AIX 6 System Administration I: Implementation

(Course code AU14)

Student Notebook

ERC 12.0

IBM certified course material

Trademarks

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

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

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

AIX®	AIX 5L™	AIX 6 (logo)™
AS/400®	CICS/6000®	DB2®
Domino®	HACMP™	iSeries®
Language Environment®	Lotus®	Magstar®
MVS™	Network Station®	Notes®
OS/2®	POWER™	POWER4™
POWER5™	POWER6™	PowerPC®
pSeries®	Requisite®	RISC System/6000®
RS/6000®	SecureWay®	SP™
System p™	System/370™	System Storage™
Tivoli®	400®	

VMware® and the VMware "boxes" logo and design, Virtual SMP and VMotion are registered trademarks or trademarks (the "Marks") of VMware, Inc. in the United States and/or other jurisdictions.

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

PS/2® is a trademark or registered trademark of Lenovo in the United States, other countries, or both.

PostScript is either a registered trademark or a trademark of Adobe Systems Incorporated in the United States, and/or other countries.

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

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

February 2008 edition

Contents

Trademarks

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

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

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

AIX®	AIX 5L™	AIX 6 (logo)™
AS/400®	CICS/6000®	DB2®
Domino®	HACMP™	iSeries®
Language Environment®	Lotus®	Magstar®
MVS™	Network Station®	Notes®
OS/2®	POWER™	POWER4™
POWER5™	POWER6™	PowerPC®
pSeries®	Requisite®	RISC System/6000®
RS/6000®	SecureWay®	SP™
System p™	System/370™	System Storage™
Tivoli®	400®	

VMware® and the VMware "boxes" logo and design, Virtual SMP and VMotion are registered trademarks or trademarks (the "Marks") of VMware, Inc. in the United States and/or other jurisdictions.

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

PS/2® is a trademark or registered trademark of Lenovo in the United States, other countries, or both.

PostScript is either a registered trademark or a trademark of Adobe Systems Incorporated in the United States, and/or other countries.

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

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

Intel is a trademark or registered trademark of Intel Corporation or its subsidiaries in the United States and other countries.

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

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

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

Course description

AIX 6 System Administration I: Implementation

Duration: 5 days

Purpose

The purpose of this course is to enable students to install, customize, and administer the AIX 6 operating system in a multiuser environment using System p server.

Audience

Anyone responsible for the system administrative duties implementing and managing an AIX 6 operating system on an System p server.

Prerequisites

The students attending this course should already 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
- Use a graphic Common Desktop Environment (CDE) interface

These skills can be acquired by taking the AIX Basics course or through equivalent AIX/UNIX knowledge.

Objectives

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

- Install the AIX 6 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 to IBM System p and AIX system administration
- 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 6 Basics course. A basic understanding of AIX environment and simple commands is recommended before taking this course.

Agenda

Day 1

Day 2

Day 3

Day 4

Day 5

Text highlighting

The following text highlighting conventions are used throughout this book:

Bold	Identifies file names, file paths, directories, user names and principals.
Italics	Identifies links to Web sites, publication titles, and is used where the word or phrase is meant to stand out from the surrounding text.
`Monospace`	Identifies attributes, variables, file listings, SMIT menus, code examples of text similar to what you might see displayed, examples of portions of program code similar to what you might write as a programmer, and messages from the system.
`Monospace bold`	Identifies commands, daemons, menu paths and what the user would enter in examples of commands and SMIT menus.
`<text>`	The text between the < and > symbols identifies information the user must supply. The text may be normal highlighting, **bold** or `monospace`, or **`monospace bold`** depending on the context.

Unit 1. Introduction to IBM System p Servers and AIX system administration

What this unit is about

This unit introduces the basic IBM System p Server 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 terminology and concepts of IBM System p Servers
- List common configurations available for IBM System p Servers
- Describe the roles of the system administrator
- Obtain **root** access with the `su` command

How you will check your progress

Accountability:

- Checkpoint questions
- Exercises

References

SG24-7559	*IBM AIX Version 6.1 Differences Guide*
SG24-2581	*Managing AIX on PCI-Based RISC System/6000 Workstations*

Unit objectives

After completing this unit, you should be able to:

- Define terminology and concepts of IBM System p servers
- List common configurations available for IBM System p servers
- Describe the roles of the system administrator
- Obtain root access with the su command

Figure 1-1. Unit Objectives AU1412.0

Notes:

What is RISC technology?

Reduced **I**nstruction **S**et **C**omputing (RISC) processors aim to:

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

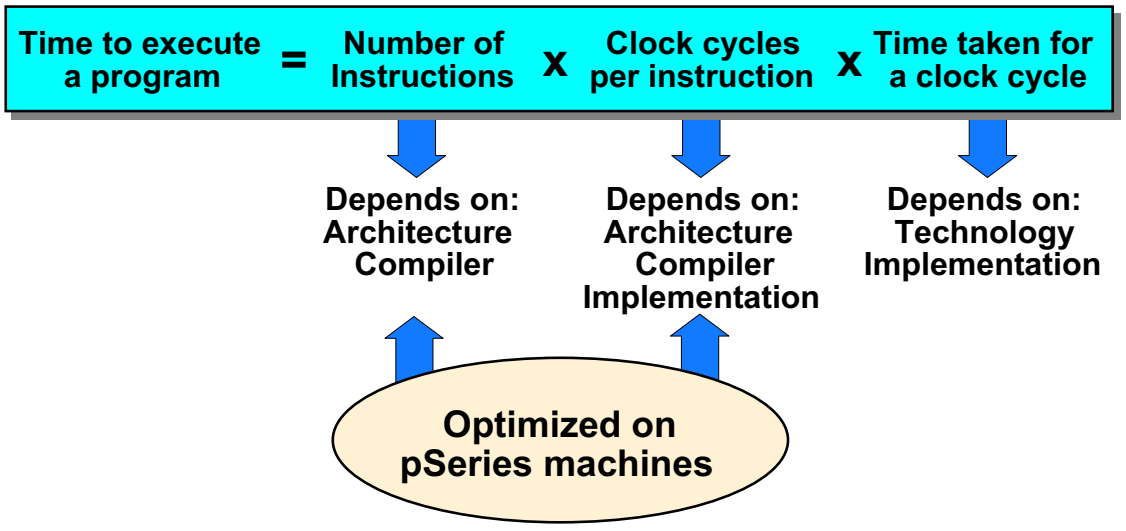


Figure 1-2. What is RISC Technology? AU1412.0

Notes:

RISC history

The IBM System p servers use RISC processors and were formally called RS/6000 systems, where the RS stood for RISC system.

Reduced **I**nstruction **S**et **C**omputing (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.

POWER architecture

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 **P**erformance **O**ptimized **W**ith **E**nhanced **R**isc **(POWER)**

architecture. The original POWER architecture has also evolved into the PowerPC, POWER4, POWER5 and POWER6 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.

64-bit architecture

Support for 64-bit architecture has been provided since 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 to AIX since AIX 5L V5.1 was the introduction of the 64-bit kernel. Server consolidation and workload scalability continues to require higher capacity hardware systems that support more memory and additional I/O devices. The 64-bit kernel is designed to support these requirements.

The 32-bit and the 64-bit kernel are available. Only 64-bit CHRP-compliant PowerPC machines are supported for the 64-bit kernel on the POWER platform. 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 reduce the number of systems needed by an organization, thereby reducing the cost and complexity of system administration

System p bus types

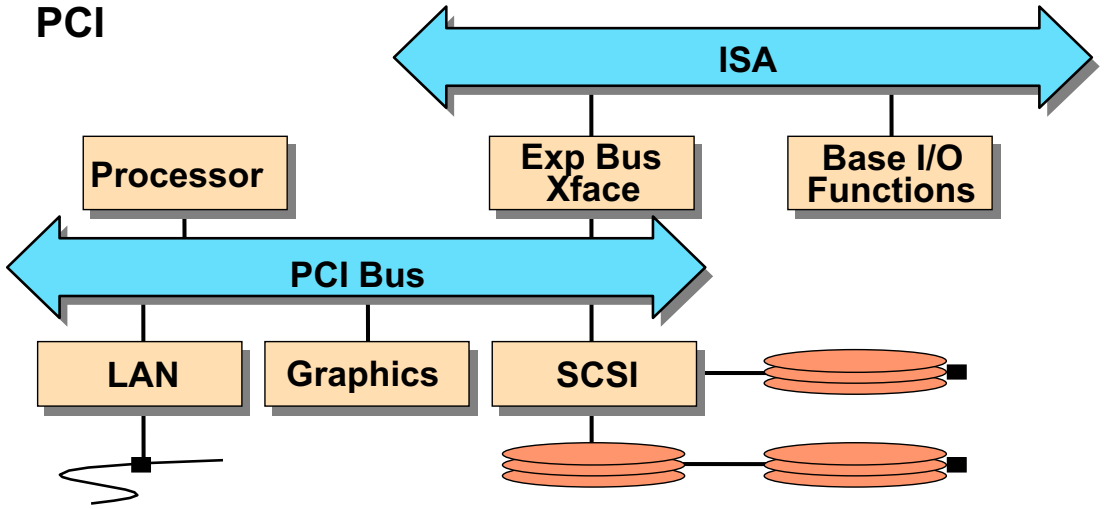


Figure 1-3. System p bus types AU1412.0

Notes:

Introduction

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

Peripheral Component Interconnect (PCI) bus

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 IBM System p platforms.

Industry Standard Architecture (ISA) bus

IBM System p also contains an Industry Standard Architecture (ISA) bus for use with some built-in devices like the diskette drive and keyboard.

Built-in ISA support remains in AIX 6.1.

Some older model PCI systems also contain ISA slots that would accept standard ISA cards. With AIX 5L V5.2 and later, ISA cards are no longer supported.

For more information

A good source for hardware information is:
http://publib.boulder.ibm.com/infocenter/pseries/v6r1/index.jsp

Workstation configuration

Single-User Graphical Workstation

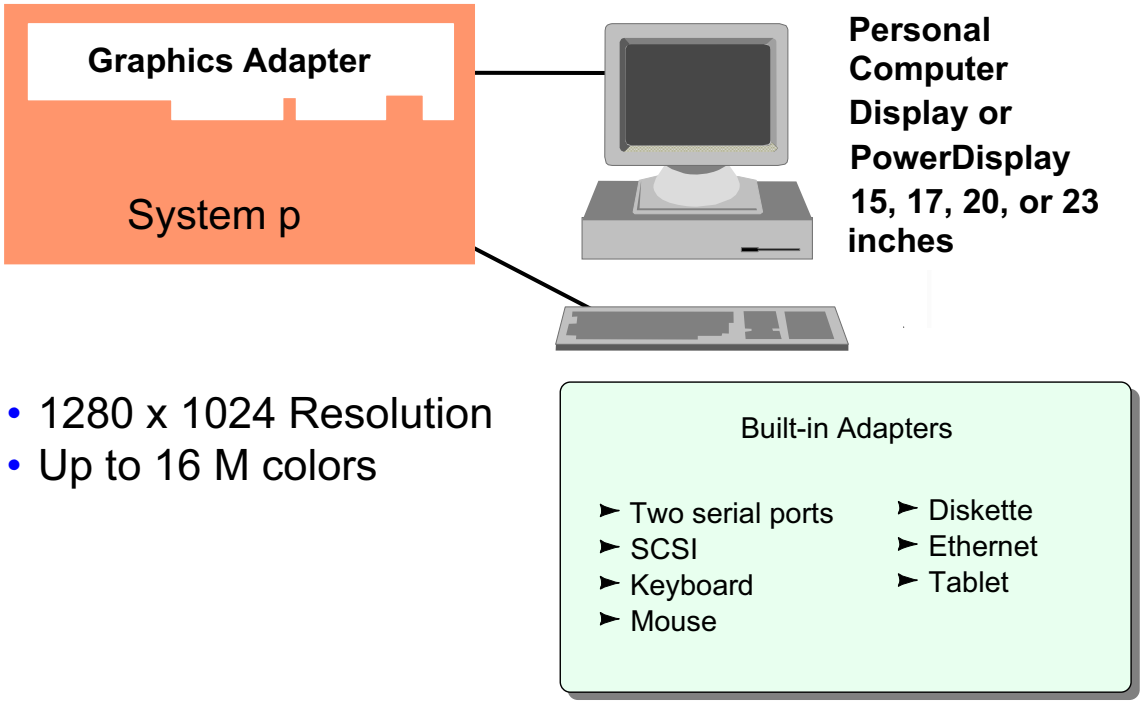


Figure 1-4. Workstation configuration AU1412.0

Notes:

Overview

One common configuration for the IBM System p servers is as a single-user graphical workstation suitable for graphics applications, such as CAD/CAM.

In this configuration, the IBM System p server has a graphical 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 IBM System p 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

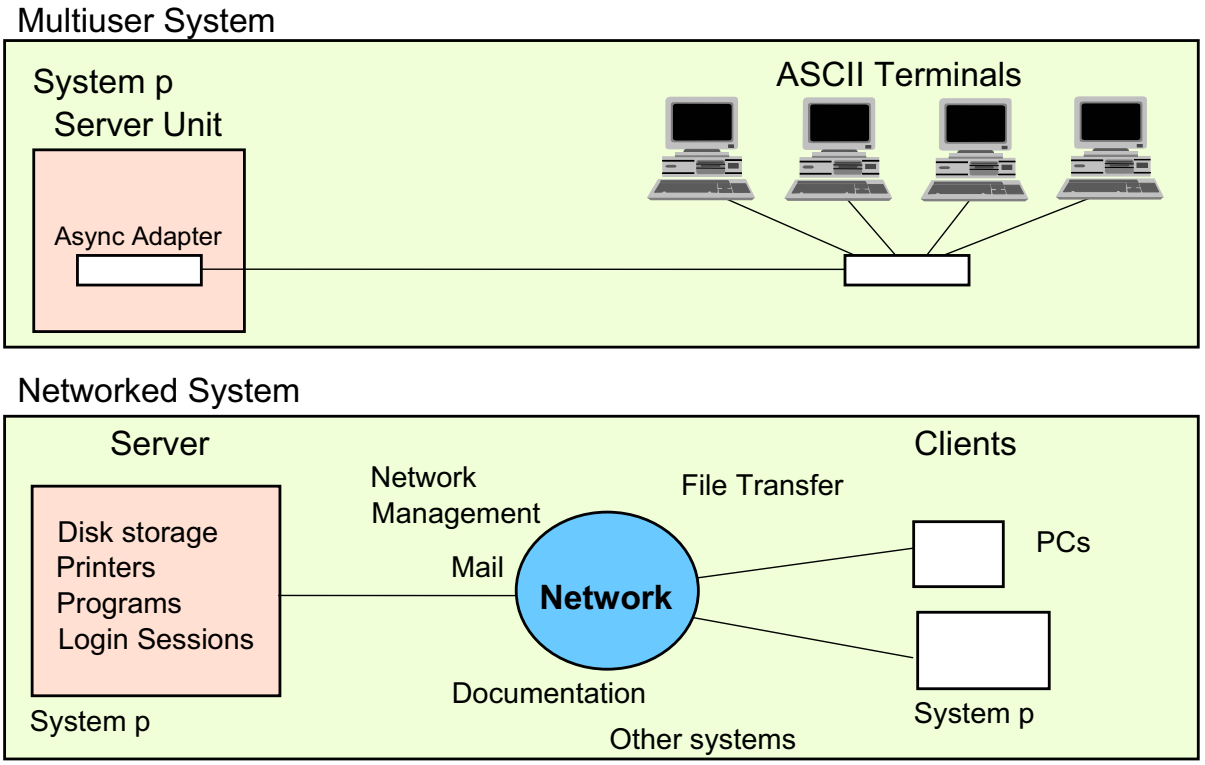


Figure 1-5. Server configurations AU1412.0

Notes:

Overview

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 IBM System p servers. All further ASCII devices require an asynchronous adapter card.

More complex systems consist of many IBM System p servers and other devices such as PCs connected over a local area network (LAN) like Ethernet or token ring. In this case, the IBM System p server requires the appropriate communications adapter card.

PC connectivity

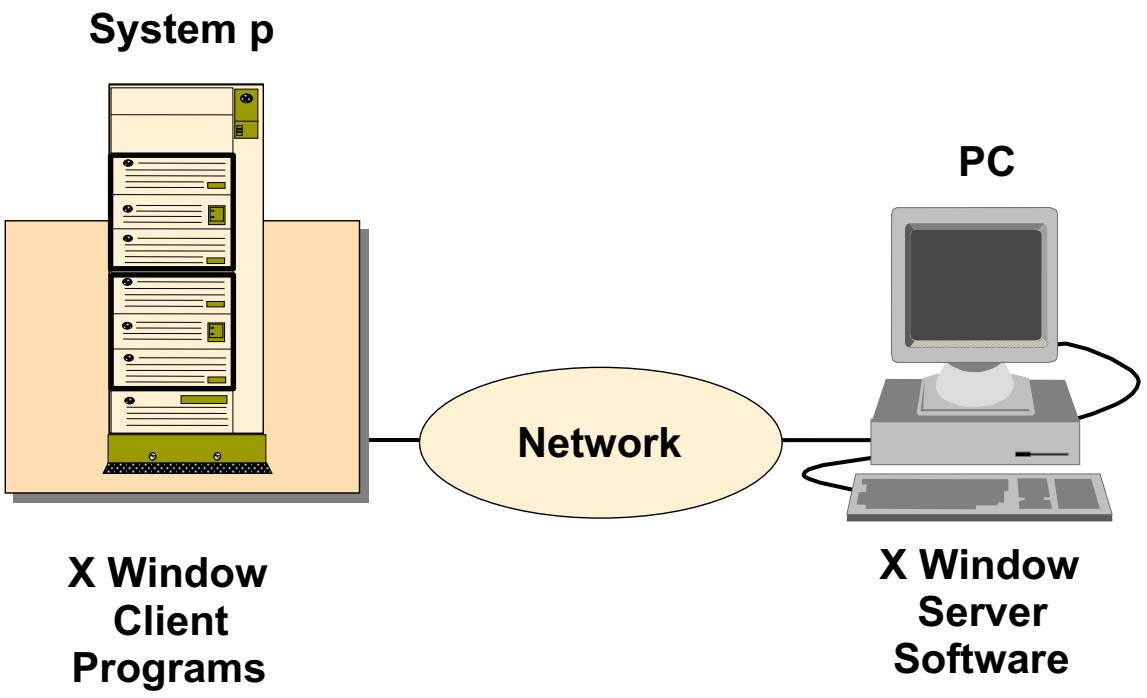


Figure 1-6. PC connectivity AU1412.0

Notes:

Overview

Very commonly, IBM System p servers are accessed via a network using PCs.

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 IBM System p server. There are many commercially available software packages for several different operating systems that provide this functionality.

Logical partitioning (LPAR)

- Resources allocated in flexible units of granularity

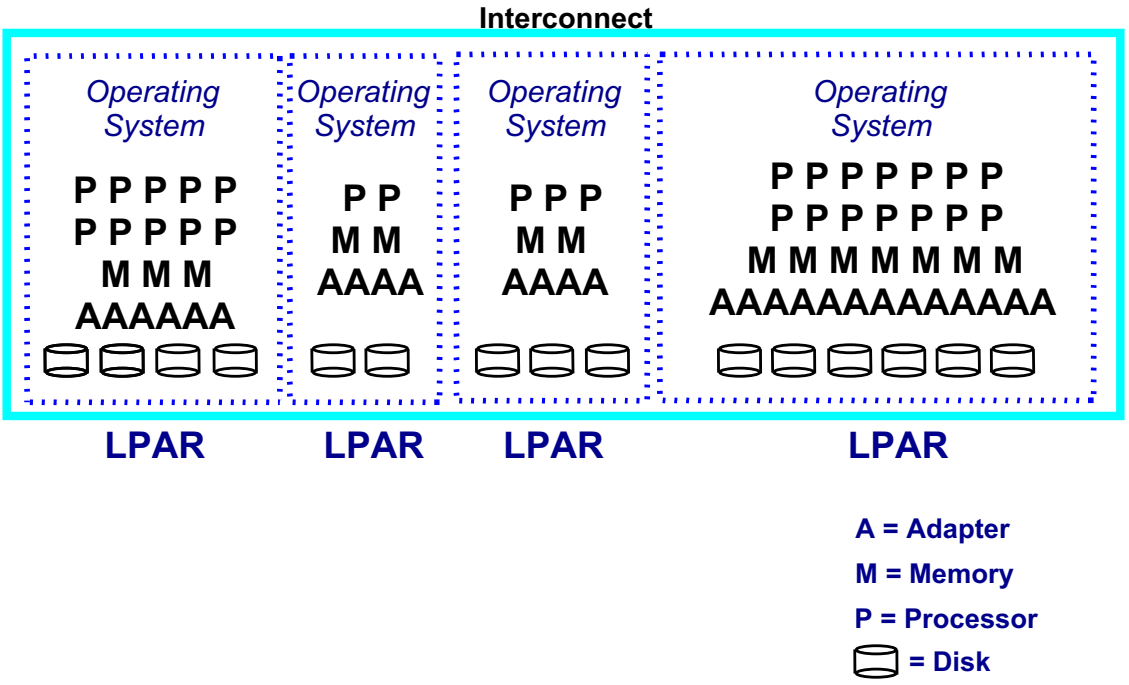


Figure 1-7. Logical partitioning (LPAR) AU1412.0

Notes:

Introduction

This visual illustrates that LPARs can have resources allocated based on the needs of the workload rather than the amount contained in a physical building block. In the diagram above, there are four partitions, each with various amounts of resources.

Adding or removing resources dynamically

On the IBM System p server implementation of LPARs, you can dynamically add and remove resources (CPUs, memory, and I/O slots) to and from a partition while the operating system is running.

For dynamic partitions, the partition must be running AIX 5L V5.2 (or later) and both the managed system and the HMC must be running a version of firmware dated October 2002 or later. All partitions running AIX 5L V5.1 and Linux are static partitions which

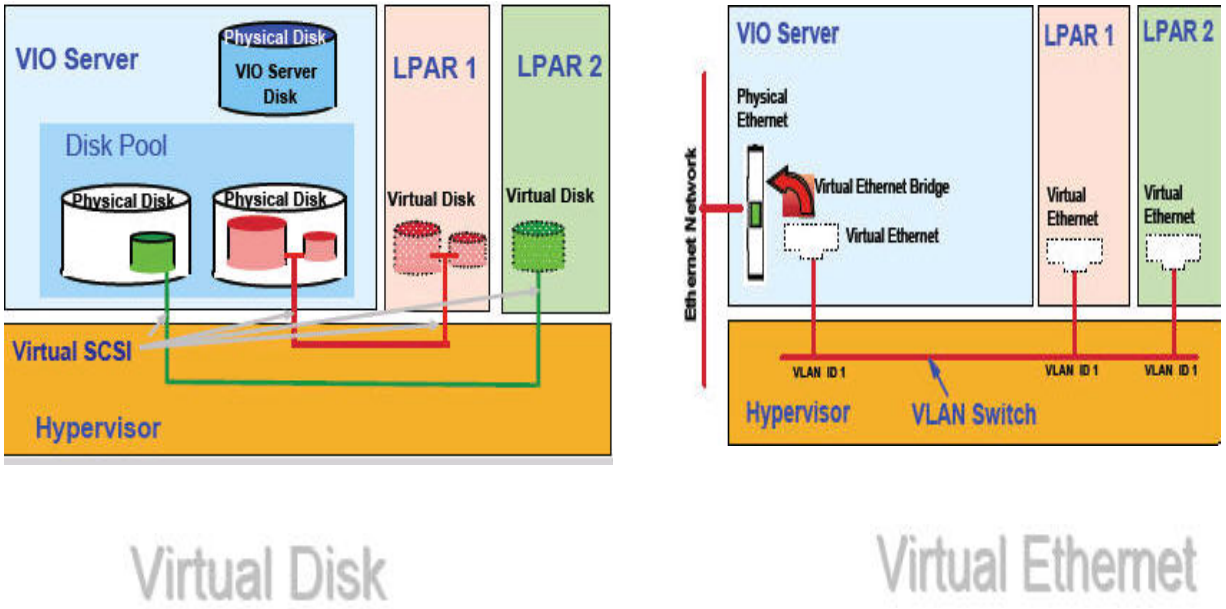
means the partitions must be reactivated (that is, rebooted) to change the resource configuration.

When memory is moved from one partition to another with dynamic LPAR, memory is written to all zeroes by the system firmware. Likewise, I/O adapters are fully reset when moved.

Allocating disks

Disks are not allocated to partitions individually. Instead, the I/O slot containing the adapter controlling one or more disks is allocated to a partition.

Logical partition virtualization



Virtual Disk

Virtual Ethernet

Figure 1-8. Logical partition virtualization AU1412.0

Notes:

Virtual Disk

The Virtual Disk concept is very popular and growing. After the virtual SCSI server adapter has been created, we can configure storage resources to be added to the adapter as virtual SCSI target devices. A virtual SCSI target disk device can be backed by either an entire physical volume or a logical volume.

Virtual Ethernet

Virtual Ethernet enables interpartition communication without the need for physical network adapters assigned to each partition. Due to the number of partitions possible on many systems being greater then the number of I/O slots, virtual Ethernet is a convenient and cost saving option to enable partitions within a single system to communicate with one another.

POWER6 system highlights

- POWER6 processor technology
  - 5th implementation of multi-core design
  - ~100% higher frequencies
- POWER6 system architecture
  - New generation of servers
  - New IO
    - PCIe, SAS / SATA
    - GX+ 12x IO drawers
  - Enhanced power management
- Enhanced virtualization
  - Partition Mobility (SoD)
  - Dedicated shared processors
  - Integrated Virtual Ethernet
- Availability
  - New RAS features
    - Processor instruction retry
  - Power management

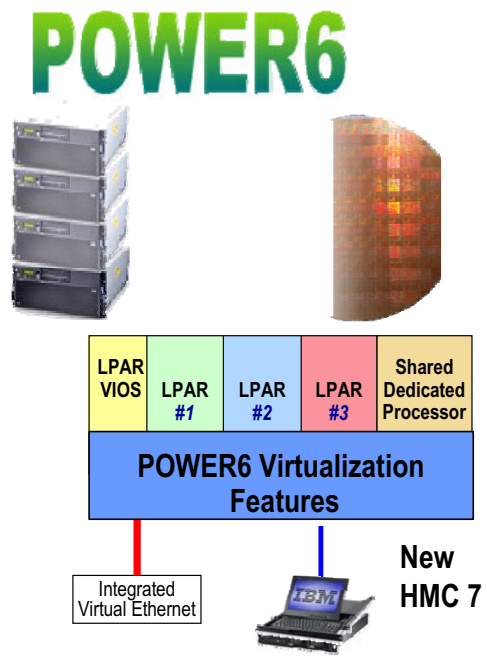


Figure 1-9. POWER6 system highlights AU1412.0

Notes:

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

- A new processor design. POWER6 Processor is the 9th generation 64-bit processor and 5th generation POWER processor
- A new system architecture
- A new virtualization plateau with enhanced capabilities
- A new HMC V7 code
- A new PHYP microcode
- Operated by a new AIX 6 version

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

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

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

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

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

AIX 6 highlights

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

Figure 1-10. AIX 6 highlights AU1412.0

Notes:

AIX 6 Overview

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

Workload Partitions

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

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

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

Security

AIX 6.1 provides several significant security enhancements.

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

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

The AIX 6.1 also provides Trusted Environment (TE) as alternative to Trusted Computing Base (TCB).

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

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

HMC management

- Hardware Management Console (HMC)
- Partition configuration and control
  - Dynamic partitioning for LPARs (AIX 5L V5.2 and later)
- Capacity Upgrade on Demand (CUoD)
- Diagnostics
- Operational management
- Remote HMC control

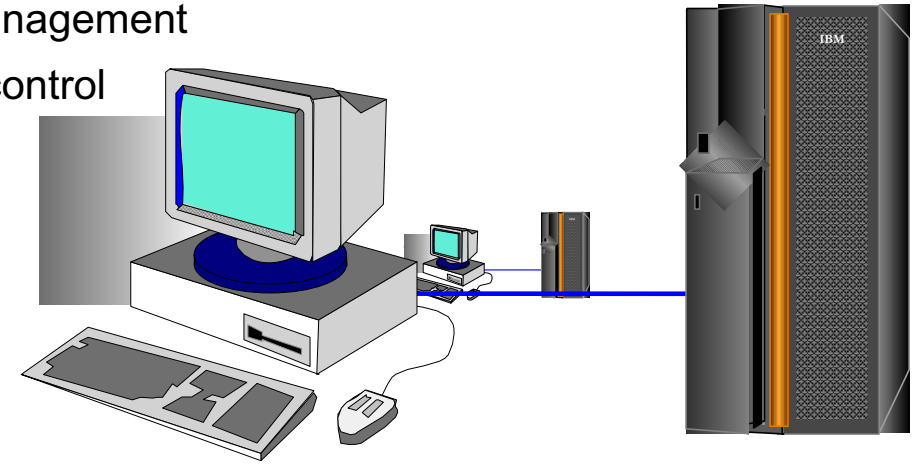


Figure 1-11. HMC management AU1412.0

Notes:

Partition configuration and control

The HMC provides the external platform to configure partitions, in an LPAR environment.

Capacity Upgrade on Demand

Capacity Upgrade on Demand (CUoD) allows you to use the HMC to non-disruptively activate extra resources while the system is operating. If you ordered a CUoD-capable system, additional resources were shipped with the system and can be enabled by using special CUoD activation codes.

Note: CUoD is not supported on systems running Linux in the full system partition.

You can use the HMC to perform the following Capacity Upgrade on Demand functions:

- Display license agreements
- Display the extra resources preinstalled on your managed system

- Type a resource activation code
- Activate extra resources
- Display CUoD status messages

Diagnostics

A challenge faced with the System p system running LPARs is standard AIX error handling. The HMC interacts with each active partition to handle problem determination functions.

Operational management

Once your partitions are active, the HMC continues to function as a management platform, handling operational tasks.

Remote HMC control

Remote access to HMC functions is provided via two paths:

- Remote the Web-based System Manager GUI:
  - From AIX 6.1 and 5L with the Web-based System Manager client installed
  - From a Microsoft Windows or Linux workstation with the Web-based System Manager client installed
  - From another HMC
- High-level commands

Remote access to the HMC

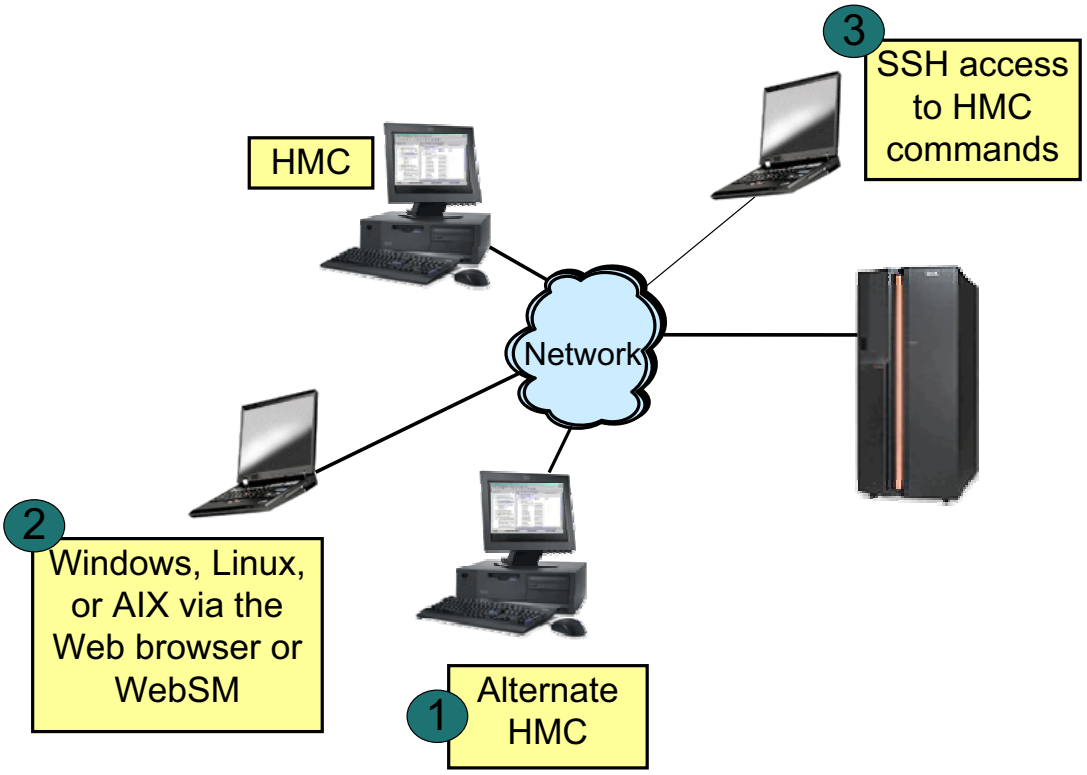


Figure 1-12. Remote access to the HMC AU1412.0

Notes:

Overview

It is often desirable to be able to access the HMC from a remote workstation. This is especially true when the managed system is a POWER4 platform where the HMC is required to be physically close to the managed system.

