

    2 Files 12,734,807 bytes
    2 Dirs  2048 bytes
```

Figure G-19. AIX Boot Image on CD-ROM

AU147.0

Notes:

CD-ROM boot image

The AIX directory on the CD-ROM contains the boot image, which includes the AIX kernel and the RAM file system. Note that the CD-ROM has no base ODM.

EFI Boot Manager

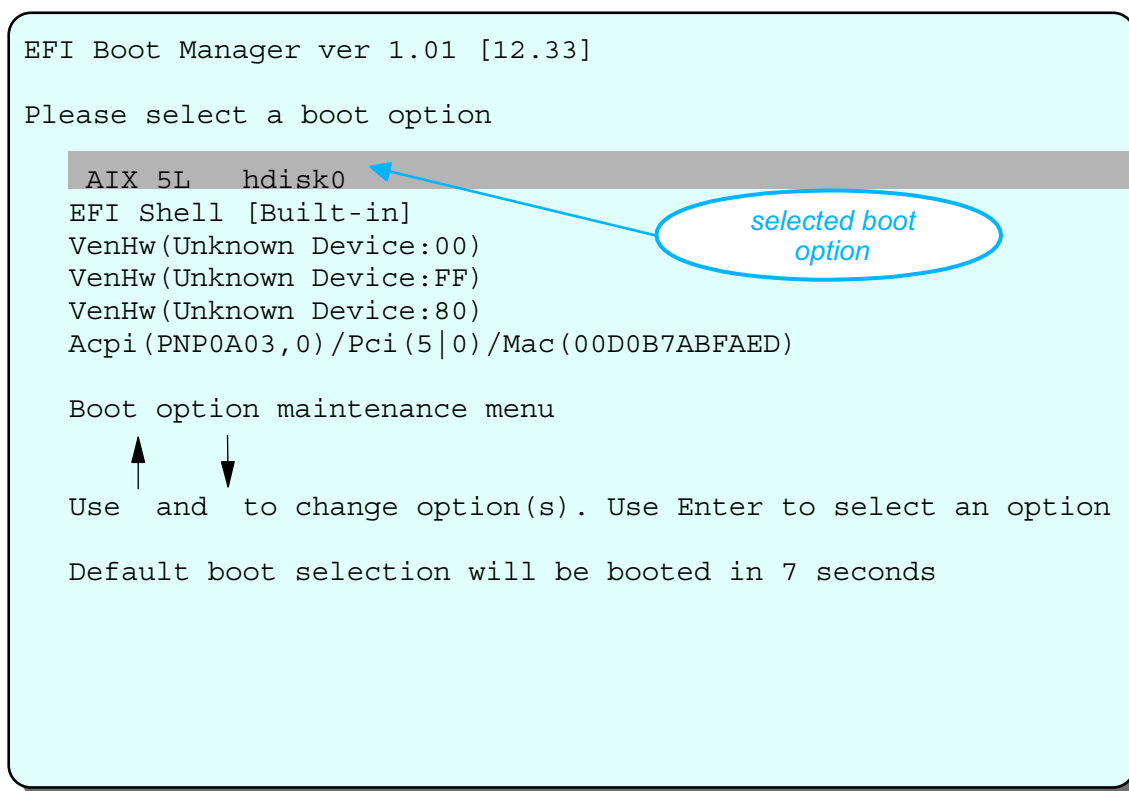


Figure G-20. EFI Boot Manager

AU147.0

Notes:

Introduction

This visual shows the **EFI Boot Manager** screen. During startup, this screen is displayed for a delay period (default delay is seven seconds). Pressing one of the arrow keys stops the timer and allows you to select a different boot option, or enter the **Boot option maintenance menu**. If you do not intervene, the system will attempt to boot using the first boot option.

If EFI is not able to boot the system from the first boot option, it will go down the list, trying each boot option in order. In the example, if EFI is unable to boot from the first option (AIX 5L hdisk0), EFI opens the EFI shell (the second boot option).

Boot options

Boot options in the **EFI Boot Manager** consist of two parts: the boot description and the device path.

Boot description

In the example above, `AIX 5L hdisk0` is a boot description. The boot description is a name for humans and can be any text. This particular boot description was created by the installation program.

EFI device path

The device path allows EFI to locate the selected boot volume. Details are provided in the next visual.

Boot options maintenance menu

Using the Arrow keys, you can select a different boot option, enter the EFI shell, or go to the **Boot options maintenance menu**. From this menu, you can add or delete boot options, change the order of the boot options, and so forth.

Installation boot option

When a New and Complete Overwrite installation is performed, the installation program creates a boot option for the new boot disk and places it first in the boot list. The visual shows an example of the entry created by the installation program, in this case: `AIX 5L hdisk0`.

EFI Device Path

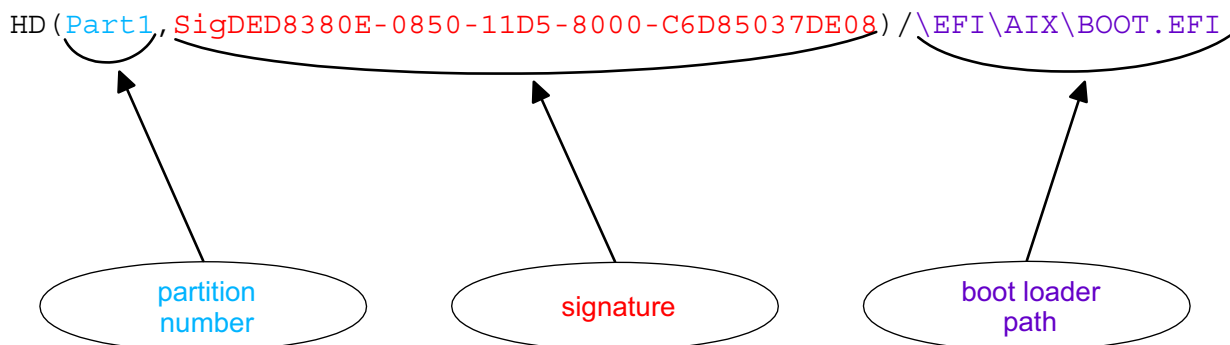


Figure G-21. EFI Device Path

AU147.0

Notes:

Device path

The device path uniquely identifies the boot device.

The **EFI Boot Manager** device path consists of:

- Partition number
In the example, `Part1` is the partition number.
- Signature
In the example, `SigDED8380E-0850-11D5-8000-C6D85037DE08` is the signature.
- File path to the boot loader program
In the example, `\EFI\AIX\BOOT.EFI` is the file path.

Adding a boot option

When you add a new boot option, you select a boot device from a list of device paths. The EFI Boot Manager prompts you for the path to the boot loader program and a boot description. Only the boot description is displayed in the EFI Boot Manager screen.

EFI boot options and system re-configurations

EFI uses the disk signature, not the boot description, the EFI short name, the EFI device number, or the location in the system (bus number and bus address) to locate a boot device. This means that the boot option does not need to be changed if the boot disk is moved to a new location in the system or if devices are renumbered by EFI because additional devices were added to the system.

EFI Boot Option Maintenance Menu

```
EFI Boot Maintenance Manager ver 1.01 [12.33]

Main Menu. Select an Operation

    Boot from a File
    Add a Boot Option
    Delete Boot Option(s)
    Change Boot Order

    Manage BootNext setting
    Set Auto Boot Timeout

    Select Active Console Output Devices
    Select Active Console Input Devices
    Select Active Console Error Devices

    Cold Reset
    Exit
```

Figure G-22. EFI Boot Option Maintenance Menu

AU147.0

Notes:

Introduction

This visual shows the operations available from the EFI Boot Maintenance Manager Main Menu.

Boot from a File

This menu allows you to navigate to a boot loader program and boot. This is similar to booting using the EFI shell.

Add a Boot Option

The table shows the steps involved in adding a boot option.

Step	Action
1.	EFI displays a list of available boot devices and prompts you to <code>Select a Volume</code> . You can use the arrow keys to select a volume.
2.	When you select a volume, EFI displays the files and directories in the root directory of the volume you selected.
3.	You can use the arrow keys to select a file or a directory. If you select a directory, EFI displays the contents of that directory (as in the previous step).
4.	When you navigate to the directory with your boot loader and select that file, EFI displays the filename and device path you have selected and prompts for a boot description. This is the short boot description which will be displayed in the EFI Boot Manager screen, (See “EFI Boot Manager” on page 33.).
5.	After entering the description, EFI prompts you to <code>Enter BootOption Data</code> . This allows you to pass arguments to the boot loader. For AIX 5.1, enter N for <code>No BootOption</code> .
6.	Finally, EFI prompts you to <code>SaveChanges to NVRAM [Y-YES N-No]</code> . If you answer Y , your new boot option will be saved.

Delete Boot Option(s)

This allows you to delete boot options.

Change Boot Order

Use this menu to change the order.

Manage BootNext Setting

This menu allows you to set a one time boot option for use on the next reboot only.

Set Auto Boot Timeout

This menu allows you to set the delay period for the **EFI Boot Manager** screen.

Select Active Console Devices

These three menus allow you to select your console device.

EFI Change Boot Order

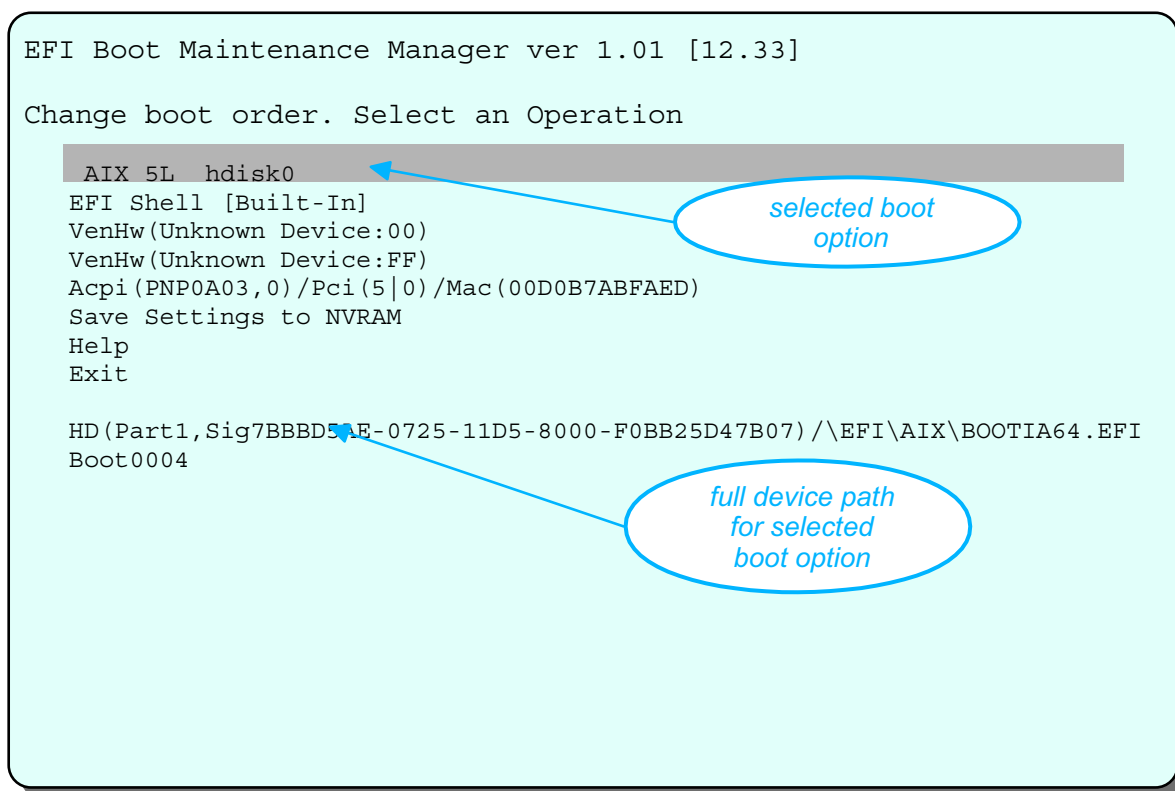


Figure G-23. EFI Change Boot Order

AU147.0

Notes:

Introduction

When you add a new boot option, EFI places it at the end of the boot option list. This visual shows the **Change boot order** menu. Use the arrow keys to select a boot option. The full device path for the selected boot option is displayed in the bottom of the screen.

Changing the order

Select the boot option using the arrow keys. Press **↑** or **↓** to move the selected boot option up or down respectively. When you have your desired order, select **Save Settings to NVRAM** and **Exit**.

Let's Review (2)

- Answer the questions using the following output:

```
Shell> map
Device mapping table
fs0    : VenHw(Unknown Device:80)/HD(Part1,Sig2AF02DE2)
blk0   : VenHw(Unknown Device:01)/HD
blk1   : VenHw(Unknown Device:80)/HD
blk2   : VenHw(Unknown Device:81)/HD
blk3   : VenHw(Unknown Device:ff)/HD
blk4   : VenHw(Unknown Device:80)/HD(Part1,Sig2AF02DE2)
blk5   : VenHw(Unknown Device:80)/HD(Part2,Sig2AF02DE2)
```

1. How many partitions are on device 80? _____
2. List the short names for partition 1 on device 80. _____
3. List the short names for partition 2 on device 80. _____
4. Assuming AIX has been installed, which partition contains the AIX boot loader? _____
5. Where would the boot image be found on device 80?

6. To boot from device 80, which partition should you add to the boot list? _____

Figure G-24. Let's Review (2)

AU147.0

Notes:

Unit Summary



- Changes to the startup process have been made in AIX 5.1 to support the Itanium architecture:
 - BIOS/EFI
 - EFI Boot Manager
 - EFI partitioning

Figure G-25. Unit Summary

AU147.0

Notes:

Appendix H. The System V Print Subsystem

What This Unit is About

This unit describes the features of the System V print subsystem which is now part of AXI 5.1.

What You Should Be Able to Do

After completing this unit, students should be able to:

- List two advantages of the System V print subsystem
- List two advantages of the AIX print subsystem
- Switch between the AIX and System V print subsystems
- Describe the process of printing a file using the System V print subsystem, including the following components:
 - Print service daemon (`lp sched`)
 - Printer configuration file
 - Terminfo database
 - Interface programs
 - Slow and fast filters
- Configure a local printer using the System V print subsystem and print to it
- Describe the steps to configure a remote System V printer

How You Will Check Your Progress

Accountability:

- In-line activities
- Checkpoint
- Machine exercises

References

- | | |
|-----------|--|
| Online | <i>AIX Guide to Printers and Printing, Chapter 6. System V Printer Configuration</i> |
| SG24-6018 | <i>Printing for Fun and Profit under AIX 5L</i>
(http://www.redbooks.ibm.com) |
| Online | <i>AIX Commands Reference</i> |

Unit Objectives

After completing this unit, students should be able to:

- List two advantages of the *System V print subsystem*
- List two advantages of the *AIX print subsystem*
- *Switch* between the AIX and System V print subsystems
- Describe the process of printing a file using the System V print subsystem, including the following components:
 - *Print service daemon*
 - *Printer configuration file*
 - *Terminfo database*
 - *Interface programs*
 - *Slow and fast filters*
- Configure a *local* printer using the System V print subsystem and print to it
- Describe the process of configuring a *remote* System V printer

Figure H-1. Unit Objectives

AU147.0

Notes:

AIX 5.1 Printing: What's New?

- System V print subsystem
- Changes to traditional AIX print subsystem
- Administration
- System management tools

Figure H-2. AIX 5.1 Printing: What's New?

AU147.0

Notes:

System V print subsystem

The classic AIX print subsystem was designed to combine the features of the System V and the Berkeley Software Distribution (BSD) printing standards, along with some unique features found only in AIX. However, these same features made the AIX print subsystem less compliant to widely used standards. With the development of AIX 5.1, a more standard print subsystem was needed. The System V print subsystem was chosen because of its wide use across many different UNIX systems.

The addition of System V printing allows system administrators with System V printing experience to easily transition to printing using AIX. Also, since the System V print subsystem is the de facto standard printing environment for UNIX systems, it will be easier for printer manufacturers to add support for AIX printing. System V printing also adds new features, such as enhanced security and support for using pre-printed forms.

Traditional AIX print subsystem changes

Both the traditional AIX print subsystem and the new System V print subsystem are available on Power-based systems. In order to support two print subsystems at the same time, some minor changes to AIX print subsystem file locations have been made (these will be described in “Print Commands Overview” on page 16).

Administration

A new user (`lp`) and group (`lp`) have been added to support System V printing.

System V print administrators need to belong to the `lp` group.

AIX print administrators need to belong to the `printq` group.

Users who belong to the `printq` group can add printer devices which can be used by either print subsystem.

System management tools

In AIX 5.1, the command line is the primary method for administering System V printing. The Web-based System Manager (WebSM) is the only menu-based tool for administering the System V print subsystem. In this release, the Web-based System Manager does not support all System V printer administration functions. Web-based System Manager support will be enhanced in later releases. The only SMIT function available for System V printing is to switch back to the AIX print subsystem.

The AIX print subsystem may be administered using either SMIT, the Web-based System Manager, or the command line.

AIX 5.1 Printing Environments

- Print directly to local printer device
- Print directly to a remote printer via a socket program
- *System V print subsystem*
- AIX print subsystem
- Infoprint Manager (or similar advanced print management system)

Figure H-3. AIX 5.1 Printing Environments

AU147.0

Notes:

Introduction

The slide gives an overview of the different approaches that can be taken to printing under AIX 5.1. In the next two slides, we'll compare System V printing to the traditional AIX print subsystem. The remainder of this unit will focus on using the System V print subsystem.

Print directly to a local printer device

This is the simplest form of printing. If your printer is directly attached to a serial or parallel port on the local machine, it is possible to print by just sending a file directly to the device. For example:

```
# cat /home/karlmi/myfile > /dev/lp02
```

In this approach, you lose the ability to serialize (spool) print requests. Only one user may print at a time. On the other hand, if a printer is being dedicated to one use, this may be a good solution. Examples might be logging to a printer or printing checks.

Print directly to a remote printer via a socket program

This is similar to printing to a device driver, except that in this case, you are sending the output to a program which makes a connection to the printer over the network.

Print using the System V print subsystem

In this environment, files to be printed are sent to the System V print service daemon (`lp sched`) using the `lp` or `lpr` commands. The print service daemon serializes the jobs so they will be printed in the order in which they were submitted. The print service may filter the file to format the data so that it matches the types of data acceptable to the printer. The print service then sends files, one at a time, to the interface program, which may do additional filtering before sending the file to the local printer driver or network printing application.

Print using the AIX print subsystem

In this environment, files to be printed are sent to the AIX print spooler daemon (`qdaemon`) using any of the AIX print commands (`enq`, `qprt`, `lp`, or `lpr`). The spooler daemon serializes the jobs. The spooler sends jobs, one at a time, to back-end programs that may filter the data and before sending it to the local printer driver or network printing application.

Print using IBM's Infoprint Manager (or similar advanced print management system)

Infoprint Manager provides serialization and filtering similar to the System V or AIX print subsystems. In addition, it adds extra capabilities of security, customization, and control not provided by either System V printing or AIX printing. For additional information, refer to the Infoprint Manager web site:

<http://www.printers.ibm.com/R5PSC.NSF/Web/ipmgraixhome>

System V Print Subsystem: Advantages

- Compatibility
- Availability of interface programs
- Security
- Support for forms
- Standard PostScript filters
- Long term strategy

Figure H-4. System V Print Subsystem: Advantages

AU147.0

Notes:

Compatibility

System administrators with experience in other UNIX variants that use System V printing will find it easy to manage printing under AIX's System V print subsystem.

Availability of interface programs

Many printer manufacturers provide interface shell scripts to support using their products under System V printing. Usually only minor modifications are required for individual UNIX variations. Because the AIX print subsystem is proprietary, an interface program written for another operating system cannot be used in the AIX print subsystem. It must be completely re-written. This has led to a limited number of printers supported under AIX. With the support of System V printing in AIX 5.1, it will be easier for manufacturers to include support for AIX printing.

Security

Controlling user access to printers can be an important issue. For example, you might need to limit access to the printer used to print checks. System V printing includes built-in capabilities for restricting user access to certain printers. Using the AIX print subsystem, the back-end program must be customized to restrict user access.

Support for forms

If you are printing to pre-printed forms, it's important that other users not be able to print while the expensive forms are loaded on the printer. The System V print subsystem provides a mechanism for mounting forms on printers and allowing or denying user access based on the form which is mounted. To provide this capability under AIX printing, you must create multiple queues and manage which queues are enabled while a form is mounted.

Standard PostScript filters

The System V print subsystem includes a number of filters for converting a number of different file formats to PostScript. Some formatting and page selection capabilities are also included.

Long term strategy

IBM's long term printing strategy for AIX is to maintain compatibility with other UNIX systems. This means that new features and functions will be added to the System V print subsystem in later releases, while the AIX print subsystem will be supported, but not enhanced in future releases.

AIX Print Subsystem: Advantages

- Powerful and flexible printer drivers
- System management tools
 - Limits fields and options validation
 - Easy printer customization
 - Single step print device and queue creation
 - Support for dial-in administration
- Customizable spooling subsystem

Figure H-5. AIX Print Subsystem: Advantages

AU147.0

Notes:

Powerful and flexible printer drivers

AIX printer drivers provide many printing options that can be easily controlled using command line options to the `qprt` command. Printer defaults can be easily managed using SMIT or the command line.

System management tools

The AIX print subsystem includes mature and powerful system management using either WebSM or SMIT, as well as the command line. System management tools for the System V print subsystem are less mature in this initial release. Some specific system management advantages using the AIX print subsystem are:

- Limits fields and options validation
Limits fields give the user or administrator a range of valid values for print options and will prevent the user from using an invalid value.
- Easy printer customization
Printers can be customized using menu selections or command line options. Under System V printing, customizing printers often requires a knowledge of shell programming.
- Single step print device and queue creation
Under System V printing, you must first add a print device and then create the print queue.
- Support for dial-in administration
Under AIX printing, administrators can dial-in and use the text-based SMIT interface to administer the print subsystem. The Web-based System Manager (which is currently the only non-command line interface for the System V print subsystem) would be hard to use over a dial-in connection.

Customizable spooling subsystem

The AIX print subsystem is specifically designed so that it can be used to serialize other types of jobs beyond just printing.

Software Packaging

- *Itanium-based* systems
 - System V print subsystem is installed as part of BOS (Basic Operating System) installation
 - AIX print subsystem is not available
- *Power* systems
 - Both print subsystems are installed as part of BOS installation
 - AIX print subsystem is enabled by default
- *System V* print subsystem filesets
 - bos.svprint.rte
 - bos.svprint.fonts
 - bos.svprint.hpnp
 - bos.svprint.ps
 - bos.terminfo.svprint.data
 - bos.msg.en_US.svprint
- *AIX* print subsystem filesets
 - bos.rte.printers
 - printers.*

Figure H-6. Software Packaging

AU147.0

Notes:

Itanium-based systems

On Itanium-based systems, only the System V print subsystem is available.

Power-based systems

Both print subsystems are installed. Only one subsystem can be active at a time. The AIX print subsystem is enabled by default.

System V print subsystem filesets

The slide shows the six filesets which comprise the System V print subsystem.

AIX print subsystem filesets

Default installation includes the base AIX print subsystem and only a few printers. When you add a printer that is not on the base install list, you will be prompted to install additional printer support from the installation media.

Switching between Print Systems

- *Only one* print subsystem can be active at a time
- Status
 - `switch.prt -d`
 - SMIT or WebSM
- Switching
 - `switch.prt -s AIX`
 - `switch.prt -s SystemV`
 - SMIT or WebSM
- What happens during the switch

Figure H-7. Switching between Print Subsystems

AU147.0

Notes:

Introduction

Either the AIX print subsystem or the System V subsystem can be active on a Power-base system, but not both at once.

Status

Use SMIT, the Web-based System Manager, or the `switch.prt` command to display the active print subsystem.

Switching

Use SMIT, the Web-based System Manager, or the `switch.prt` command to switch subsystems.

Switching from AIX to System V printing

When you switch from AIX to System V, the `switch.prt` command performs the following actions:

Step	Action
1.	Checks for active print jobs. If there are, exits with error message: <code>All print jobs must be terminated.</code>
2.	Stops <code>qdaemon</code> , <code>writesrv</code> , and <code>lpd</code> daemons.
3.	Modifies <code>/etc/inittab</code> so that the AIX daemons will not be started on next boot and so that the System V daemon will be started on next boot.
4.	Disables AIX printing SMIT menus as much as possible (some AIX printing menus are removed; others give an error message if you try to use them).
5.	Switches Web-based System Manager plug-ins.
6.	Changes lock files from AIX to System V.
7.	Removes AIX links and add System V links for the common commands (This will be described in "Print Commands Overview" on page 16).
8.	Launches the System V print daemon (<code>/usr/lib/lp/lpsched</code>).

Switching from System V to AIX printing

When you switch from System V to AIX, the `switch.prt` command performs the following actions:

Step	Action
1.	Checks for active print jobs. If there are, exits with error message: <code>All print jobs must be terminated.</code>
2.	Stops <code>lpsched</code> using the <code>lpshut</code> command.
3.	Modifies <code>/etc/inittab</code> so that <code>lpsched</code> will not be started on next boot and so that the AIX daemons will be started on next boot.
4.	Enables AIX printing SMIT menus.
5.	Switches Web-based System Manager plug-ins.
6.	Changes lock files from System V to AIX.
7.	Removes System V links and add AIX links for the common commands. (This will be described in "Print Commands Overview" on page 16).
8.	Launches the AIX print daemons.

Disabled queues or printers

If there are disabled queues or printers with waiting jobs, they will remain disabled if the print subsystem is switched. If the original print subsystem is reactivated, they will remain disabled. If the queue or printer is then enabled, the jobs will be printed.

User submits job using `enq` or `qprt` when System V printing is active

If a user submits a job using the AIX print commands when the System V print subsystem is active, the user will receive this error message:

```
Cannot awaken qdaemon (request accepted anyway)
```

If the AIX print subsystem is reactivated, the jobs will be queued and will print.

Print Commands Overview

- *Common commands* in /usr/bin
 - cancel
 - disable
 - enable
 - lp
 - lpq
 - lpr
 - lprm
 - lpstat
- AIX print subsystem active
 - common commands linked to /usr/*aix*/bin
- System V print subsystem active
 - common commands linked to /usr/*sysv*/bin
- *Man pages cover both versions*

Figure H-8. Print Commands Overview

AU147.0

Notes:

Introduction

Both print subsystems share a number of commands, but command behavior and option flags differ for the same command, depending on which subsystem is active. AIX handles this by linking commands from /usr/bin to either /usr/aix/bin or /usr/sysv/bin.

Man pages

The man page for each common command includes information about both versions of the command. You need to make sure you are reading the correct part of the man page for the print subsystem you are using.

AIX print subsystem command information

The portion of the man page pertaining to the AIX print subsystem version of the command begins with the following heading:

```
<command_name> Command on PowerPC Platform
```

System V print subsystem command information

The portion of the man page pertaining to the System V print subsystem version of the command begins with the following heading:

```
<command_name> Command on PowerPC and IA-64 Platforms
```

```
Note: This is a System V Print Subsystem command.
```

System V Printing Overview

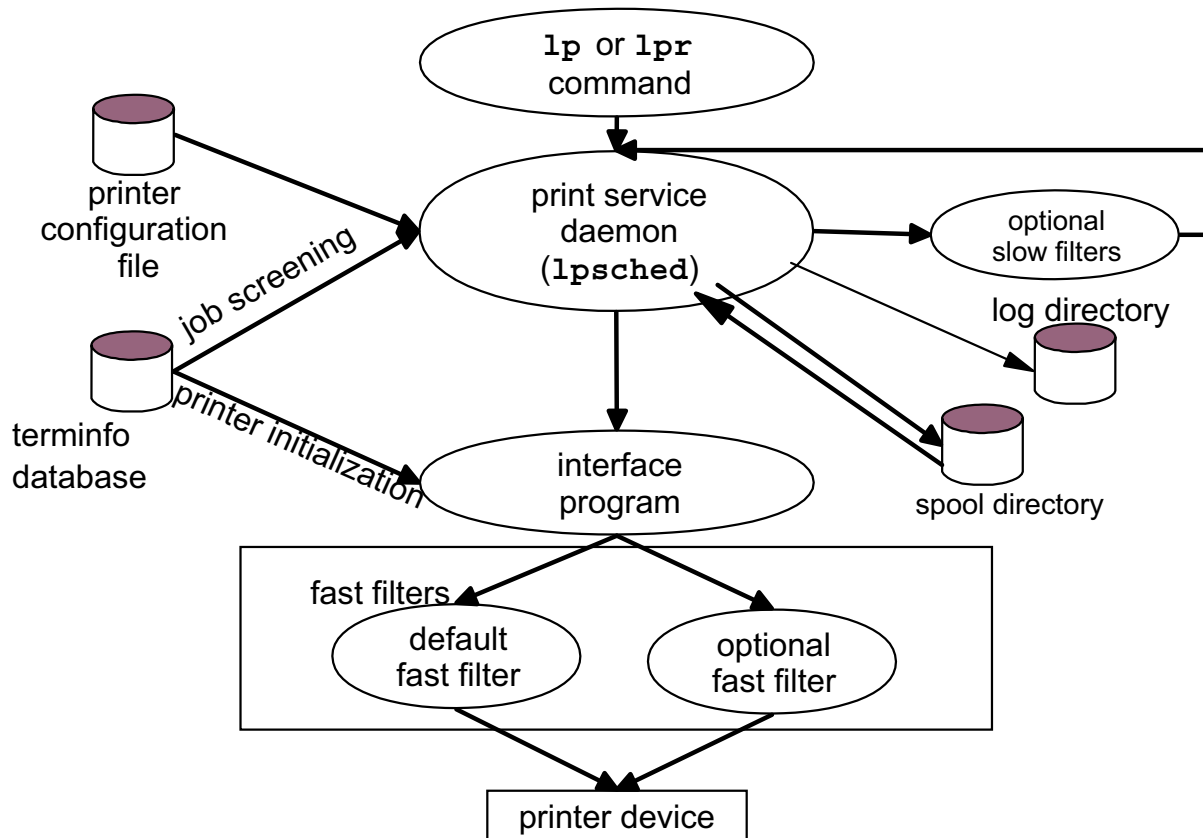


Figure H-9. System V Printing Overview

AU147.0

Notes:

Introduction

This slide provides an overview of the System V printing process. In the following pages, we'll provide additional details.

Job submission (`lp` or `lpr`)

Print jobs can be submitted using either the `lp` or `lpr` commands. Users can specify the printer or class of printers they want to use and a number of attributes which control how the job will be printed.

`lpsched`

`lpsched` is the print service daemon. It is started at boot time from the `/etc/inittab` file if the System V print subsystem is active.

Job screening (`lp sched`)

When `lp sched` receives a print job, the first task is to screen the job to see if it can be printed. This includes checking to see if the requested printer is accepting jobs and if the printer is capable of printing the type of job with the attributes requested by the user. `lp sched` uses the printer configuration file and information from the terminfo database for this purpose. If the job cannot be printed as requested by the user, it will be rejected.

Printer configuration file

When you create (or modify) a System V printer, printer configuration file is created (or modified). This file describes the printer, including:

- Content types this printer will accept
- Device name
- Source of interface script
- Printer type in the terminfo database
- Banner and form feed requirements

terminfo database

The terminfo database contains data describing characteristics of different printer types. This data is used in two ways. `lp sched` uses the data to determine if the job can be printed. Later in the process, the interface program uses this same information to initialize the printer.

Job spooling (`lp sched`)

If the job can be printed, `lp sched` assigns it a unique *request ID* and creates a request file (which describes the print job) in the spool directory.

The request ID is formed using the printer name and a unique number. For example, a request ID for a printer named `hqps` might be `hqps-01`. The request ID is used when requesting status or canceling a job.

Printers can be grouped into *classes*. If the user has requested printing to a class, `lp sched` will send it to the queue for the first available printer in the class that is capable of printing the job.

Filters

Filters are used by the System V print subsystem to perform three functions:

- Converting file content:
This could include tasks such as adding carriage returns to line feeds, mapping one set of control characters to another set, and so forth. For example, converting a simple text file to PostScript so that it can be printed by a PostScript printer.
- Interpreting special print modes requested by the user:
This could include print modes such as landscape page orientation, reverse page order, and so forth.
- Detecting printer faults.

There are two types of filters:

- Slow filters are filters that incur a lot of overhead and do not need to be connected to the printer while they run. `lp sched` runs slow filters in the background so that the printer is not tied up while they perform file conversion.
- Fast filters interact directly with the printer. They can control the printer and receive status back from the printer. Some fast filters also perform file conversion tasks like slow filters.

Filtering

`lp sched` determines which filters must be used based on:

- Printer type
- Content of the file to be printed (as specified by the user)
- Types of content the printer will accept (from the printer configuration file)
- Any special mode options requested by the user
- Capabilities of the available filters (registered using the `lpfilter` command)

`lp sched` may decide to use a combination of several filters. Slow filters are run directly by `lp sched`. Fast filters are run by the interface program, as directed by `lp sched`. Several filters may be piped together to achieve the desired file format.

Printing (`lp sched` and the interface program)

When a job moves to the top of a queue, `lp sched` passes the job to the interface program which has been defined for that printer.

interface program

The interface program is a shell script that manages the printer. When you create a System V printer, you specify which interface script you wish to use. Two interface scripts are provided with the System V print subsystem or you can write your own interface scripts. In addition, some printer manufacturers provide interface scripts specifically for their printers.

The interface script performs the following tasks:

- Initializes the printer port, if necessary, and printer hardware using terminfo data
- Invokes the fast filter to print a banner page, if required
- Invokes the fast filter to print requested number of copies of the file to be printed

Logging

`lp sched` is responsible for monitoring job status and updating files in the log directory.

System V Terminology

Term	Description	Examples
Printer device	The <i>device driver</i> this printer queue uses	/dev/lp01
System V printer	The printer <i>queue</i>	myprinter
Printer type	The <i>terminfo entry</i> used for this printer	PS
Content type	The <i>types of files</i> this printer can handle	postscript
Interface type	The <i>interface script</i> to use with this printer	/usr/lib/lp/model/PS
Class	A class is a <i>group of printers</i>	bldg5

Sample command to create a printer:

```
# lpadmin -p myprinter -v /dev/lp01 -c bldg5 -T PS \
-I postscript -m PS
```

Figure H-10. System V Terminology

AU147.0

Notes:

Introduction

One of the most confusing things about System V printing is the terminology. For example, many different things are referred to as “types”. The table in the slide describes some System V terms. The `lpadmin` command at the bottom of the slide shows how these terms are used when defining a System V printer.

Printer device

The term *printer device* usually refers to the actual printer device driver. Printer devices are created using `mkdev` and associated with a System V printer using the `-v device_name` flag to the `lpadmin` command.

In the example, the printer `myprinter` is configured to use printer device `/dev/lp01`.

System V printer

The term *System V printer*, or even just *printer* by itself, usually refers to the printer queue, which is defined using the `-p printer_name` flag to the `lpadmin` command.

In the example, the System V printer is named `myprinter`.

Printer type

The *printer type* associates a printer to an entry in the terminfo database. Use the `-T printer_type` flag to `lpadmin` to specify the printer type. The information in the terminfo database is used by the interface program to initialize the printer.

In the example, the printer type is `PS`, which is one of several terminfo entries for PostScript printers.

Content type

The content type identifies what kind of content the printer can handle. This can be a list of content types. For example, some laser printers can accept both PostScript and PCL (Printer Command Language). Use the `-I content_type` flag to `lpadmin` to specify printer content types.

In the example, the content type is `postscript`.

Content type of files to be printed

The user specifies the content type of a file to be printed using the `-T content_type` flag to the `lp` command. If a content type is not specified, the default content type is `simple`. (See the man page for `lpadmin` for a definition of the `simple` content type.) When you submit a print job, `lpsched` screens the job to see if the requested printer will accept the content of the file to be printed. If not, it checks to see if there are registered filters which can be used to convert the file to a content type the printer can handle. If the printer cannot accept the content directly and there are no registered filters which can convert the content, the print job will be rejected.

Interface or model type

The *interface* specifies which interface program will be used by the printer. You specify the interface using one of the following `lpadmin` flags:

Flag	Description
<code>-i interface_path</code>	Copy the script specified by <i>interface_path</i> (full path name) and use it as the interface script for this printer.
<code>-e printer_name</code>	Copy the interface script already defined for <i>printer_name</i> and use it for this printer.
<code>-m model</code>	Copy the file <i>model</i> in <code>/usr/lib/lp/model</code> and use it as the interface script for this printer.

In the example, `lpadmin` will copy the `/usr/lib/lp/model/PS` interface script to be used for `myprinter`.

Class

Printers can be grouped into *classes*. A class is an arbitrary group of printers. If a user submits a job to a class of printers, the print service will print it on the first available printer that can handle the job. Printers are added to a class using the `-c class_name` flag to `lpadmin`. If the class does not exist, it is created.

In the example, `myprinter` is added to class `bdg5`.

Let's Review (1)

1. What command is used to display which print subsystem is active? _____
2. When the System V print subsystem is active, /usr/bin/cancel is linked to _____.
3. The _____ or _____ commands can be used to submit print jobs to the System V print service.
4. _____ is the System V print service daemon.
5. _____ filters are executed by `lp sched` and do NOT interact with the printer.
6. _____ filters are executed by the interface program and DO interact with the printer.
7. The printer type associates a printer with an entry in _____.

Figure H-11. Let's Review (1)

AU147.0

Notes:

Adding a System V Printer

1. Create printer device
 - `mkdev`
 - SMIT
 - WebSM
2. Create System V printer
 - `lpadmin`
 - WebSM

Figure H-12. Adding a System V Printer

AU147.0

Notes:

Introduction

Creating a System V printer is done in two steps:

- Creating the printer device
- Creating the System V printer

Printer Device Overview

- Exact match of printer to printer device is *not critical*
- List *defined* printer devices


```
# lsdev -Cc printer
lp0 Available 00-00-0P-00 Lexmark Optra Color 1200 printer
```
- List *supported* printer devices


```
# lsdev -Pc printer
...
printer lexOptraC1200 parallel Lexmark Optra Color 1200 printer
printer lexOptraC1200 rs232 Lexmark Optra Color 1200 printer
...
```
- Printer *device attributes are not used* when printing from AIX or System V print subsystem


```
# splp lp0
device = /dev/lp0      (+ yes      ! no)
CURRENT FORMATTING PARAMETERS (ignored by qprt, lpr, and lp commands)
Note: -p + causes the other formatting parameters to be ignored.
-p !   pass-through?           -c +   send carriage returns?
-l 64  page length (lines)     -n +   send line feeds?
-w 80  page width (columns)    -r +   carriage rtn after line feed?
-i 0   indentation (columns)  -t !   suppress tab expansion?
-W !   wrap long lines?       -b +   send backspaces?
-C !   convert to upper case? -f +   send form feeds?
CURRENT ERROR PROCESSING PARAMETERS
-T 300 timeout value (seconds) -e !   return on error?
```

Figure H-13. Printer Device Overview

AU147.0

Notes:

Introduction

Printer devices can be used by either print subsystem. Printer devices may be added using SMIT, the Web-based System Manager, or the command line (`mkdev`).

Connecting printers

Local printers may be connected in one of two ways: serial or parallel. Network-attached printers may be connected directly to the network, or they may be connected to a remote print server host that is accessed over the network.

Choosing a printer device

The printer device that you choose determines the buffer size and some timing parameters for the device driver. However, it is not critical that you find an exact match between your printer and the printer device driver software. You should choose a printer device that is:

- A similar kind of printer, for example: laser, ink jet, and so forth
- Similar in speed to your actual printer

In the example in the visual, lp0 has been configured using the parallel port and the Lexmark Optra Color 1200 printer device driver; however, the physical printer is actually a Canon Bubble Jet. These printers are similar enough that the printer device operates correctly for the Canon printer.

Listing printer devices

Use `lsdev -Cc printer` to list printer device which have already been defined.

Use `lsdev -Pc printer` to list supported printer devices. If you do not find an appropriate device for the printer you want to add, you may need to install additional printer software. Use `smit install_package` fastpath.

Printer device attributes

You can list printer device attributes using `lsattr -El printer_device_name` or using `splp printer_device_name` as shown in the slide. However, the printer device attributes shown (such as page length, page width, indentation, and so forth) are only used when printing by sending a file directly to the printer device. If you are using the System V or AIX print subsystem, the printer device is put into pass-through mode. The print subsystem now controls how the printer will operate.

For System V printers, defaults for these attributes are usually defined by the terminfo entry (printer type). Depending on the printer, it may be possible to override the defaults when submitting a print job using the `-o` flag to the `lp` command.

Creating a Local System V Printer

```
# lpadmin -p myprinter -v /dev/lp0 -T bj-300
```

The diagram consists of a cylinder icon labeled 'printer configuration file' with an arrow pointing to an oval labeled 'print service daemon (lpsched)'.

- Printer configuration file

```
# cat /etc/lp/printers/myprinter/configuration
Banner: on:Always
Content types: simple
Device: /dev/lp0
Interface: /usr/lib/lp/model/standard
Printer type: bj-300
Modules: default
Form feed: on
```

Figure H-14. Creating a Local System V Printer

AU147.0

Notes:

Introduction

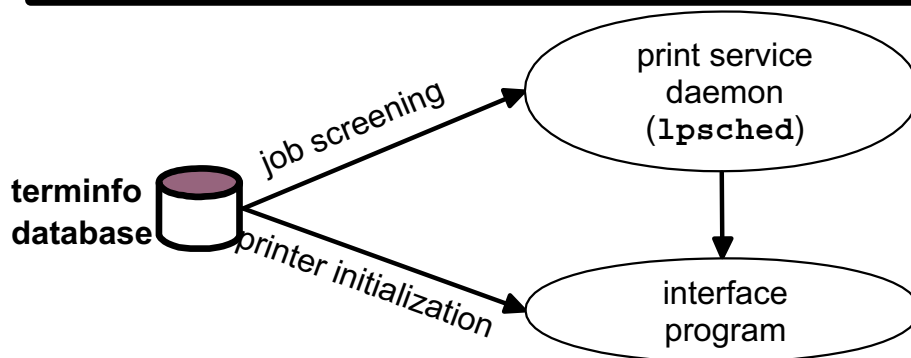
System V printers are added, or modified, using `lpadmin` or the Web-based System Manager. This slide shows a printer created using `lpadmin`. See the `lpadmin` man page for a complete description of the many options to this command.

Printer configuration file

The printer configuration file is created by `lpadmin` when you create a printer. This file defines the printer to `lpsched`.

In the example, the printer name, printer device and printer type were specified. The other attributes in the printer configuration file were not specified, so the defaults were used.

Printer Types



- terminfo *source* for printers:
/usr/share/lib/terminfo/svprint.ti
- *Compiled* terminfo file for a Canon Bubble Jet (printer type bj-300):
/usr/share/lib/terminfo/b/bj-300
- *To compile* terminfo source:
tic svprint.ti
- *To view* bj-300 terminfo entry:
infocmp bj-300

Figure H-15. Printer Types

AU147.0

Notes:

Introduction

System V printer types are defined in the terminfo database. Printer types are similar in function to the virtual printer definition files used by the AIX print subsystem. Unlike AIX virtual printer definitions, one terminfo entry may be used for a number of different System V printers.

Purpose

Printer type information is used by `lp sched` to perform job screening and by the interface program to initialize the printer.

Organization

Terminfo entries are binaries which are compiled from terminfo source files. The database resides at `/usr/lib/terminfo`. By convention, source files reside in `/usr/share/lib/terminfo` and are named `*.ti`. For example, the source file for the System V printer types supplied with AIX 5.1 is `svprint.ti`. Each compiled terminfo entry is a separate file which resides in `/usr/lib/terminfo/X`, where `X` is the first letter of the terminfo name. For example, the terminfo entry for printer type `bj-300`, is `/usr/lib/terminfo/b/bj-300`.

Commands

Use the `tic` command to compile a terminfo source file. Use `infocmp` to display a terminfo entry, or to compare two entries.

Contents of a terminfo entry

Terminfo entries contain information about the printer. For example, a terminfo entry might include:

- Printer characteristics, such as:
buffer size, number of pins in the print head, vertical and horizontal resolution, and so forth
- Printer control characters, such as the characters required to perform:
carriage return, form feed, line feed, set underline mode, as so forth

No printer type

If you do not specify a printer type, it will default to `unknown`. Depending on how you are using the printer, this may not be a problem. It does mean that

- Your printer will not be initialized by the interface program
- Any `-o` options on the `lp` command line (such as `-o cpi`, `-o width`, `-o length`, and so forth) cannot be used.
- Some simple control characters may not function correctly.

The exception to this would be if you have a printer specific interface script which generates the command sequences internally in the script without consulting terminfo.

Listing available printer types

You can view available printer types in the `/usr/share/lib/terminfo/svprint.ti` file. Or if you are using the Web-based System Manager, you will be presented with a list of printer types. You can also view the `/usr/lpp/sysmgt.websm/inst_root/var/websm/data/model.stz` file. This is a stanza-format file which associates a printer model with terminfo entry (printer type), interface script, and content type.

Choosing a printer type

It is not critical that you find an exact match for your printer model, but it should be a similar kind of printer. Here are some guidelines:

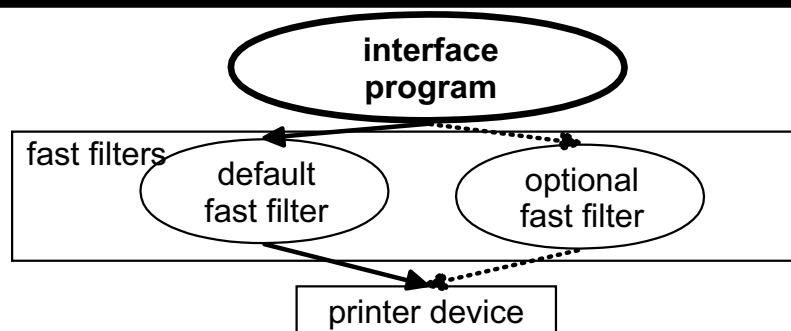
- If you are using a PCL printer, you can usually use the `hplaserjet` printer type. Set the content type to `pcl`.
- If you are using a PostScript printer, set the content type to `postscript` and choose one of the following printer types:
 - For serial connected printers: `PS`, `PSR`, or `PS-r`
 - For parallel connected printers: `PS-b` or `PS-br`
 - Use the `-r` types (`PS-r` or `PS-br`) to print pages in reverse order, with the banner page last.

In the case of these PostScript printer types, `lpsched` uses the printer type to choose the correct fast filter.

More information

If you believe that you need to create a new terminfo entry for your printer, see *Printing for Fun and Profit under AIX 5L* (Redbook), *Appendix C. Virtual printer colon files and System V terminfo*.

Interface Programs



- Available interface programs
 - /usr/lib/lp/model/standard
 - /usr/lib/lp/model/PS
 - Manufacturer or user created
- Copied to: /etc/lp/interfaces/printer_name
- Functions
 - Initialize printer port
 - Initialize printer hardware
 - Print banner, if requested
 - Print number of copies requested
 - Return exit status to `lp sched`

Figure H-16. Interface Programs

AU147.0

Notes:

Introduction

The interface is responsible for performing the functions as listed in the slide:

- Initialize the printer port - using `stty`.
- Initialize the printer hardware - using commands from `terminfo`.
- Print the banner page.
- Print copies of the print job - using `filter`.
- Handle any printer errors from the filter and return exit status (success or failure) back to `lp sched`.

lpsched

`lpsched` calls the printer's interface program for local print requests. `lpsched` passes information to the interface program. Some of the information sent to the interface program includes:

- Terminfo entry to use
- Fast filter to use
- Character set (optional)
- Number of copies
- Files to print

Note: Interface programs are sometimes mistakenly referred to as “print drivers”.

Available interface programs

The System V print subsystem includes two interface scripts: `standard` and `PS`. Some manufacturers supply printer specific interface scripts, or you can create your own.

Choosing an interface program

Unless you have a manufacturer's interface for your printer, in general, you can use the `PS` interface for PostScript printers and the `standard` interface for all other printers. If you do not specify an interface, `lpadmin` will select the `standard` script.

Administrative concerns

When a printer is created, a copy is made of the interface script for that printer. For example, if printer `myprinter` is defined to use the `PS` interface, the `/usr/lib/lp/model/PS` file will be copied to `/etc/lp/interfaces/myprinter`.

If you need to modify a printer's interface script, modify the copy in `/etc/lp/interfaces`. If you want to change the template for all future printers created, modify the source file.

More information

If you need to create a custom interface script, you can use the `standard` or `PS` script as a template. For more information, see:
Printing for Fun and Profit under AIX 5L (Redbook),
Chapter 4. System V advanced printing.

Spool and Log Files