Remote operation is possible using a Web-based System Manager graphic interface or using line commands via a Secure Shell facility. The Web-based System Manager client may be another HMC, an AIX system running the Web-based System Manager, or a Linux/Windows platform with the Remote Client software installed.

Installing remote client

You may install the Web-based System Manager remote client software on a Windows PC or on a Linux workstation. From the Web-based System Manager, you may connect to any HMC and access its functions. You must login to the HMC (both to install the

remote client and to access the HMC each time). The Web-based System Manager client can be loaded from the AIX standard distribution software or directly from the HMC for Windows PCs and Linux (on Intel) workstations. To install the client on Windows PCs and Linux workstations from the HMC, open a Web browser to the following Web address where *hmc-hostname* is the actual hostname of the HMC:

http://hmc-hostname/remote_client.html

HMC default console view

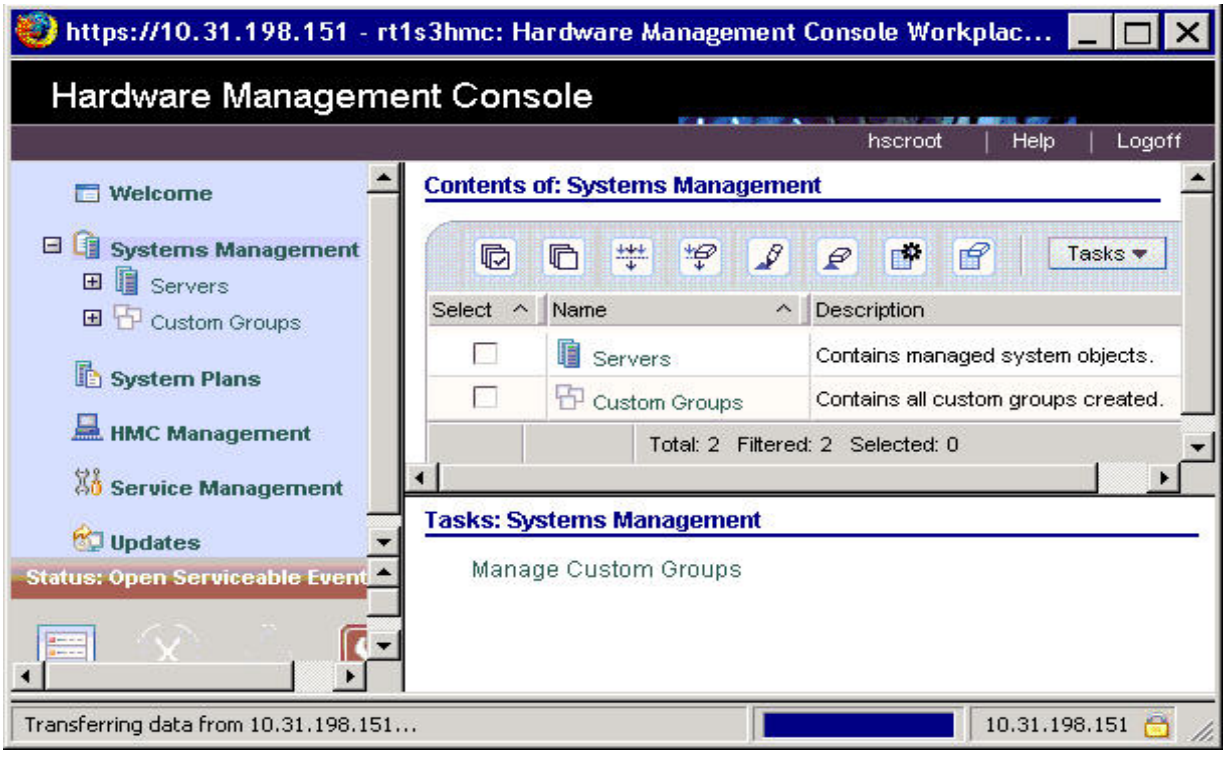


Figure 1-13. HMC default console view AU1412.0

Notes:

Default HMC console view

When you log in to the HMC, the HMC Graphical User Interface (GUI) management window opens and selects the management environment automatically. This window is divided into two main areas: the Navigation area and the Contents area.

The panel on the left (the Navigation Area) displays a hierarchy of icons that represent collections of computers, individual computers, managed resources, and tasks. Each Navigation area icon identifies a tool. At the highest point, or root of the tree, is the Systems Management. The Systems Management tool contains one or more host computer tools that are managed by the console. Each computer tool contains multiple application tools that contain managed objects, tasks, and actions for a related set of system entities or resources.

Role of the system administrator

- Pre-installation planning of:
  - User accounts/groups
  - Storage allocation/paging space
  - Subsystem (printing, networks, and so forth)
  - 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 1-14. Role of the system administrator AU1412.0

Notes:

Overview

There are a number of distinct tasks which the system administrator on a UNIX or AIX system must perform. Often there is 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 the **root** password secure
- Some tasks can be performed by other users in special groups such as **system**, **security**, **printq**, and **lp**
- The **su** command allows you to obtain **root**'s permissions or permissions of any user whose password you know

```
$  su root
```

or

```
$  su  -  root
```


Figure 1-15. Who can perform administration tasks? AU1412.0

Notes:

Limiting access to administrative tasks

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 AIX Print Subsystem printer management, and **lp** for System V Print Subsystem 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.

Obtaining root privileges

To obtain superuser **(root)** privileges while logged in as a normal user, you can use the **su** command. This prompts 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 prevents 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 "**-**" (dash) 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.

Checkpoint

1. What type of adapter are you likely to require for communicating from a logical partition?
   a. Asynchronous
   b. Graphics
   c. Ethernet

2. True or False? The adapters seen by the AIX operating system, in an LPAR, may be either physical or virtual.

   _____

3. True or False? The su command allows you to get root authority even if you signed on using another user ID.

Figure 1-16. Checkpoint AU1412.0

Notes:

Exercise 1: root login methods

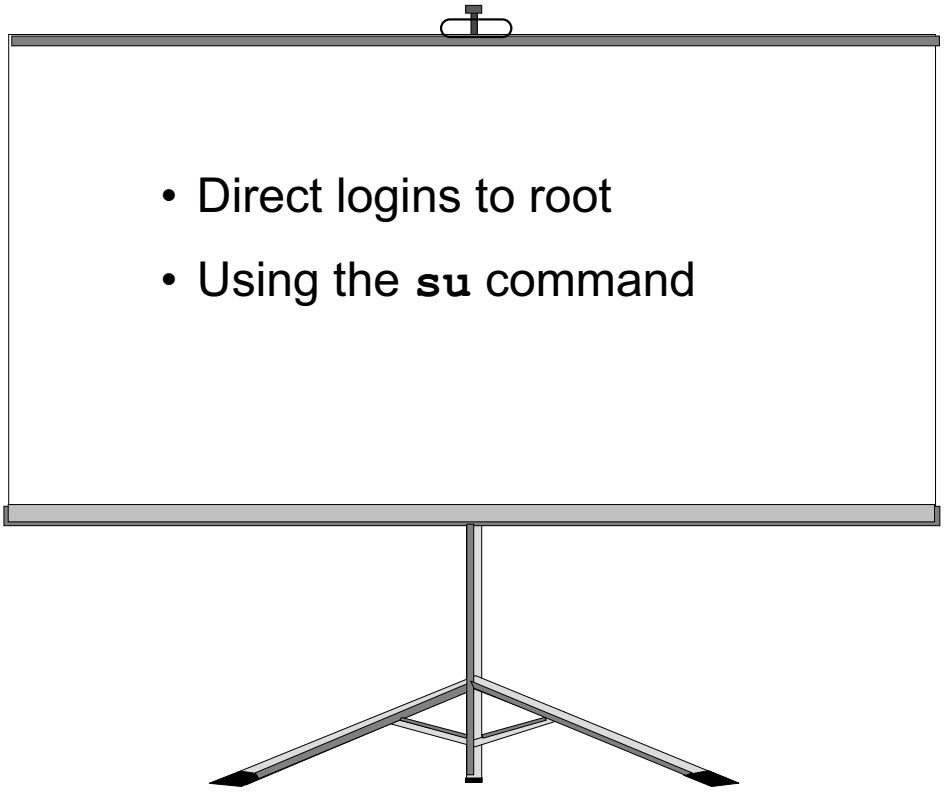


Figure 1-17. Exercise 1: root login methods AU1412.0

Notes:

Introduction

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

Unit summary

- Common configurations
  - Single-user graphics workstation
  - Multiuser ASCII
  - Networked system
  - X Window-enabled PC
- New features for:
  - POWER6
  - AIX 6
- System administrator's role:
  - Pre-installation planning
  - Install hardware, software, network
  - Manage user accounts, system resources, licenses
  - Backup/recovery
  - Define subsystems
  - Performance monitoring, capacity planning

Figure 1-18. Unit summary AU1412.0

Notes:

Unit 2. System Management Interface Tool (SMIT)

What this unit is about

This unit covers the use of SMIT.

What you should be able to do

After completing this unit, you should be able to:

- Describe the benefits of the system management tools available with AIX V6.1
- Discuss the functionality of SMIT
- Explain how SMIT activity is logged

How you will check your progress

Accountability:

- Checkpoint questions
- Exercise

References

Online	AIX Version 6.1 Commands reference
Online	AIX Version 6.1 Operating System and Device management
Online	AIX Version 6.1 Systems Director Console for AIX

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

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

Unit objectives

After completing this unit, you should be able to:

- Describe the benefits of the system management tools available with AIX version 6.1
- Discuss the functionality of SMIT
- Explain how SMIT activity is logged

Figure 2-1. Unit objectives AU1412.0

Notes:

Early system administration

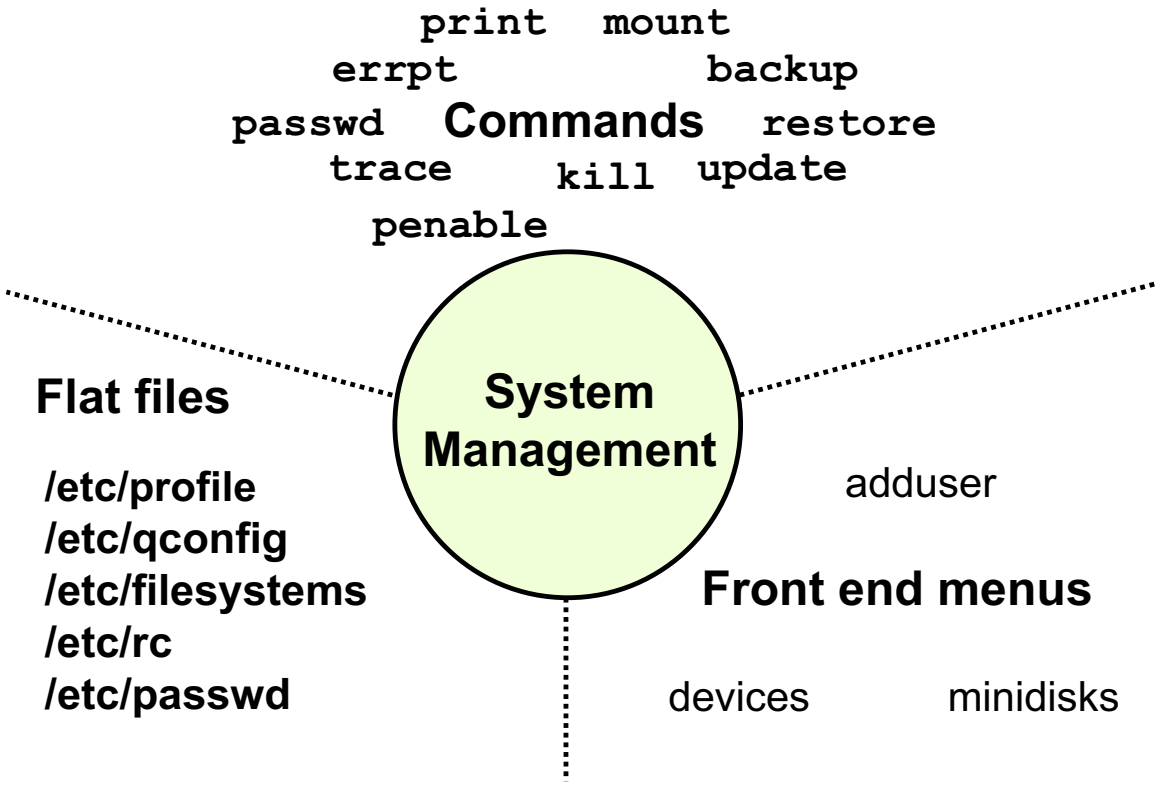


Figure 2-2. Early system administration AU1412.0

Notes:

Problems with early system administration

The key problems with system administration on UNIX and AIX systems before AIX V3 were the following:

- There was not a consistent common interface for performing system administration tasks
- Use of the available methods required the administrator to be very knowledgeable about how the system worked and about the format of various configuration files

Early system administration techniques

The following techniques were used to perform system administration tasks on early UNIX and AIX systems:

- *Commands* - A number of commands were available which performed some system management functions. These had various origins (for example, from AT&T, from Berkeley, and from 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 process was very prone to typing errors and also required knowledge of one of the system editors.

Development of front ends for AIX systems

The first front ends available for AIX systems were for device handling and user creation. Even after these early front ends were introduced, however, most system administration tasks still had to be performed using one of the methods described above.

Now, however, most basic administrative tasks can be performed through use of front ends.

System management objectives

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

Figure 2-3. System management objectives AU1412.0

Notes:

Minimize time and resources spent managing systems

Organizations seek to minimize the time and resources spent managing systems, that is, to manage computer systems efficiently. AIX helps with tools such as SMIT, the Web-based System Manager and AIX 6.1 Systems Director.

Maximize reliability, performance, and productivity

Organizations also wish to maximize system reliability and performance in order to maximize the productivity of the *users* of computer systems. AIX helps with features, such as the logical volume manager, that help avoid the need for the system to be brought down for maintenance.

Provide remote system management solutions

Today's information technology environment also creates a need for 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 also be done with the **`telnet`** program and SMIT.

AIX administration

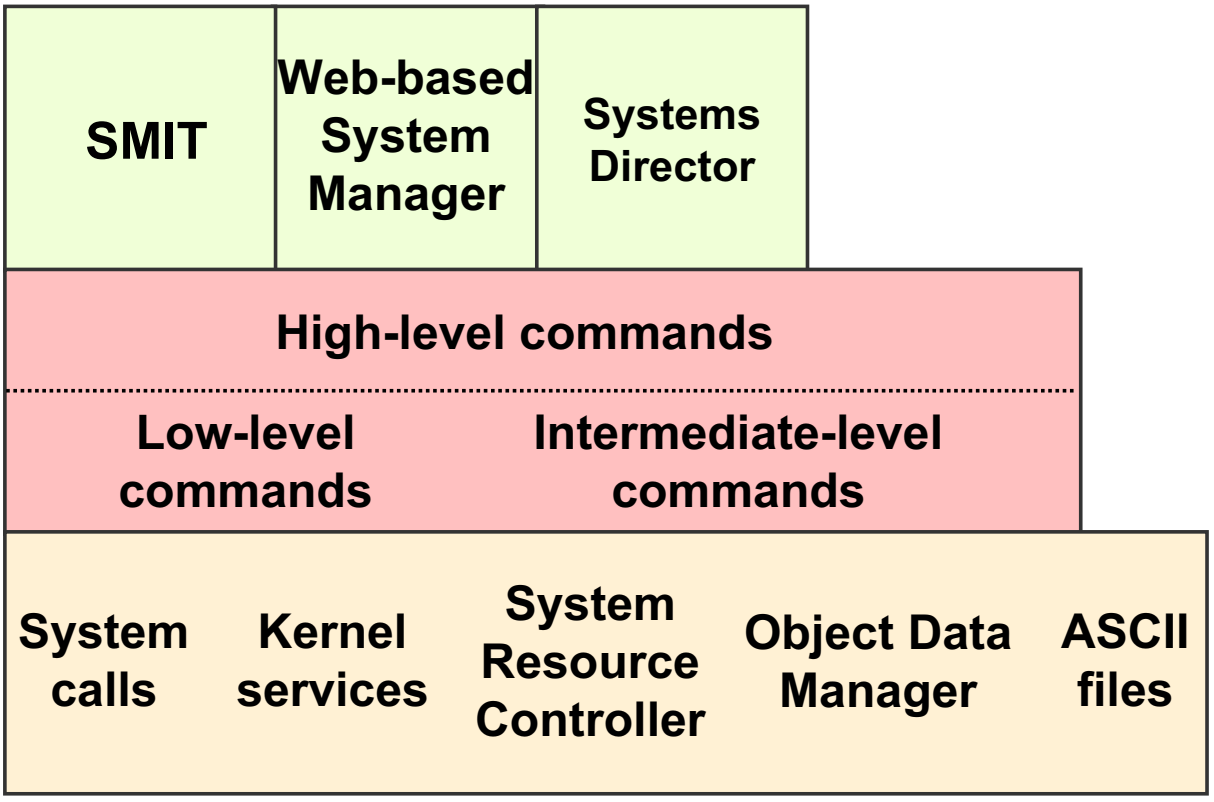


Figure 2-4. AIX administration AU1412.0

Notes:

Overview of SMIT

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.

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

Types of commands

The following classification of commands may be helpful in understanding the operation of SMIT:

- *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. SMIT constructs high-level commands from the user's selections and then executes these commands on demand.
- *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 to AIX system calls or kernel services. (They are not normally executed directly by a user.)

Overview of Web-based System Manager

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. In this unit we focus on SMIT, but the Web-based System Manager is discussed in further detail later in this course.

System Management Interface Tool (SMIT)

ASCII or AIXwindows (Motif) user interface components

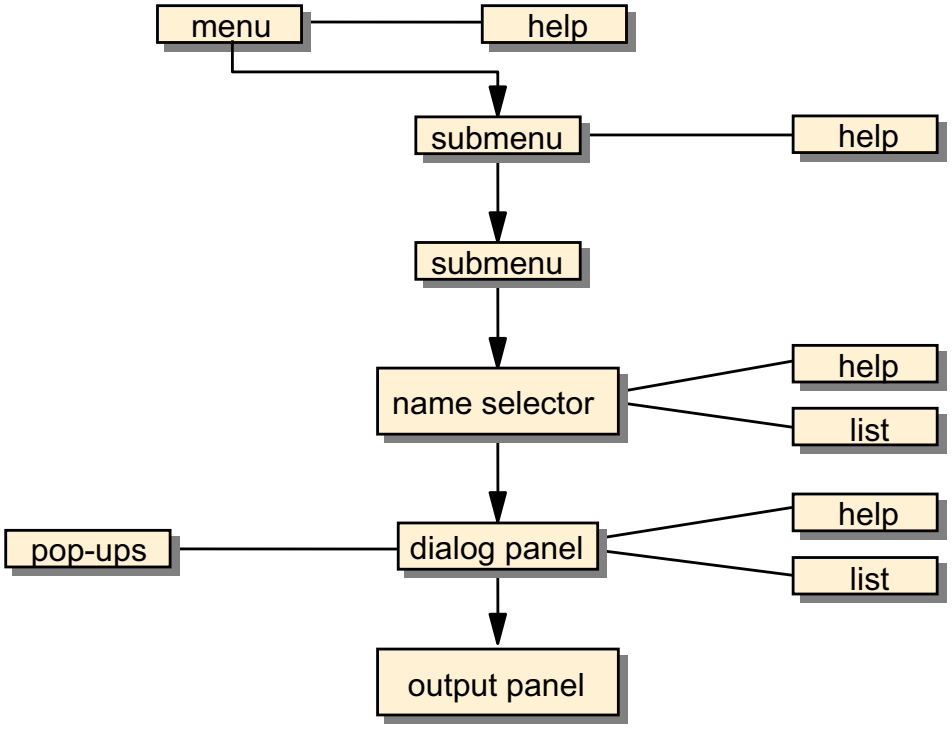


Figure 2-5. System Management Interface Tool (SMIT) AU1412.0

Notes:

Available interfaces

SMIT provides a flexible user environment. The user can use an ASCII or an AIXWindows-based interface. These interfaces provide the same facilities, but the interaction is slightly different.

Components of user interface

The SMIT 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* allows you to control the way in which the task is performed (for example, to set the attributes for the drive, or to 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 that guides you through the use of SMIT, and also provides contextual information about each submenu, dialog screen, and also each field within a dialog screen.

SMIT main menu (ASCII)

```
# smit
```

```
                          System Management

 Move cursor to desired item and press Enter.

   Software Installation and Maintenance
   Software License Management
   Devices
   System Storage Management (Physical & Logical Storage)
   Security & Users
   Communications Applications and Services
   Workload Partition Administration
   Print Spooling
   Advanced Accounting
   Problem Determination
   Performance & Resource Scheduling
   System Environments
   Processes & Subsystems
   Applications
   Installation Assistant
   Cluster Systems Management
   Using SMIT (information only)

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


Figure 2-6. SMIT main menu (ASCII) AU1412.0

Notes:

Main menu selections

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.

Use of keys

In the ASCII mode, in order to select from the menus, you have to use the up and down arrow keys. This moves 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.

Importance of `TERM` environment variable

When using SMIT in the ASCII mode, the menus and dialog panels sometimes come up distorted. That is the result of not having an appropriate TERM variable value. Setting and exporting this variable can solve the problem. For example, executing the command **`export TERM=vt320`** might solve the problem.

SMIT main menu (Motif)

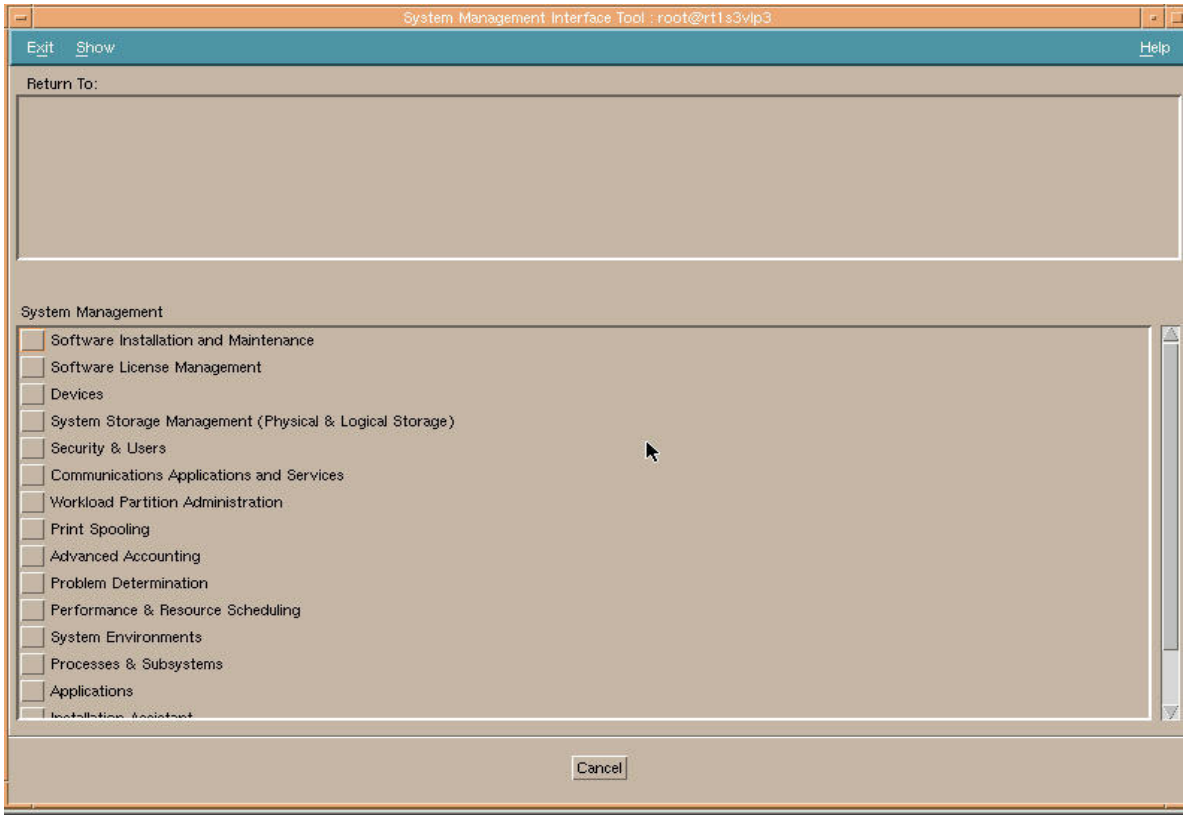


Figure 2-7. SMIT main menu (Motif) AU1412.0

Notes:

Need for graphical environment

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

Working with the graphical version of 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. You can also move back to any previous screen in the menu hierarchy by selecting 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.

Graphical and ASCII SMIT differences

Notice that, in the graphical version of SMIT, the function keys have been removed. The layout of the menu is slightly different too. In the Motif version of SMIT, you must use the mouse to click the desired options, whereas the ASCII version uses the cursor keys (as the mouse feature is disabled).

Dialog screen

```
                          Schedule a Job

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

                                              [Entry Fields]
  YEAR                                            [07]              #
  MONTH                                           [Jun]            +
  DAY (1-31)                                      [22]              #
* 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 AU1412.0

Notes:

Dialog screens and selector screens

A *dialog screen* allows you to enter values that are used in the operation performed. Some fields are already 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 is acted upon by the subsequent dialog and AIX command.

Entering data

To enter data, move the highlighted bar to the value you want to change. 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

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

*	This is 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 is not 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-l>`** to get a pop-up window to select from.

Use of particular keys

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 is executed
F7 (or ESC-7)	Edit - 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-l	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 AU1412.0

Notes:

Fields on first line of output

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

The value of the **stdout** field indicates whether there is standard output, that is, whether there is output produced as a result of running the command. The output is displayed in the body section of this screen.

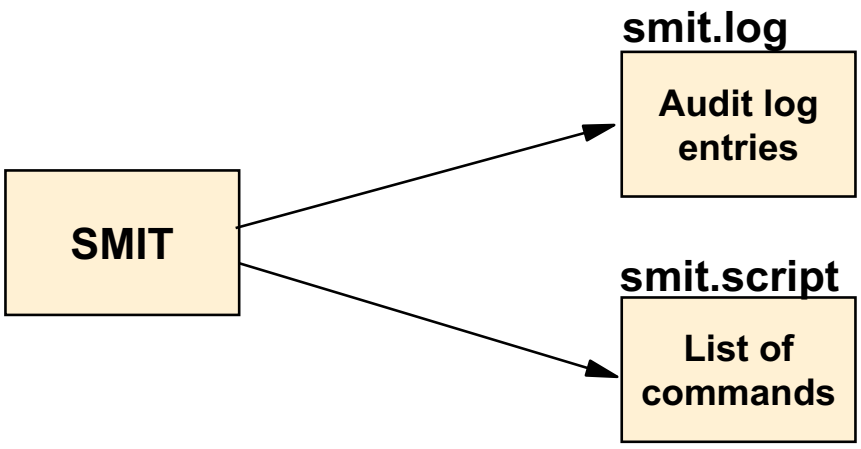
The value of the **stderr** field indicates whether there are error messages. In this case, there are no error messages.

Note that, in the Motif version of SMIT, a representation of a man in the top right-hand corner of the screen is used to indicate the values of the **Command** field.

Body of the screen

The body of the screen holds the output/error messages from the command. In this example, there is output, but there are no error messages.

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 AU1412.0

Notes:

Overview

SMIT creates two files in the **$HOME** directory of the user running SMIT. If these files already exist, then SMIT appends 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

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

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 it can be used as the basis for more complex operations.

`smit` command options

- General syntax:

  `smit [-options] [ FastPath ]`

- Invoke ASCII version:

  # **`smitty`**

  or

  # **`smit -C`**

- Log (but do not actually run) commands:

  # **`smit -x`**

- Redirect the log file and script file:

  # **`smit -s /u/team1/smit.script -l /u/team1/smit.log`**

  # **`smit -s /dev/pts/1 -l /dev/pts/2`**

Figure 2-11. `smit` command options AU1412.0

Notes:

Introduction

The command `smit` is used to invoke SMIT. It is not particularly common to run `smit` with any options, although a number of them do exist. Some of the more commonly used options are described here.

Using a fastpath

Using a SMIT *fastpath* can be very helpful. Fastpaths are names that specify 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 (for the current screen) by pressing **F8 (or Esc+8) - Image**.

Specifying the ASCII version of SMIT

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 option is used fairly often.

Specifying that commands should not actually be executed

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

Specifying the log and script files

Since **smit.log** and **smit.script** are created in the user's home directory, the *natural* growth of these files can create a problem if you log in directly as **root**. Recall that **root**'s home directory is /. Later, you learn that filling the root area of your disk can cause your machine to crash. You can tell SMIT to log **smit.log** and **smit.script** information elsewhere using **`smit -l filename`** to specify the location of **smit.log** and **`smit -s filename`** to specify the location of **smit.script**.

IBM Systems Director Console for AIX

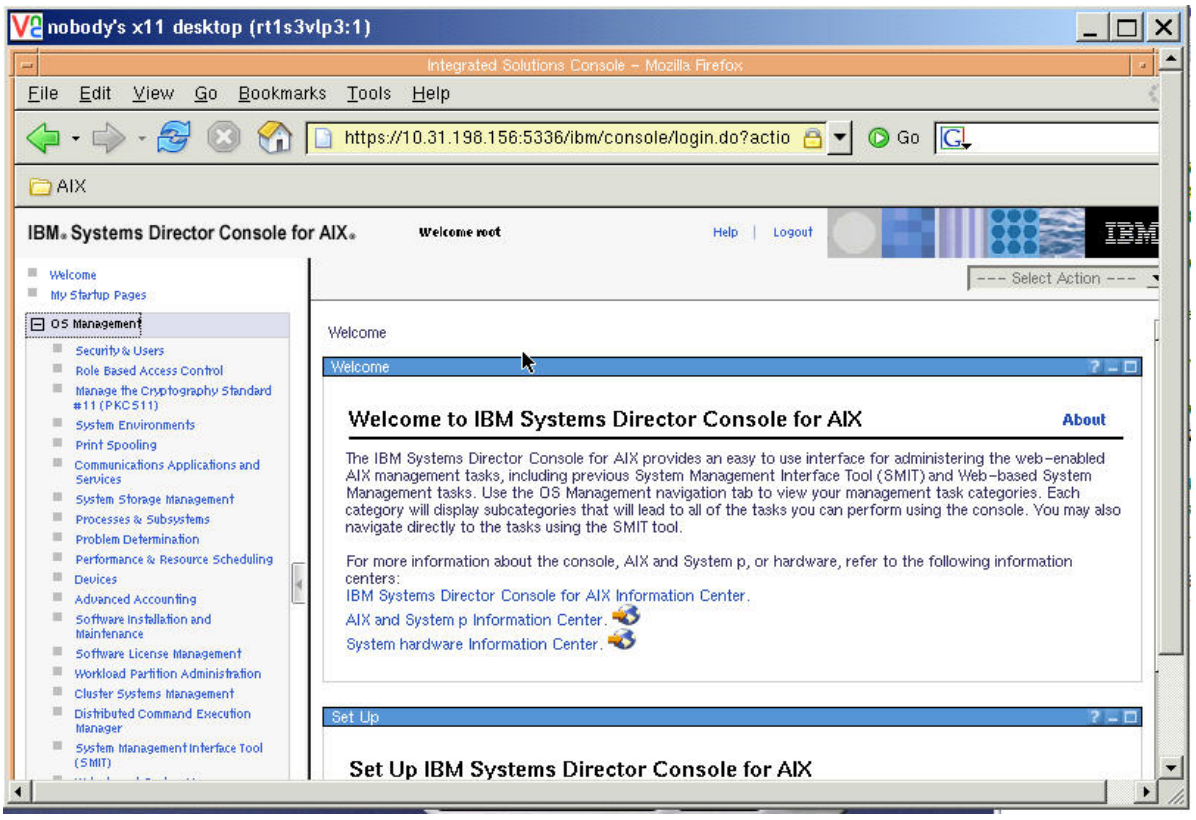


Figure 2-12. IBM Systems Director Console for AIX AU1412.0

Notes:

IBM Systems Director Console

The **IBM Systems Director Console for AIX** is a new management tool in AIX 6.1. It is automatically installed after AIX 6.1 installation is completed. When the network configuration is finished, you can access it using a Web browser. Internet Explorer Version 7, and Mozilla Firefox are the supported browsers. You need to enter your user name and password. The URLs are:

- **http://<hostname>:5335/ibm/console**
- **https://<hostname>:5336/ibm/console**

IBM Systems Director Console provides an easy to use interface for AIX management tasks. The management task categories are listed under the navigation tab. Each category has subcategories that include all the tasks you can perform using the console.

The Layout of the IBM Systems Director Console for AIX

The IBM Systems Director Console for AIX has a **Console toolbar, Navigation area, and Work area**.

Console tool bar across top:

- User name (for example. “Welcome root”)
- Help - Infocenter - ISC Help - *AIX Administrator Guide* - Logout

Navigation area:

- Tasks categories
- Welcome - My Startup Pages - OS Management - Health - Settings

Work area:

- Page bar - Multiple pages/tabs - Action Selection List
- Portlets - Administrators can operate any tasks on the portlets

Checkpoint

1. Specify the SMIT function keys that can be used for the following:
   a) List the command that will be run: ____
   b) List the screen name which can be used for the fastpath: ____
   c) Take a screen image: ___
   d) Break out into a shell: ___
   e) Return to the previous menu: ___
2. Specify two ways you can request the ASCII character version of SMIT from an X-windows environment command prompt:
   – _____
   – _____

Figure 2-13. Checkpoint AU1412.0

Notes:

Exercise 2: Using SMIT

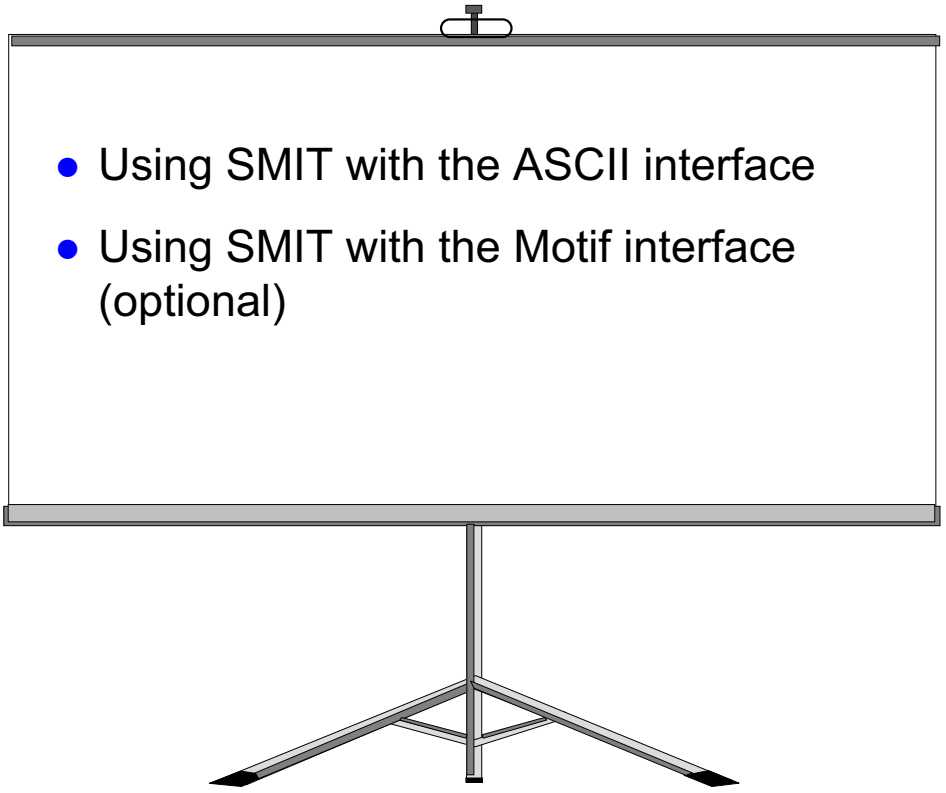


Figure 2-14. Exercise 2: Using SMIT AU1412.0

Notes:

Introduction

This exercise allows you to get familiar with SMIT.

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

Unit summary

- Most system administration tasks can be completed using either the ASCII or graphical (Motif) version of SMIT
- SMIT provides logging of activities and generated commands
- SMIT has useful fastpaths for bypassing the menu structures

Figure 2-15. Unit summary AU1412.0

Notes:

Unit 3. System startup and shutdown

What this unit is about

This unit describes the system startup and shutdown processes and 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
- Explain 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 *AIX 6.1 Operating system and device management*

Note: References listed as "Online" above are available at the following address:

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

Unit objectives

After completing this unit, you should be able to:

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

Figure 3-1. Unit objectives AU1412.0

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 AU1412.0

Notes:

Normal mode

When you power on your system, one of the first things it does is determine which device it should use to boot the machine. By default, the machine uses the normal boot list, which usually contains one or more hard drives. When the machine does a normal boot, it completes the full AIX boot sequence and start processes, enables terminals and generates 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.

System Management Services

Another option 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 into the firmware. This resource 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 or selecting SMS boot mode using the HMC console.

Maintenance mode

If your system does not boot or you have lost the **root** password, you need to boot your machine using bootable media other than the hard drive (like an installation CD or bootable backup - **`mksysb`** tape). This boots 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 are given backdoor access to your system.

Typically, by pressing the **F5** key, you use the default firmware bootlist, which always contains the CD as boot media. Pressing **F5** causes the machine to use the default bootlist on some of the smaller older machines. On some other machines, this action causes use of the service bootlist. The following example illustrates building of the service bootlist:

```
# bootlist -m service cd0 hdisk0 hdisk1
```

Diagnostics

There is one other boot option - to boot into *diagnostics*. This can be accomplished by using bootable media specifically designed for diagnostics or by invoking the diagnostic mode when the hard drive is the boot device during a service boot. The System Administration II class (Q1316/AU16) includes a discussion of diagnostics.

Boot lists

All machines have a normal boot list and one or more service boot lists. By default, the normal boot list is used. The service boot list is invoked (like SMS) during the initial stages of the boot sequence using function key **F6**.

Use of numeric keys

Not all consoles use function keys, such as **F1**, **F5**, and **F6** for controlling the boot mode. On serial attached tty consoles (which do not have function keys) and on many newer pSeries machines (even with graphics terminals), the equivalent numeric keys are used for this purpose. Thus, for a service boot using the default bootlist, you might need to press the numeric **5** key rather than the **F5** key; similarly, for access to SMS, you might need to press the numeric **1** key instead of the **F1** key.

Starting System Management services

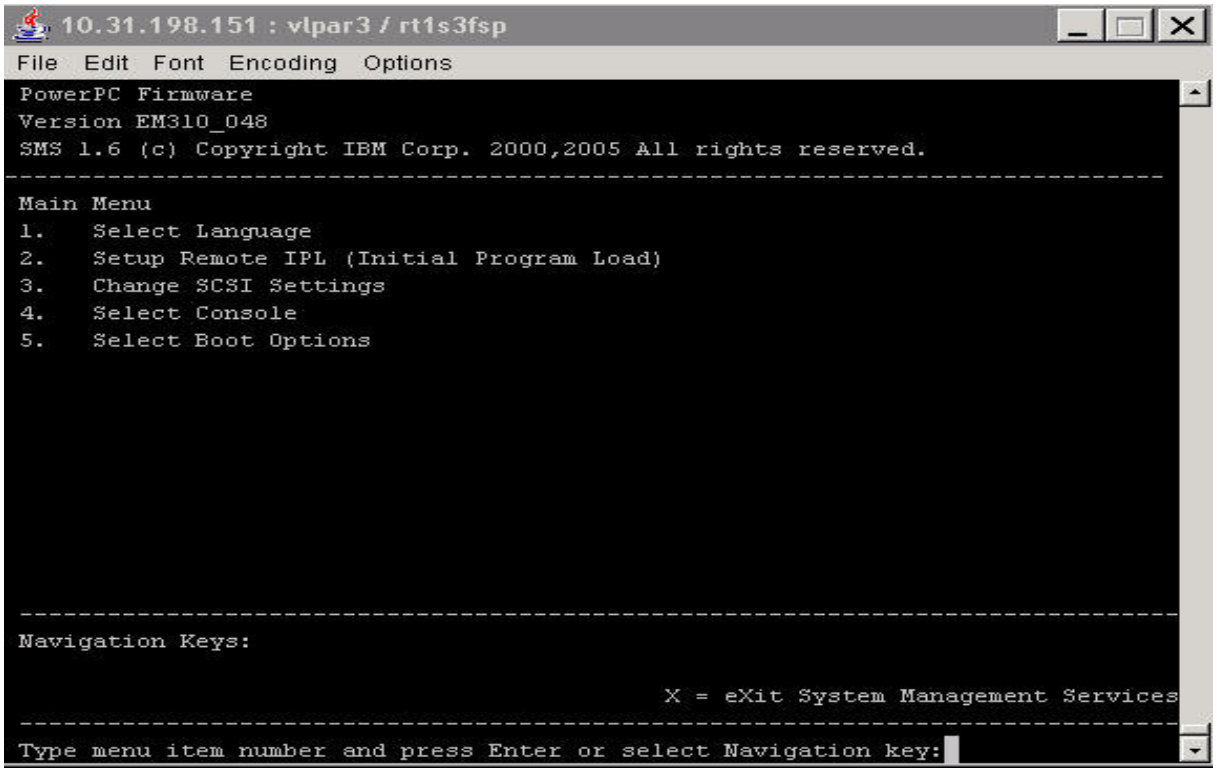


Figure 3-3. Starting system management services AU1412.0

Notes:

Introduction

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*.

Invoking SMS

To invoke SMS, power on (or reboot) the system. You hear one beep when the machine first powers on. About 30 to 45 seconds later, when POST discovers the keyboard, you hear a different tone. This is what you are listening for. Also, you 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. If you are in a noisy room, and you do see the keywords or

icons for the discovered devices on the screen (for example, on an LED display), wait for the keyword or icon for the keyboard to be displayed before pressing F1. *Timing is everything!*

SMS main menu

The visual shows an example of a System Management Service main menu. The exact configuration of the menu varies depending on the model of PCI RS/6000, pSeries, or System p server being used. The main services in this example are the following:

- *Select Language*
- *Setup Remote IPL*
- *Change SCSI XSettings*
- *Select Console*
- *Select Boot Options*
- *Exit* - SMS

System p server start up process overview

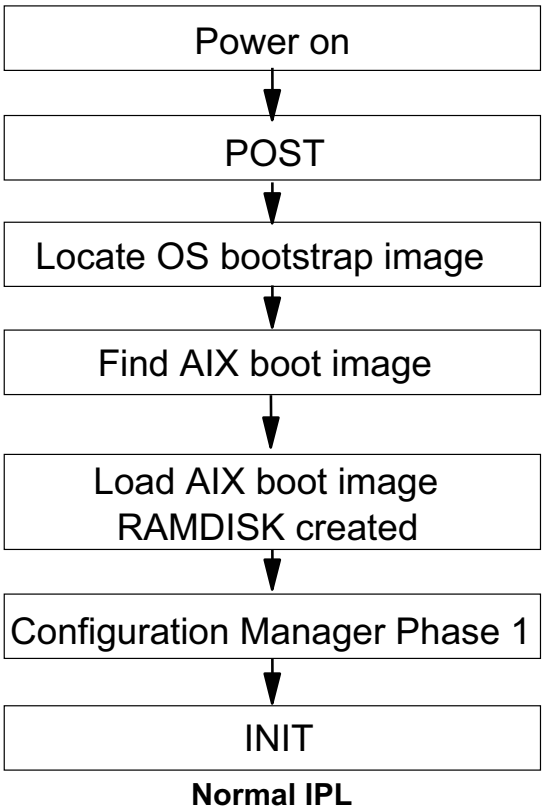


Figure 3-4. PCI RS/6000 start up process overview AU1412.0

Notes:

Introduction

During the boot process, a number of steps must be completed. The LED panel provides information on the boot progress. Some LED values displayed are model specific. These values can be found in the *Service Guide* for that specific model. Other codes, particularly those displayed in later stages of the boot process, are generic AIX codes, which are the same on all AIX systems.

Power-on Self Test (POST)

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

Bootstrap code

Once the POST is completed, the system locates and loads bootstrap code. This stage is completed by System Read Only Storage (System ROS) 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.

Passing control to the 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 stage are generic AIX codes. These are the same on all AIX systems.

The `bootinfo` command

- To view the architecture type:

```
# bootinfo -p
```

rs6k MCA model
rspc PCI model (POWER Reference Platform)
chrp PCI model (Common Hardware Reference)

- To view the bit addressing:

```
# bootinfo -y
```

32 32-bit
64 64-bit

Figure 3-5. The bootinfo command AU1412.0

Notes:

Supported architecture types

AIX 5L V5.1 supports the rs6k, rspc and chrp architecture types. However, AIX 5L V5.2 to AIX 6.1 support only chrp. These architecture types are described briefly in the table below:

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

Determining the architecture type

The architecture type for a given machine can be determined using the following command:

```
# bootinfo -p
```

The `-y` option of the `bootinfo` command

The following command can be used to determine whether your system has 32-bit or 64-bit hardware:

```
# bootinfo -y
```

The `alog` command

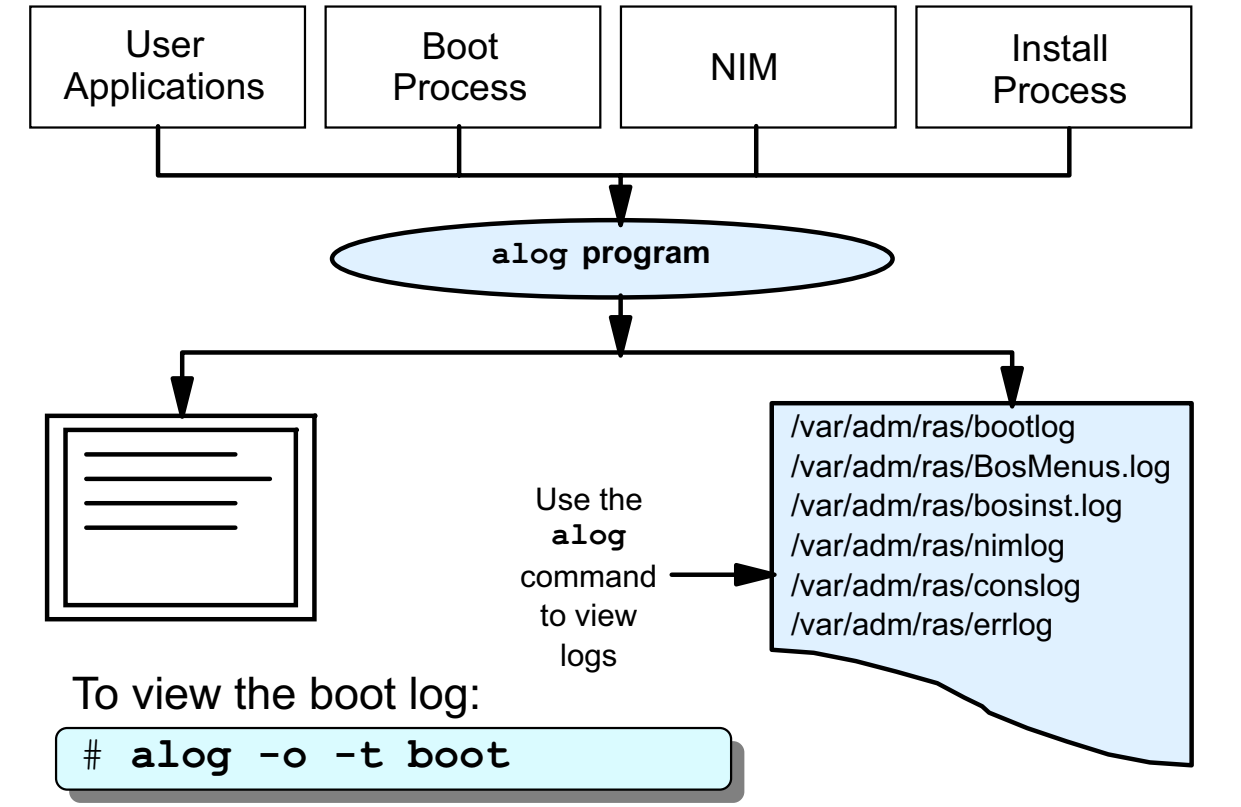


Figure 3-6. The alog command AU1412.0

Notes:

Overview

The **`alog`** command is a BOS feature that provides a general-purpose logging facility 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 log file

The file is treated as a circular log. This means that, 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, nim,** and **console**.

Use in boot process

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 does not boot, boot the machine into maintenance mode and view the **boot** log contents.

Viewing logs with SMIT

You can also use SMIT to view the different system-supported logs. Use the following command:

```
# smit alog
```

/etc/inittab

Format of the line: `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 ...
mkatmpvc:2:once:/usr/sbin/mkatmpvc >/dev/console 2>&1
atmsvcd:2:once:/usr/sbin/atmsvcd >/dev/console 2>&1
load64bit:2:wait:/etc/methods/cfg64 >/dev/console 2>&1 # Enable 64-bit execs
tunables:23456789:wait:/usr/sbin/tunrestore -R > /dev/console 2>&1 ...
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 ...
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
cron:23456789:respawn:/usr/sbin/cron
piobe:2:wait:/usr/lib/lpd/pio/etc/pioinit >/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 # High availability daemon
l2:2:wait:/etc/rc.d/rc 2
l3:3:wait:/etc/rc.d/rc 3
l4:4:wait:/etc/rc.d/rc 4
. . .
```


Figure 3-7. **/etc/inittab** AU1412.0

Notes:

Introduction

The **/etc/inittab** file lists the processes that `init` starts, and it also specifies when to start them.

If this file gets corrupted, the system cannot boot properly. Because of this, it is a good idea to keep a backup of this file.

Format of entries

The individual line entries in **/etc/inittab** contain the following fields:

- `id` - 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 is 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

The **`telinit`** command can be used to cause **`init`** to re-read the **/etc/inittab file**. For example, you may need to do this if **`init`** stops respawning the **`getty`** process on a TTY due to line errors. More information regarding use of the **`telinit`** command is given in the next few paragraphs.

Run levels

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`** places your machine into a single user mode terminating all logins. Once the machine reaches the single user mode, you are prompted to enter **root** password. When you are ready to return to normal mode, type **`telinit 2`**.

Making changes to /etc/inittab

Because **/etc/inittab** 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**. These commands 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 reread the file by using the **`telinit q`** command.

System resource controller

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

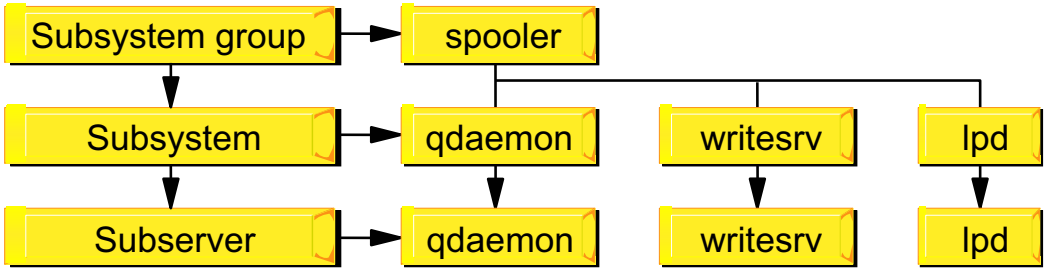


Figure 3-8. System resource controller AU1412.0

Notes:

Purpose of the System Resource Controller

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. Subsystems 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 subsystems or groups of subsystems with a few commands.

Subsystem groups, subsystems, and subservers

The relationship between the *subsystem group* and subsystem is easily seen from the output of **`lssrc -a`**. The visual shows the relationship between the *spooler* subsystem group and its subsystems *qdaemon*, *writesrv*, and *lpd*. Some subsystems have subservers. To illustrate, the *tcpip* subsystem 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 AU1412.0

Notes:

Introduction

In this section, we discuss some examples of SRC commands.

Listing SRC status

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

Specifying a subsystem or subsystem group

The **-s** and **-g** flags are used to specify subsystems or subsystem groups, respectively. As shown on the visual, these flags can be used with various SRC-related commands.

Other examples on the visual

In the remaining examples, you 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 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 AU1412.0

Notes:

Deciding how to stop a process

Because some processes are *started* using SRC, they should be *stopped* using SRC.

If you are not sure how a process was started, you can run **lssrc** to view what is controlled by SRC. As an alternative, you can determine the same information by viewing the output from **ps -ef**.

Discussion of output shown on visual

In the output on the visual, **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 **/usr/bin/program**.

System shutdown

- The **shutdown** command:
  - Gracefully stops all activity on the system and advises all logged on users
  - Warns users of an 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 AU1412.0

Notes:

Introduction

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

If used with no options, **shutdown** displays a message on all enabled terminals (using the **wall** command), then (after one minute) disables all terminals, kills all processes on the system, syncs the disks, unmounts all file systems, and then halts the system.

Some commonly used options

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** flag specifies a “pretend” shutdown. It appears to all users that the machine is about to shut down, but no shutdown actually occurs.

Shutting down to single-user mode

Use the following commands to shut down the system to single-user mode:

```
# shutdown -m
```

Creating a customized shutdown sequence

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, that is, before normal shutdown processing begins. This is useful if, for example, you need to close a database prior to a shutdown. If **rc.shutdown** fails (non-zero return code value), the shutdown is terminated.

Flags

The following flags can be used with the **shutdown** command:

- **-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** Since AIX 5L V5.1, this option 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 is not captured.

Note regarding the -t option

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
   AIX Security Expert
  Assign the Console
  Change / Show Date, Time, and Time Zone
  Manage Language Environment
  Change / Show Characteristics of Operating System
  Change / Show Number of Licensed Users
  Broadcast Message to all Users
  Manage System Logs
  Change / Show Characteristics of System Dump
  Change/Show Documentation Services
  Change System User Interface
  Change/Show Default Browser
  Change/Show Documentation Services
  Web-based System Manager
  Enable 64-bit Application Environment
  Manage Remote Reboot Facility
  Manage System Hang Detection



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


Figure 3-12. Manage the system environment AU1412.0

Notes:

Introduction

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

Functions available from System Environments selection

The following functions can be chosen from the **Systems Environments** selection in SMIT:

- **Stop the System** - Runs the `shutdown` command.
- **AIX Security Expert** - Allows setting the system security level.
- **Assign the Console** - Allows assignment or reassignment of the system console. A reboot is required for it to take effect.

- **Change/Show Date, Time, and Time Zone** - 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 setting.
- **Manage Language Environment** - 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.
- **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.
- **Change/Show Documentation Services** - Allows the **root** user to specify values for the environment variable which configure the infocenter documentation services.
- **Change System User Interface** - Determines whether CDE or command-line login is used.
- **Change/Show Default Browser** - manages system default Web browser.
- **Change/Show Documentation Services** - Allows configuration for a documentation server.
- **Web-based System Manager** - Configures Web-based SMIT for remote configuration.
- **Enable 64-bit Application Environment** - Allows the 64-bit application environment to be enabled either immediately or at system restart.
- **Manage Remote Reboot Facility** - Identifies a serial port and special string to invoke remote reboot.
- **Manage System Hang Detection** - Configures automated action for when a defined set of processes get no cycles.

Manage Language Environment

```
# smit mlang
```

```
                    Manage Language Environment

 Move cursor to desired item and press Enter.

   Change/Show Primary Language Environment
   Add Additional Language Environments
   Remove Language Environments
   Change/Show Language Hierarchy
   Set User Languages
   Change/Show Applications for a Language
   Convert System Messages and Flat Files




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


Figure 3-13. Manage Language Environment AU1412.0

Notes:

Language environment management with SMIT

As shown on the visual, the fastpath **`smit mlang`** can be used to access a number of language environment management functions.

The `LANG` variable

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.

The `chlang` command

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.

Code sets

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.

Installation default 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.

Conversion from one code page to another

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

Euro currency symbol support

The Euro currency symbol “**€**” is supported in the ISO.8859-15, UTF-8 and IBM-1252 codesets.

Hardware Management Console

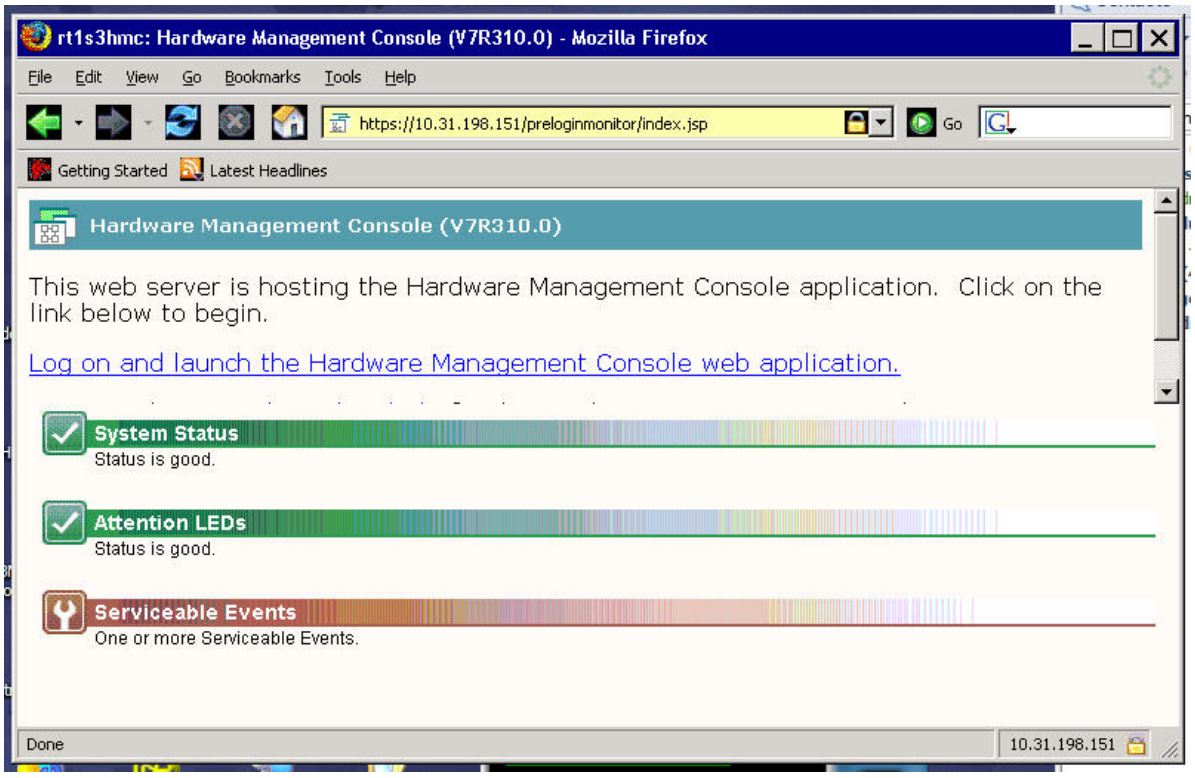


Figure 3-14. Hardware Management Console AU1412.0

Notes:

Hardware Management Console

The HMC is primarily a desktop PC workstation used to provide several functions for configuring, installing, and operating System p servers. There is a graphic user interface (GUI) or command line interface. The HMC can be used for logical partitioning management and displaying system resources booting. It also is used for starting, stopping, resetting, and shutting down a partition.

Working with Logical Partitions

Booting, starting, and shutting down a logical partition are our focus in this chapter. By booting in SMS mode, you can identify the 1st boot device. Starting the system with the network adapter selected as the 1st device, you should be able to install out logical partition using a NIM server. To access your HMC, it is assumed that your managed system is attached. Enter the IP- address using the URL protocol https in a Web browser (https://##.##.##.##).

HMC – LPAR operations menu

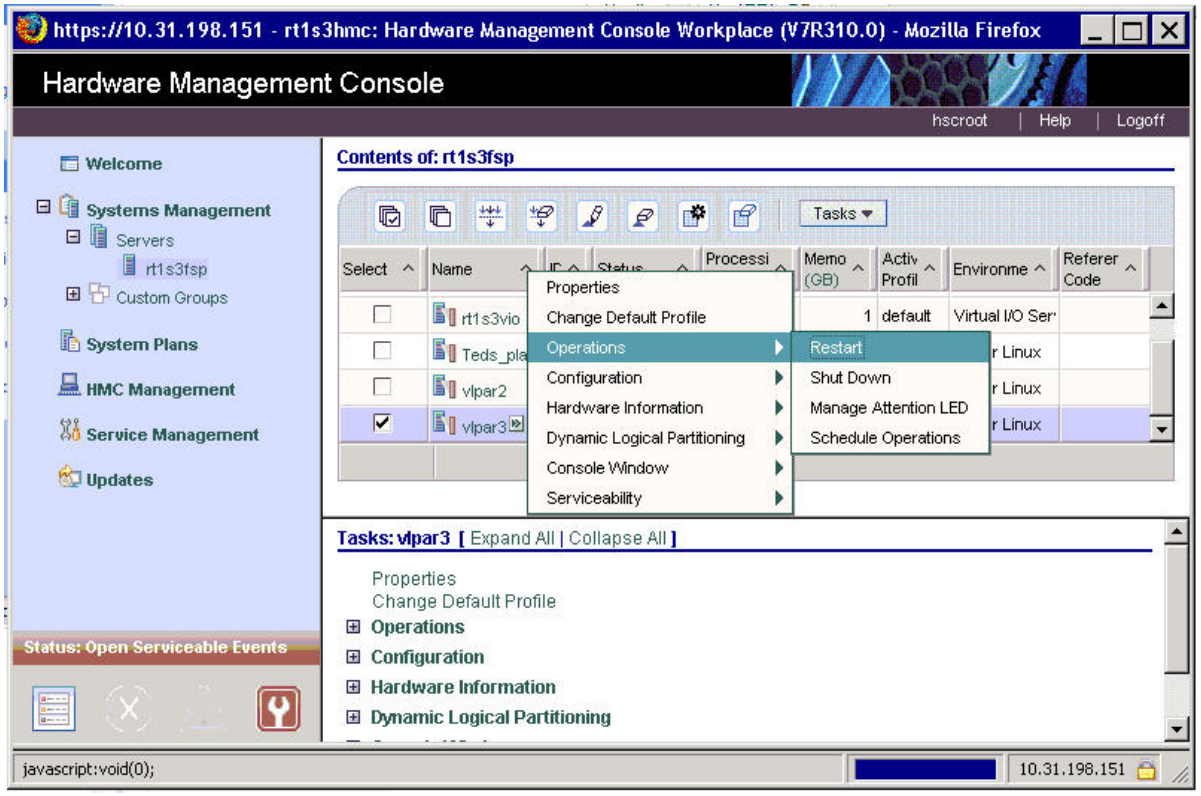


Figure 3-15. HMC - LPAR operations menu AU1412.0

Notes:

HMC Management Tasks

After starting the HMC from a Web browser, then you need to access your managed server. Viewing the image above, you can see that the Systems Management and the Servers option were expand. All the managed system for this HMC is listed.