```

$ lp -d canon /etc/motd
request id is canon-10 (1 file)
$ lp -d canon -c /etc/passwd
request id is canon-11 (1 file)
$ su -
# cd /var/spool/lp/tmp/kca48
# ls -l
total 5
-rw-r--r--  1 root    system      11 Mar 16 16:18 .SEQF
-rw-----  1 lp      lp          87 Mar 16 16:08 10-0
-rw-----  1 lp      lp         109 Mar 16 16:18 11-0
-rw-----  1 lp      lp         366 Mar 16 16:18 11-1
-rw-----  1 lp      lp          88 Mar 16 13:43 8-0
# cat 10-0
C 1
D canon
F /etc/motd
O locale=C flist='/etc/motd:880'
. . .
# cat 11-0
C 1
D canon
F /var/spool/lp/tmp/kca48/11-1
O locale=C flist='/etc/passwd:366'
. . .

```

Figure H-17. Spool and Log Files

AU147.0

Notes:

Introduction

Each time a user sends a print job to a printer, the print service creates one or more files in the spool directory (`/var/spool/lp/tmp/<hostname>`) that describe the job request. These files remain in this directory while the job is in the queue waiting to be printed. When the job is finished printing, information in the files is appended to the log file `/var/lp/logs/requests` and the files are removed from the spool directory.

Copying files

Normally, if you send a file to the print service using the `lp` command, the print service does not copy your file to the spool directory, but instead just reads from the original. This means that if you delete the original copy after you submit the print request, but before it is printed, the print request will fail. In some circumstances a copy is created. It is also possible to request that the print service create a copy of the print file in the spool directory before printing. Files will be copied under the following circumstances:

- The job is submitted using `lp -c`. (The default for the `lp` command is NOT to copy.)
- The job is submitted using `lpr`, without the `-s` flag. (The default for the `lpr` command IS to copy.)
- The job is received from a remote system. (In this case, the file does not exist on the print server system, and so must be copied.)
- Copying files has been enabled as the default by issuing the `lpadmin -O copy` command.

Note: this flag sets the value of the `copy-files` parameter in the `/etc/default/lp` file to `on`.

Files in `/var/spool/lp/tmp/<hostname>`

This directory contains the following files:

File	Description
<code>.SEQF</code>	This file is used to keep track of the next job number.
<code>X-0</code>	These files are the actual request files, where <code>X</code> is the job number. Notice that the printer name is not part of the request file name, but rather is stored within the file.
<code>X-N</code>	If files are being copied to the spool directory, there may be additional files (<code>X-1</code> , <code>X-2</code> and so forth) that contain the actual data to be printed.

Example

In the example in the slide on page 35, job 10 was created without copying while copying was requested for job 11. Excerpts from the request files are shown in the slide. In the request file:

- C indicates the number of copies requested.
- D indicates the name of the printer.
- F indicates the name of the file to print.
- O indicates additional information. In this case, the locale and the name and permissions of the original file (`flist=`).

Notice that for job 11, `/var/spool/lp/tmp/kca48/11-1` is the file to print. This would be a copy of the original file, in this case: `/etc/passwd`.

Log file (`/var/lp/logs/requests`)

Each time a print job completes, information from the request file is appended to `/var/lp/logs/requests`. This file will grow until it fills the file system unless you manage it.

Syntax reference

For a complete description of the syntax used in the request and log files, see:
AIX 5L Version 5.1 Guide to Printers and Printing,
Chapter 6. System V Printer Configuration.

File system size

If your machine is a print server for remote clients or if users are routinely copying files to the spool directory, you may need to increase the size of the var file system.

Alternatively, you could create a new file system dedicated to print spooling and link `/var/spool/lp/tmp` to the new file system.

Managing Printers

- `enable printer`
start printing
- `disable [-c] [-r reason] [-W] printer`
stop printing
- `accept printer`
start queueing
- `reject [-r reason] printer`
stop queueing
- Creating and enabling a new printer
`mkdev`
`lpadmin`
`accept`
`enable`

Figure H-18. Managing Printers

AU147.0

Notes:

Introduction

The System V print subsystem allows you to control queueing and printing separately.

`enable / disable`

The `enable` and `disable` commands control whether jobs in the queue will be printed. For example, if you need to perform service on the physical printer, or need to mount a form, use `disable` to stop printing. This allows users to continue to submit jobs, but nothing is printed while you perform service on the printer. When service is complete, use `enable` to restart printing and jobs from the queue are again printed.

disable syntax

The table shows options to **disable**.

Option	Description
-c	Cancel any requests currently printing on any of the designated printers. Cannot be used with -w .
-r <i>reason</i>	Assign a reason for disabling the printers. The reason is reported by lpstat -p . <i>reason</i> must be quoted if it includes spaces.
-w	Wait for any currently printing job to complete before disabling printers. Cannot be used with -c .

accept / reject

The **accept** and **reject** commands control whether the printer adds print requests to the printer queue. For example, use **reject** to stop queueing for a printer if you need to change queue parameters. Any jobs remaining in the queue are printed. When the queue is empty, make the desired changes and then use **accept** to restart the queue using the new parameters.

reject syntax

The table shows options to **reject**.

Option	Description
-r <i>reason</i>	Assign a reason for rejecting requests. The reason is reported by lpstat -a . <i>reason</i> must be quoted if it includes spaces.

Creating a new printer

After you create a new System V printer, you must remember to use **accept *printer*** to turn on queueing and **enable *printer*** to turn on printing.

If you create a class, you must turn on queueing to the class using **accept *class***.

Using the Print Service

- **Submit** print jobs
`lp -d dest [print-options] file_list`
- **Modify** print jobs
`lp -i request-id [print-options]`
- **Cancel** print jobs
`cancel request-id-list`
`cancel printer`
`cancel -u user-list [printer-list]`
- **Check** status
`lpstat [flags] [object-list]`

Figure H-19. Using the Print Service

AU147.0

Notes:

Introduction

This slide summarizes the commands (accessible by any user) to utilize the System V print service. See the respective man page for detailed information on the many options available.

Submit print jobs

The basic syntax to submit jobs is shown in the slide. *dest* can be either a printer or a class of printers. If a class is specified, the print service will choose the first available printer in the class that can handle all the print options requested.

Modify existing print jobs

This form of the `lp` command can be used to change the options for a previously submitted print request. You can get the request-id using `lpstat`. If the job has not

yet started printing, the changes will be accepted, if the printer can handle them. If the job has started printing, it will be stopped and restarted from the beginning. If the job has finished, the change is rejected.

Cancel print jobs

The `cancel` command is used to cancel print jobs, as shown in the table. Regular users can only cancel their own jobs. `root` or `lp` can cancel any user's jobs.

Command	Description
<code>cancel request-id-list</code>	Cancel the jobs specified. You can get the request-ids using <code>lpstat</code> .
<code>cancel printer-list</code>	The user's currently printing job for the requested printer will be cancelled.
<code>cancel -u user-list [printer-list]</code>	Cancel all jobs for specified users. If <code>printer-list</code> is specified, only cancel the users' jobs for the listed printers.

Check status

Use `lpstat` to check status. There are many options. Several of the most useful ones are shown in the table. Options can be combined to get the output you need. If the `list` argument is omitted, `lpstat` reports on all of that type of object. If you have many printers, omitting `list` may make the output of `lpstat` unreasonably long.

Option	Description
<code>-o [list] [-1]</code>	Reports the status of print requests. <code>list</code> can be printers, classes, or request-ids. if <code>-1</code> is used, additional status for each job is reported.
<code>-p [list] [-D] [-1]</code>	Reports printing status: <ul style="list-style-type: none"> • Enabled/disabled • What is currently printing • Device status (available/defined) With <code>-D</code> , a brief description of each printer is included. With <code>-1</code> , a full description is included. <code>list</code> is a list of printers.
<code>-a [list]</code>	Reports queue status (accepting/rejecting). <code>list</code> can be printers or classes.
<code>-u [list]</code>	Reports status for users in <code>list</code> .
<code>-t [list]</code>	Reports total status (similar to combined output of <code>-o</code> , <code>-p</code> , and <code>-a</code>).

Let's Review (2)

1. A _____ is a group of printers.
2. Use the _____ command to enable a printer to begin accepting print requests.
3. Use the _____ command to enable a printer to begin printing print requests.
4. AIX 5.1 includes two interface programs _____ and _____.
5. If a user wants the print service to copy her file to the spool directory, she should use the _____ option to the lp command.
6. Use _____ to display status of outstanding print requests.
7. The System V print log file is _____.

Figure H-20. Let's Review (2)

AU147.0

Notes:

Using Filters

- **Purpose**
 - Convert file content
 - Interpret special print modes
 - Handle printer faults
- **Filter types**
 - Slow filters run in background
 - Fast filters interact with printer
- **Using filters**
 - Filters must be registered
 - Printer content must be set
 - File content must be set
- **Managing filters**
 - Filter definition files
/etc/lp/fd/*.fd
 - Registering a filter
`lpfilter -f filter_name -F filter_definition_filename`
 - Listing filters
`lpfilter -f [filter_name | all] -l`
 - Removing a filter
`lpfilter -f filter_name -x`

Figure H-21. Using Filters

AU147.0

Notes:

Purpose

We'll start by reviewing the purpose of using filters. Filters are used by the System V print subsystem to perform three functions:

- **Converting file content:**
For example, converting a simple text file to PostScript so that it can be printed on a PostScript printer.
- **Interpreting special print modes requested by the user:**
This could include print modes such as landscape page orientation, reverse page order, and so forth.
- **Detecting printer faults:**
Printer faults include such things as printer out of paper or printer off line.

Filter types

There are two types of filters:

- Slow filters are filters that incur a lot of overhead and do not need to be connected to the printer while they run.
- Fast filters interact directly with the printer. They can control the printer and receive status back from the printer, Some fast filters also perform file conversion tasks like slow filters.

Using filters

In order for filters to work correctly, a number of things need to be taken care of:

- Filters must be registered
While the System V print subsystem includes a number of useful filters, the print service will not use them until they are registered. Use `lpfilter` to register a filter.
- Printer content must be set
The printer content types must be set correctly (using `lpadmin -I content_type_list`) so that the print service knows what types of files the printer can accept without filtering. If not set, printer content type defaults to `simple`.
- File content must be set
If a print job contains content other than `simple`, the file content must be set when the print job is submitted (using `lp -T content_type`) so that the print service knows what the file's content type is. If not set, the print service assumes the file content is `simple`.

Filter definition files

The filter definition file describes the filter's capabilities and how it can be used. The following is a partial list of the information that can be included in this file.

Item	Description
Command	File name of the filter program
Input types	Content this filter will accept as input
Output types	Content this filter can provide as output
Printer types	Printer types that may use this filter
Printers	Normally a filter would work with all printers that accept the output type, however you can restrict which printers may use a filter if this is desirable

Standard filters

The AIX 5.1 System V print subsystem includes a number of filters. Filter definitions for these filters are `/etc/lp/fd/*.fd`.

Managing filters

Use the `lpfilter` command to register a filter. For example, the `dpost` filter is used to convert troff files to PostScript. The `dpost` filter definition file is `/etc/lp/fd/dpost.fd`. To register the `dpost` filter, enter:

```
# lpfilter -f dpost -F /etc/lp/fd/dpost.fd
```

Registered filter definitions are stored in the `/etc/lp/filter.table` file, however, you should not directly edit this file. Use `lpfilter` to manage the registered filters.

If you wish to change how a filter is used, edit the filter definition file and re-enter the `lpfilter` command.

To list a registered filter (for example to list the `dpost` filter):

```
# lpfilter -f dpost -l
```

To list all registered filters:

```
# lpfilter -f all -l
```

To remove a registered filter (for example to remove the `dpost` filter):

```
# lpfilter -f dpost -x
```

More information

For more information on managing filters or creating your own filters, see:

- *Printing for Fun and Profit under AIX 5L (Redbook), Chapter 4. System V advanced printing*
- *AIX 5L Version 5.1 Guide to Printers and Printing, Chapter 6. System V Printer Configuration*
- man page for `lpfilter` command

Using Forms

- *Registering* forms with the print service
- *Requesting* a form for a print job
- *Alerting* the operator to mount a form
- *Mounting* a form
- *Unmounting* a form
- *Controlling access* to forms
- *Displaying information*

Figure H-22. Using Forms

AU147.0

Notes:

Introduction

A form is a pre-printed sheet of paper which can be loaded into a printer in place of plain paper. Some examples are company letterhead, checks, invoices, receipts, and so forth.

The System V print subsystem facilitates printing to forms by providing the functions shown in the slide. We will summarize the procedures for using forms here. For complete details, see:

Printing for Fun and Profit under AIX 5L (Redbook)

Note: The print service does not position print output on a form; this is the responsibility of the application.

Registering forms with the print service

Forms are managed in a similar way as filters. The first step to using forms is to create a form definition file and register the form with the print service. The definition file describes the form, including page length, page width, number of pages, line pitch, character pitch, alignment pattern, and so forth. The alignment pattern is sample output that can be used to correctly position the form when it is mounted. Once you have created the definition file, register the form using:

```
# lpforms -f form_name -F form_definition_file
```

Requesting a form for a print job

Users can request that a print job use a particular form using the `-f` flag to `lp`, for example:

```
# lp -f form_name -d printer file_to_print
```

The print job will be queued, but will not be printed until the form has been mounted on the requested destination.

Alerting the operator to mount a form

To configure the print service to notify the operator when jobs requesting a form have been queued:

```
# lpforms -f form_name -A alert_type [-Q number] [-W interval]
```

The table explains the options.

Option	Description
<code>-A <i>alert_type</i></code>	Send alerts to user <code>lp</code> when <i>form_name</i> is requested. <i>alert_type</i> can be: <ul style="list-style-type: none"> • <code>mail</code>: Send mail to user <code>lp</code>. • <code>write</code>: Send message to terminal where <code>lp</code> is logged in. • <code>none</code>: Do not alert. • <code>shell-command</code>: Execute named command. • <code>quiet</code>: Do not send any more messages for current form request.
<code>-Q <i>number</i></code>	Send alerts after <i>number</i> of form requests have accumulated in the queue. Default is one.
<code>-W <i>interval</i></code>	Repeat alert every <i>interval</i> minutes. Default is zero, which indicates alerting once.

Mounting a form

Mounting a form lets the print service know that the specified form is now loaded onto the printer. Any queued jobs using that form can now proceed. Use the following steps to mount a form:

Step	Action
1.	Disable the printer. # <code>disable printer_name</code>
2.	Physically load the form in the printer
3.	Inform the print service that the form is ready: # <code>lpadmin -p printer -M -f form_name [-a] [-o filebreak]</code>

Option	Description
<code>-M -f form_name</code>	Informs print service that <i>form_name</i> is mounted.
<code>-a</code>	Prints the alignment pattern (if defined in the form definition file). The operator can then adjust the form and press <Enter> for another alignment pattern. This can be repeated as many times as needed to get the alignment right. Type <q> to quit printing alignment patterns.
<code>-o filebreak</code>	Inserts a form feed at the end of each alignment pattern. If not specified, no form feed is added.

Step	Action
4.	Align the form, if required. (See <code>-a</code> option in table above.)
5.	Physically load the form in the printer
6.	Enable the printer. Queued jobs for this form will now be printed. # <code>enable printer_name</code>

Unmounting a form

To unmount a form, follow these steps:

Step	Action
1.	Disable the printer. # <code>disable printer_name</code>
2.	Physically remove the form from the printer.
3.	Inform the print service that the form is removed: # <code>lpadmin -p printer -M -f none</code>
4.	Enable the printer. # <code>enable printer_name</code>

Controlling access to forms

Use the following commands to control which users can submit print requests for a particular form:

```
# lpforms -f form_name -u allow:user_list
```

```
# lpforms -f form_name -u deny:user_list
```

Where *user_list* is a comma-separated list of AIX users. The allow and deny lists function in the same way as the cron.allow and cron.deny files. See the `lpforms` man page for details.

Displaying form information

Two commands are available for displaying information about a form:

- For user root or lp

```
# lpforms -f form_name -l
```

This command displays all the information in the form definition file, including the alignment pattern and user allow/deny lists.
- For any user

```
# lpstat -f form_name -l
```

This command displays information in the form definition file, excluding the alignment pattern and user allow/deny lists.

Planning a Local System V Printer

- Printer name
- Printer device
- Printer type
- Class
- Content types and filters
- Alerts
- Banner pages
- Forms
- Access policy

Figure H-23. Planning a Local System V Printer

AU147.0

Notes:

Introduction

This slide lists a number of issues which need to be considered when defining a local System V printer.

Printer name

The printer name should make it easy for users to identify the printer. You can use any name you wish, with the following restrictions:

- The name must be a valid file name for the file system you are using
- The name cannot begin with a dash (-), although a dash can be used in other positions in the name

Printer device

If you are configuring a local printer, you must decide what printer device driver to use. If AIX or your printer manufacturer does not provide a printer device specific to your printer, you can probably use a printer device for a similar printer. This is discussed in “Choosing a printer device” on page 28.

Printer type

This is discussed in “Choosing a printer type” on page 32. In general, it is not critical that you find an exact match for your printer model, just that it be a similar kind of printer.

Class

Do you want to include this printer in an existing printer class or define a new class? Classes can give users flexibility. By printing to a class of printers, any of which meet their requirements, they may be able to get their job printed more quickly.

Content types and filters

Questions to consider are:

- What types of content will you be sending to this printer?
- What types of content can this printer accept?
- If you have a need to print content that the printer can't handle, are there filters available and registered?

These issues are discussed in “Using Filters” on page 43.

Alerts

When there is a problem with a printer, how should the print service alert the print administrator? Printer alerts are configured on a per printer basis using the `-A` flag to `lpadmin`. The alert options are similar to the options for form alerts discussed in “Alerting the operator to mount a form” on page 47. See the `lpadmin` man page for complete details.

Forms

Do you require any special forms? If so, consider these questions:

- How should the print service alert the print administrator?
- Do you need to control access to any forms?

These issues are discussed in “Using Forms” on page 46.

Banner pages

The System V print subsystem allows you to control the printing of banner pages. The default is to print a banner page with every print job. Users can request no banner page. The print service will reject this request unless enabled to allow skipping the banner using the `-o nobanner` option to `lpadmin`. The table summarizes the relevant command options.

Option	Description
<code>lpadmin -o banner</code>	Banners are required (default).
<code>lpadmin -o nobanner</code>	Users are allowed to request that the banner not be printed.
<code>lp -o nobanner</code>	Request print job be printed without a banner.

Access policy

Do you need to control access to a printer? The System V print subsystem allows you to control access to printers using an allow-list, a deny-list, or both. These lists can be created using the `-u allow:user-list` or `-u deny:user-list` options to `lpadmin` and function similarly to the `cron.allow` and `cron.deny` files. See the `lpadmin` man page for complete details.

System V Network Printing

- Print server:
Serving print requests from remote clients using LPD protocol (RFC 1179)
- Print client:
Printing to remote *LPD printers or servers*
 - network attached printers running LPD
 - local printers on a server running LPD
- `lpNet`
- `/etc/lp/Systems` and `lpssystem`
- Printing to *JetDirect-attached printers*

Figure H-24. System V Remote Printing

AU147.0

Notes:

Introduction

This slide provides an overview of the network printing capabilities of the System V print subsystem. More details about configuring a AIX system as a print server or print client are provided in the next two slides.

Print server

The System V print subsystem can be configured so that a locally attached printer on your system (the *print server*) can be used to print requests from remote machines (the *print clients*) which are running the LPD protocol as defined in Request for Comments (RFC) 1179.

Print client

You can configure the System V print subsystem to print to any network destination that supports LPD as defined in Request for Comments (RFC) 1179. A network destination in this sense can be a:

- Printer (directly connected to the network) that is running LPD
- System (with locally attached printers) that is running LPD

lpNet

The **lpNet** daemon is used by both network printing clients and servers. **lpNet** is automatically started by **lp sched**.

On client machines, **lp sched** sends the print request to **lpNet** for transmission to the print server. No formatting or filtering is done on the client side.

On server machines, **lpNet** receives the remote print request and sends it to **lp sched**. If the request can be printed, **lp sched** processes the print request as it would any local request. Printer type, filtering for content and other formatting is all done on the server.

/etc/lp/Systems and lp system

Remote systems with which you want the print service to communicate (client or server) must be registered in the `/etc/lp/Systems` file. The **lp system** command is used to manage entries in this file.

JetDirect-attached printers

The System V print subsystem also supports printing to printers that attach to the network using the Hewlett-Packard JetDirect interface. Configuring the print subsystem for JetDirect printers is not included in this class. See *Printing for Fun and Profit under AIX 5L* (Redbook) for additional information about this capability.

Configuring a Network Print Server

- Register remote systems

```
# lpsystem [-T timeout] [-R retry] [-y comment] system_name
```

- Grant or deny access

```
# lpadmin -p printer -u allow:user_list | -u deny:user_list
```

Figure H-25. Configuring a Network Print Server

AU147.0

Notes:

Introduction

Configuring a network print server is done in two steps:

- 1) Registering the remote systems (clients) allowed to use the server
- 2) Granting or denying access to individual remote users or groups of remote users

Register remote systems

Use the `lpsystem` command to register the remote systems.

lpsystem syntax

```
# lpsystem [-T timeout] [-R retry] [-y comment] system_name
# lpsystem -l system_name
# lpsystem -r system_name
```

The table explains the usage of the various options.

Option	Description
<i>system_name</i>	This parameter specifies the name or IP address of the remote system. <i>System_name</i> can be * to allow access from any system.
-T <i>timeout</i>	This option specifies the length of time the print service will allow a network connection to be idle. If idle time exceeds <i>timeout</i> , the connection is dropped. It will be re-established if there are more requests. <i>timeout</i> can be: <ul style="list-style-type: none"> • <i>n</i>: never timeout. This is the default. • 0: timeout immediately. • <i>N</i>: timeout after <i>N</i> minutes.
-R <i>retry</i>	This option specifies the length of time to wait to re-establish a connection if the connection was abnormally dropped. <i>retry</i> can be: <ul style="list-style-type: none"> • <i>n</i>: do not retry until there is more work. • 0: try to reconnect immediately. • <i>N</i>: try to reconnect after <i>N</i> minutes. The default is 10.
-y <i>comment</i>	This option allows you to add a free form comment. The comment must be quoted if it contains spaces.
-l <i>system_name</i>	This option lists the parameters defined for <i>system_name</i> , including any comment.
-r <i>system_name</i>	This option removes <i>system_name</i> from the list of registered systems.

Grant or deny access

Use `lpadmin` to control access for a user on a remote client as you would for a local user. The difference is that `user_list` can contain the remote system name.

Syntax

```
# lpadmin -p printer -u allow:user_list | -u deny:user_list
```

The table shows the syntax for this usage of the `lpadmin` command.

Option	Description
<code>-p printer</code>	Specifies the name of the printer on the server.
<code>-u allow</code>	This parameter specifies the users who are allowed access.
<code>-u deny</code>	This parameter specifies the users who are denied access.
<code>user_list</code>	This is a comma or space separated (must be quoted if space separated) list of users to allow or deny. The list can include any of the following: <ul style="list-style-type: none"> • <code>userID</code>: a user on the local system • <code>system_name!userID</code>: a user on <code>system_name</code> • <code>system_name!all</code>: all users on <code>system_name</code> • <code>all!userID</code>: a user on all systems • <code>all!all</code>: all users on all systems

Configuring a Remote Print Client

- Register the server system

```
# lpsystem [-T timeout] [-R retry] [-y comment] server_name
```

- Define the printer queue on the client

```
# lpadmin -p local_name -s  
server_name[!server_printer_name]
```

- **accept** and **enable** the printer queue on the client

Figure H-26. Configuring a Remote Print Client

AU147.0

Notes:

Introduction

Configuring your system to print using a remote LPD printer is done in three steps:

- Register remote system (the printer or print server) on the client
- Define the printer queue on the client
- **accept** and **enable** the printer queue on the client

Of course the print server or network attached printer must already have been configured to accept your requests.

Register the server system

Registering the server system is done using `lpsystem` as explained in “Configuring a Network Print Server” on page 55, except that in this case you use the name or IP address of the server system.

Define the printer queue on the client

Use the following command to define the print queue on the client system:

```
# lpadmin -p local_name -s server_name[!server_printer_name]
```

The table shows the syntax for this usage of the `lpadmin` command.

Option	Description
<code>-p <i>local_name</i></code>	Specifies the name of the print queue on the client.
<code>-s <i>server_name</i></code>	Specifies the name or IP address of the remote print server. This could be a remote system or a printer directly connected to the network.
<code><i>server_printer_name</i></code>	Specifies the name of the printer on the server. The <code><i>local_name</i></code> and the <code><i>server_printer_name</i></code> do not have to agree.

Name of printer on the server

In the case of a remote system, this is the name of the remote print queue on the server. In the case of a network attached printer running LPD, this is the name of the print queue within the printer. Consult your printer's documentation for details.

Printer type and content type

Printer type (`-T printer_type`) and content type (`-I content_type`) may be specified when defining a remote printer on the client. However, this information is not used by the client. Printer type and content type are only used by `lpsched` when printing to local printers. This work is done on the print server. However, you may still want to define these values on the client system so that `lpstat` on the client system will give users a correct understanding of the purpose and usage of this printer.

The Web-based System Manager requires that you enter printer type and content type when defining a remote printer. These values are not actually used, so you can enter anything. However, as mentioned above, entering correct values for the remote printer will make status listings more useful to users.

accept and enable

Use `accept` to cause the print queue on the client to begin accepting requests.

Use `enable` to cause `lpNet` on the client to begin sending requests to the remote printer.

System V Administrative Command Summary

Command	Description
<code>accept/</code> <code>reject</code>	Permits jobs to be queued for specified destination (printer or class). Prevents jobs from being queued for specified destination.
<code>enable/</code> <code>disable</code>	Activates the named printers so they will print from the queue. Deactivates named printers.
<code>cancel</code>	Cancels print jobs.
<code>lpadmin</code>	Create or modify printer configuration.
<code>lpfilter</code>	Manages filters.
<code>lpforms</code>	Manages forms. (Use <code>lpadmin</code> to mount a form.)
<code>lpmove</code>	Move print jobs to another destination.
<code>lpsched/</code> <code>lpshut</code>	Start the print service. Stop the print service.
<code>lpssystem</code>	Register remote systems with the print service.
<code>lpusers</code>	Manages default priority and priority limits for printer service users.
<code>lpstat</code>	Report print service status.

Figure H-27. System V Administrative Command Summary

AU147.0

Notes:

Introduction

This slide provides a brief summary of all the System V administrative commands. A summary of the options to `lpadmin` is included on the next page. See the relevant man page for complete information.

For a comparison of the commands for the System V print subsystem and the AIX print subsystem, see:

Printing for Fun and Profit under AIX 5L (Redbook)
Appendix A. Print Tasks and Commands

lpadmin syntax

The most frequently used administrative command is `lpadmin`. The following table summarizes the command syntax. Again, see the man page for a complete description.

Adding or changing a printer:

```
# lpadmin -p printer [options]
```

Removing a destination (printer or class):