Server Operations

Clicking in the Select box to the left of LPAR you want to manage produces arrows to the right side of the name. Clicking these arrows displays a menu as shown above. The Operations menu offers the various operation you may perform on your partition. These include activating or stopping the partition, or starting a virtual terminal to access the system console for the partition.

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 3-16. Checkpoint AU1412.0

Notes:

Exercise 3: System startup and shutdown

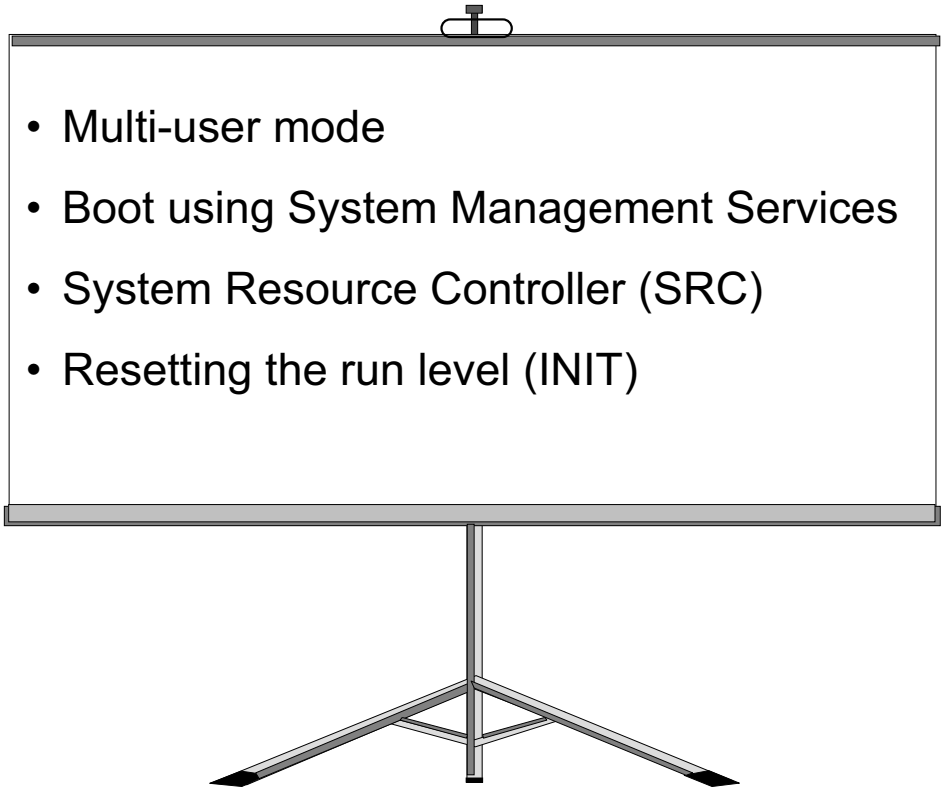


Figure 3-17. Exercise 3 - System startup and shutdown AU1412.0

Notes:

Introduction

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

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 shut down 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.

Figure 3-18. Unit summary AU1412.0

Notes:

Unit 4. 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

SG24-7559	*AIX Version 6.1 Differences Guide*
Online	*AIX 6.1 Installation and Migration*

Note: References listed as "Online" above are available at the following address:

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

Unit objectives

After completing this unit, you should be able to:

- Define the package definitions and naming conventions
- Identify how software products and updates are installed and managed on the system

Figure 4-1. Unit objectives AU1412.0

Notes:

AIX product offerings

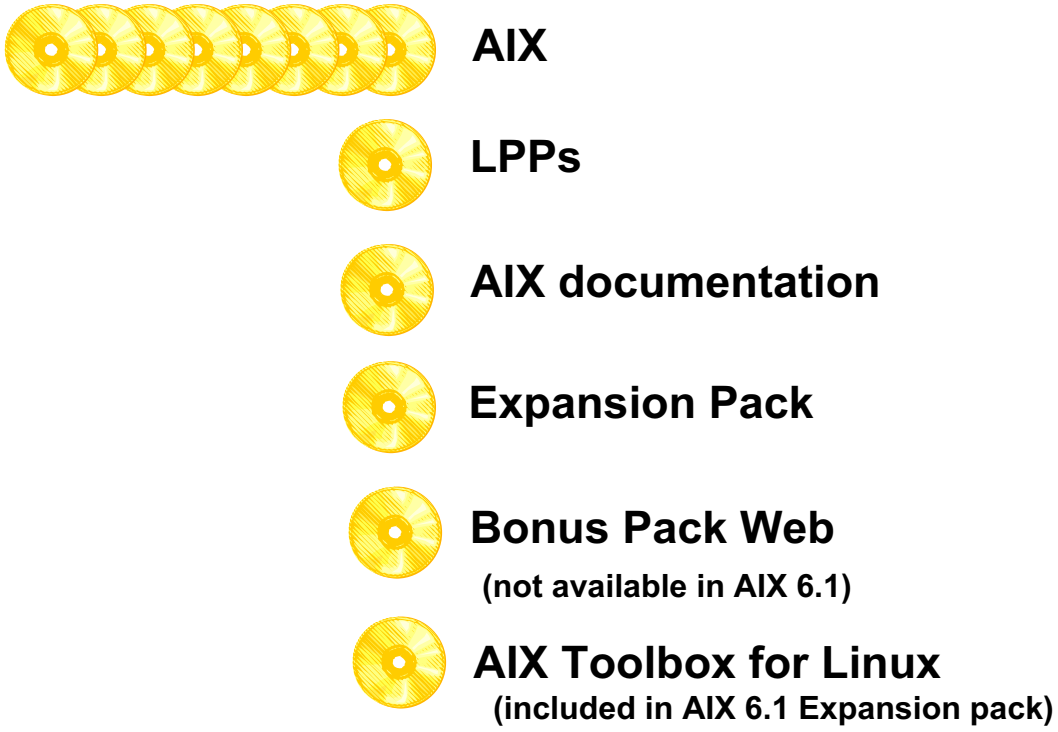


Figure 4-2. AIX product offerings AU1412.0

Notes:

Introduction

The AIX 6.1 operating systems are delivered on multiple CDs. During the ordering process, it is necessary to indicate the system type.

Licensed Program Products (LPPs) are separately orderable products that run on the AIX operating system.

The contents of the Expansion Pack and Linux toolbox vary over time. Their purpose is to acquaint users with tools and products that may be valuable in their business environment.

Expansion Pack

An Expansion Pack is included with every new order of AIX at no additional charge when media is selected, or can be ordered separately for existing AIX licenses. Typical releases may include: database software, development tools, software supporting

e-business, interoperability support, browsers, Java and Internet application development tools, network management utilities, and country-specific security encryption. Linux toolbox is now included in the Expansion pack.

AIX documentation

The AIX 6.1 documentation comes on a two CDs or one DVD set. It contains the full AIX documentation library in many different languages, in addition to the infocenter run time environment. These CDs/DVD are listed as being available at the end of 2007. They are not part of the base system installation.

Packaging definitions

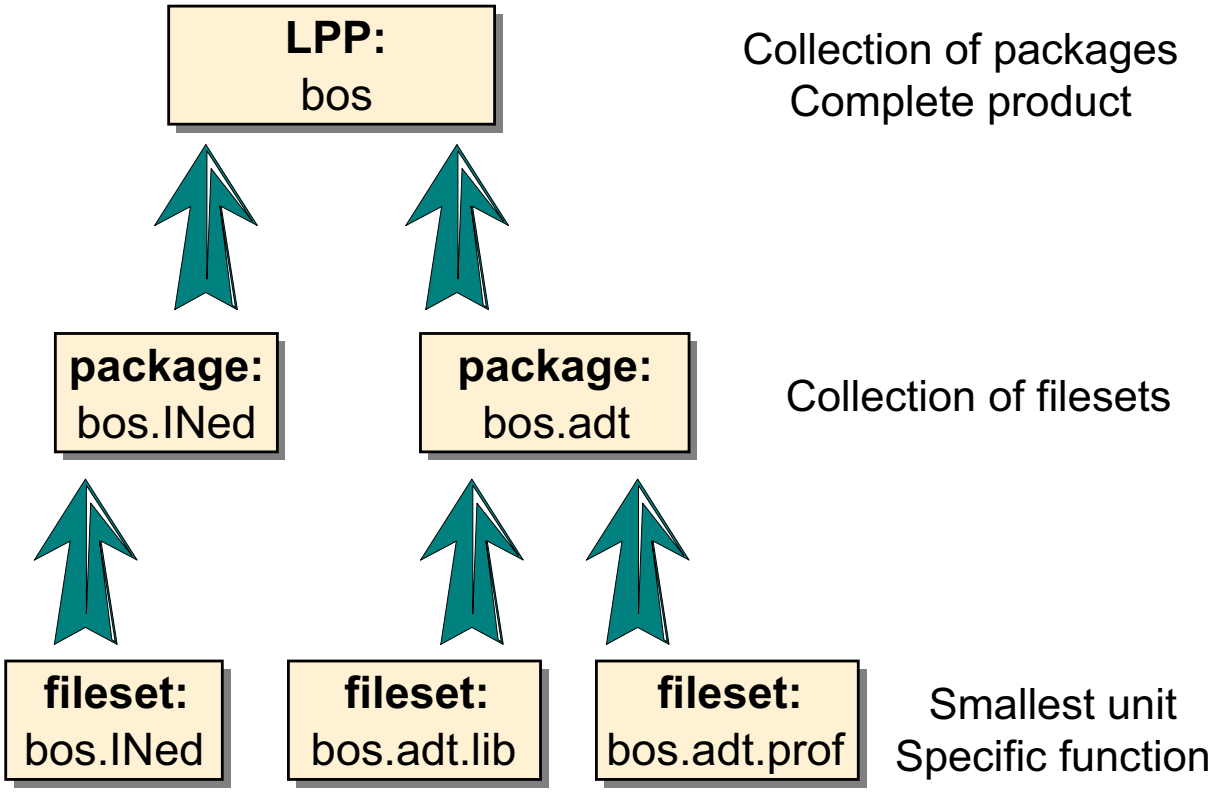


Figure 4-3. Packaging definitions AU1412.0

Notes:

Licensed Program Product (LPP)

A Licensed Program Product (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.

Package

A package contains a group of filesets with a common function. It is a single, installable image.

Fileset

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.

Bundles

- A *bundle* is a collection of packages and filesets suited for a particular environment
- Predefined system bundles in AIX include:
  - AllDevicesKernels
  - Alt_Disk_Install
  - App-Dev
  - CC_Eval.Graphics
  - CDE
  - GNOME
  - Graphics
  - KDE
  - Kerberos_5
  - Media-Defined
  - Mozilla
  - PerfTools
  - Server
  - cas_client and cas_server
  - Devices
  - Infocenter
  - openssh_client and openssh_server
  - wsm_remote

Figure 4-4. Bundles AU1412.0

Notes:

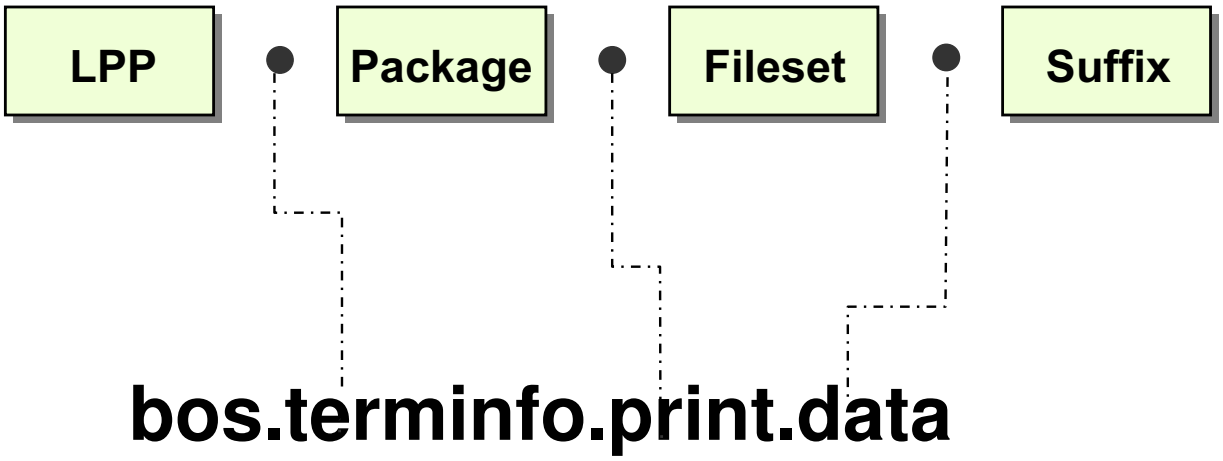
Overview

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 are only installed if the prerequisite hardware is available. For example, a graphic adapter is needed to run AIXWindows.

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 and AIX version to AIX version.

Fileset naming



Message convention:

LPP.msg[.lang].package.fileset

Figure 4-5. Fileset naming AU1412.0

Notes:

Fileset naming convention

Filesets follow a standard naming convention. The Licensed Program Product name is 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 is 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
.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.

Software updates

```
# oslevel
```

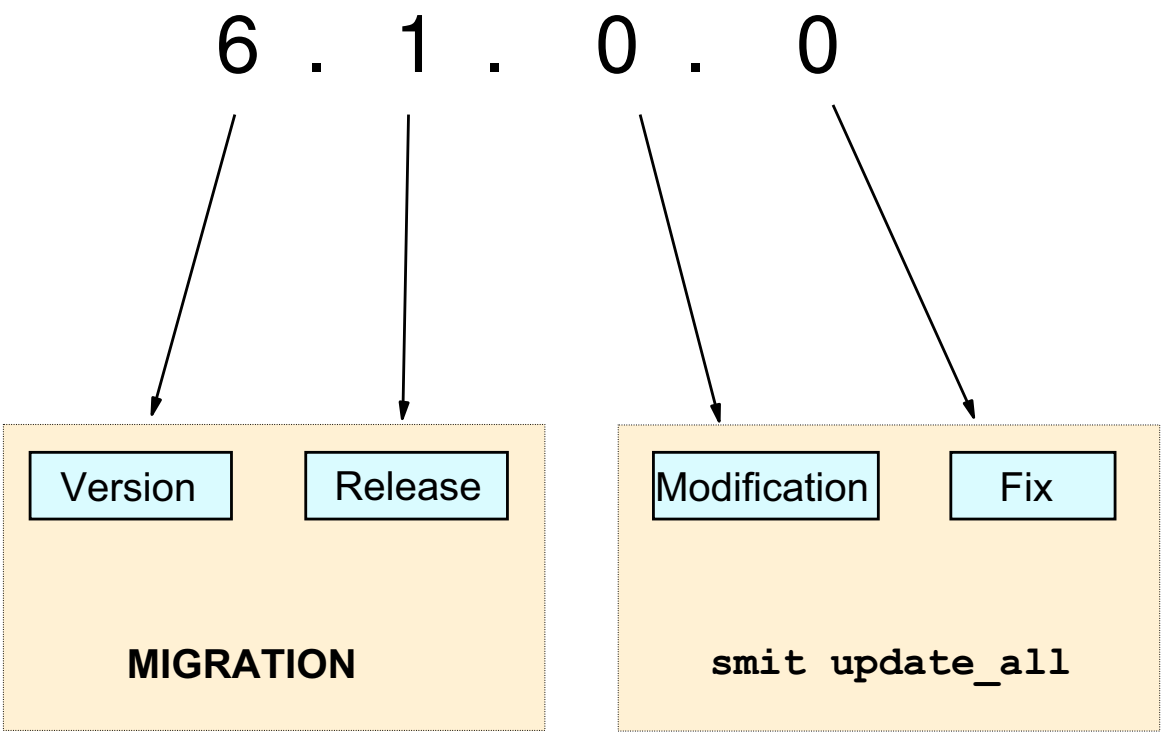


Figure 4-6. Software updates AU1412.0

Notes:

Introduction

As new software is created for AIX, you 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 four parts:

- Version
- Release
- Modification
- Fix

You can see this using the **`oslevel`** command.

The `oslevel` command

The **`oslevel`** command reports the level of the operating system using a subset of all filesets installed on your system. It also prints information about maintenance levels, including which filesets are not at a specified maintenance level.

The command syntax is:

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

where:

`-l` ***`Level`***	Lists file sets at levels earlier than maintenance level specified by the ***`Level`*** parameter
`-g`	Lists file sets at levels later than the current maintenance level
`-q`	Lists names of known maintenance levels that can be specified using the **`-l`** flag
`-r`	Applies all flags to the recommended maintenance levels
`-f`	Forces the **`oslevel`** command to rebuild the cache for this operation
`-s`	Applies all flags to service pack

Examples are:

- To get the actual BOS level:

```
# oslevel
6.1.0.0
```

- To get actual AIX BOS maintenance level:

```
# oslevel -r
6100-01
```

Types of upgrades

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 discussion in the *AIX 6.1 Installation* unit. If you want to make a change to the modification or fix levels, use the **`smit update_all`** command. 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 upgrades are available at no charge. They are provided on CD (order via AIX Support Center) or they can be downloaded from the Web. AIX updates are available at *http://www.ibm.com/servers/eserver/support/pseries/aixfixes.html*. This Web site is discussed in more detail later in this unit.

Software states

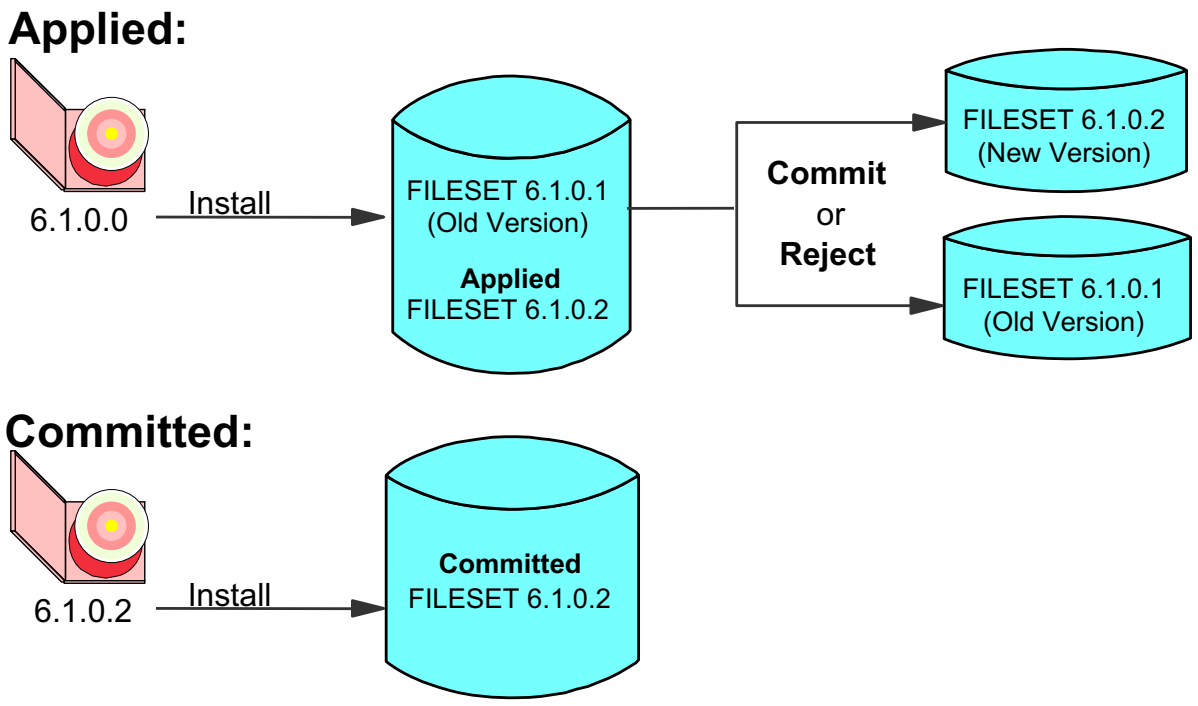


Figure 4-7. Software states AU1412.0

Notes:

Committed state and the initial install

AIX has a number of software states. When you are installing software for the first time, the software automatically installs to a *committed* state. This means there is only one level of that software product installed on your system.

Applied state versus committed state for maintenance

When you are installing a fix or a maintenance level upgrade to your system, you 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 removes the old version from the disk.

If the newer version is causing a problem, you can reject it which removes the newer version and recommits the old version.

Removing software products

With committed (or applied) software products, you can also remove them. This causes the product's files to be deleted from the system. Requisite software (software dependent on this product) is also 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

```
                Software Installation and Maintenance

 Move cursor to desired item and press Enter.

   Install and Update Software
   List Software and Related Information
   Software Maintenance and Utilities
   Software Service Management
   Relocatable Software Installation and Maintenance
   Network Installation Management
   EZ NIM (Easy NIM Tool)
   System Workload Partition Software Maintenance
   System Backup Manager
   Alternate Disk Installation
   EFIX Management
   Thin Server Maintenance


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


Figure 4-8. Software Installation and Maintenance AU1412.0

Notes:

Overview

Use the SMIT fast path `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

```
                    Install and Update Software

 Move cursor to desired item and press Enter.

   Install Software
   Update Installed Software to Latest Level (Update All)
   Install Software Bundle
   Update Software by Fix (APAR)
   Install and Update from ALL Available Software









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


Figure 4-9. Install and Update Software AU1412.0

Notes:

Introduction

Use the **`smit install_update`** fast path 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 you system.

Update Installed Software to Latest Level

This option is the **`smit update_all`** fast path. It enables you to update all of your currently installed software products. Only the existing installed products are updated; no new optional software is installed. This is the most commonly used method to install a maintenance level (service) update. This option does not allow you to install fixes that are on the media, but are older than the most recent fix available on the media.

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 from the Web site: http://www-912.ibm.com/eserver/support/fixes/fixcentral/main/pseries/aix

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. This option shows you the base function filesets plus all levels of fixes on the media.

Install Software

```
                          Install Software

Type or select values in entry fields.
Press Enter AFTER making all desired changes.
                                                        [Entry Fields]
* INPUT device / directory for software                 /dev/cd0
* SOFTWARE to install                                  [_all_latest]    +
  PREVIEW only? (install operation will NOT occur)      no              +
  COMMIT software updates?                              yes             +
  SAVE replaced files?                                  no              +
  AUTOMATICALLY install requisite software?             yes             +
  EXTEND file systems if space needed?                  yes             +
  OVERWRITE same or newer versions?                     no              +
  VERIFY install and check file sizes?                  no              +
  Include corresponding LANGUAGE filesets?              yes             +
  DETAILED output?                                      no              +
  Process multiple volumes?                             yes             +
  ACCEPT new license agreements?                        no              +
  PREVIEW new LICENSE agreements?                       no              +

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


Figure 4-10. Install Software AU1412.0

Notes:

Introduction

The SMIT **Install Software** dialog screen allows you to install all or selected software from the installation media. If any updates exist for these products, they are also installed.

To perform an **update_all**, the SMIT screen is identical except in the **SOFTWARE to install** line you see **[update_all]**.

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.

SOFTWARE to install: _all_latest

If **_all_latest** if left in the **SOFTWARE to install** line, everything on the installation media is 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 fast path **`smit install_latest`**.

PREVIEW only?

The **PREVIEW only?** option allows you to preview the results of the installation without actually performing the software install. The system displays information on space requirements and a list of software products and updates that are installed.

COMMIT software updates?

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.

License agreements

Beginning with AIX 5L V5.1, 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.

Using the `geninstall` command

Using the **`geninstall`** command is also a way to install AIX LPP packages. The **`geninstall`** command calls the **`installp`** command to install additional AIX LPP packages. An example is:

```
# geninstall -d /usr/sys/inst.images/installp/ppc bos.games
```

Do not specify the version, release, modification or fix level of the fileset, otherwise the installation fails.

Software inventory

```
# smit list_installed
```

```
          List Installed Software and Related Information

  Move cursor to desired item and press Enter.

    List Installed Software
    List Installed Software by Bundle
    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       F4=List
    F5=Reset     F6=Command        F7=Edit         F8=Image
    F9=Shell     F10=Exit          Enter=Do
```

lslpp command:

- **-L** Lists the installed software
- **-h** Shows the history of a software product

Figure 4-11. Software inventory AU1412.0

Notes:

Introduction

Use the SMIT fast path **smit list_installed** to access the **List Installed Software and Related Information** 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 **lslpp** 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
- **-b** List software for the specified bundle name

List installed software

```
# lslpp -l "bos.*"

  Fileset              Level  State      Description
  ----------------------------------------------------------------------
Path: /usr/lib/objrepos
  bos.64bit         6.1.0.10  COMMITTED  Base Operating System 64 bit
                                         Runtime
  bos.acct          6.1.0.10  COMMITTED  Accounting Services
  bos.adt.base      6.1.0.10  COMMITTED  Base Application Development
                                         Toolkit
  bos.adt.include   6.1.0.11  COMMITTED  Base Application Development
                                         Include Files
  bos.adt.lib       6.1.0.10  COMMITTED  Base Application Development
                                         Libraries
  bos.alt_disk_install.boot_images
                    6.1.0.10  COMMITTED  Alternate Disk Installation
                                         Disk Boot Images
  bos.alt_disk_install.rte
                    6.1.0.10  COMMITTED  Alternate Disk Installation
                                         Runtime
  bos.cdmount       6.1.0.0   COMMITTED  CD/DVD Automount Facility
  bos.content_list  6.1.0.0   COMMITTED  AIX Release Content List
  bos.diag.com      6.1.0.10  COMMITTED  Common Hardware Diagnostics
  bos.diag.rte      6.1.0.10  COMMITTED  Hardware Diagnostics
  bos.diag.util     6.1.0.10  COMMITTED  Hardware Diagnostics Utilities
    ... (rest of output omitted) ...
```


Figure 4-12. List installed software AU1412.0

Notes:

Overview

The **lslpp** command is used to list the installed software on the system. The various options of the **lslpp** command allow you to view selected information on the software installed.

The output of the **lslpp** command displays the fileset name, the level of the product, its state (applied or 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

Fix repository

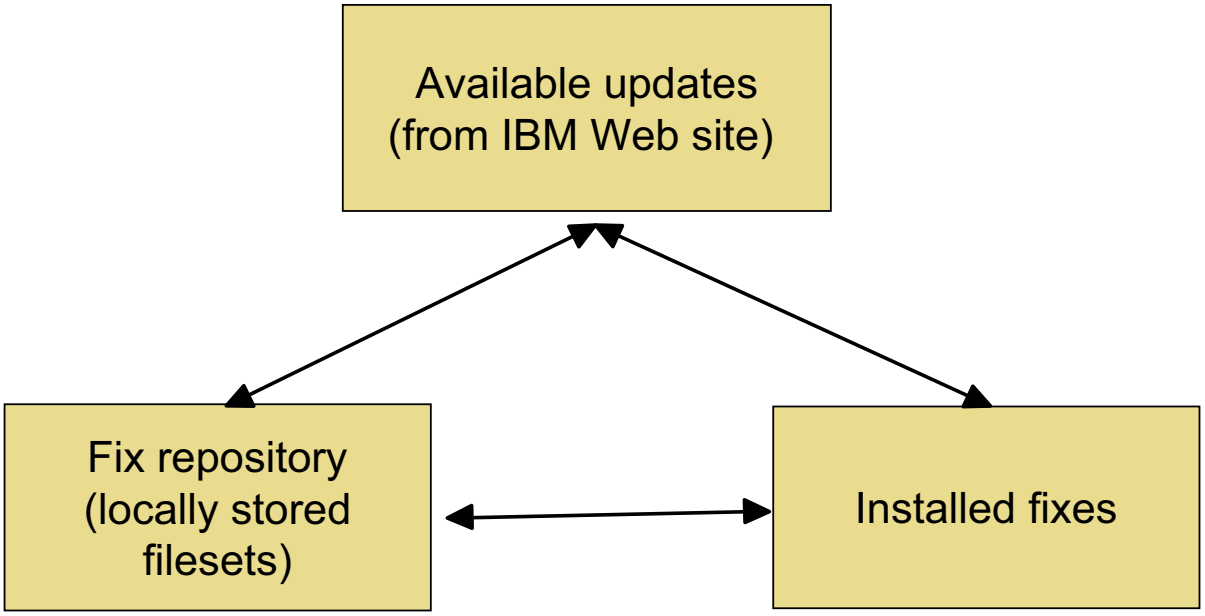


Figure 4-13. Fix repository AU1412.0

Notes:

Overview

When working with modifications or fixes, it is common to download the fixes to a directory on your local hard drive before installation. This location is commonly referred to as the **fix** repository.

While **/usr/sys/inst.images** is a standard location for storing software images and appears in the SMIT **Input Device/Directory** (**F4)** list, you may choose to use any directory for this purpose.

Managing the fix repository includes knowing how up to date it is relative to what is available and what in the repository has been installed.

AIX provides a reporting facility to compare the installed software, repository software and IBM Web site available software. Any two of these can be compared to obtain a report.

Fix Central Web site

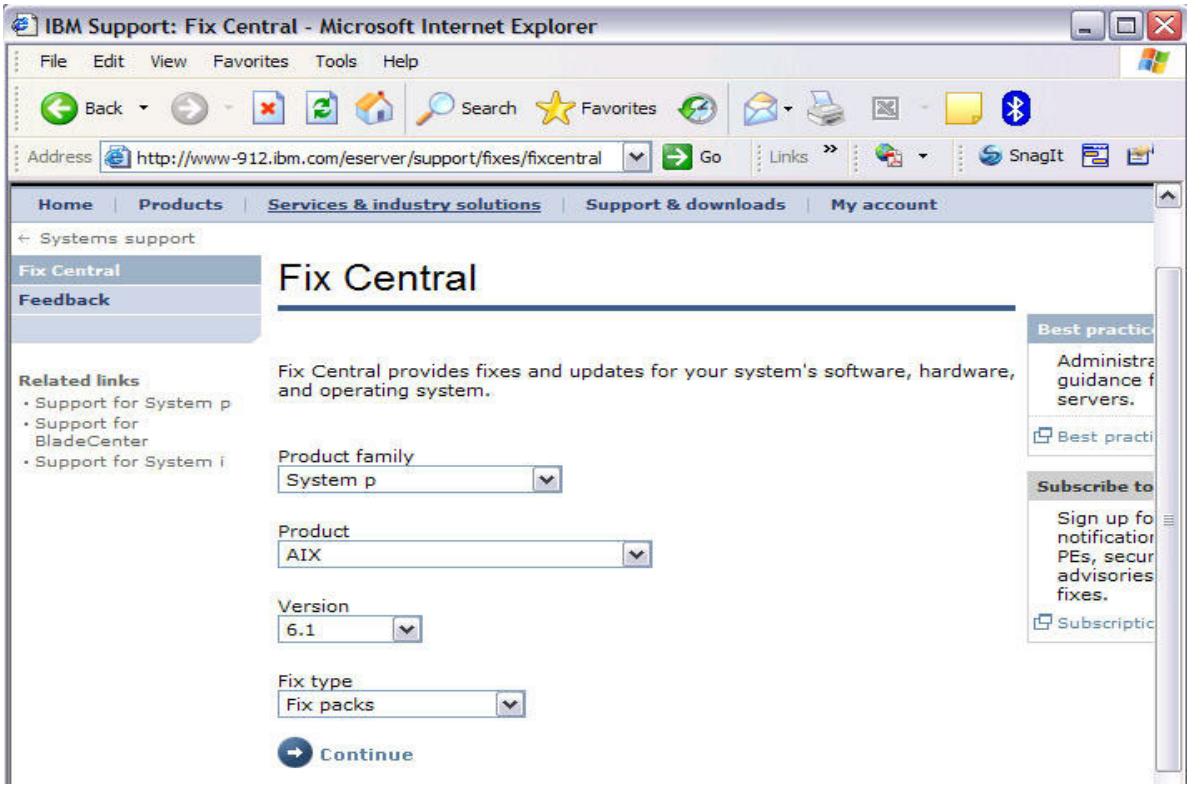


Figure 4-14. Fix Central Web site AU1412.0

Notes:

Introduction

IBM System p support provides a Web site interface (Fix Central) to access fix information. One way to navigate to this Web page is:

- Via Fix Central at:
  http://www-912.ibm.com/eserver/support/fixes/fixcentral/main/pseries/aix

  Identify the **Product family** as **System p, Product** as **AIX, Version** as **6.1** and **Fix type** as **Fix packs**. Clicking **Continue**, to reach the next visual.

Fix pack

Fix packs for AIX 6.1 operating system are provided as Technology Level packages or Service Packs. These generally available updates have been tested to operate best

when all updates in a fix pack are installed. IBM recommends installing the complete fix pack.

Fix recommendations

Recommendation tools and services for AIX administrators allow the system administrator to be able to determine recommended levels of hardware and software.

FLRT - Fix **L**evel **R**ecommendation **T**ool is a planning tool to help administrators determine what key components of the System p server are at the minimum recommended fix level.

Server Optimization and Integration Services - Healthcheck is designed to help you proactively detect configuration abnormalities that may be keeping your System p server environment from performing optimally.

Fix search

Find downloads for single software product.

Managing updates

IBM provides several tools which allow the system administrator of AIX machines to be able to manage downloading and installing fixes to the AIUX operation system filesets.

Security advisories

This service provides technical information for IT professionals who maintain System p servers. As support bulletins are released, you receive email containing links to the bulletin.

More fix services screen (from Fix Central)

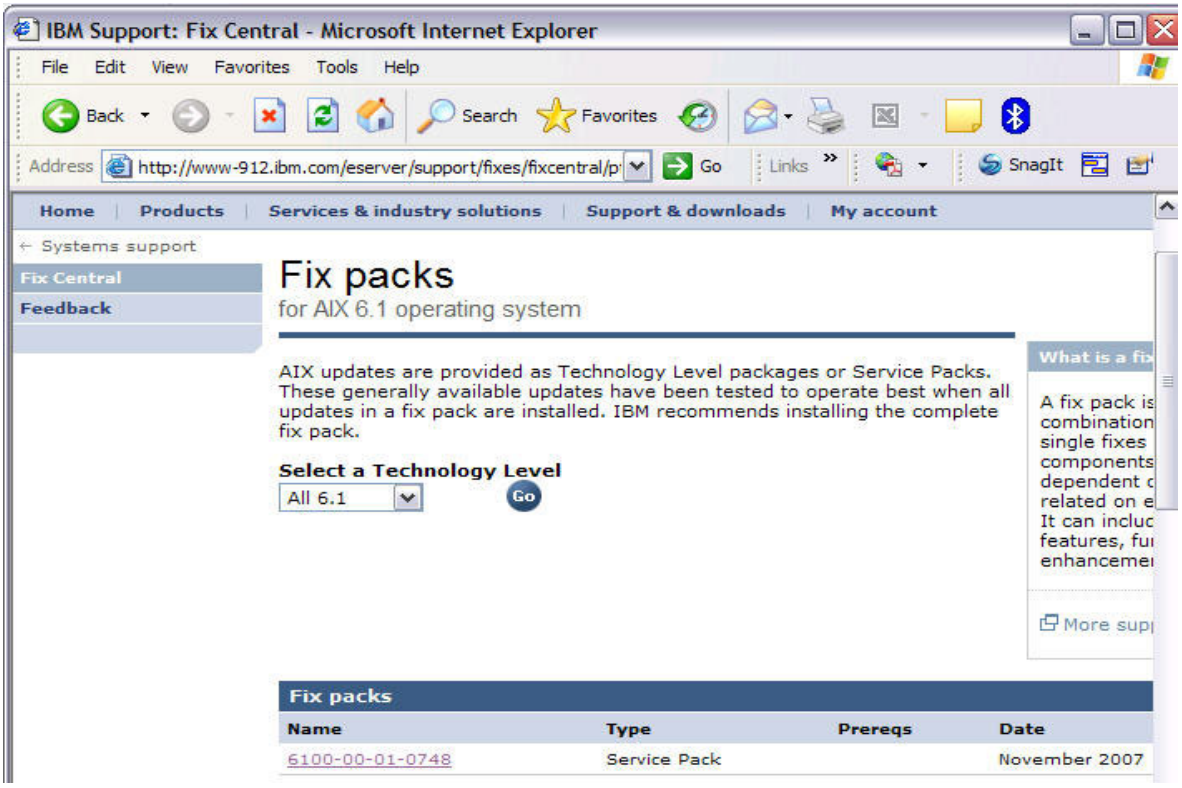


Figure 4-15. More fix services screen (from Fix Central) AU1412.0

Notes:

Introduction

Service Packs are cumulative. Newer Service packs in a Technology Level contain all the updates from previous Service Packs. A fix pack is a combination of many single fixes for product components that are dependent on or related on each other. It can include new features, functions, or enhancements.

Available information

After selecting the **Fix pack** by double-clicking the Name link, you are presented with three tabs. The **Package tab** displays the Release date and Package name. **The Package details tab** lists the available new APARs, their description, and known problems with the package. The **Installation tips tab** identifies all the required steps before, during and after installation.

Software Service Management

```
# smit service_software
```

```
                Software Service Management

    Move cursor to desired item and press Enter.

      Service Update Management Assistant (SUMA)
      Comparison Reports
      Rename Software Images in Repository
      Clean Up Software Images in Repository





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


Figure 4-16. Software Service Management AU1412.0

Notes:

Introduction

The SMIT **Software Service Management** menu allows access to Service Update Management Assistant (SUMA) functions, which significantly simplify the system update process by allowing policy-based automatic downloads of maintenance updates from the Web. The **Software Service Management** menu also allows generation of reports to manage filesets installed on a system, filesets contained in a repository, and filesets available from the IBM Fix Central Web site. It also provides a way to clean up and rename software images in a repository.

Service Update Management Assistant (SUMA)

SUMA is fully integrated into the AIX 6.1 Base Operating System and supports scheduled and unattended task-based download of Authorized Program Analysis Reports (APARs), Program Temporary Fixes (PTFs), and recommended maintenance levels (MLs). SUMA can also be configured to periodically check the availability of

specific new fixes and entire maintenance levels, so that the time spent on such system administration tasks is reduced. The SUMA implementation allows for multiple concurrent downloads to optimize performance and has no dependency on any Web browser.

SUMA policies can be run without extensive configuration. Filtering options allow comparisons against an installed software inventory, a fix repository, or a maintenance level to ensure only desired fixes are downloaded. SUMA provides the option to send an e-mail notification containing a list of what's available for download, as well as detailed summary statistics of a download.

SUMA can be accessed through the **`suma`** command or through the **`smit suma`** fast path.

Comparison Reports menu

This menu allows you to generate a comparison report to identify which fix filesets that are available in a fix repository (**lpp_source** or a directory of fixes) have not been installed on the system.

Rename Software Images in Repository option

This option allows you to rename updates that have FIX ID numbers for names, to more meaningful fileset names like those generated when updates are copied to hard disk for future installation. This action renames all filesets in the indicated directory with the same format. This option is available using the SMIT **`rename_software`** fast path. You can also use the **`bffcreate`** command to rename software images in a directory. This allows you to determine exactly what updates are contained in a directory because the fileset name is recognizable.

Clean Up Software Images in Repository option

This option allows you to perform various tasks to clean up software image directories on stand-alone systems. This allows you to remove duplicate or unnecessary filesets from image repositories, easing management of the images and reducing the amount of space required to store them.

Comparison Reports

```
# smit compare_report
```

```
                    Comparison Reports

Move cursor to desired item and press Enter.

  Compare Installed Software to Fix Repository
  Compare Installed Software to List of Available Updates
  Compare Fix Repository to List of Available Updates



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


Figure 4-17. Comparison Reports AU1412.0

Notes:

Introduction

The SMIT **Comparison Reports** menu or the **`compare_report`** command allows you to generate several comparison reports to identify which of the available fix filesets are either installed or in a local repository. Originally, one side of the comparison could include a downloaded list from the IBM Fix Central Web site. That is no longer an option. At this time, you can only compare between what is installed and what is at a fix repository, such as an **lpp_source** or a directory of fixes.

You can perform these actions in the **`smit compare_report`** fast path or using the **`compare_report`** command.

Compare Installed Software to Fix Repository

To compare installed software filesets to filesets contained in a fix directory, you can use:

- The SMIT **Compare Installed Software to Fix Repository** menu which can be accessed by the `smit instofix_compare` fast path.
- The `compare_report` command with the following options:

```
compare_report -s -i FixDir {[-l ] [-h ] [-m ] [-n ]}-t ReportDir -Z |-v]
```

The following report lists are generated:

- Filesets on the system that are back-level (**lowerlevel.rpt)**
- Filesets on the system that are at a later level (**higherlevel.rpt**)
- Filesets in the fix repository that are not installed on the system (**notinstalled.rpt**)
- Filesets installed on the system that are not in the fix repository (**no_update_found.rpt**)

Compare Installed Software to List of Available Updates

This is no longer an option. AIX fix central no longer supports it.

Compare Fix Repository to List of Available Updates

This is no longer an option. AIX fix central no longer supports it.

Compare a List of Installed Software on a Base System to Another System

You can also compare a list of installed software on a base system to another system. This option allows you to compare the filesets installed on a system to another system. The `lslpp -Lc` output from one system is saved to a file and compared with the `lslpp -Lc` output from another system.

To compare a list of installed software on a base system to another system use the `compare_report` command with the following options:

```
compare_report -b BaseRp -o OtherRp{[-l] [-h] [-m] [-n]} [[[-t ReportDir] [-Z]] |-v]
```

The following report lists are generated:

- A list of base system installed software that is at a lower level (**baselower.rpt**)
- Filesets not installed on the base system, but installed on the other system (**otheronly.rpt**)
- A list of base system installed software that is at a higher level (**basehigher.rpt**)
- Filesets installed on the base system that are not installed on the other system (**baseonly.rpt**)

Software Maintenance and Utilities

```
# smit maintain_software
```

```
                Software Maintenance and Utilities

 Move cursor to desired item and press Enter.

   Commit Applied Software Updates (Remove Saved Files)
   Reject Applied Software Updates (Use Previous Version)
   Remove Installed Software

   Rename Software Images in Repository
   Clean Up Software Images in Repository

   Copy Software to Hard Disk for Future Installation
   Copy Software Bundle to Hard Disk for Future Installation

   Check Software File Sizes After Installation
   Verify Software Installation and Requisites

   Clean Up After Failed or Interrupted Installation

   Service Update Management Assistant (SUMA)


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


Figure 4-18. Software Maintenance and Utilities AU1412.0

Notes:

Overview

The fast path **`smit maintain_software`** allows you to commit, reject and remove software. You might also find the other menu items useful.

Copying software

With the **Copy Software to Hard Disk for Future Installation** and **Copy Software Bundle to Hard Disk for Future Installation** options, 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. To **Copy *all* Software to Hard Disk for Future Installation** use the following command:

```
# gencopy -d /dev/cd0 -t /usr/sys/inst.images all
```


Checking software

If you are experiencing problems with your software, the **Check Software File Sizes After Installation** and **Verify Software Installation and Requisites** checks and verifies the system by analyzing it to determine if there is problem. It compares information stored on the disk to the information stored in ODM.

Cleanup

The **Clean Up After Failed or Interrupted Installation** 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 then need to start your installation/update over.

Repository management

The **Rename Software Images in Repository**, **Clean Up Software Images in Repository**, and **Service Update Management Assistant (SUMA)** options are also on the SMIT **Software Service Management** menu. These were already discussed in this unit.

instfix command

- Installs a fix:

```
# instfix -k  IY58143 -d /dev/cd0
```

- Searches for a fix:

```
# instfix -ik IY58143
All filesets for IY58143 were found.
```

- Searches for a fix by keyword:

```
# instfix -s SCSI -d /dev/cd0
```

- Lists which AIX BOS maintenance levels are partly or fully installed:

```
# instfix -i | grep ML
All filesets for 6.1.0.0_AIX_ML were found.
All filesets for 6100-01_AIX_ML were found.
```

- Lists which filesets are missing in a partly installed AIX BOS maintenance level:

```
# instfix -ciqk 6100-01_AIX_ML | grep :-:
```


Figure 4-19. **instfix** command AU1412.0

Notes:

Introduction

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 is 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.

Options for the **instfix** command

Some of the options for the **instfix** command are:

- **-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** results 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**).

-c Output should be in colon delimited format.

Examples

The examples on the visual do the following:

- Install all filesets associated with fix IY58143 from the CD in the /**dev**/**cd0** drive:
  ```
  # instfix -k IY58143 -d /dev/cd0
  ```
- Inform the user whether fix IY58143 is installed:
  ```
  # instfix -ik IY58143
  All filesets for IY58143 were found.
  ```
- List all keyword entries on the CD containing the string SCSI:
  ```
  # instfix -s SCSI -d /dev/cd0
  ```
- List which AIX BOS Maintenance Levels are partly or fully installed:
  ```
  # instfix -i | grep ML
   All filesets for 6.1.0.0_AIX_ML were found.
  ```
- List which filesets are missing in a partly installed AIX BOS Maintenance level:
  ```
  # instfix -ciqk 6100-01_AIX_ML | grep :-:
  ```

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 4-20. Checkpoint AU1412.0

Notes:

Checkpoint solutions

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?
   `lslpp -l`
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)**?
   Install Software by default installs 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.

Exercise 4: AIX software installation

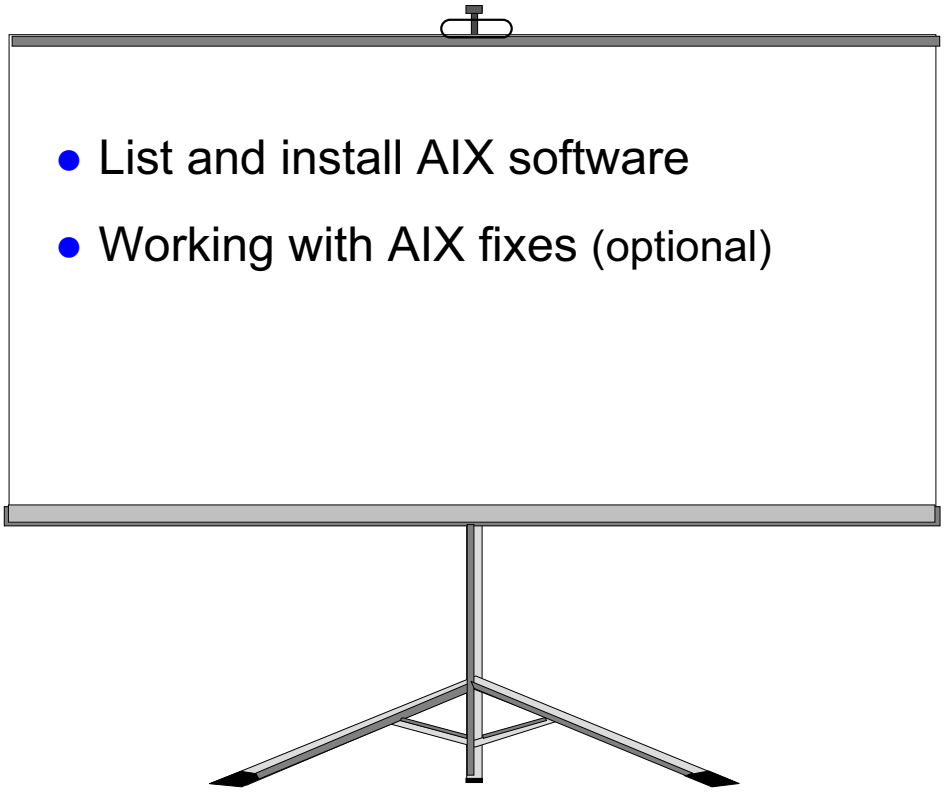


Figure 4-21. Exercise 4: AIX Software Installation AU1412.0

Notes:

Introduction

This lab gives you the opportunity to install filesets and show software installation history.

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

Unit summary

- AIX package naming conventions include the following terms:
  - LPP
  - Package
  - Fileset
  - Suffix
- The easiest way to install software is to use SMIT. The **`geninstall`** and **`installp`** commands are also available.
- Use the **`lslpp`** command, SMIT or the Web-based System Manager to list all software products installed on the system.

Figure 4-22. Unit summary AU1412.0

Notes:

Unit 5. AIX 6 installation

What this unit is about

This unit describes the process of installing the AIX 6.1 operating system.

What you should be able to do

After completing this unit, you should be able to:

- List the different installation and media options available
- List the steps necessary to install the AIX 6.1 base operating system
- Identify the tasks that can be carried out using the Configuration Assistant

How you will check your progress

Accountability:

- Checkpoint questions
- Exercise

References

SG25-7559	*IBM AIX Version 6.1 Differences Guide*
SC23-6629	*AIX Version 6.1 Release Notes*
SC23-6630	*AIX Version 6.1 Expansion Pack Release Notes*
Online	*AIX Version 6.1 Installation and migration*

Note: References listed as "Online" above are available at the following address:

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

Unit objectives

After completing this unit, you should be able to:

- List the different installation and media options available
- List the steps necessary to install the AIX version 6.1 base operating system
- Identify the tasks that can be carried out using the Configuration Assistant

Figure 5-1. Unit objectives AU1412.0

Notes:

Installation methods

- CD-ROM
- Tape (not available for AIX 6.1 installation)
  - 4 mm
  - 8 mm
- Preinstallation option (for a new system order)
- Network Installation Manager (NIM)
  - Token Ring
  - Ethernet
  - FDDI

Figure 5-2. Installation methods AU1412.0

Notes:

Required memory

In AIX 6.1, 256 MB is a minimum required to install the base operating system.

In AIX 5L V5.2 and AIX 5L V5.3, 128 MB of RAM is required to install the base operating system (BOS).

Platform type

Beginning with AIX 5L V5.2, the Common Hardware Reference Platform (CHRP) is the only supported platform. Execute **`bootinfo -p`** to get your hardware platform and **`bootinfo -y`** to check. As of AIX 6.1, the 32-bit kernel has been deprecated. Therefore, 64-bit hardware is required run AIX 6.1 (POWER4, POWER5, or POWER6 systems only).

Media type

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 from the tape.

Preinstallation option

The preinstallation option is only valid if accompanied by a hardware order that includes the preinstalled AIX.

Network Installation Manager

Network installations are carried out using the AIX Network Installation 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 System p 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)

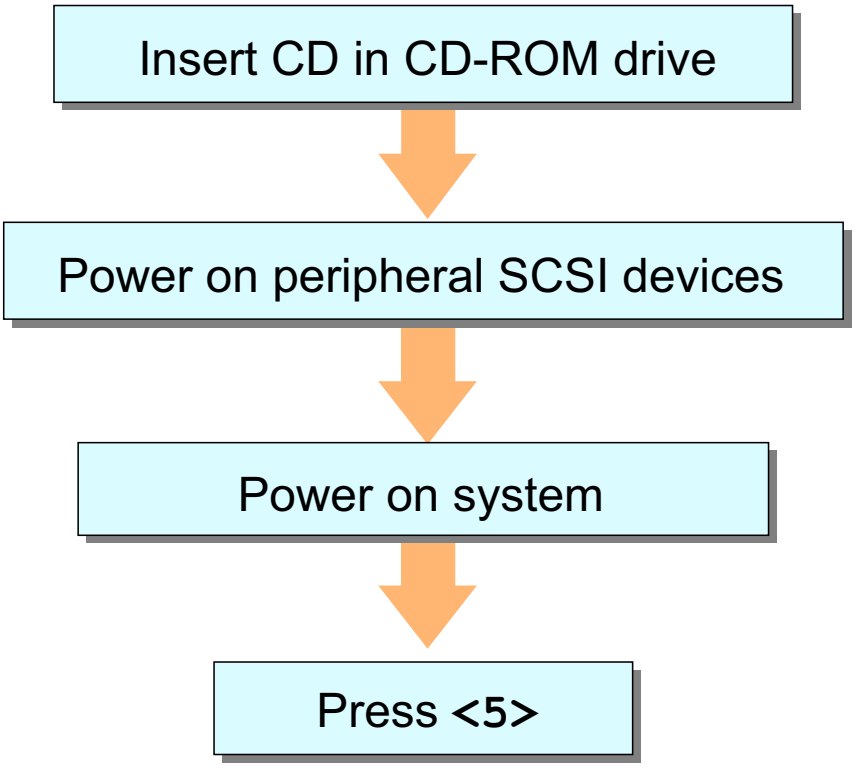


Figure 5-3. Installation process (from CD) AU1412.0

Notes:

Installation media and peripheral devices

The system needs 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 does not recognize it. It is best to power on all peripheral devices, because during the installation all recognized devices are configured.

The CD and tape devices must be powered on to open the door to the device. If they are internal, you need to power on the system before inserting the installation media.

Power on the system

Power on the system to start the boot sequence. The LEDs displays numbers indicating the system components that are being tested. Also, if you are using a graphical display,

you see icons of the hardware devices appear on the screen. The machine is completing a power on self test (POST).

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 searches the boot list for a bootable image. When it finds the bootable image, you can see menus appear on the screen.

When to press the numeric 5 key

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 numeric **<5>** key on the keyboard. This invokes the default service boot list. The CD is on that list. If you are attempting to install by tape, you need to add a tape device to a bootlist. This is done via the System Management Services (SMS) program. This is discussed later. Older AIX servers and AIX releases may require using the <F5> key instance of the numeric <5> key.

Installation process (from NIM)

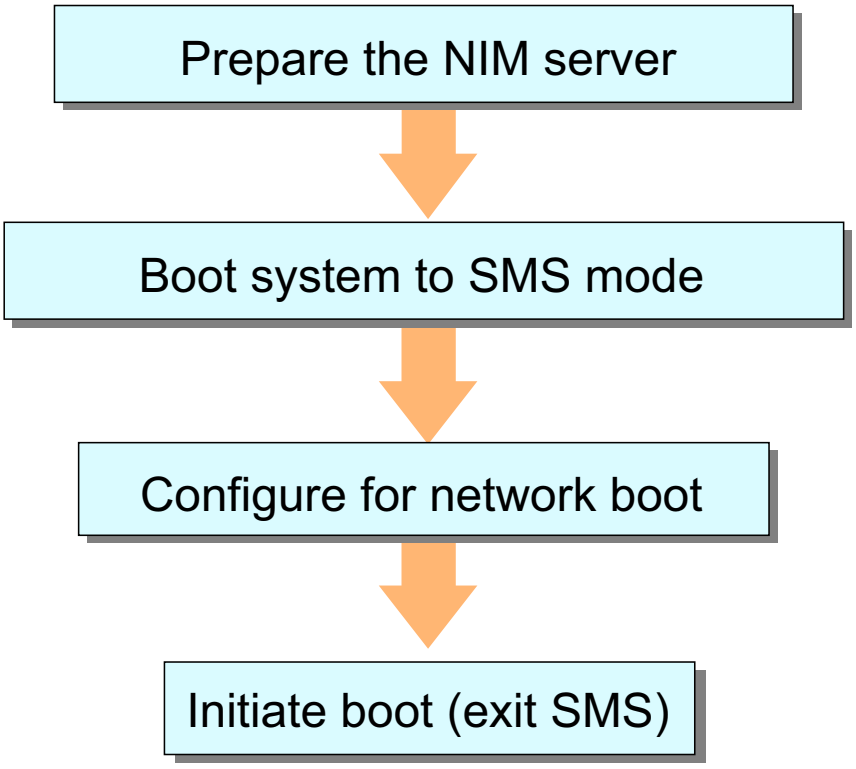


Figure 5-4. Installation process (from NIM) AU1412.0

Notes:

Overview

In many environments, especially when working with many logical partitions, the best way to handle installations is by using a Network Installation Manger (NIM) server. This way, you do not need to move the CD drive from partitions to partition or need to keep multiple sets of physical media. Instead, the base operating system filesets are stored on the NIM server and accessed via NFS. Once setup, the NIM server can be used to install multiple systems across the network in parallel.

NIM server setup

The installation and configuration of a NIM server is outside the scope of this class. But essentially, the installation media is copied into a directory and processed to create resource objects to be server to the NIM clients. The NIM clients are defined as machine objects, each uniquely identified by either MAC address or IP address. Then each client machine is allocated the appropriate resources. Finally, a bos install

operation is run on the client object to place it in a state where the NIM server responds to a network boot request.

SMS boot

For a non-partitioned machine, you would power on the machine and press numeric <1> at the right moment in order to cause the firmware to start the SMS facility.

For a partitioned machine, you would use the associated HMC to control the boot and request that the SMS facility be started.

In either case, an SMS menu is displayed on the system console. In the case of a partitioned machine, you would use an HMC virtual terminal window to see the console for your partition.

1. Shutdown the lpar system.
2. After the system is down, you see the **Activate** selection, click it.
3. Check the **Open a terminal window or console session,** and click the Advanced button.
4. At the **Boot mode** drop-down menu, select **SMS**.
5. Click **OK** and the system boots to the **SMS menu**.

Configure for network boot

In SMS, you would need to configure the NIM address and the client address as the Remote IPL parameters and then identify the network adapter as the boot device to be used.

a. Select 2. **Setup Remote IPL**, select the **Logical LAN** by typing in **1** and pressing **<Enter>**.

b. Select **1. IP Parameters**. Enter the number for the IP address you need to change and press **<Enter>**. This enables you to type the correct IP address at the bottom of the menu. Afterward, press the **<Esc>** key until you reach the **Main Menu**.

c. Select **5. Select Boot Option**.

d. Select **2. Configure Boot Device Order**.

e. Select **1. Select 1st Boot device**.

f. Select **6. Network**.

g. Select **1. Logical LAN** (the number by vary depending your on configuration).

h. Select **2. Set Boot Sequence: Configure as 1st Boot Device**.

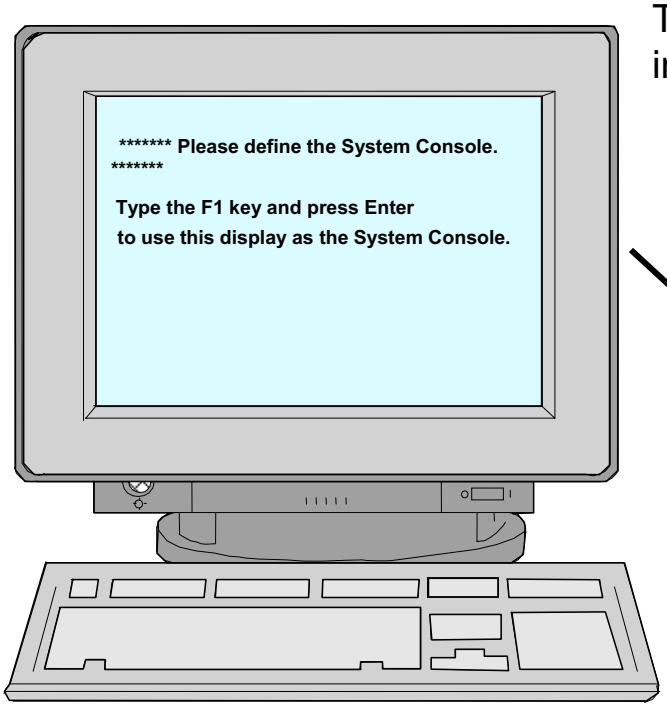
Initiate Boot

When you exit SMS, it automatically starts a boot sequence using the configuration you have setup. In this case, a network boot request is sent to the NIM server which sends a response identifying the boot image file to be retrieved. The system then downloads the boot image file and executes it.

The boot image NFS mounts the assigned resources and display a maintenance menu which allows the system administrator to install the operating system, just as if we had booted off of install media in the CD drive.

Console and language definition

Select your console



This message is displayed in different languages to:

- All native graphics displays
- Terminal on serial

NEXT:

Select the language for installation

Figure 5-5. Console and language definition AU1412.0

Notes:

Selecting the console

Each native (graphics) display and the ASCII terminal attached to the first built-in serial port (**S1**) displays the console messages. Whichever display you respond to becomes the console display during the installation. The console display can be changed at a later time if required.

Graphic displays 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 need to press **2** and then **<Enter>**.

Configuring an ASCII terminal as a console

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 assumes 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 does 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.

Selecting the language for installation

During the installation, you are also prompted to select the language to be used for the messages and the status information during the installation process. This language does not have to 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:

```
Welcome to Base Operating System
Installation and Maintenance

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

      1 Start Install Now with Default Settings
  >>> 2 Change/Show Installation Settings and Install
      3 Start Maintenance Mode for System Recovery
      4 Configure Network Disks (iSCSI)


   88 Help  ?
   99 Previous Menu
 >>> Choice [1]: 2
```


Figure 5-6. Installation and Maintenance menu AU1412.0

Notes:

Overview

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 and Settings

```
                    Installation and 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 . . . . . .  English (United States)
        Language . . . . . .  English (United States)
        Keyboard  . . . . . . . . . . . . . . . . . English (United States)
        Keyboard Type . . . . . . . . . . . . . Default
3 Security Model  . . . . . . . . . . . . . . . . . . . . . Default

4 More Options        (Software install 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 5-7. Installation and Settings AU1412.0

Notes:

Overview

The **Installation Settings** screen allows you to:

- Set the type of installation:
  - Migration
  - Preservation
  - New and Complete Overwrite
- Determine the installation disk
- Set the primary language environment
- Set more options

Method of installation

Option 1 of the **Installation and Settings** menu:

```
                    Change Method of Installation

   Type the number of your choice and press Enter.

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

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

3  Migration Install
   Upgrades the Base Operating System to current release. Other product
   (application) files and configuration data are saved.