```
# lpadmin -x destination
```

Option	Description
<code>-p <i>printer</i></code>	Specifies the name of the printer. When adding a printer, you must specify either: -v (for a local printer) or -s (for a remote printer)
<code>-v <i>device</i></code>	Used to configure a local printer. Associates a device with a printer.
<code>-s <i>server</i></code> <code>[!<i>server_printer_name</i>]</code>	Used to configure a remote printer. <i>server</i> specifies the name or IP address of the remote print server. This could be a remote system or a printer directly connected to the network. <i>server_printer_name</i> specifies the name of the printer queue on the server.
<code>-x <i>destination</i></code>	Remove <i>destination</i> (which can be a printer or a class) from the print service.
<code>-i <i>interface</i></code> <code>-m <i>model</i></code> <code>-e <i>printer_name</i></code>	Used to specify the printer interface. Only one of these options can be specified. If none of these are specified, the <code>standard</code> interface is used. <code>-i <i>interface</i></code> specifies a full file path. <code>-m <i>model</i></code> specifies one of the supplied interface programs (a file in <code>/usr/lib/lp/model</code>). <code>-e <i>printer_name</i></code> directs <code>lpadmin</code> to copy the interface used for <i>printer_name</i> to the printer being added or changed (specified with <code>-p <i>printer</i></code>).
<code>-T <i>printer_type</i></code>	Identifies an entry in the terminfo database, which is used by the interface program and some filters.
<code>-I <i>content_type_list</i></code>	Identifies one or more types of content that this printer can handle without filtering. If not specified, default is <code>simple</code> .

Option	Description
-c <i>class</i>	Add <i>printer</i> to <i>class</i> . Create <i>class</i> if it does not already exist.
-r <i>class</i>	Remove <i>printer</i> from <i>class</i> . If <i>printer</i> is the last member of <i>class</i> , then remove <i>class</i> .
-O {<i>copy</i> <i>nocopy</i>}	<p>-O <i>copy</i> specifies that files should always be copied to the spool directory. -O <i>nocopy</i> specifies that files should not be copied to the spool directory unless requested by user or otherwise required (for example: a remote print request or input piped to the lp command).</p> <p>This flag sets the value of the <code>copy-files</code> parameter in the <code>/etc/default/lp</code> file to <code>on</code> (<i>copy</i>) or <code>off</code> (<i>nocopy</i>).</p>
-A <i>alert_type</i> [-W <i>minutes</i>]	<p>Specifies the type of alert used to notify the administrator of printer faults. The default is to send the alert message via mail.</p> <p>-W <i>minutes</i> specifies the interval between alerts. <code>0</code> or <code>once</code> is the default, which indicates sending only one alert for a fault.</p>
-M -f <i>form_name</i> [-o <i>filebreak</i>]	<p>Mount <i>form_name</i> on <i>printer</i>.</p> <p>-o <i>filebreak</i> specifies that a form feed be inserted between each copy of the alignment pattern.</p>
-f <i>allow:form_list</i> -f <i>deny:form_list</i>	Allow or deny the forms in <i>form_list</i> to be printed on <i>printer</i> . By default, all forms are denied.
-u <i>allow:user_list</i> -u <i>deny:user_list</i>	Allow or deny the users in <i>user_list</i> to access <i>printer</i> .

System V User Command Summary

Command	Description
<code>cancel</code>	Cancel print jobs
<code>lp</code>	Submit a print job to a printer
<code>lpstat</code>	Report print service status

Figure H-28. System V User Command Summary

AU147.0

Notes:

Introduction

This slide provides a brief summary of the System V user commands. Options for the `lp` command are summarized on the next page. See the relevant man page for complete information.

`cancel`

A summary of the `cancel` command is shown in “Cancel print jobs” on page 41.

`lpstat`

A summary of the `lpstat` command is shown in “Check status” on page 41.

lp syntax

Use the `lp` command to submit jobs to the System V print service. There are many options. The following table summarizes the most commonly used options.

Options can be entered in any order, however the files to be printed must occur at the end of the command line.

```
# lp -d destination [options] files
```

Option	Description
<code>-d destination</code>	Specifies the printer destination (printer or class) where the job is to be printed.
<code>files</code>	Specifies one or more files to be printed. Files will be printed in the order specified. Use <code>-</code> to specify standard input.
<code>-c</code>	Copy files to spool directory before printing. Default is not to copy unless <code>lpadmin -O copy</code> has been used.
<code>-n number</code>	Print <i>number</i> copies. Default is one.
<code>-T content_type</code>	Specifies content type of the file. If the requested printer destination cannot handle this content, the print service looks for a filter to convert the file. If no acceptable combination of filter/printer can be found, the job is rejected.
<code>-f form_name</code>	Print the job on form <i>form_name</i> . If the requested printer destination is not allowed to use this form, the job is rejected. If the form is not mounted, an alert is sent to the administrator. (How form alerts are handled is configured by the <code>lpform</code> command.)
<code>-o options</code>	<code>-o</code> specifies a printer-dependant list of options. Supported options are defined by the printer type (terminfo entry). Options can include items such as: page length, page width, line pitch, character pitch, and so forth.
<code>-m</code>	Send notification via mail when job has been printed. Default is no mail.

Option	Description
<code>-y mode_list</code>	Print the job according to the modes in <i>mode_list</i> . This option may only be used if there is a filter available to handle the requested modes; otherwise the print job will be rejected. The allowed modes are locally defined (in the filter definition files). Modes can include such items as: reverse order, landscape mode, print only selected page numbers, and so forth.

Checkpoint

1. List two advantages of the System V print subsystem.

2. List two advantages of the AIX print subsystem.

3. What command is used to switch from AIX to System V printing? _____
4. `lp sched` uses information in _____ and _____ to screen print jobs.
5. The interface program uses commands in _____ to initialize the printer.
6. _____ are used to convert file content.
7. Use the _____ command to manage filters.
8. _____ is used to create or modify a System V printer.
9. _____ is used to create a printer device.

Figure H-29. Checkpoint

AU147.0

Notes:

Unit Summary



- AIX 5.1 supports both the AIX print subsystem and the System V subsystem
- Only System V is available on Itanium-based system
- Either System V or AIX (not both) can be active on Power-based systems
- The System V print subsystem provides compatibility with printing solutions on many other UNIX variants
- The System V print subsystem provides the capability of supporting a wide range of printers and printing needs, but system management is somewhat complex
- System management of the System V print subsystem will be enhanced in future releases of AIX

Figure H-30. Unit Summary

AU147.0

Notes:

Appendix I. Checkpoint Solutions

Unit 1: Introduction to RS/6000 System Administration

1. What type of adapter are you likely to require for a single-user graphics workstation?

Correct Answer:

C, Graphics

2. What is the difference between UP and SMP machines?

Correct Answer:

Uniprocessors only have one microprocessor. SMP-symmetric Multiprocessing machines have multiple microprocessors.

3. T/ F: The su command allows you to get root authority even if you signed on using another userID.

Correct Answer:

True, but you must still know the root password.

Unit 2: System Management Tools and Documentation

1. Define the SMIT function keys that can be used for the following:

- a. List the command that will be run **F6**
- b. List the screen name which can be used for the fastpath **F8**
- c. Take a screen image: **F8**
- d. Break out into a shell: **F9**
- e. Return to the previous menu: **F3**

2. T/F: AIX Web-based documentation can be used to reference information in different ways, such as searching for a command, searching for a task or viewing information in a book like manner.

Correct Answer:

True

3. T/F: The AIX V5.1 documentation is viewed using a Web browser.

Correct Answer:

True

4. T/F: WebSM is available for client access automatically after the BOS is installed.

Correct Answer:

False. The WebSM server must be configured and enabled for client access.

5. Which of the statements are true regarding the Web-based System Manager?
- An AIX V5.1 system can be managed from a remote PC with appropriate JAVA and Web-browser code installed.
 - In stand-alone mode use the wsm command to access the Web-based System Manager.
 - It is possible to manage an AIX 5.1 system from a remote AIX 5.1 system using an ASCII terminal.
 - The Web-based System Manager includes TaskGuides that direct the user through complex tasks.

Correct Answer:

a, b, d

C is false. However, it is possible with a graphics terminal, to manage different systems simultaneously by adding the remote systems in the Navigation window of WebSM.

Unit 3: System Startup and Shutdown

1. What is the first process that is created on the system and which file does it reference to initiate all the other processes that have to be started?

Correct Answer:

The first process is init and it will look in the /etc/inittab file

2. Which AIX feature can be used to stop and start groups of daemons or programs?

Correct Answer:

The System Resource Controller (SRC)

3. T/F: You can only execute the shutdown command from the console.

Correct Answer:

False. It is not where you type the command as much as who you are. Ordinary users cannot run the shutdown command.

Unit 4: Devices

1. Is it possible to use SCSI ID 7 for a new tape drive?

Correct Answer:

No. The SCSI adapter itself uses ID 7. So, it cannot be used for other devices.

2. What will happen if we attempt to add another device with the SCSI address set to 4?

Correct Answer:

The operation will fail as there is already a device (SCSI Disk Drive) configured at this location.

3. Can the 8 mm tape drive be currently used? Why?

Correct Answer:

No, because it is in the defined state. You have to first make it available by either using SMIT or the mkdev command.

4. Where is the printer connected?

Correct Answer:

The parallel port.

5. The token-ring adapter is installed in what slot?

Correct Answer:

It is installed in slot 4 on the PCI bus.

Unit 5: System Storage Overview

1. How many different PP sizes can be set within a single VG?

Correct Answer:

One

2. By default, how big are PPs?

Correct Answer:

4 MB

3. How many VGs can a PV belong to?

Correct Answer:

b.

4. T/F: All VGDA information on your system is identical, regardless of how many VGs exist.

Correct Answer:

False. All VGDA's within a VG are the same.

5. With which logical volume is the /home file system associated?

Correct Answer:

/dev/hd1

6. What type of file systems are being displayed?

Correct Answer:

Journalled file systems (jfs), enhanced journalled file systems, and CD-ROM (cdrfs)

7. What is the mount point for the file system located on the /dev/lv00 logical volume?

Correct Answer:

/home/john

8. Which are the system supplied logical volumes and their associated file systems?

Correct Answer:

/dev/hd4 /
/dev/hd1 /home
/dev/hd2 /usr
/dev/hd9var /var
/dev/hd3 /tmp

9. Which file system is used primarily to hold user data and home directories?

Correct Answer:

/home

10. Which of the logical volumes above are examples of logical volumes with journaled file systems on them?

Correct Answer:

hd9var,hd3,lv00

Unit 6: Working with the Logical Volume Manager

1. T/F: An LV can span more than one physical volume.

Correct Answer:

TRUE

2. T/F: An LV can span more than one volume group.

Correct Answer:

FALSE

3. T/F: The contents of a PV can be divided between two VGs.

Correct Answer:

FALSE

4. T/F: If mirroring LVs, it is not necessary to perform a backup.

Correct Answer:

FALSE. You still need to back up to external media.

5. T/F: SMIT can be used to easily increase or decrease the size of a logical volume.

Correct Answer:

FALSE. SMIT can only be used to increase a file system. Decreasing one requires backing up the file system, removing it, recreating it, and then restoring.

6. T/F: Striping is done at a logical partition level.

Correct Answer:

FALSE. It is done at a stripe unit level.

Unit 7: Working with File Systems

1. Will the size of the file system change when the size of the logical volume it is on is increased?

Correct Answer:

No

2. If a file system is the same size as the logical volume on which it sits, will the size of the logical volume increase when the size of the file system that is sitting on it increases?

Correct Answer:

Yes

3. If you remove a logical volume, will the file system that is sitting on it be removed as well?

Correct Answer:

The contents will be removed, but the information about the file system contained in `/etc/filesystems` will not.

Unit 8: Managing File Systems

1. What command can you use to determine if a file system is full?

Correct Answer:

`df`

2. What two commands can be used to find the files and users that are taking the most disk space?

Correct Answer:

`du` and `ls -l`

3. T/F: It is good practice to run `fsck -y` on all file systems, even if they are mounted.

Correct Answer:

False

Unit 9: Paging Space

1. What problems can you conclude from the following listing?

Correct Answer

Obviously it is difficult to come to any conclusion regarding the state of this system just by looking at a snapshot picture as above. However, at a first glance the following potential problems can be noticed:

paging00 is underutilized and it is too large.
This needs to be reduced in size.

paging01 is overutilized and the size seems to be too small.
This needs to be increased.

Both user-defined paging spaces are on the same disk.
It would be better if one of them is moved onto a disk which is less utilized.

2. T/F: The size of paging00 (in the above example) can be dynamically decreased. Why?

Correct Answer

True. In AIX V5.1, you can use the `chps -d` command to dynamically decrease the size of a paging space.

Unit 10: AIX V5.1 Installation

1. AIX can be installed from which of the following: (select all that are correct)

Correct Answer:

b

2. T/F: A 'Preservation Install' will preserve all data on the disks.

Correct Answer:

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

3. What is the console used for during the installation process?

Correct Answer:

The console is used to display all the system messages and interact with the installation.

Unit 11: AIX Software Installation and Maintenance

1. Which of the following states can your software be in, in order for you to be able to use it? (select all that apply)

Correct Answer:

a, d

2. What command is used to list all installed software on your system?

Correct Answer:

lspp -l

3. Which of the following can you install as an entity? (select all that apply)

Correct Answer:

a, b, c, d

4. What is the difference between the SMIT menus: **Install Software** and **Update Installed Software to Latest Level (Update All)**?

Correct Answer:

Install Software will by default install everything from the installation media (except printer and devices) onto the system.

Update Installed Software to Latest Level (Update All) installs only updates to filesets already installed on your system.

Unit 12: Backup and Restore

1. What is the difference between A and B?

A: `find /home/fred | backup -ivf /dev/rmt0`

B: `cd /home/fred; find . -print | backup -ivf /dev/rmt0`

Correct Answer:

A will backup the files using the full path names, whereas B will backup the file names using the relative path names, and so B's files can be restored into any directory.

2. On a **mksysb** tape if you entered **tctl rewind** and then **tctl -f/dev/rmt0.1 fsf 3** which element on the tape could you look at?

Which command could you use to restore these files?

Correct Answer:

You would be at the start of the backed up images of the files, having skipped over the boot portion of the tape. The files are backed up using the **backup** command so you would have to use the **restore** command.

3. SMIT **mksysb** will backup all file systems, provided they are mounted.

Correct Answer:

False. **mksysb** only backs up rootvg file systems. To backup other VGs, you must use the **savevg** command.

Unit 13: Security and User Administration

1. What are the benefits of using the **su** command to switch user to **root** over logging in as **root** ?

Correct Answer

A log is kept in the sulog file of all the users executing the su command, which can be monitored

2. Why is a umask of 027 recommended?

Correct Answer

This value will remove all permission bits for the “others ” section, which is desirable

3. As a member of the security group, which password command would you use?

Correct Answer

pwdadm as this will not prompt for roots password or the users old password.

4. Which password change command does SMIT use?

Correct Answer

.passwd command

5. T/F: When you delete a user from the system, all the user’s files and directories are also deleted.

Correct Answer

False. You must remember to delete them if they are obsolete.

6. If an ordinary user forgets their password, can the system administrator find out by querying the system as to what the user’s password was set to? Why?

Correct Answer

No, because the passwords are held on the system in encrypted format, so even the system administrator cannot tell what the password was set to.

7. Password restrictions are set in which of the following files?
- a./etc/passwd
 - b./etc/security/passwd
 - c./etc/security/restrictions
 - d./etc/security/user

Correct Answer

d

8. Which of the following statements are true?
- a.A user can only belong to one group.
 - b.A member of the security group can administer user accounts.
 - c.An admin user is a user whose account cannot be administered by any member of the security group.
 - d.The **chmod g+s** command sets the SUID permission of a file.
 - e.The root user, commonly known as the superuser has UID=0 and GID=0.

Correct Answer

b,c,e

Unit 14: Scheduling

1. T/F: The **at.allow** and **at.deny** files must be used to specify which users are allowed and denied use of the **at** command.

Correct Answer

False. These files are mutually exclusive and only one or the other should be used.

2. Using **cron**, how would you specify a job to run every Thursday at 10 past and 30 minutes past every hour?

Correct Answer

10,30 ***

3. How would you schedule the script "myscript" to run 10 minutes from now?

Correct Answer

```
#at now +10 mins  
myscript  
ctrl -d  
#
```

Unit 15: Printers and Queues

1. T/F: One of the advantages of queues is that each user can have a different default queue set up for them.

Correct Answer:

True. This can be accomplished using the PRINTER environment variable.

2. The `/etc/qconfig` file is read by the **backend** program to determine what the queue discipline is.

Correct Answer:

False, it is read by **qdaemon**.

3. All printer software is automatically installed when you install the base operating system.

Correct Answer:

False, only a handful of printer software is installed by default.

4. What is the difference between these two commands?