   88 Help  ?
   99 Previous Menu

>>> Choice [2]:  1
```


Figure 5-8. Method of installation AU1412.0

Notes:

Changing the method of installation

When you select Option `1` in the **Installation and Settings** menu to change the method of installation, the **Change Method of Installation** submenu shown in the visual is 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 the **Preservation Install** 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 removes only the contents of /**usr,** / **(root),** /**var** and /**tmp**. The **Preservation Install** option preserves page and dump devices as well as /**home** and other user-created file systems. System configuration has to be done after doing a preservation installation.

Migration Install

Migration prior to AIX V4.2.1 is not supported. Use the **Migration Install** method to upgrade an AIX V4.2.1 or later system to an AIX 5L or later version, 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 4.x.

The installation process determines which optional software products are 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 >>>.

                                          Size      VG
        Name        Location Code        (MB)     Status     Bootable
>>>1    hdisk0      10-80-00-4,0         2063     rootvg        yes
     2  hdisk1      10-80-00-5,0         2063     rootvg        no

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

>>> Choice   [0]:
```


Figure 5-9. Installation disks AU1412.0

Notes:

Selecting installation disks

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 is 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>`.

Erasure Options for Disks

Erasure Options for Disks

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

```
    1  Number of patterns to write........... 0
    2  Pattern #1........................... 00
    3  Pattern #2........................... ff
    4  Pattern #3........................... a5
    5  Pattern #4........................... 5a
    6  Pattern #5........................... 00
    7  Pattern #6........................... ff
    8  Pattern #7........................... a5
    9  Pattern #8........................... 5a
>>> 0 Continue with choices indicated above
   88  Help ?
   99  Previous Menu
>>> Choice[0]:
```


Figure 5-10. Erasure Options for Disks AU1412.0

Notes:

Introduction

There are times when you may want to reuse a disk that previously contained some sensitive material and you want to be sure that information is no longer accessible.

If this is an overwrite installation, you can specify to erase the disks chosen to be installed before the installation occurs by typing **55** and pressing the **<Enter>** key for the **More Disk Options** option shown on the previous visual.

Erasure Options for Disks menu

The **More Disk Options** option opens a new menu (**Erasure Options for Disks** shown in the visual) that prompts for the number of patterns to write, which is the number of times the drive is overwritten. If you choose **0** for the number of patterns to write, the disks are not erased prior to installation.

This menu also prompts for the patterns to be used for each disk erasure. The patterns are a choice of the hexadecimal values `00`,`a5`,`5a`, or `ff`. For example, a pattern of `00` writes all zeros to the drive. Erasing a drive is a time-consuming process and only drive types that are supported by the **`diag`** command can take advantage of this option (for example, erasure of IDE drives are not supported).

Primary language environment

Option 2 of the **Installation and Settings** 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)
        -----
        -----

    10. MORE CHOICES ......

   88   Help ?
   99   Previous menu

    Choice [1]:
```


Figure 5-11. Primary language environment AU1412.0

Notes:

Overview

At this point in the installation process, you can change the language and cultural convention that is used on the system after installation. This screen might 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 obtained from the installation media.

Cultural convention determines the way numeric, monetary, and date and time characteristics are displayed.

The **Language** field determines the language used to display text and system messages.

Install Options

Option 4 of the **Installation and Settings** menu:

```
                    Install Options

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

     1. Graphics Software..................................................... Yes
     2. System Management Client  Software ……..... Yes
     3. Create JFS2 File Systems …….……. Yes
     4. Enable System Backups to install any system ….. Yes
        ( Install all devices)



>>>  5.  Install More Software

      0  Install with the current settings listed above.

88 Help ?
99 Previous Menu

>>>  Choice [5]: _
```


Figure 5-12. Install Options AU1412.0

Notes:

Introduction

When installing, the installation software detects that and presents some additional installation options.

Install Options

When **Graphics Software** is yes, X11, WebSM, Java, and other software dependent on these packages is installed. **System Management Client Software** includes WebSM, Java, service agent, lwi and pconsole. You may optionally select to have **JFS file systems**, instead of JFS2 file systems. **Enabling System Backups to install on other systems** installs all devices. Otherwise only device drivers necessary to your system hardware configuration are installed. To install more software (which may require CD swapping) press Enter.

Install More Software

```
                    Install More Software

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

    1.  Firefox (Firefox CD) ..................................................... No
    2.  Kerberos_5 (Expansion Pack)....................................... No
    3.  Server (Volume 2)......................................................... No


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

   88 Help ?
   99 Previous Menu

>>>  Choice [0]: _
```


Figure 5-13. Install More Software AU1412.0

Notes:

Overview

The **Install More Software** option is available in the New and Complete Overwrite installation method, as well as the Preservation installation method. Select **Install More Software** to choose additional software to install after the BOS installation process finishes. A software bundle file corresponds to each selection that contains the required packages and filesets.

Begin installation

Installing Base Operating System

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 5-14. Begin installation AU1412.0

Notes:

Overview

The installation media contains information stored on it to determine the sizes that the standard AIX file systems have. These are set large enough for the installation to succeed but do 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 are 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 takes some time but can be left unattended. After the BOS has installed, the appropriate locale optional program is also installed.

Once the installation has completed, the system automatically reboots from the newly installed operating system on disk.

Configuration Assistant menu

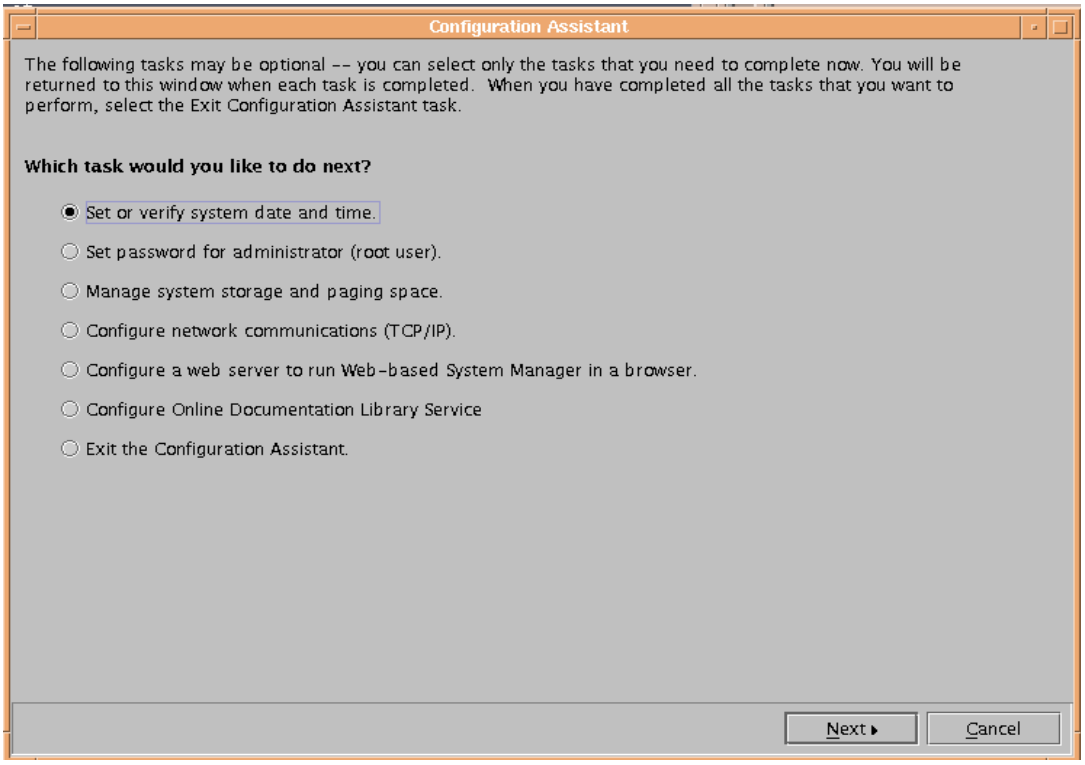


Figure 5-15. Configuration Assistant menu AU1412.0

Notes:

Introduction

After installing AIX, the operating system runs 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.

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.

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, you have at first to accept the license agreement and only the tasks that apply to your type of installation are 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.

After the initial execution

When you have completed your work using the Configuration Assistant / Installation Assistant, you can indicate that you are done working with the program. This prevents this program from being displayed at the next reboot.

To run the Configuration Assistant or Installation Assistant at a later time:

- 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.

You must have **root** user authority to use the Configuration Assistant / Installation Assistant.

Checkpoint

1. AIX 5 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 preserves all data on the disks.
3. What is the console used for during the installation process?

   _____

   _____

Figure 5-16. Checkpoint AU1412.0

Notes:

Checkpoint solutions

1. AIX V6.1 can that be installed from which of the following? (Select all are correct)
   a. 8 mm tape
   b. CD-ROM
   c. Diskette
   d. 4 mm tape
2. True or False? A Preservation Install preserves all data on the disks.

   Preserves SOME of the existing data on the disk selected for installation. Warning: This method overwrites the user (**/usr**), variable (**/var**), temporary (**/tmp**), and root (**/**) file systems. Other product (application) files and configuration data are destroyed.
3. What is the console used for during the installation process?

   The console is used to display all the system messages and interact with the installation.

Exercise 5: AIX install and Configuration Assistant

Figure 5-17. Exercise 5: AIX install and Configuration Assistant AU1412.0

Notes:

Introduction

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

Unit summary

- AIX V6.1 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 5-18. Unit summary AU1412.0

Notes:

Unit 6. Web-based System Manager

What this unit is about

This unit covers the process of installing, configuring, and using the AIX Web-based System Manager.

What you should be able to do

After completing this unit, you should be able to:

- Use the Web-based System Manager to manage AIX
- Install and configure the Web-based System Manager

How you will check your progress

Accountability:

- Checkpoint questions
- Exercises

References

Online	AIX 6.1 Documentation
SC23-4920	*AIX 5L Version 5.3 Web-based System Manager Administration Guide*

Unit objectives

After completing this unit, you should be able to:

- Use the Web-based System Manager to manage AIX
- Install and configure the Web-based System Manager

Figure 6-1. Unit objectives AU1412.0

Notes:

Web-based System Manager

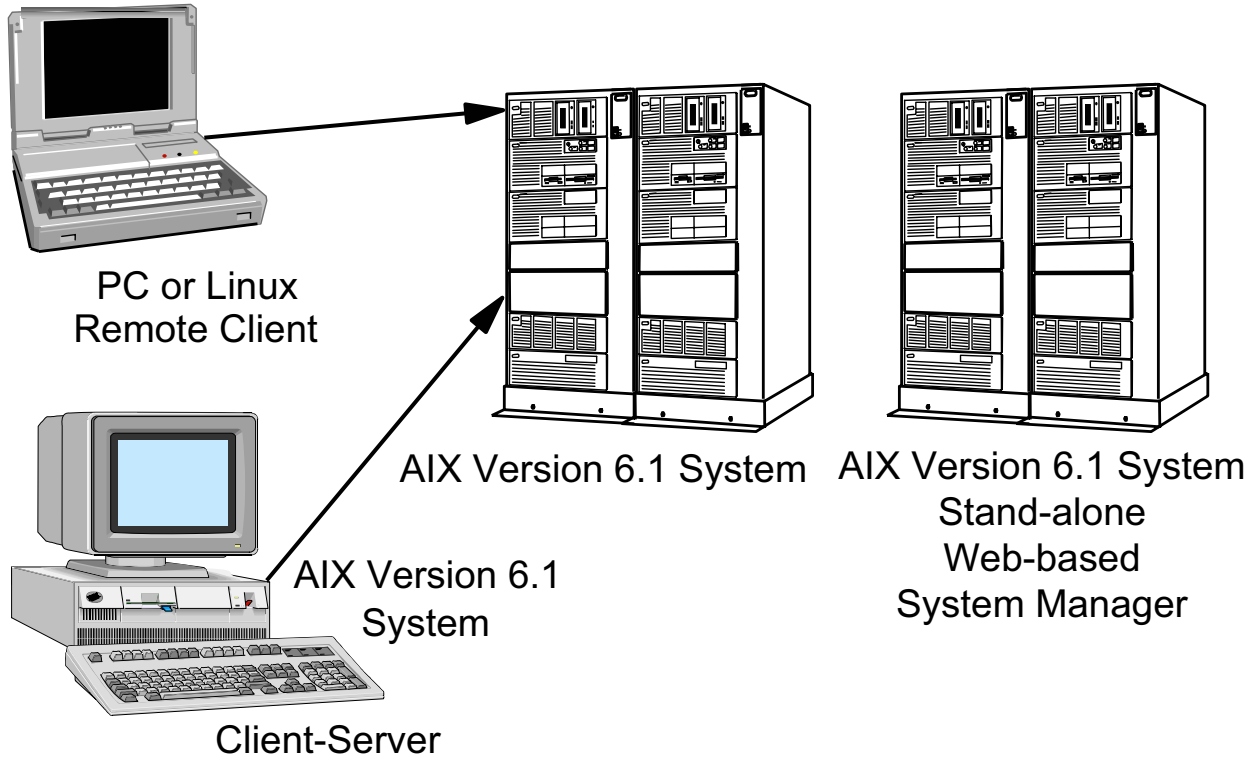


Figure 6-2. Web-based System Manager AU1412.0

Notes:

Introduction

AIX V4.3 introduced the Web-based System Manager, which is the next step in the evolution of AIX system administration tools. There have been many enhancements to the Web-based System Manager since AIX 5L V5.1. It is considered the default system administration tool for AIX 5L and later.

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.

The Web-based System Manager also supports a remote management. 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:

- Client-server mode by running the `wsm` command on an AIX client
- Remote client mode on either Windows or Linux clients

Web-based System Manager objectives

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 with a browser that supports Java V1.4 or use downloaded client code from an AIX 6.1 code
- Enable AIX systems to be administered remotely
- Provide a system administration environment that provides a similar look and feel to the Windows NT/2000/XP, LINUX 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 needs to be used in an ASCII environment.

Client requirements

Supported Microsoft Windows clients for AIX are:

- Windows 2000 Professional version
- Windows XP Professional version
- Windows Server 2003

Supported Linux clients are PCs running:

- Red Hat Enterprise Version 3
- SLES 8 or SLES 9
- Suse 8.0, Suse 8.1, Suse 8.2, and Suse 9.0 using desktops KDE or GNOME only

The PC Web-based System Manager Client installation needs a minimum of 300 MB free disk space, 512 MB memory (1 GB preferred) and a 1 GHz CPU.

To download Web-based System Manager Client code from an AIX host, use the address: *http://<hostname>/remote_client.html*

Accessing the Web-based System Manager

- Stand-alone mode:
  # wsm

- Client-server mode:
  – From the AIX 6.1 client:
  # wsm -host <managed-host>

- Remote client mode:
  – From Windows or Linux desktop:
    - Double-click the Web-based System Manager
    - Remote Client icon

Figure 6-3. Accessing the Web-based System Manager AU1412.0

Notes:

Introduction

The Web-based System Manager can be configured to run in several different modes. The operating environments in which Web-based System Manager can be started are stand-alone application, client-server, applet, and remote client.

Stand-alone mode

To access the Web-based System Manager from the command line, use the **wsm** command.

To start the Web-based System Manager console from the Common Desktop Environment (CDE), do the following:

1. Select the **Application Manager** icon in the CDE front panel
2. Select the **System_Admin** icon
3. Select the **Management Console** icon

Client-server mode

You can manage your local machine or machines that have been configured for remote management from the Web-based System Manager console. You can select a different host than your local machine as the managed host. To do this, use the following command: **`/usr/websm/bin/wsm -host [managed-host]`**. The managed host you specify as [***`managed-host`***] displays under the Navigation Area as the first name under the list of hosts that can be managed. This host is also used to load the Web-based System Manager user preference file (**$HOME/WebSM.pref**). Using the **`-host`** argument displays the console to the machine you are using, but uses the preferences file of the remote host you specify.

Remote client mode

Remote client mode allows you to run the Web-based System Manager console on a Windows or Linux system and manage remote AIX systems.

On a Windows system:

1. Double-click the **Web-based System Manager Remote Client** icon located on the Windows desktop to open the application
2. Click the **Start** button in the Task bar, then select **Programs -> Web-based System Manager -> Web-based System Manager Remote Client**
3. From an MS-DOS prompt, run the **`wsm.bat`** command from the Remote Client **bin** directory
4. Using Windows Explorer, double-click the **`wsm.bat`** icon in the Remote Client **bin** folder

On a Linux system running the Gnome Desktop:

1. Click the **Gnome** menu button in the Task Bar, then select **Programs -> Web-based System Manager Remote Client**
2. From an xterm, run the **`wsm`** command from the Remote Client **bin** directory

On a Linux system running the KDE Desktop:

1. Click the **KDE** menu button in the Task Bar, then select **Programs -> Web-based System Manager Remote Client**
2. From an xterm, run the **`wsm`** command from the Remote Client **bin** directory

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

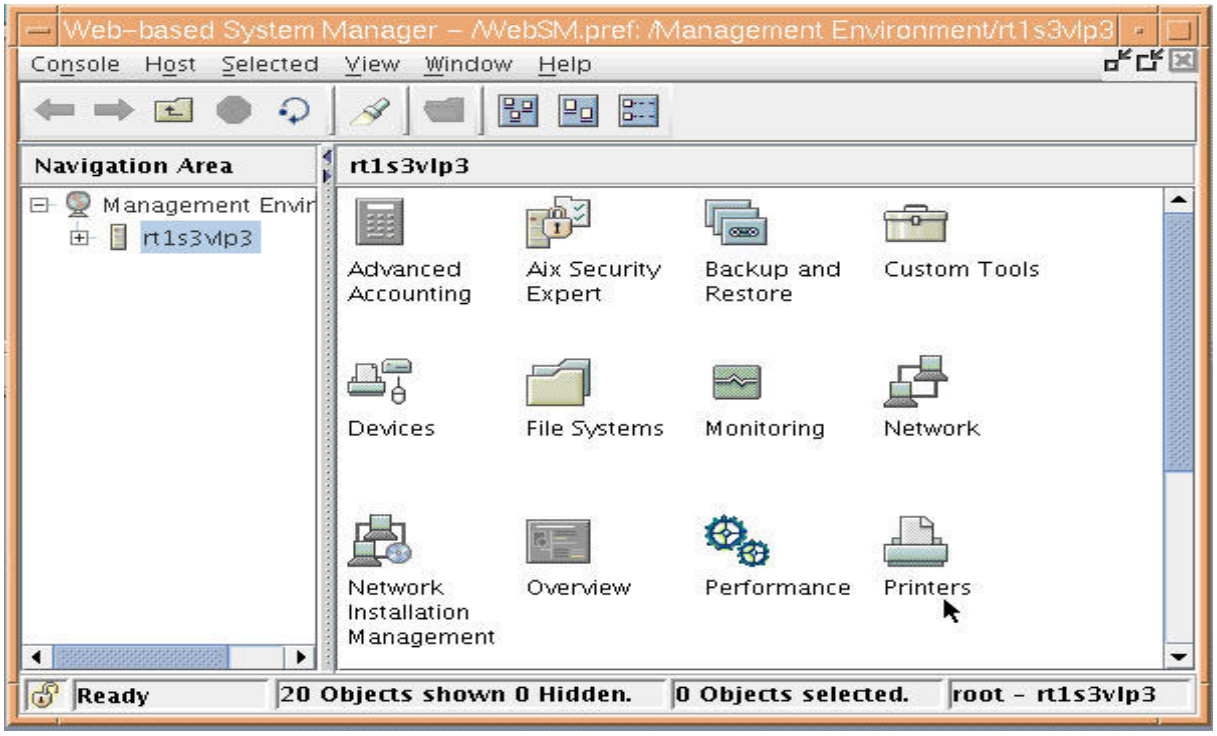


Figure 6-4. Using the Web-based System Manager (1 of 3) AU1412.0

Notes:

Console window

The visual 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.

Session log

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. To view the session log, select **Console -> Session Log**.

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

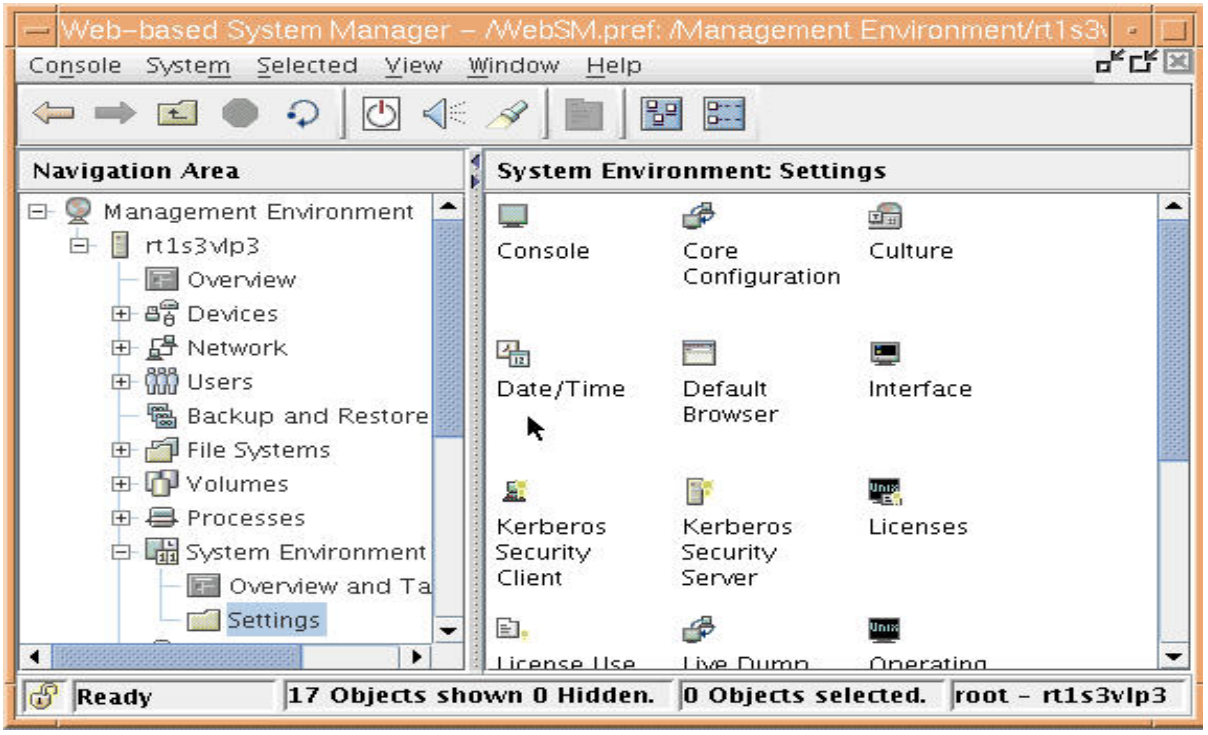


Figure 6-5. Using the Web-based System Manager (2 of 3) AU1412.0

Notes:

Toolbar functions

The visual shows a Web-based System Manager window with the **System Environment: Settings** plug-in running in the Contents area. Above the Navigation Area and Contents Area, there is a toolbar with symbols for managing the window. 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.

Date/Time icon

If you select the **Date/Time** icon, this allows you to set the date and time on the system.

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

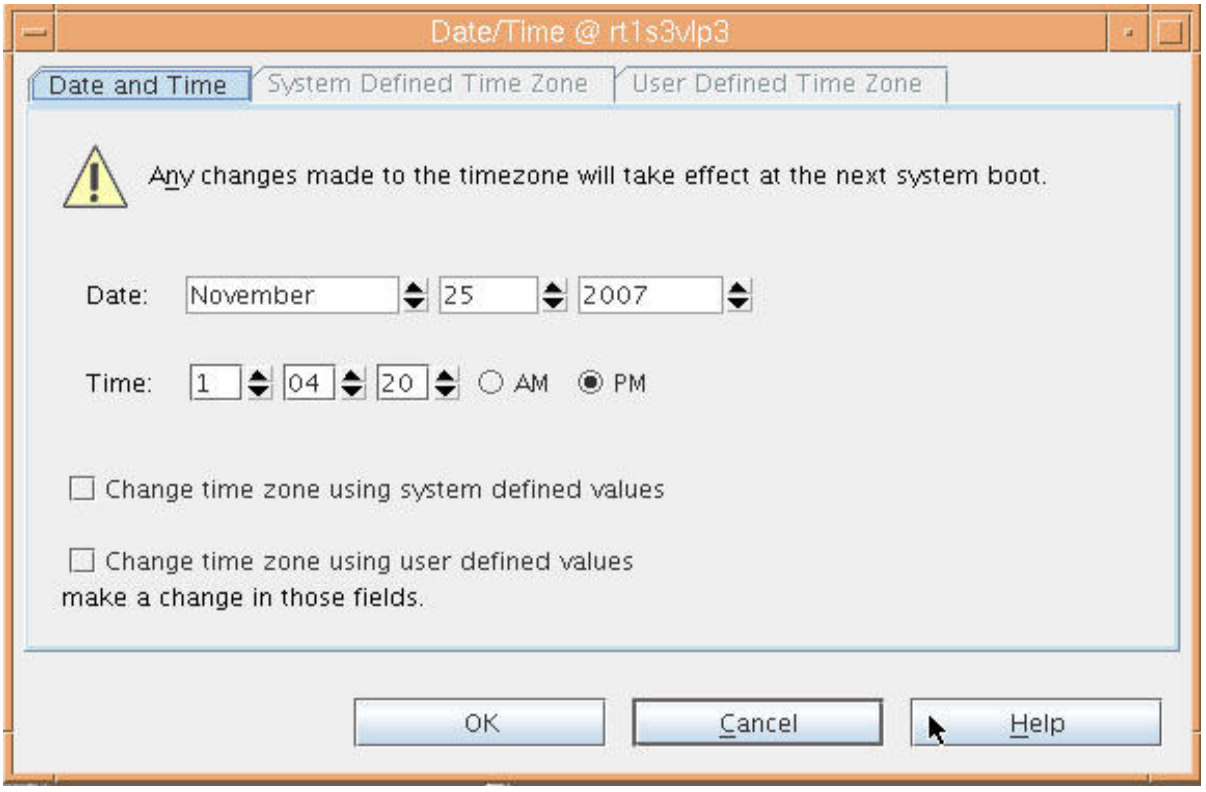


Figure 6-6. Using the Web-based System Manager (3 of 3) AU1412.0

Notes:

Overview

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-1.ibm.com/servers/aix/wsm/* .

Installation for a remote client

- Install the Web-based System Manager (usually done by default with the base)
- Install an HTTP server:
  – IBM HTTP Server (IHS2) on AIX Version 6.1 Expansion Pack
- Configure the HTTP server (more detail on the next visual)
  – **`configassist`**
- Test the HTTP server with a browser
- Enable the Web-based System Manager server:
  `# /usr/websm/bin/wsmserver -enable`
- Install the Web-based System Manager client on Windows or Linux platforms (more detail coming up)

Figure 6-7. Installation for a remote client AU1412.0

Notes:

Installing the Web-based System Manager

The Web-based System Manager must be installed on the client and on any managed systems that will be using it. If you have graphics installed on your machine, you probably have Web-based System Manager installed. The fileset that needs to be installed is **sysmgt.websm.framework**.

To verify that the fileset is installed, type the following:

```
# lslpp -h sysmgt.websm.framework
```

If Web-based System Manager is not installed, you see a message similar to the following:

```
lslpp: Fileset sysmgt.websm.framework not installed.
```


If Web-based System Manager is installed, you see output similar to the following:

```
# lslpp -h sysmgt.websm.framework
  Fileset         Level     Action       Status       Date         Time

----------------------------------------------------------------------
Path: /usr/lib/objrepos
  sysmgt.websm.framework
 6.1.0.0   COMMIT       COMPLETE 10/18/07 13:39:50