```
# qprt -Pasc file1
# qprt -c -Pasc file1
```

Correct Answer:

The `-c` flag produces a spool file.

5. What methods can be used to find out what the system default queue is?

Correct Answer:

- a) First entry in `/etc/qconfig` file
- b) The output from the `qchk` command with no options is for the default queue
- c) The first queue listing from the `lpstat` command is for the default queue

6. Can any user bring the print queues down? Name a few people who can.

Correct Answer:

No, only system administrators, or root or members of the `printq` group can.

7. T/F: Once the queue is down, no more jobs can be submitted to the printer.

Correct Answer:

False. Jobs can be submitted to the queue. However, they will not be printed until the queue is brought up again.

8. Can users hold all their print jobs in a specific queue? If so, how?

Correct Answer:

Yes, they can by only specifying a queue name and not individual job numbers.

Unit 16: Networking Overview

1. What are the following commands used for?

Correct Answer:

ftp transfer files from one machine to another
rexec execute a command on a remote system
telnet login to another system

2. What is the difference (if any) between a **host** and a **gateway**.

Correct Answer:

A host is an individual machine connected to a network, whereas a gateway is a special kind of host which links two or more physical network segments together.

3. T/F: Each machine in a TCP/IP network must have a unique hostname and TCP/IP address.

Correct Answer:

True

4. Which file holds the name and the TCP/IP address of each host?

Correct Answer:

/etc/hosts

Appendix F: Serial Devices

Review Solution One:

1. T/F: If a device, like a TTY, is left for **cfgmgr** to configure automatically, it will pick up the default values which might not be desirable.

Correct Answer:

False, TTYs and other serial devices are not self-configurable and so will not be detected by **cfgmgr**.

2. T/F: If TTYs are connected via concentrator boxes, they must all be connected in sequence on the concentrator box otherwise they will not be configured.

Correct Answer:

False, TTYs can be connected in any order on the concentrator boxes. However, the management of these will obviously be more difficult.

3. T/F: /dev/tty0 indicates that the TTY is connected to port 0, /dev/tty1 to port 1 and so on.

Correct Answer:

False, When a TTY is added to the system, you have to specify to which port the TTY is connected. As they can be connected in any order on the concentrator boxes, there is no relationship between the /dev/tty name, which is the name allocated to the device by the operating system (and will always be the lowest number not allocated) and the port number which you specify. So, for example, tty1 can be connected to port 15.

4. What environment variable holds the terminal type for a terminal?

Term

Appendix G: Itanium-based System Startup

1. On Itanium-based systems, hardware initialization is performed by _____ and _____ .

Correct Answer:

BIOS

EFI

2. Early in the boot process, you can use the _____ menu to view hardware status and configure the IDE bus controller.

Correct Answer:

Setup

3. You can manage the boot list using the...

Correct Answer:

EFI Boot Manager

4. The _____ contains the AIX boot loader.

Correct Answer:

EFI system partition

5. The physical volume partition contains...

Correct Answer:

rootvg logical volumes

6. What is the AIX device name for the EFI system partition?

Correct Answer:

/dev/hdiskX_s0

7. What is the AIX device name for the physical volume partition?

Correct Answer:

/dev/hdiskX

8. List the three components of the boot logical volume.

Correct Answer:

kernel, RAM file system, base ODM

Review Solution Two:

1. How many partitions are on device 80?

Correct Answer:

Two

2. List the short names for partition 1 on device 80.

Correct Answer:

fs0 and blk4

3. List the short names for partition 2 on device 80.

Correct Answer:

blk5

4. Assuming AIX has been installed, which partition contains the AIX boot loader?

Correct Answer:

fs0 (or partition 1 or blk4)

5. Where would the boot image be found on device 80?

Correct Answer:

blk5 (or partition 2) - the physical volume partition

6. To boot from device 80, which partition should you add to the boot list?

Correct Answer:

partition 1, which contains the boot loader

Appendix H: The System V Print Subsystem

1. List two advantages of the System V print subsystem.

Correct Answer:

Compatibility, Availability of interface programs, Security, Support for Forms, Standard PostScript filters, Long term direction

2. List two advantages of the AIX print subsystem.

Correct Answer:

Powerful and flexible printer drivers, mature system management tools, Customizable spooling subsystem

3. What command is used to switch from AIX to System V printing?

Correct Answer:

switch.prt -s SystemV

4. **lpsched** uses information in _____ and _____ to screen print jobs.

Correct Answer:

the printer configuration file
terminfo

5. The interface program uses commands in _____ to initialize the printer.

Correct Answer:

terminfo

6. _____ are used to convert file content.

Correct Answer:

Filters

7. Use the _____ command to manage filters.

Correct Answer:

lpfilter

8. _____ is used to create or modify a System V printer.

Correct Answer:

lpadmin

9. _____ is used to create a printer device.

Correct Answer:

mkdev

Glossary

This glossary includes terms and definitions from:

- The *American National Standard Dictionary for Information Systems*, ANSI X3.172-1990, copyright 1990 by the American National Standards Institute (ANSI). Copies may be purchased from the American National Standards Institute, 11 West 42nd Street, New York, New York 10036. Definitions are identified by the symbol (A) after the definition.
- The ANSI/EIA Standard— 440-A, *Fiber Optic Terminology*. Copies may be purchased from the Electronic Industries Association, 2001 Pennsylvania Avenue, N.W., Washington, DC 20006. Definitions are identified by the symbol (E) after the definition.
- The *Information Technology Vocabulary*, developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC JTC1/SC1). Definitions of published parts of this vocabulary are identified by the symbol (I) after the definition; definitions taken from draft international standards, committee drafts, and working papers being developed by ISO/IEC JTC1/SC1 are identified by the symbol (T) after the definition, indicating that final agreement has not yet been reached among the participating National Bodies of SC1.
- The Network Working Group Request for Comments: 1208.

The following cross-references are used in this glossary:

Contrast with: This refers to a term that has an opposed or substantively different meaning.

Synonym for: This indicates that the term has the same meaning as a preferred term, which is defined in its proper place in the glossary.

Synonymous with: This is a backward reference from a defined term to all other terms that have the same meaning.

See: This refers the reader to multiple-word terms that have the same last word.

See also: This refers the reader to terms that have a related, but not synonymous, meaning.

Deprecated term for: This indicates that the term should not be used. It refers to a preferred term, which is defined in its proper place in the glossary.

A

access mode A matrix of protection information stored with each file specifying who may do what to a file. Three classes of users (owner, group, all others) are allowed or denied three levels of access (read, write, execute).

access permission See **access mode**.

access privilege See **access mode**.

address space The address space of a process is the range of addresses available to it for code and data. The relationship between real and perceived space depends on the system and support hardware.

AIX Advanced Interactive Executive. IBM's implementation of the UNIX Operating System.

AIX Family Definition IBM's definition for the common operating system environment for all members of the AIX family. The AIX Family Definition includes specifications for the AIX Base System, User Interface, Programming Interface, Communications Support, Distributed Processing, and Applications.

alias The command and process of assigning a new name to a command.

ANSI American National Standards Institute. A standards organization. The United States liaison to the International Standards Organization (ISO).

application program A program used to perform an application or part of an application.

argument An item of information following a command. It may, for example, modify the command or identify a file to be affected.

ASCII American Standard Code for Information Interchange. A collection of public domain character sets considered standard throughout the computer industry.

awk An interpreter, included in most UNIX operating systems, that performs sophisticated text pattern matching. In combination with shell scripts, awk can be used to prototype or implement applications far more quickly than traditional programming methods.

B

background (process) A process is "in the background" when it is running independently of the initiating terminal. It is specified by ending the ordinary command with an ampersand (&). The parent of the background process does not wait for its "death".

backup diskette A diskette containing information copied from another diskette. It is used in case the original information is unintentionally destroyed.

Berkeley Software Distribution Disseminating arm of the UNIX operating system community at the University of California at Berkeley; commonly

abbreviated "BSD". Complete versions of the UNIX operating system have been released by BSD for a number of years; the latest is numbered 4.3. The phrase "Berkeley extensions" refers to features and functions, such as the C shell, that originated or were refined at UC Berkeley and that are now considered a necessary part of any fully configured version of the UNIX operating system.

bit bucket The AIX file "/dev/null" is a special file which will absorb all input written to it and return no data (null or end of file) when read.

block A group of records that is recorded or processed as a unit.

block device A device that transfers data in fixed size blocks. In AIX, normally 512 or 1024 bytes.

block special file An interface to a device capable of supporting a file system.

booting Starting the computer from scratch (power off or system reset).

break key The terminal key used to unequivocally interrupt the foreground process.

BSD Berkeley Software Distribution.

- BSD 2.x - PDP-11 Research
- BSD 4.x - VAX Research
- BSD 4.3 - Current popular VAX version of UNIX.

button

1. A word, number, symbol, or picture on the screen that can be selected. A button may represent a command, file, window, or value, for example.
2. A key on a mouse that is used to select buttons on the display screen or to scroll the display image.

byte The amount of storage required to represent one character; a byte is 8 bits.

C

C The programming language in which the UNIX operating system and most UNIX application programs are written. The portability attributed to UNIX operating systems is largely due to the fact that C, unlike other higher level languages, permits programmers to write systems-level code that will work on any computer with a standard C compiler.

change mode The **chmod** command will change the access rights to your own files only, for yourself, your group or all others.

character I/O The transfer of data byte by byte; normally used with slower, low volume devices such as terminals or printers.

character special file An interface to devices not capable of supporting a file system; a byte oriented device.

child The process emerging from a fork command with a zero return code, as distinguished from the parent which gets the process id of the child.

client User of a network service. In the client/server model, network elements are defined as either using (client) or providing (server) network resources.

command A request to perform an operation or run a program. When parameters, arguments, flags, or other operands are associated with a command, the resulting character string is a single command.

command file A data file containing shell commands. See **shell file**, or **shell script**.

command interpreter The part of the operating system that translates your commands into instructions that the operating system understands.

command or **previous command** key.

concatenate The process of forming one character string or file from several. The degenerate case is one file from one file just to display the result using the **cat** command.

console The only terminal known explicitly to the Kernel. It is used during booting and it is the destination of serious system messages.

context The hardware environment of a process, including:

- CPU registers
- Program address
- Stack
- I/O status

context The entire context must be saved during a process swap.

control character Codes formed by pressing and holding the **control** key and then some other key; used to form special functions like **End Of File**.

control-d See **eof** character.

cooked input Data from a character device from which backspace, line kill, and interrupt characters have been removed (processed). See **raw input**.

current directory The currently active directory. When you specify a file name without specifying a directory, the system assumes that the file is in your current directory.

current subtree Files or directories attached to the current directory.

curses A C subroutine library providing flexible screen handling. See **Termlib** and **Termcap**.

cursor A movable symbol (such as an underline) on a display, usually used to indicate to the operator where to type the next character.

customize To describe (to the system) the devices, programs, users, and user defaults for a particular data processing system.

D

DASD Direct Access Storage Device. IBM's term for a hard disk.

device driver A program that operates a specific device, such as a printer, disk drive, or display.

device special file A file which passes data directly to/from the device.

directory A type of file containing the names and controlling information for other files or other directories.

directory pathname The complete and unique external description of a file giving the sequence of connection from the root directory to the specified directory or file.

diskette A thin, flexible magnetic plate that is permanently sealed in a protective cover. It can be used to store information copied from the disk.

diskette drive The mechanism used to read and write information on diskettes.

display device An output unit that gives a visual representation of data.

display screen The part of the display device that displays information visually.

E

echo To simply report a stream of characters, either as a message to the operator or a debugging tool to see what the file name generation process is doing.

editor A program used to enter and modify programs, text, and other types of documents.

environment A collection of values passed either to a C program or a shell script file inherited from the invoking process.

escape The backslash "\" character specifies that the single next character in a command is ordinary text without special meaning.

Ethernet A baseband protocol, invented by the XEROX Corporation, in common use as the local area network for UNIX operating systems interconnected via TCP/IP.

event One of the previous lines of input from the terminal. Events are stored in the (Berkeley) History file.

event identifier A code used to identify a specific event.

execution permission For a file, the permission to execute (run) code in the file. A text file must have execute permission to be a shell script. For a directory, the permission to search the directory.

F

field A contiguous group of characters delimited by blanks. A field is the normal unit of text processed by text processes like sort.

field separator The character used to separate one field from the next; normally a blank or tab.

FIFO "First In, First Out". In AIX, a FIFO is a permanent, named pipe which allows two unrelated processes to communicate. Only related processes can use normal pipes.

file A collection of related data that is stored and retrieved by an assigned name. In AIX, files are grouped by directories.

file index Sixty-four bytes of information describing a file. Information such as the type and size of the file and the location on the physical device on which the data in the file is stored is kept in the file index. This index is the same as the AIX Operating System i-node.

filename expansion or generation A procedure used by the shell to generate a set of filenames based on a specification using metacharacters, which define a set of textual substitutions.

file system The collection of files and file management structures on a physical or logical mass storage device, such as a diskette or minidisk.

filter Data-manipulation commands (which, in UNIX operating systems, amount to small programs) that take input from one process and perform an operation yielding new output. Filters include editors, pattern-searchers, and commands that sort or differentiate files, among others.

fixed disk A storage device made of one or more flat, circular plates with magnetic surfaces on which information can be stored.

fixed disk drive The mechanism used to read and write information on a fixed disk.

flag See **Options**.

foreground (process) An AIX process which interacts with the terminal. Its invocation is not followed by an ampersand.

formatting The act of arranging text in a form suitable for reading. The publishing equivalent to compiling a program.

fsck A utility to check and repair a damaged file structure. This normally results from a power failure or hardware malfunction. It looks for blocks not assigned to a file or the free list and puts them in the free list. (The use of blocks not pointed at cannot be identified.)

free list The set of all blocks not assigned to a file.

full path name The name of any directory or file expressed as a string of directories and files beginning with the root directory.

G

gateway A device that acts as a connector between two physically separate networks. It has interfaces to more than one network and can translate the packets of one network to another, possibly dissimilar network.

global Applying to all entities of a set. For example:

- A global search - look everywhere
- A global replace - replace all occurrences
- A global symbol - defined everywhere.

grep An AIX command which searches for strings specified by a regular expression. (Global Regular Expression and Print.)

group A collection of AIX users who share a set of files. Members of the group have access privileges exceeding those of other users.

H

hardware The equipment, as opposed to the programming, of a system.

header A record at the beginning of the file specifying internal details about the file.

heterogeneous Descriptor applied to networks composed of products from multiple vendors.

hierarchy A system of objects in which each object belongs to a group. Groups belong to other groups. Only the "head" does not belong to another group. In AIX this object is called the "Root Directory".

highlight To emphasize an area on the display screen by any of several methods, such as brightening the area or reversing the color of characters within the area.

history A list of recently executed commands.

home (directory). 1. A directory associated with an individual user.

home (directory). 2. Your current directory on login or after issuing the cd command with no argument.

homogeneous Descriptor applied to networks composed of products from a single vendor.

hypertext Term for on-line interactive documentation of computer software; to be included with AIX.

I

IEEE Institute of Electrical and Electronics Engineers. A professional society active in standards work, the IEEE is the official body for work on the POSIX (Portable Operating System for Computer Environments) open system interface definition.

index See **file index**.

indirect block A file element which points at data sectors or other indirect blocks.

init The initialization process of AIX. The ancestor of all processes.

initial program load The process of loading the system programs and preparing the system to run jobs.

i-node A collection of logical information about a file including owner, mode, type and location.

i number The internal index or identification of an i-node.

input field An area into which you can type data.

input redirection The accessing of input data from other than standard input (the keyboard or a pipe).

interoperability The ability of different kinds of computers to work well together.

interpreter A program which "interprets" program statements directly from a text (or equivalent) file. Distinguished from a compiler which creates computer instructions for later direct execution.

interrupt A signal that the operating system must reevaluate its selection of which process should be

running. Usually to service I/O devices but also to signal from one process to another.

IP Internet Protocol.

ipl See **initial program load**.

ISO International Standards Organization. A United Nations agency that provides for creation and administration of worldwide standards.

J

job A collection of activities.

job number An identifying number for a collection of processes devolving from a terminal command.

K

kernel The part of an operating system that contains programs that control how the computer does its work, such as input/output, management and control of hardware, and the scheduling of user tasks.

keyboard An input device consisting of various keys allowing the user to input data, control cursor and pointer locations, and to control the user/work station dialogue.

kill To prematurely terminate a process.

kill character The character which erases an entire line (usually @).

L

LAN Local Area Network. A facility, usually a combination of wiring, transducers, adapter boards, and software protocols, which interconnects workstations and other computers located within a department, building, or neighborhood. Token-Ring and Ethernet are local area network products.

libc A basic set of C callable routines.

library In UNIX operating systems, a collection of existing subroutines that allows programmers to make use of work already done by other programmers. UNIX operating systems often include separate libraries for communications, window management, string handling, math, etc.

line editor An editor which processes one line at a time by the issuing of a command. Usually associated with sequential only terminals such as a teletype.

link An entry in an AIX directory specifying a data file or directory and its name. Note that files and directories are named solely by virtue of links. A name is not an intrinsic property of a file. A file is uniquely identified only by a system generated identification number.

lint A program for removing "fuzz" from C code. Stricter than most compilers. Helps former Pascal programmers sleep at night.

Local Area Network (LAN) A facility, usually a combination of wiring, transducers, adapter boards, and software protocols, which interconnects workstations and other computers located within a

department, building, or neighborhood. Token-Ring and Ethernet are local area network products.

login Identifying oneself to the system to gain access.

login directory See **home directory**.

login name The name by which a user is identified to the system.

logout Informing the system that you are through using it.

M

mail The process of sending or receiving an electronically delivered message within an AIX system. The message or data so delivered.

make Programming tool included in most UNIX operating systems that helps "make" a new program out of a collection of existing subroutines and utilities, by controlling the order in which those programs are linked, compiled, and executed.

map The process of reassigning the meaning of a terminal key. In general, the process of reassigning the meaning of any key.

memory Storage on electronic memory such as random access memory, read only memory, or registers. See **storage**.

message Information displayed about an error or system condition that may or may not require a user response.

motd "Message of the day". The login "billboard" message.

Motif The graphical user interface for OSF, incorporating the X Window System. Behavior of this interface is compatible with the IBM/Microsoft Presentation Manager user interface for OS/2. Also called OSF/Motif.

mount A logical (i.e., not physical) attachment of one file directory to another. "remote mounting" allows files and directories that reside on physically separate computer systems to be attached to a local system.

mouse A device that allows you to select objects and scroll the display screen by means of buttons.

move Relinking a file or directory to a different or additional directory. The data (if any) is not moved, only the links.

multiprogramming Allocation of computer resources among many programs. Used to allow many users to operate simultaneously and to keep the system busy during delays occasioned by I/O mechanical operations.

multitasking Capability of performing two or more computing tasks, such as interactive editing and complex numeric calculations, at the same time. AIX and OS/2 are multi-tasking operating systems; DOS, in contrast, is a single-tasking system.

multiuser A computer system which allows many people to run programs "simultaneously" using multiprogramming techniques.

N

named pipe See **FIFO**.

Network File System (NFST) A program developed by SUN Microsystems, Inc. for sharing files among systems connected via TCP/IP. IBM's AIX, VM, and MVS operating systems support NFS.

NFST See **Network File System**.

NIST National Institute of Science and Technology (formerly the National Bureau of Standards).

node An element within a communication network.

- Computer
- Terminal
- Control Unit

null A term denoting emptiness or nonexistence.

null device A device used to obtain empty files or dispose of unwanted data.

null string A character string containing zero characters.

O

object-oriented programming Method of programming in which sections of program code and data are represented, used, and edited in the form of "objects", such as graphical elements, window components, etc., rather than as strict computer code. Through object-oriented programming techniques, toolkits can be designed that make programming much easier. Examples of object-oriented programming languages include Pareplace Systems, Inc.'s Smalltalk-80T, AT&T's C++T, and Stepstone Inc.'s Objective-CR.

oem original equipment manufacturer. In the context of AIX, OEM systems refer to the processors of a heterogeneous computer network that are not made or provided by IBM.

Open Software Foundation (OSF) A non-profit consortium of private companies, universities, and research institutions formed to conduct open technological evaluations of available components of UNIX operating systems, for the purpose of assembling selected elements into a complete version of the UNIX operating system available to those who wish to license it. IBM is a founding sponsor and member of OSF.

operating system The programs and procedures designed to cause a computer to function, enabling the user to interact with the system.

option A command argument used to specify the details of an operation. In AIX an option is normally preceded by a hyphen.

ordinary file Files containing text, programs, or other data, but not directories.

OSFT See **Open Software Foundation**.

output redirection Passing a programs standard output to a file.

owner The person who created the file or his subsequent designee.

P

packet switching The transmission of data in small, discrete switching "packets" rather than in streams, for the purpose of making more efficient use of the physical data channels. Employed in some UNIX system communications.

page To move forward or backward on screen full of data through a file usually referring to an editor function.

parallel processing A computing strategy in which a single large task is separated into parts, each of which then runs in parallel on separate processors.

parent The process emerging from a Fork with a non#zero return code (the process ID of the child process). A directory which points at a specified directory.

password A secret character string used to verify user identification during login.

PATH A variable which specifies which directories are to be searched for programs and shell files.

path name A complete file name specifying all directories leading to that file.

pattern-matching character Special characters such as * or ? that can be used in a file specification to match one or more characters. For example, placing a ? in a file specification means that any character can be in that position.

permission The composite of all modes associated with a file.

pipes UNIX operating system routines that connect the standard output of one process with the standard input of another process. Pipes are central to the function of UNIX operating systems, which generally consist of numerous small programs linked together into larger routines by pipes. The "piping" of the list directory command to the word count command is `ls | wc`. The passing of data by a pipe does not (necessarily) involve a file. When the first program generates enough data for the second program to process, it is suspended and the second program runs. When the second program runs out of data it is suspended and the first one runs.

pipe fitting Connecting two programs with a pipe.

pipeline A sequence of programs or commands connected with pipes.

portability Desirable feature of computer systems and applications, referring to users' freedom to run application programs on computers from many vendors without rewriting the program's code. Also known as "applications portability", "machine-independence", and "hardware-independence"; often cited as a cause of the recent surge in popularity of UNIX operating systems.

port A physical I/O interface into a computer.

POSIX "Portable Operating Systems for Computer Environments". A set of open standards for an operating system environment being developed under the aegis of the IEEE.

preprocessor The macro generator preceding the C compiler.

process A unit of activity known to the AIX system, usually a program.

process 0 (zero) The scheduler. Started by the "boot" and permanent. See **init**.

process id A unique number (at any given time) identifying a process to the system.

process status The process's current activity.

- Non existent
- Sleeping
- Waiting
- Running
- Intermediate
- Terminated
- Stopped.

profile A file in the users home directory which is executed at login to customize the environment. The name is **.profile**.

prompt A displayed request for information or operator action.

protection The opposite of permission, denying access to a file.

Q

quotation Temporarily cancelling the meaning of a metacharacter to be used as an ordinary text character. A backslash (\) "quotes" the next character only.

R

raw I/O I/O conducted at a "physical" level.

read permission Allows reading (not execution or writing) of a file.

recursive A recursive program calls itself or is called by a subroutine which it calls.

redirection The use of other than standard input (keyboard or pipe output) or standard output (terminal display or pipe). Usually a file.

regular expression An expression which specifies a set of character strings using metacharacters.

relative path name The name of a directory or file expressed as a sequence of directories followed by a file name, beginning from the current directory.

RISC Reduced Instruction Set Computer. A class of computer architectures, pioneered by IBM's John Cocke, that improves price#performance by minimizing the number and complexity of the operations required in the instruction set of a computer. In this class of architecture, advanced compiler technology is used to provide operations, such as multiplication, that are infrequently used in practice.

root directory The directory that contains all other directories in the file system.

S

scalability Desirable feature of computer systems and applications. Refers to the capability to use the same environment on many classes of computers, from personal computers to supercomputers, to accommodate growth or divergent environments, without rewriting code or losing functionality.

SCCS Source Code Control System. A set of programs for maintaining multiple versions of a file using only edit commands to specify alternate versions.

scope The field of an operation or definition. Global scope means all objects in a set. Local scope means a restriction to a subset of the objects.

screen See **display screen**.

scroll To move information vertically or horizontally to bring into view information that is outside the display screen or pane boundaries.

search and replace The act of finding a match to a given character string and replacing each occurrence with some other string.

search string The pattern used for matching in a search operation.

sed Non-interactive stream editor used to do "batch" editing. Often used as a tool within shell scripts.

server A provider of a service in a computer network; for example, a mainframe computer with large storage capacity may play the role of database server for interactive terminals. See **client**.

setuid A permission which allows the access rights of a program owner to control the access to a file. The program can act as a filter for user data requests.

shell The outermost (user interface) layer of UNIX operating systems. Shell commands start and control other processes, such as editors and compilers; shells can be textual or visual. A series of system commands can be collected together into a "shell script" that executes like a batch (.BAT) file in DOS.

shell program A program consisting of a sequence of shell commands stored in an ordinary text file which has execution permission. It is invoked by simply naming the file as a shell command.

shell script See **shell program**.

single user (mode) A temporary mode used during "booting" of the AIX system.

signal A software generated interrupt to another process. See **kill**.

sockets Destination points for communication in many versions of the UNIX operating system, much as electrical sockets are destination points for electrical plugs. Sockets, associated primarily with 4.3 BSD, can be customized to facilitate communication between separate processes or between UNIX operating systems.

software Programs.

special character See **metacharacter**.

special file A technique used to access I/O devices in which "pseudo files" are used as the interface for commands and data.

standard error The standard device at which errors are reported, normally the terminal. Error messages may be directed to a file.

standard input The source of data for a filter, which is by default obtained from the terminal, but which may be obtained from a file or the standard output of another filter through a pipe.

standard output The output of a filter which normally is by default directed to the terminal, but which may be sent to a file or the standard input of another filter through a pipe.

stdio A "Standard I/O" package of C routines.

sticky bit A flag which keeps commonly used programs "stick" to the swapping disk for performance.

stopped job A job that has been halted temporarily by the user and which can be resumed at his command.

storage In contrast to memory, the saving of information on physical devices such as fixed disk or tape. See **memory**.

store To place information in memory or onto a diskette, fixed disk, or tape so that it is available for retrieval and updating.

streams Similar to sockets, streams are destination points for communications in UNIX operating systems. Associated primarily with UNIX System V, streams are considered by some to be more elegant than sockets, particularly for interprocess communication.

string A linear collection of characters treated as a unit.

subdirectory A directory which is subordinate to another directory.

subtree That portion of an AIX file system accessible from a given directory below the root.

suffix A character string attached to a file name that helps identify its file type.

superblock Primary information repository of a file system (location of i-nodes, free list, etc.).

superuser The system administration; a user with unique privileges such as upgrading execution priority and write access to all files and directories.

superuser authority The unrestricted ability to access and modify any part of the Operating System. This authority is associated with the user who manages the system.

SVID System V Interface Definition. An AT&T document defining the standard interfaces to be used by UNIX System V application programmers and users.

swap space (disk) That space on an I/O device used to store processes which have been swapping out to make room for other processes.

swapping The process of moving processes between main storage and the "swapping device", usually a disk.

symbolic debugger Program for debugging other programs at the source code level. Common symbolic debuggers include sdb, dbx, and xdbx.

sync A command which copies all modified blocks from RAM to the disk.

system The computer and its associated devices and programs.

system unit The part of the system that contains the processing unit, the disk drive and the disk, and the diskette drive.

System V AT&T's recent releases of its UNIX operating system are numbered as releases of "UNIX System V".

T

TCP Transmission Control Protocol. A facility for the creation of reliable bytestreams (byte-by-byte, end#to#end transmission) on top of unreliable datagrams. The transmission layer of TCP/IP is used to interconnect applications, such as FTP, so that issues of re-transmission and blocking can be subordinated in a standard way. See **TCP/IP**.

TCP/IP Transmission Control Protocol/Internet Protocol. Pair of communications protocol considered defacto standard in UNIX operating system environments. IBM TCP/IP for VM and IBM TCP/IP for MVS are licensed programs that provide VM and MVS users with the capability of participating in networks using the TCP/IP protocol suite.

termcap A file containing the description of several hundred terminals. For use in determining communication protocol and available function.

termlib A set of C programs for using **termcap**.

tools Compact, well designed programs to perform specific tasks. More complex processes are performed by sequences of tools, often in the form of pipelines which avoid the need for temporary files.

two-digit display Two seven-segment light-emitting diodes (LEDs) on the operating panel used to track the progress of power-on self-tests (POSTs).

U

UNIX® Operating System A multi-user, multi-tasking interactive operating system created at AT&T Bell Laboratories that has been widely used and developed by universities, and that now is becoming increasingly popular in a wide range of commercial applications. See Kernel, Shell, Library, Pipes, Filters.

user interface The component of the AIX Family Definition that describes common user interface functions for the AIX PS/2, AIX/RT, and AIX/370 operating systems.

/usr/grpR One of the oldest, and still active, user groups for the UNIX operating systems. IBM is a member of /usr/grp.

uucp A set of AIX utilities allowing

- Autodial of remote systems
- Transfer of files
- Execution of commands on the remote system
- Reasonable security.

V

vi Visual editor. A character editor with a very powerful collection of editing commands optimized for ASCII terminals; associated with BSD versions of the UNIX operating system.

visual editor An optional editor provided with AIX in which changes are made by modifying an image of the file on the screen, rather than through the exclusive use of commands.

W

wild card A metacharacter used to specify a set of replacement characters and thus a set of file names. For example "*" is any zero or more characters and "?" is any one character.

window A rectangular area of the screen in which the dialog between you and a given application is displayed.

working directory The directory from which file searches are begun if a complete pathname is not specified. Controlled by the cd (change directory) command.

workstation A device that includes a keyboard from which an operator can send information to the system, and a display screen on which an operator can see the information sent to or received from the computer.

write Sending data to an I/O device.

write permission Permission to modify a file or directory.

X

X/OpenT An international consortium, including many suppliers of computer systems, concerned with the selection and adoption of open system standards for computing applications. IBM is a corporate sponsor of X/Open. See **Common Application Environment**.

X Windows IBM's implementation of the X Window System developed at the Massachusetts Institute of Technology with the support of IBM and DECT, that gives users "windows" into applications and processes not located only or specifically on their own console or computer system. X-Windows is a powerful vehicle for distributing applications among users on heterogeneous networks.

Y

yacc "Yet Another Compiler# Compiler". For producing new command interfaces.

Z

zeroth argument The command name; the argument before the first.