Path: /etc/objrepos
  sysmgt.websm.framework
 6.1.0.0   COMMIT       COMPLETE 10/18/07 13:44:07
```

The following optional filesets can be installed to add additional functionality to Web-based System Manager. They are not installed by default.

- **sysmgt.msg.*Locale Language*.websm.apps**

  Enables the locale language to be used if the `LANG` environment variable is set or if the **`-lang`** argument is used with the **`wsm`** command.

- **sysmgt.websm.security**

  Adds support for Secure Socket Layer communication between client and server. This fileset supports 40-bit encryption and is available on the Expansion Pack.

- **sysmgt.websm.security-us**

  Adds support for Secure Socket Layer communication between client and server. This fileset supports 128-bit encryption and is available on the Expansion Pack. Export and import laws could make this fileset unavailable in some countries.

Installation Requirements to Support Remote Client Mode

To install the Web-based System Manager Remote Client over a network, you must have the **sysmgt.websm.webaccess** fileset installed on at least one AIX system. This fileset is installed automatically with the base operating system. The machine used to install Web-based System Manager Remote Client must be set up as an HTTP Server. This is done by installing and configuring the HTTP Server of your choice.

The next visual shows the configuration of the HTTP server.

Enabling a Web-based System Manager Server

Client-server mode needs to be enabled on the servers that are to be managed as remote machines. Enabling and disabling a machine to act as a Web-based System Manager Server can be done through the **`/usr/websm/bin/wsmserver`** command.

To enable a machine to be a Web-based System Manager server, type the command: **`/usr/websm/bin/wsmserver -enable.`** This can also be done through SMIT using the fastpath **`smit web_based_system_manager`**.

To disable a machine so that it cannot be managed from a Web-based System Manager client, type the command: **`/usr/websm/bin/wsmserver -disable.`** This does not terminate existing Web-based System Manager server processes.

Configure the HTTP (Web) server

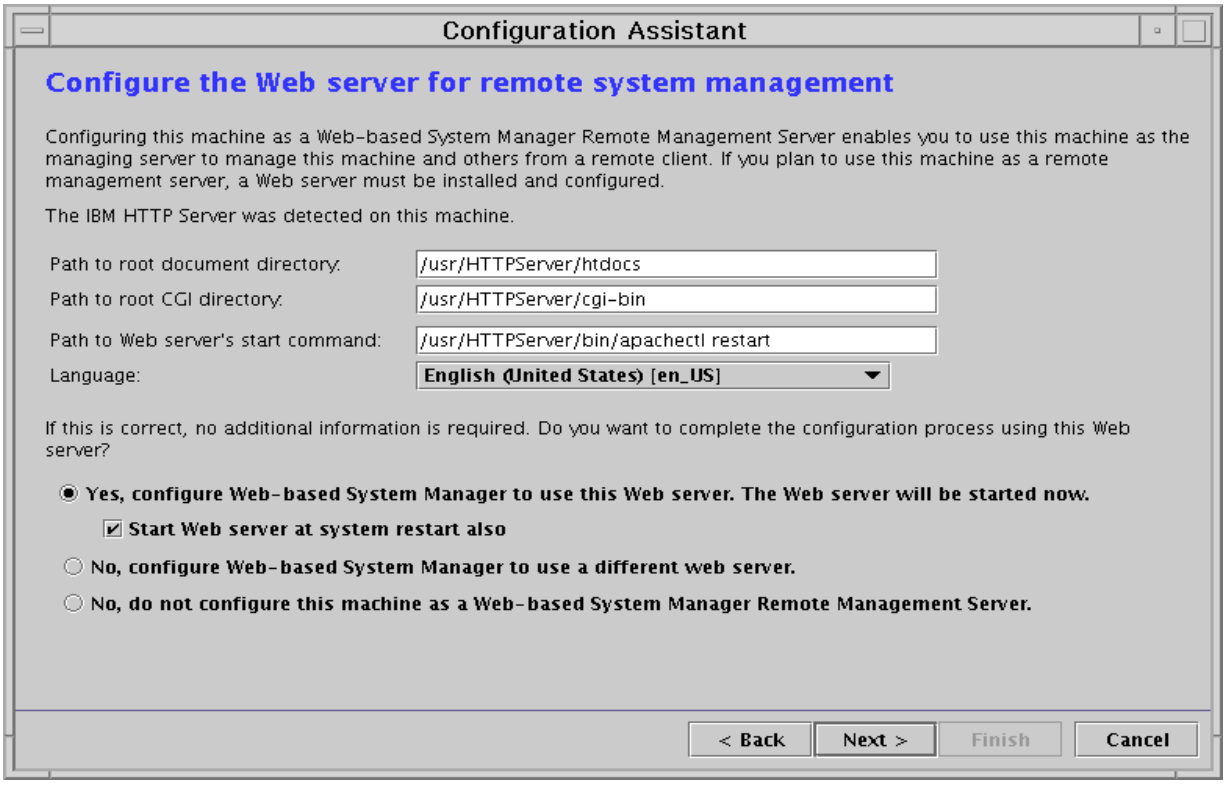


Figure 6-8. Configure the HTTP (Web) server AU1412.0

Notes:

Configuring the HTTP server

You can configure the HTTP server (Web server) for Web-based System Manager remote management in two ways:

- In the Configuration Assistant (**`/usr/websm/bin/configassist`** command), select **Configure the Web server for Web-based System Manager remote management**. You see the screen as shown in the visual.
- You can also configure the Web-based System Manager from SMIT. The fastpath is: **`smit web_based_system_manager`**.

The default values that you see are generally good, but if you are using some Web Server software other than the HTTP Server you may wish to change the directories and start commands. You may also select what language you use on the interface.

If you wish to always have the Web server ready for use, select the **Start Web server at system restart also** button to have the start command placed in the **inittab** file.

Remote client installation

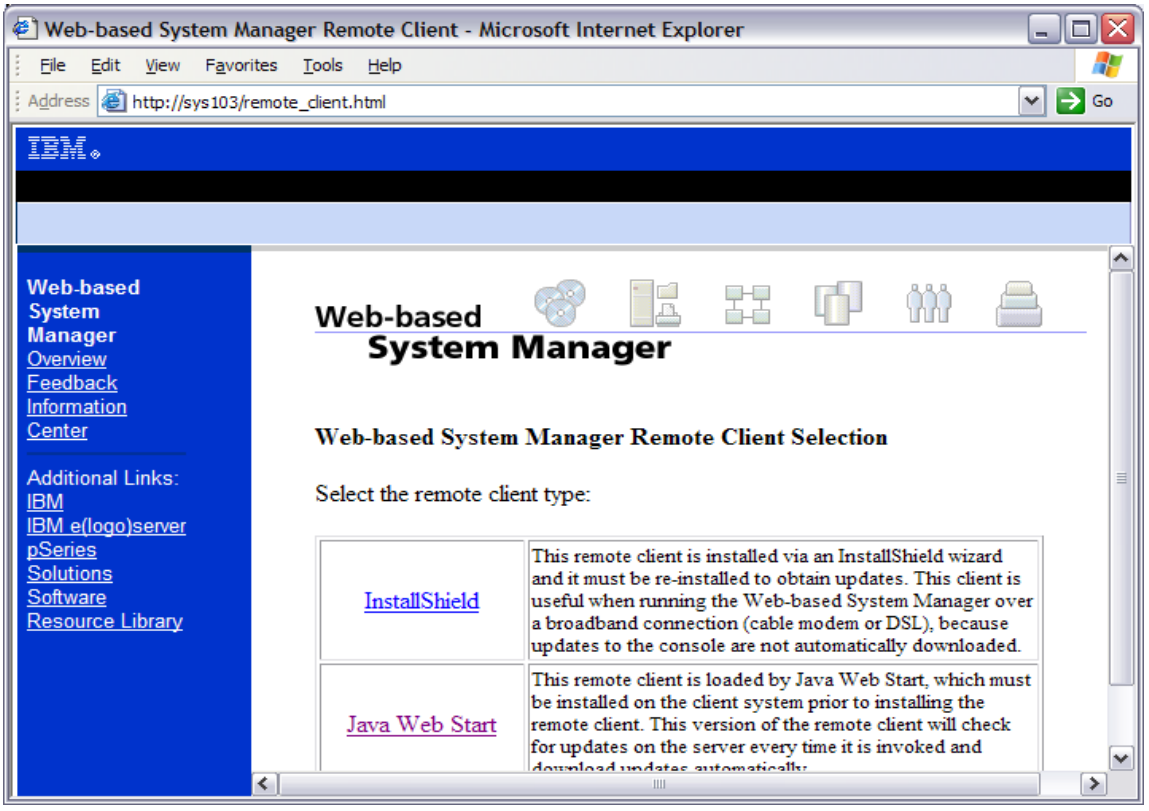


Figure 6-9. Remote client installation AU1412.0

Notes:

Installation options

To install the Web-based System Manager remote client on a PC running Windows or Linux, access the AIX Web Server from the client's browser, with the URL of:

http://<managed-host>/remote_client.html

The Web page provides two options for installation of remote client software:

- **InstallShield**
- **Java Web Start** (beginning with AIX 5L V5.2.3.0)

The **InstallShield** is pretty straightforward. It downloads the code and installs using the InstallShield standard.

The advantage of **Java Web Start** is that every time the client application runs, it checks to see if there is a remote server application software update and automatically downloads the changes.

Java Web Start security

If you are going to use **Java Web Start**, then you must install and configure the security package which otherwise would be an option in using the remote client. You first have to install the security package on the AIX server.

- **sysmgmt.websm.security** (regular strength)
- **sysmgmt.websm.security-us** (stronger encryption)

The URL for installing the client part of the security package is:
http://hostname/remote_client_security.html

Checkpoint

1. True or False? The Web-based System Manager is available for client access automatically after the BOS is installed.

   _____

   _____

2. Which of the statements are true regarding the Web-based System Manager?
   a) An AIX V6.1 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 V6.1 system from a remote AIX V6.1 system using an ASCII terminal.

   _____

   _____

   _____

   _____

Figure 6-10. Checkpoint AU1412.0

Notes:

Exercise 6: Web-based System Manager

- Use the Web-based System Manager
- Configure the Web-based System Manager for AIX client access (optional)
- Configure the Web-based System Manager for remote client access (optional)
- Configure the Web-based System Manager for client access applet mode (optional)

Figure 6-11. Exercise 6: Web-based System Manager AU1412.0

Notes:

Introduction

This lab allows you to set up the Web-based System Manager 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 the Web-based System Manager.

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

Unit summary

- The Web-based System Manager supports system administration tasks in a stand-alone or client-server environment.
- The Web-based System Manager can be used either locally or remotely from either another Web-based System Manager installed AIX platform or a PC with the Web-based System Manager application.
- Remote access may be enabled or disabled.

Figure 6-12. Unit summary AU1412.0

Notes:

Unit 7. 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 the 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

SG24-7559	*AIX Version 6.1 Differences Guide*
Online	*AIX 6.1 Installation and migration*
Online	*AIX 6.1 Operating System and device management*

Note: References listed as "Online" above are available at the following address:

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

Unit objectives

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 the different states of a device
- Describe the format of device location codes
- Use SMIT to add/show/change/delete devices

Figure 7-1. Unit objectives AU1412.0

Notes:

Device terminology

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

Figure 7-2. Device terminology AU1412.0

Notes:

Device terminology

In order to attach peripherals such as terminals and printers to an AIX system, you must tell AIX the characteristics of these devices so that the operating system can send the correct signals to the adapter where 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 is sent to the appropriate device driver. Data read from logical devices is 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--rw      1       root    system    20,0  Oct 29 02:25    fd0
brw-rw--rw      1       root    system    20,64 Oct 29 02:26    fd1
crw-rw--rw      1       root    system    20,0  Oct 29 02:25    rfd0
crw-rw--rw      1       root    system    20,64 Oct 29 02:26    rfd1
:
:
crw-r--r--      1       root    system    22,0  Oct 29 02:25    rmt0
crw-r--r--      1       root    system    22,1  Oct 29 02:25    rmt0.1
:
:
brw-------      1       root    system    14,1  Oct 29 02:44    hdisk0
brw-------      1       root    system    14,2  Nov  1 05:31    hdisk1
crw-------      2       root    system    14,1  Oct 29 02:44    rhdisk0
crw-------      1       root    system    14,2  Nov  1 05:31    rhdisk1
```

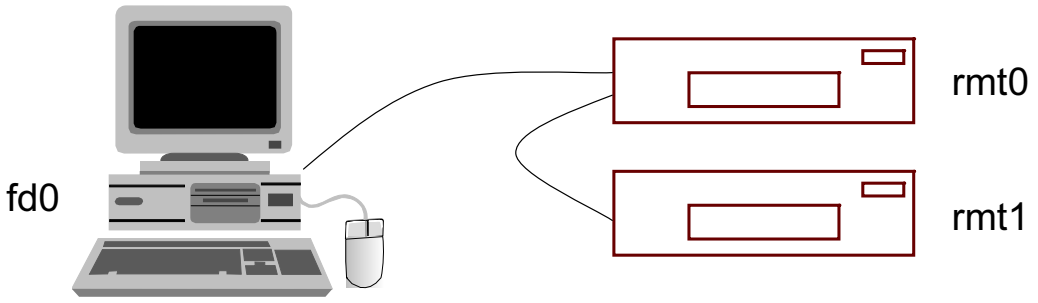


Figure 7-3. Listing of /dev directory AU1412.0

Notes:

Types of devices

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. This is usually only used for 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.

`ls -l /dev` output

The `ls -l` command allows you to see the type of a file. A special file (in the **/dev** directory) is 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 22 and a minor device number of 1. This indicates that the code to handle major device 22 must already be in the kernel, and it must handle device number 1 correctly. While not shown here, there would be files for **rmt0** with minor numbers of 0 through 7, each of which must be handled correctly by the device driver. 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 or the operation mode of a device of that type.

Examples of block devices

Following are examples of block devices:

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

Examples of character (raw) devices

Following are 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

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	23100020	pci	IBM 10/100Mbps Ethernet PCI Adapter (23100020)
adapter	14101800	pci	IBM PCI Tokenring Adapter (14101800)
adapter	chrp_ecp	isa_sio	CHRP IEEE1284 (ECP) Parallel Port Adapter
adapter	keyboard	kma_chrp	Keyboard Adapter

Customized Configuration Database

Name	Status	Location	Description
sa0	Available	01-S1	Standard I/O Serial Port
sioka0	Available	01-K1-00	Keyboard Adapter
rmt0	Available	10-80-00-0.0	SCSI 4mm Tape Drive
hdisk0	Available	10-80-00-4,0	16 Bit SCSI Disk Drive
hdisk1	Available	10-80-00-5,0	16 Bit SCSI Disk Drive
mem0	Available		Memory
ent0	Available	10-60	IBM 10/100 Mbps Ethernet PC Adapter (23100020)

lft	lft	node	Low Function Terminal Subsystem
diskette	fd	siofd	Diskette Drive
printer	ibm4019	parallel	IBM 4019 LaserPrinter

Figure 7-4. Device configuration database AU1412.0

Notes:

Introduction

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

Predefined database

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.

Customized database

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

Configuration Manager

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)

```
  # lsdev -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
..
..

  # lsdev -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 7-5. List all supported devices AU1412.0

Notes:

Introduction

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

Devices are classified by:

- `Class` indicates what the device does
- `Type` indicates what model it is
- `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.

Listing supported devices and device attributes

To find out what devices are listed in the predefined database, use the command: **`lsdev -P`** or 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.

To find out the default attributes of a predefined device, use the command **`lsattr -D`** with the device class or SMIT -> **Devices** -> **List Devices** -> **Show Characteristics of a Supported Device**.

Some options for the **`lsattr`** command include:

`-H` shows the headers for the output

`-c` specifies the class of device

List all defined devices

CuDv (Customized Devices)

```
# lsdev -C -H
name      status         location     description
sys0      Available      System       Object
pci0      Available      PCI Bus
isa0      Available      10-58        ISA Bus
sa0       Available      01-S1        Standard I/O Serial Port
scsi0     Available      10-80        Wide/Fast-20 SCSI I/O Controller
cd0       Available      10-80-00-3,0 SCSI Multimedia CD-ROM Drive
rmt0      Defined        10-80-00-6,0 4.0 GB 4mm Tape Drive
hdisk0    Available      10-80-00-4,0 16 Bit SCSI Disk Drive
hdisk1    Available      10-80-00-5,0 16 Bit SCSI Disk Drive
mem0      Available                   Memory
ent0      Available      10-60        IBM 10/100 Mbps Ethernet PCI
tok0      Available      10-90        IBM PCI Tokenring Adapter

# 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 7-6. List all defined devices AU1412.0

Notes:

Introduction

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

Listing customized device information

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 contains:

- `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 because the device is powered off or 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 -l [`*`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 **`-l`** 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 is listed such as EC level, FRU number, part number, and so forth. The output also displays the model architecture and bus type.

Device states

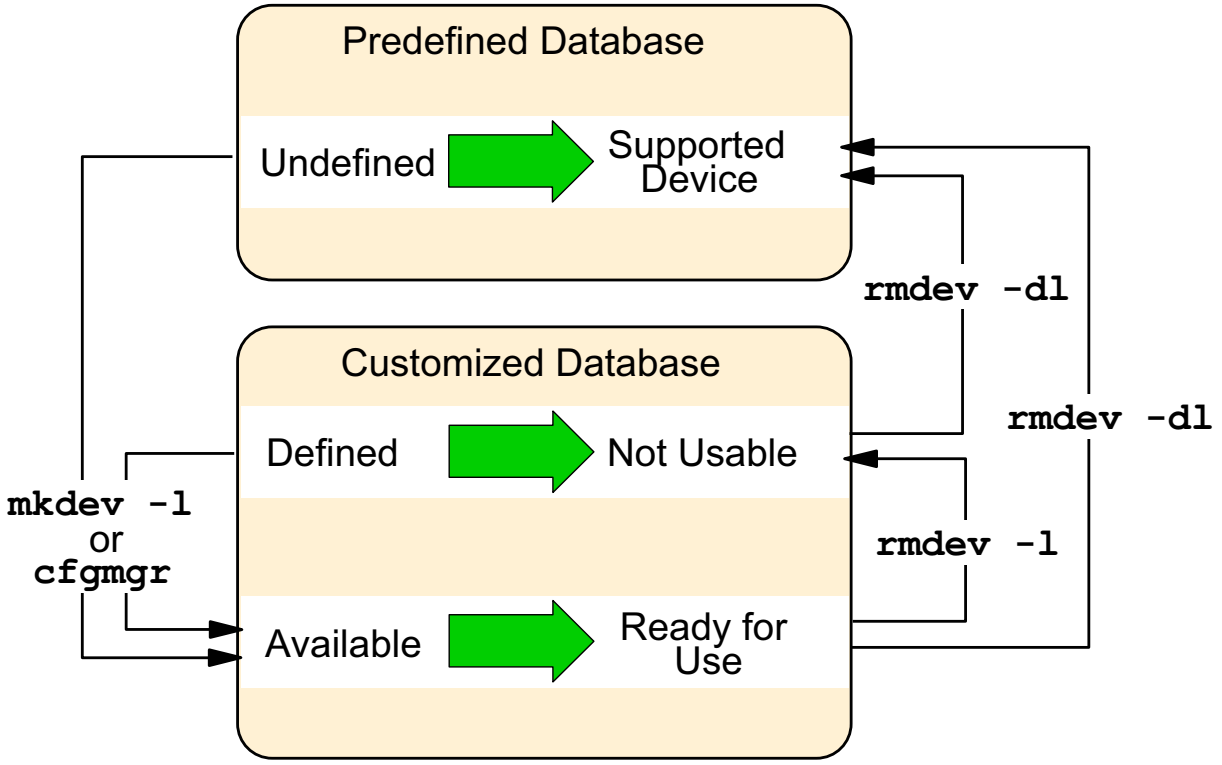


Figure 7-7. Device states AU1412.0

Notes:

Device states

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 appears 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.

Changing device states

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, you can use either:

- SMIT **-> Devices -> Configure/Unconfigure Devices -> Configure a Defined Device**
- The command: **mkdev -l rmt0**

To move an available tape device to defined:

- SMIT **-> Devices -> Configure/Unconfigure Devices -> Unconfigure a Device** then set **KEEP definition in database** to **yes**
- The command: **rmdev -l rmt0**

To permanently remove an available or define tape device:

- SMIT **-> Devices -> Configure/Unconfigure Devices -> Unconfigure a Device** then set **KEEP definition in database** to **no**
- The command: **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 the command **rmdev -dl xxxx** to remove the device from the ODM.

cfgmgr

The command that is executed at boot time to discover and configure attached devices is **cfmgr**. After booting the system, if a device is either removed/powered-off or attached/powered-on, the state of the device in the ODM is not automatically changed. Rather than running **mkdev** or **rmdev** on the specific device, you would typically execute **cfgmgr**. **cfgmgr** would discover any new devices or re-assesses the state of any devices already defined in the ODM customized device database.

Self-configuring devices

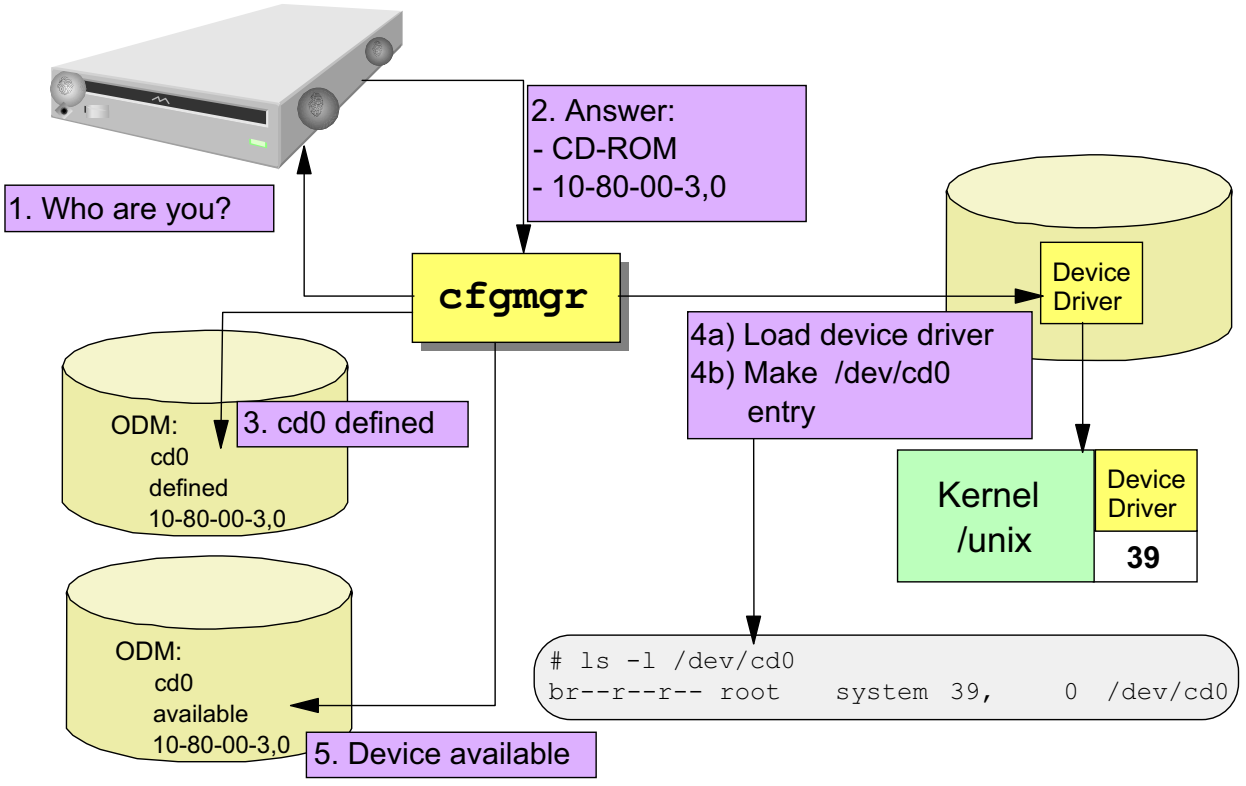


Figure 7-8. Self-configuring devices AU1412.0

Notes:

Configuring devices

cfgmgr 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. **cfgmgr** identifies 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 **cfgmgr** runs during the boot process, no more intervention is required by the administrator.

You see that for SCSI devices, you 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
              MPIO Management
              Fixed Disk
              Disk Array
              CD ROM Drive
              Read/Write Optical Drive
              Diskette Drive
              Tape Drive
              Communications
              Graphic Displays
              Graphic Input Devices
              Low Function Terminal (LFT)
              SCSI Initiator Device
              SCSI Adapter
              FC Adapter
              IDE Adapter
              iSCSI
              Asynchronous I/O
              Multimedia
              List Devices
              Configure/Unconfigure Devices
              Install Additional Device Software
              PCI Hot Plug Manager
              SSA Disks
              SSA RAID Arrays
```

Figure 7-9. SMIT Devices menu AU1412.0

Notes:

Introduction

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 varies depending on what you have configured or installed on your system.

Devices can also be managed using the Web-based System Manager.

SMIT device functions

Some of the SMIT 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**

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

- **PTY**

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

- **Communications**

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

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 7-10. Device addressing AU1412.0

Notes:

Location code

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.

Location code format

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 (only)
	XY	Resources attached to the XY PCI bus (For example - 10 or 1P)
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 7-11. Location code format for PCI devices AU1412.0

Notes:

Introduction

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, hard disks 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 Guide* for your model. These can be found online at the IBM Information Center:
http://publib.boulder.ibm.com/infocenter/pseries/v6r1/index.jsp

PCI location code format (AB)

The location code format in the visual is an example of what you might find on a 43P Model 150 System p system.

In older machines there was only a single PCI bus which used an **AB** value of **04**. In current machines these are multiple PCI buses which are assigned **AB** values which identify the bus, such as **10** or **IP**.

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 where either there is only one PCI bus or where the PCI bus cannot be determined.
- **XY** defines resources attached to the **XY** parent PCI bus, where **XY** is a two character identify for the bus determined by the machine designer. For example, a machine may have several PCI buses each numbered **10**, **20**, and so forth.

PCI location code format (CD)

The second set of digits, **CD**, identify a slot or adapter number. Again, how this position is used may vary from machine/model to machine/model.

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 discover. 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 are attached to one of the two PCI buses. Slots 2 and 3 are on the primary bus and 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 Guides.* To give one example, a card in slot 1 has an address of **1P-08**.

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

- **1P-08 Slot 1**
- **10-b0 Slot 2**
- **10-90 Slot 3**
- **1P-18 Slot 4**
- **1P-10 Slot 5**

For integrated devices, like the built-in keyboard port, the C position is a letter A-Z and the D position is 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 in which they are discovered during configuration. Each integrated device is assured a unique value.

PCI location code format (EF)

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

PCI location code format (GH)

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

Location code example: Non-SCSI

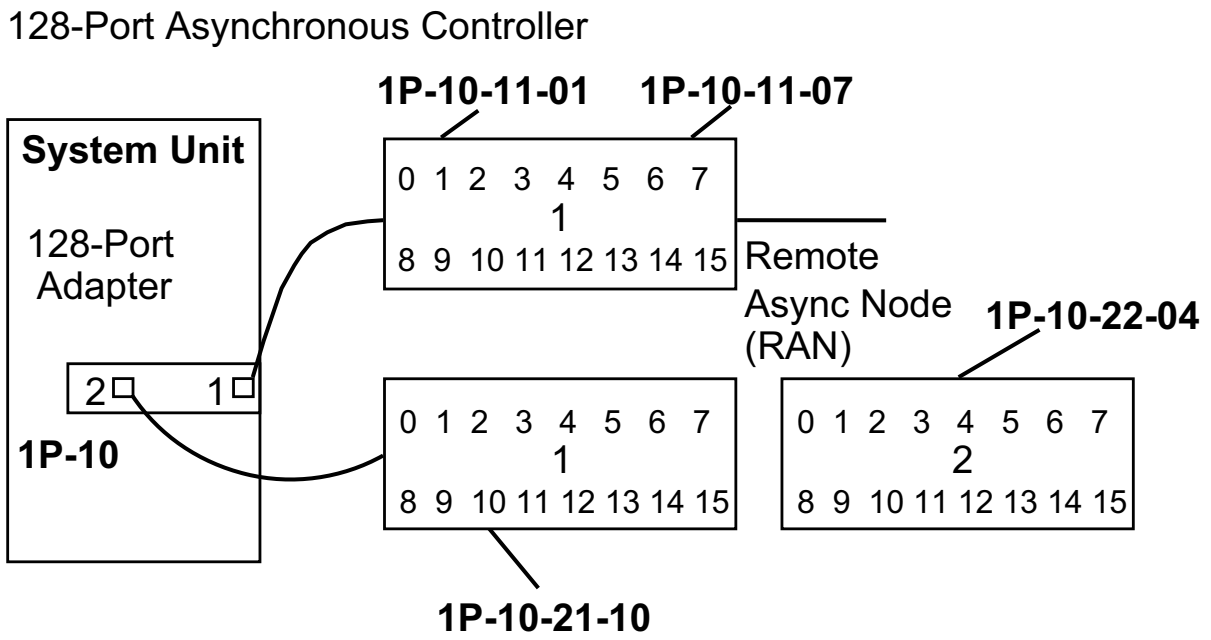


Figure 7-12. Location code example: Non-SCSI AU1412.0

Notes:

Non-SCSI example

The example in the visual illustrates non-SCSI device location codes for a System p server.

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.

Location code

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 7-13. Location code format for SCSI devices AU1412.0

Notes:

SCSI device example

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

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

AB-CD position

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

EF position

The **EF** position identifies the SCSI bus. If the controller provides only a single SCSI bus, the **EF** position is **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**.

G,H position

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 recognizes 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)

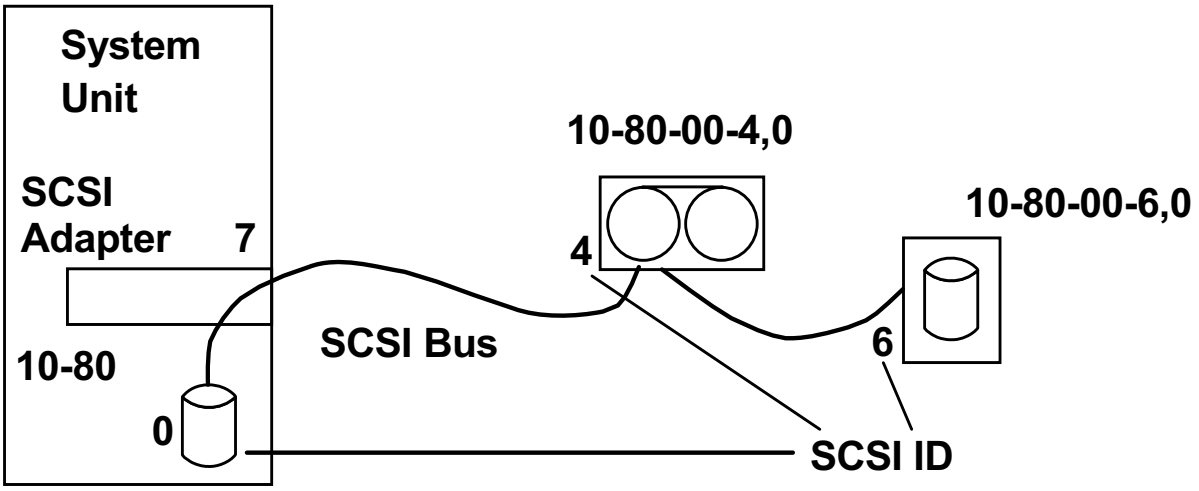


Figure 7-14. Location code example for SCSI device AU1412.0

Notes:

What is the location code?

This example shows several SCSI devices attached to a single SCSI bus on a 43P Model 150. 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 integrated 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.

Physical location codes

- Assigned by system firmware
- Used to uniquely identify hardware for:
  - Assigning adapters to logical partitions
  - Identifying field replaceable units (FRU)
- Structure of a physical location code:
  - <enclosure>.<planar>.<slot>-<port>-<logical location>
  - Enclosure is usually:
    <machine type>.<model>.<serial#>
  - Example, **U787A.001.DNZ0713-P1-C3**
- Displayed by default with **`lscfg`** command

Figure 7-15. Physical location codes AU1412.0

Notes:

Introduction

Physical addressing has been in place throughout the history of the RS/6000, pSeries, and System p family of products. An important change with the partitionable System p servers is that these codes are now, not only of concern to manufacturing engineers and IBM equipment service personnel, but also to the administrator of these systems.

This physical location code is generated by the system firmware and the scheme used to assign the location code can vary from one machine to the next.

Uses of physical location codes

With the System p servers, the unique identifier of a physical component is the physical location code. This identifier is used in problem isolation, identifying a failed component, and specifying the field replaceable unit (FRU) that is needed to fix the problem.

When partitioning a system, devices are assigned to each logical partition on an adapter by adapter basis and the physical location code is the unique identifier of each adapter. Since each partition may have a different operating system, the identification of components cannot depend upon an O/S unique location code such as the AIX location code.

Addressing scheme

The first part of the physical location code is the enclosure identification. On the older power4 platforms (ex p690) this would be the rack# and the position of a drawer in the enclosure. In most System p servers the enclosure is identified by noting the machine-type, model and serial number of that enclosure.

The next part of the location code is the physical location of a component in that enclosure. This would typically be the planar (or internal bus) and the connection to that bus (expressed as a slot position). The location code up to this point could be identifying an adapter. Optionally, the physical location may include a port# if that adapter has multiple ports.

Following this physical portion of the location code we can have a logical portion. This is used when a component is located by addressing on the adapter's bus rather than a fixed physical path. For example, the SCSI address of a device on a SCSI bus.

Important AIX commands

- List all adapters and their AIX location codes: **`lsdev -Cc adapter`**
- Show information about device including physical location code and the AIX location code: **`lscfg -vl`** ***`xyz`*** where ***`xyz`*** is the name of a device such as **ent0** or **ssa0**

Listing device physical locations

CuDv Customized Devices

```
# lsdev -C -H -F "name status physloc location description"
```

name	status	physloc	location	description
en1	Defined		01-08	Standard Ethernet Network
ent1	Defined	U789D.001.DQDWAYT-P1-C4-T1	01-08	10/100/1000 Base-TX
et1	Defined		01-08	IEEE 802.3 Ethernet
hdisk2	Defined	U7311.D20.107F67B-P1-C04-A8	02-08-01-8,0	16 Bit LVD SCSI Disk
hdisk3	Defined	U7311.D20.107F67B-P1-C04-A9	02-08-01-9,0	16 Bit LVD SCSI Disk
scsi0	Defined	U7311.D20.107F67B-P1-C04	02-08-00	PCI X Dual Channel
scsi1	Defined	U7311.D20.107F67B-P1-C04	02-08-01	PCI X Dual Channel
ses0	Defined	U7311.D20.107F67B-P1-C04-AF	02-08-01-15,0	SCS Enclosure Services
sisscsia0	Defined	U7311.D20.107F67B-P1-C04	02-08	PCI XDDR Dual Channel

Figure 7-16. Listing device physical locations AU1412.0

Notes:

Using `lsdev` to list location codes

The use of physical location codes is becoming more common, especially in working on problem determination involving the physical devices in a System p server.

By default the **`lsdev`** command only shows the traditional AIX locations codes, but it does allow us to ask for additional information.

The **`lsdev -C -H -F "name status physloc location description"`** displays the output in a user-specified format. The **`physloc`** format option provides the physical location code of a device and the **`location`** format option provides the AIX location code of a device.

Adding an ASCII terminal

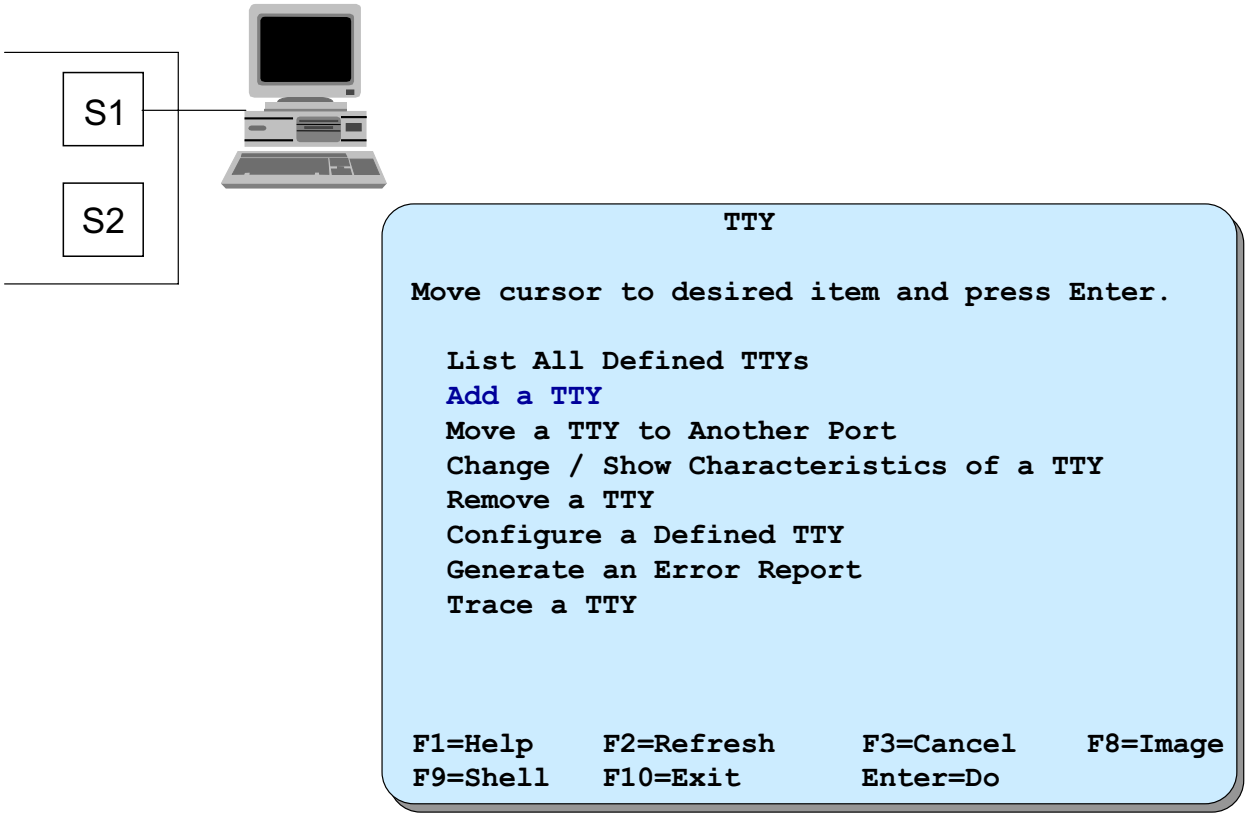


Figure 7-17. Adding an ASCII terminal AU1412.0

Notes:

Introduction

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

First steps

First, physically attach the terminal to the serial port. Be sure to note which serial port it is attached to. We 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   1P-03-11 16-Port RAN EIA-232 for 128-Port adapter
 sa3   Available   1P-03-12 16-Port RAN EIA-232 for 128-Port adapter
 sa4   Available   1P-03-13 16-Port RAN EIA-232 for 128 Port adapter
```


Figure 7-18. Attachment AU1412.0

Notes:

Introduction

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

TTY type

In this example, 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**.

Location code

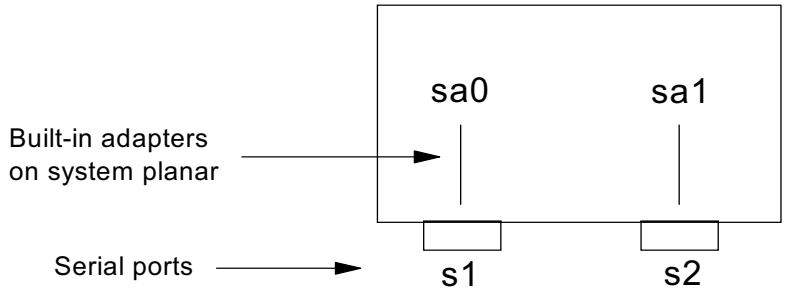
The location code is also displayed. **01-S1** is, in fact, the location code of serial port 1.

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.

Device nomenclature

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



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

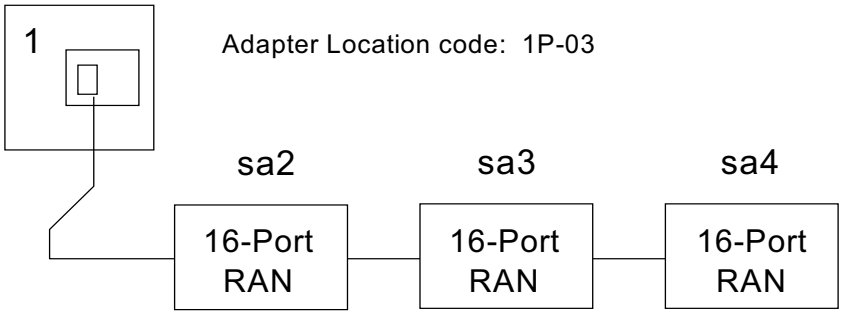


Figure 7-19. Device nomenclature AU1412.0

Notes:

Pictorial view

This visual shows a picture and the associated nomenclature for the scenario we are discussing on adding a TTY.

Add a TTY

```
                                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 LOGIN                                   disable            +
  BAUD rate                                     []                  +
  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...31]

F1=Help             F2=Refresh          F3=Cancel           F4=List
Esc+5=Reset         Esc+6=Command       Esc+7=Edit          Esc+8=Image
Esc+9=Shell         Esc+0=Exit          Enter=Do
```


Figure 7-20. Add a TTY AU1412.0

Notes:

PORT number

There is only one mandatory field on this screen and that is the **PORT number.** The **F4** key provides 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**

Enable LOGIN

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

Asynchronous line characteristics

The asynchronous line characteristics must be specified: **BAUD rate**, **PARITY**, **BITS per character**, **Number of 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.

TERMINAL type

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
- **`getconf -a`**
  - Provides the values of all system configuration variables

Figure 7-21. Documenting hardware configuration AU1412.0

Notes:

Commands to help document device configurations

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

The following commands are useful to help document your device configuration:

- **`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 such as the 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.

- **getconf -a**

  Writes the values of all system configuration variables to standard output.

getconf -a command examples

Following are examples of the **getconf** command:

```
# getconf BOOT_DEVICE
hdisk0

# getconf MACHINE_ARCHITECTURE
chrp

# getconf KERNEL_BITMODE
64

# getconf HARDWARE_BITMODE
64

# getconf REAL_MEMORY
131072

# getconf DISK_PARTITION /dev/hdisk0
16

# getconf DISK_SIZE /dev/hdisk0
8678
```

Checkpoint (1 of 2)

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

   _____

   _____

2. Use the output on the next visual (**`lsdev -C -H`**) to answer the following four questions.

   a) What happens if we attempt to add another device with the SCSI address set to 4?

   _____

   _____

   b) Can the 8 mm tape drive be currently used? Why?

   _____

   _____

   c) Where is the printer connected? _____

   d) The Ethernet adapter is installed in what slot?

   _____

   _____

Figure 7-22. Checkpoint (1 of 2) AU1412.0

Notes:

Checkpoint (2 of 2)

```
# lsdev -C -H
name    status     location      description
sys0    Available                System Object
pci0    Available                PCI Bus
isa0    Available  10-58         ISA Bus
ppa0    Available  01-R1         Standard I/O Parallel Port Adapter
lp0     Available  01-R1-00-00   IBM 4039 LaserPrinter
sa0     Available  01-S1         Standard I/O Serial Port 1
tty0    Available  01-S1-00-00   Asynchronous Terminal
mem0    Available                Memory
scsi0   Available  10-80         Wide SCSI I/O Controller
rmt0    Defined    10-80-00-3,0  5.0 GB 8 mm Tape Drive
hdisk0  Available  10-80-00-4,0  SCSI Disk Drive
ent0    Available  10-60         IBM PCI 10/100 Ethernet Adapter
```


Figure 7-23. Checkpoint (2 of 2) AU1412.0

Notes:

Exercise 7: Devices

- List device configuration
- List and change system parameters
- Configure a tape device
- Configure a CD-ROM device

Figure 7-24. Exercise 7: Devices AU1412.0

Notes:

Introduction

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

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

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`**.

Figure 7-25. Unit summary AU1412.0

Notes:

Unit 8. 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
- Exercise

References

SG24-7559	*AIX Version 6.1 Differences Guide*
Online	*AIX 6.1 Operating System and device management*

Note: References listed as "Online" above are available at the following address:

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

Unit objectives

After completing this unit, you 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 8-1. Unit objectives AU1412.0

Notes:

Components of AIX storage

- Files
- Directories
- File systems
- Logical storage
- Physical storage
- Logical Volume Manager (LVM)

Figure 8-2. Components of AIX storage AU1412.0

Notes:

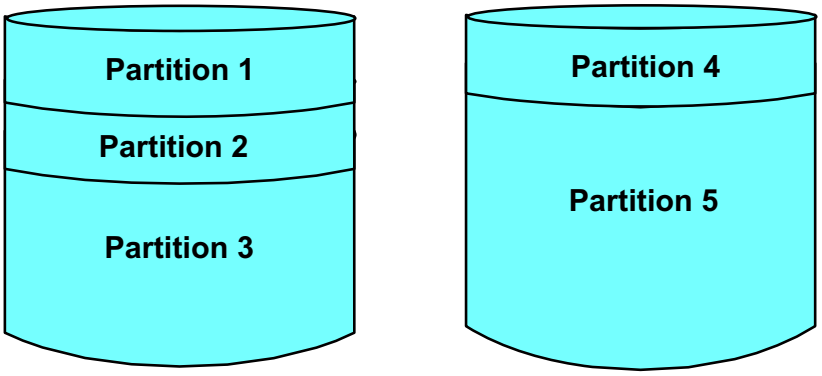
Components

The basic components or building blocks of AIX storage are:

- Files
- Directories
- File systems
- Logical storage
- Physical storage
- Logical Volume Manager (LVM)

As a user, you work with files and directories. As a system administrator, you 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 8-3. Traditional UNIX disk storage AU1412.0

Notes:

Issues with traditional UNIX disk storage

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 was on a partition on the hard disk.

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

A major limitation to partitions was that each partition had to consist of contiguous disk space. This characteristic limited the partition to reside on a single physical drive. It could not span multiple hard disks. Since file systems were always contained within a partition, no file system could be defined larger than the largest physical drive. This meant that no single file could exist larger than the largest physical drive.

Benefits of the LVM

- Logical volumes solve noncontiguous space problems
- Logical volumes can span disks
- Logical volume sizes can be dynamically increased
- Logical volumes can be mirrored
- Physical volumes are easily added to the system
- Logical volumes can be relocated
- Volume group and logical volume statistics can be collected

These tasks can be performed dynamically!

Figure 8-4. Benefits of the LVM AU1412.0

Notes:

Constraints virtually eliminated

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 in the visual can be performed while users are on the system.

Logical Volume Manager components

- Volume group (VG)
- Physical volume (PV)
- Physical partition (PP)
- Logical volume (LV)
- Logical partition (LP)

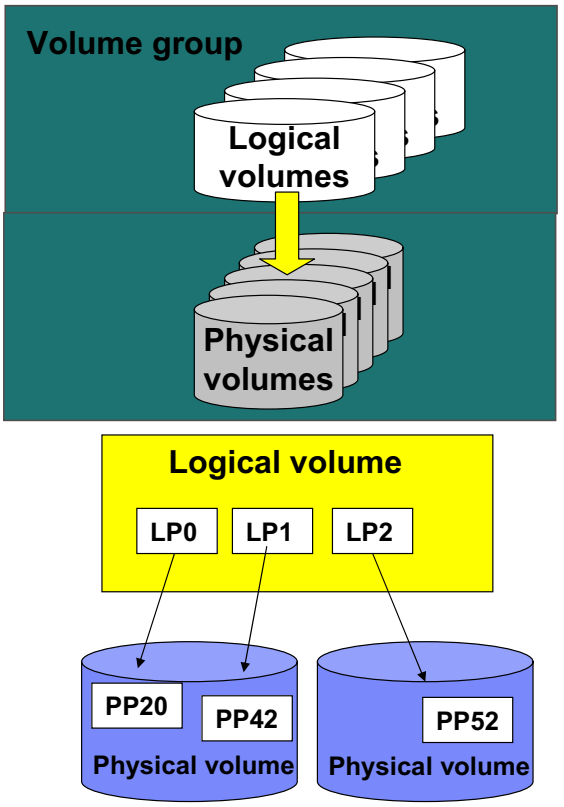


Figure 8-5. Logical Volume Manager components AU1412.0

Notes:

Introduction

The AIX Logical Volume Manager controls disk storage resources by mapping data between a simple and flexible logical view of storage space and the actual physical disks.

This visual and these notes provide a brief overview of the basic components of LVM.

Components

A hierarchy of structures is used to manage disk storage:

- Volume groups
- Physical volumes
- Physical partitions
- Logical volumes
- Logical partitions

Volume group (VG)

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.

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

Physical volume (PV)

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

For a disk to be used by LVM, the disk 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 volume group (VG).

Physical partition (PP)

All of the physical volumes in a volume group are divided into physical partitions (PPs). All the physical partitions within a volume group are the same size, although different volume groups can have different PP sizes.

Logical volume (LV)

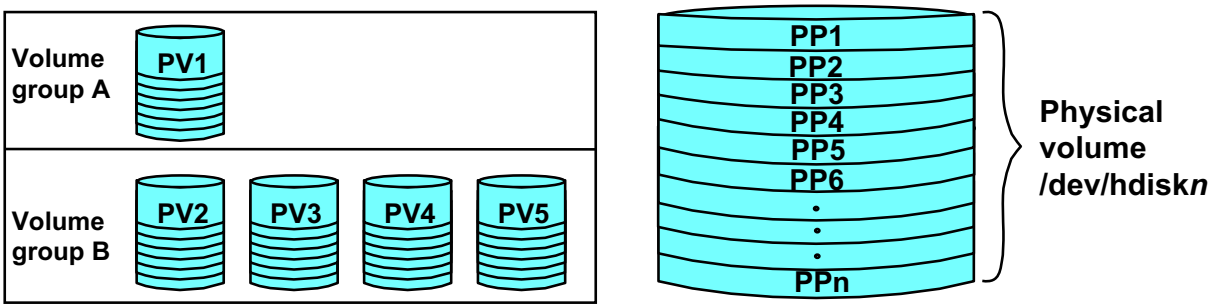
Within each volume group, one or more logical volumes (LVs) are defined. Logical volumes are groups of information located on physical volumes. Data on logical volumes appears to be contiguous to the user but can be non-contiguous on the physical volume or can even be located on several physical volumes.

Logical partition (LP)

Each logical volume consists of one or more logical partitions (LPs). Logical partitions are the same size as the physical partitions within a volume group. Each logical partition is mapped to at least one physical partition. Although the logical partitions are numbered consecutively, the underlying physical partitions are not necessarily consecutive or contiguous.

This allows file systems, paging space, and other logical volumes to be resized or relocated, to span multiple physical volumes, and to have their contents replicated for greater flexibility and availability in the storage of data.

Physical storage



Normal volume groups			Big volume groups	
-t factor	Disks (PVs)	PPs per PV	Disks (PVs)	-t factor
1	32	1016	128	1
2	16	2032	64	2
4	8	4064	32	4
8	4	8128	16	8
16	2	16256	8	16
N/A	N/A	32512	4	32
N/A	N/A	65024	2	64

Figure 8-6. Physical storage AU1412.0

Notes:

Introduction

Disk space on a physical volume (PV) is allocated to logical volumes (LVs) in chunks called physical partitions (PPs). Each physical partition size is the same across all the disks in a volume group (VG). The PP size is set at the time the VG is created. The size is set in megabytes on power of two boundaries (for example: 4 MB, 8 MB, 16 MB, and so forth). The default is 4 MB.

In AIX 5L V5.2 and later, LVM defaults the PP size of a new VG to the smallest PP size (equal or greater than 4 MB) which allows full addressing of the largest disk in the VG given the selected maximum number of PPs per PV (defaults to 1016). The smallest PP size is 1 MB, which is supported by using a larger number of PPs per PV.

When a PV is added to a system, a file called **hdisk*n*** 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.

Normal volume group

Originally AIX supported VGs with a maximum of 32 PVs, no more than 1016 PPs per disk and an upper limit of 256 LVs per VG. This VG type is commonly referred to as the original, normal or small volume group.

As disks increased in size, this meant that the PP size had to increase to use the entire disk space and stay within the 1016 PPs per disk limit. Larger PPs means less flexibility in allocating space for LVs, and potentially more wasted space.

For example, for an 18 GB disk, you must have a PP size of 32 MB (a PP size of 16 MB would require 1152 PPs, over the limit).

Volume group -t factor

To handle the increase in hard disk drive capacity over time, AIX V4.3.1 implemented a new volume group factor which can be specified by the **`-t`** flag of the **`mkvg`** command that allows you to increase the maximum number of PPs per disk proportional to the given integer multiplier value. The maximum number of PVs decreases proportional to the specified -t factor.

For example, if you wanted to use an 8 MB PP size with our 18 GB disks, you would need at least 2304 PPs per disk. Setting the -t factor to 4 would allow 4064 PPs per disk, but would limit us to 8 disks in the VG.

Big volume group

AIX V4.3.2 expanded the LVM scalability by introducing big volume groups. A big VG can have up to 128 physical volumes and a maximum of 512 LVs defined with it. The volume group -t factor can also be used with the big VG.

Using our 18 GB disk example, setting the -t factor to 4, would allow us to have a VG with a PP size of 8 MB and 32 disks.

Volume groups

- Volume group types:
  - Normal
  - Big
  - Scalable
- Limits

Volume Group Type	Max PVs	Max LVs	Max PPs per VG	Max PP Size
Normal	32	256	32512 (1016 * 32)	1 GB
Big	128	512	130048 (1016 * 128)	1 GB
Scalable	1024	4096	2097152	128 GB

- New physical volumes:
  - Add to existing volume groups
  - Create new volume group
- Why create new volume groups?
  - Separate user data from operating system files
  - Disaster recovery
  - Data portability
  - Data integrity and security

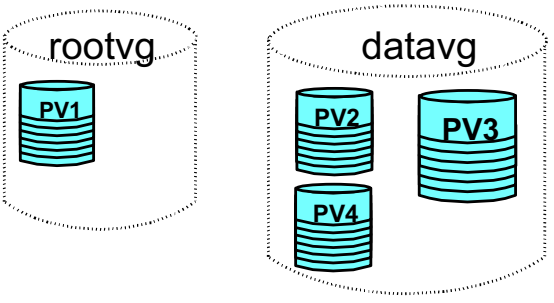


Figure 8-7. Volume groups AU1412.0

Notes:

Volume group types

With successive versions of AIX, new type of volume groups have been introduced which allow for greater capacities and greater flexibility:

- **Normal volume groups**

  When creating a volume group with SMIT or using the **mkvg** command, normal volume groups are the default.

- **Big volume groups**

  Big volume groups were introduced with AIX V4.3.2. A big volume group must be created using the command line command **mkvg -B**. Besides increasing the number of PVs per VG, the big volume group also doubled the maximum number of LVs per VG from 255 to 512. Support for creating big volume groups via SMIT was introduced in AIX 5L V5.3.

- **Scalable volume groups**

  Scalable volume groups were introduced with AIX 5L V5.3. A scalable VG can accommodate a maximum of 1024 PVs and raises the limit for the number of LVs to 4096. The -t factor does not apply to the scalable VG type.

  The maximum number of PPs is no longer defined on a per disk basis but applies to the entire VG. This opens up the prospect to configure VGs with a relatively small number of disks but fine grained storage allocation options through a large number of PPs which are small in size. The scalable VG can hold up to 2097152 (2048 KB) PPs. Optimally, the size of a physical partition can also be configured for a scalable VG.

  A scalable volume group is be created using the line command **`mkvg -S`**.

Existing and new volume groups

When the system is installed, the root volume group (**rootvg**) is created. **rootvg** consists of a base set of logical volumes and physical volumes required to start the system and any other logical volumes you specify to the installation script.

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.

Why create separate volume groups?

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 (VGDA)

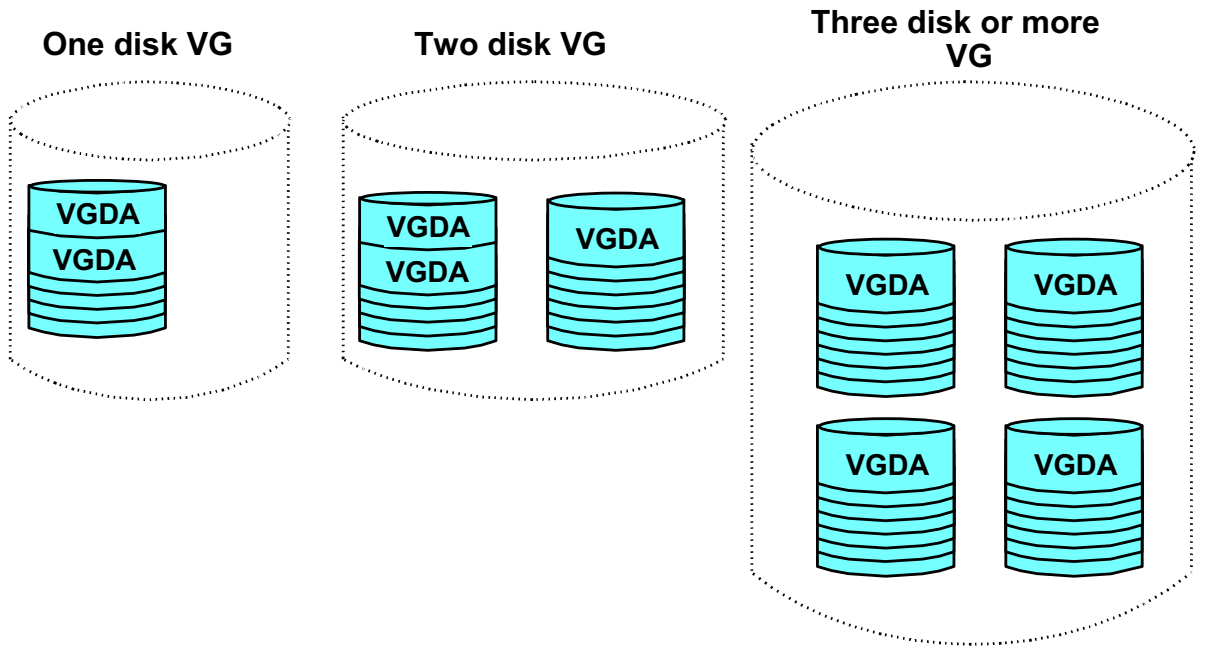


Figure 8-8. Volume group descriptor area (VGDA) AU1412.0

Notes:

Volume Group Descriptor Area (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 with either one or two disks (as shown in the visual).

Quorum

There must be a quorum of VGDAs available to activate the volume group and make it available for use (with the **`varyonvg`** command). 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 VGDAs available.

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

Logical storage

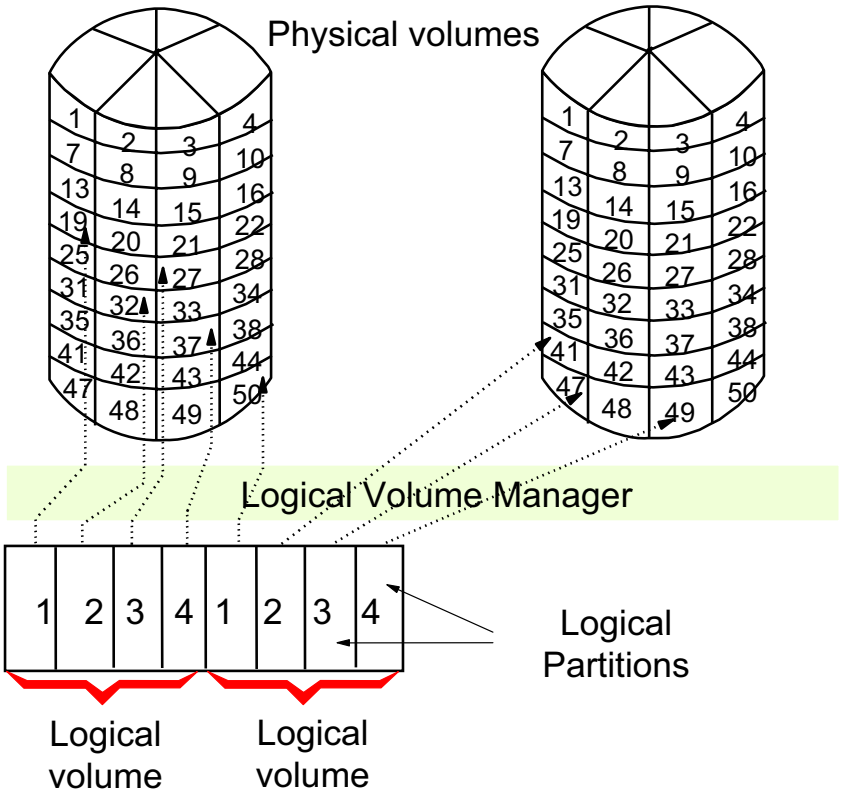


Figure 8-9. Logical storage AU1412.0

Notes:

Logical partition

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.

Logical volume

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 re-created smaller logical volume.

The mapping of which logical partition corresponds to which physical partition is maintained in the VGDA for the volume group. It is both a physical view and a logical view.

LVM mapping

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:
  - Journaled (JFS) or enhanced journaled file system (JFS2)
  - Journal log (**/dev/hd8**)
  - Paging space (**/dev/hd6**)
  - Boot logical volume (**/dev/hd5**)
  - Dump device
  - Nothing (raw logical volume)
- Examples of JFS/JFS2 logical volumes:

/dev/hd1	**/home**
/dev/hd2	**/usr**
/dev/hd3	**/tmp**
/dev/hd4	**/**
/dev/hd9var	**/var**
/dev/hd10opt	**/opt**
/dev/hd11admin	**/admin**
/dev/lv00	**/myfilesystem**

Figure 8-10. Uses of logical volumes AU1412.0

Notes:

Introduction

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 **`mklv`** command or go through the SMIT menus. This command allows you to specify the name of the logical volume and to define its characteristics.

JFS and JFS2 file systems

The native file system on AIX is the journaled file system (JFS), or the enhanced journaled 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.

Journal log

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. Journaled file systems and enhanced journaled file systems is discussed in greater detail later in the course.

Paging space

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

Boot logical volume

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

Dump device

When you install the operating system, the dump device is automatically configured for you. By default, the primary device is /**dev**/**hd6**, which is the paging logical volume, and the secondary device is /**dev**/**sysdumpnull**. For systems migrated from versions of AIX earlier than V4.1, the primary dump device is what it formerly was, /**dev**/**hd7**.

Raw logical volume

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

What is a file system?

- A file system is:
  - Method of storing data
  - Hierarchy of directories
- Seven types supported:
  - Journaled File System (JFS)
  - Enhanced Journaled File System (JFS2)
  - CD-ROM File System (CDRFS)
  - DVD-ROM File System (UDFS)
  - Network File System (NFS)
  - Common Internet Filesystem (CIFS)
  - Proc File System (PROCFS)
- Different file systems are connected together via directories to form the view of files users see

Figure 8-11. What is a file system? AU1412.0

Notes:

Introduction

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.

Supported file systems

AIX supports seven 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
- **UDFS** - Universal Disk Format (UDF) file system on DVD
- **CIFS** - Common Internet File System accessed across a network (via AIX Fast Connect)
- **NFS** - Network File System accessed across a network
- **PROCFS** - Proc file system maps processes and kernel data structures to corresponding files

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

Why have multiple 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 8-12. Why have multiple file systems AU1412.0

Notes:

Benefits

A file system is a structure that allows you to organize your data. It is one level in the hierarchy of you 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 provide the best performance.

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

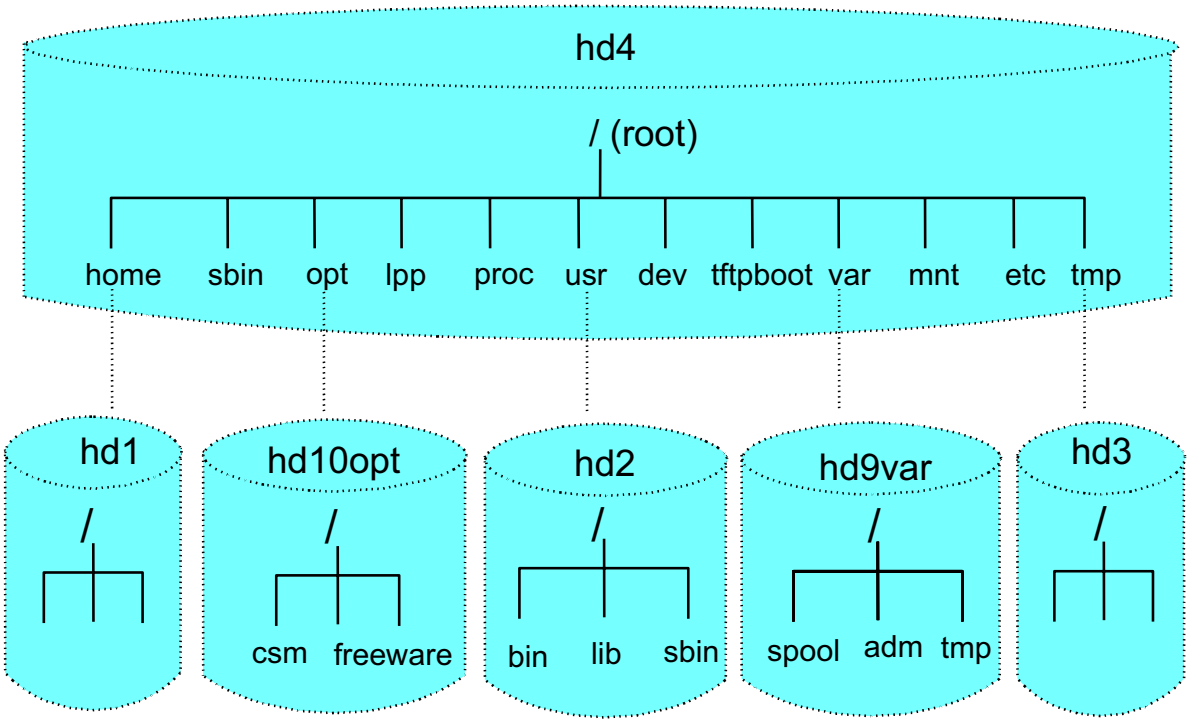
Limit disk usage

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.

Data is not all in one place

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 are not affected. Also, administrators can take a file system offline without affecting 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 volumes

Figure 8-13. Standard file systems in AIX AU1412.0

Notes:

Initial file systems

When AIX is first installed on a stand-alone system there are only seven journaled file systems and one pseudo file system (/**proc**) 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.

- **/opt** = **/hd10opt**

  Special file system to store freeware files.

- **/proc** = **/proc**

  Special pseudo file system kept in memory to support threads, or light weight processes. This file system is not designed to store user files. It is a type of file system which is different from a journal file system.

- **/admin** = **/hd11admin**

  There are two empty directories: lost_found and tmp. The permissions on this /admin/tmp directory is 755 and is owned by root. This tmp directory has more security for applications to use.

Let's review

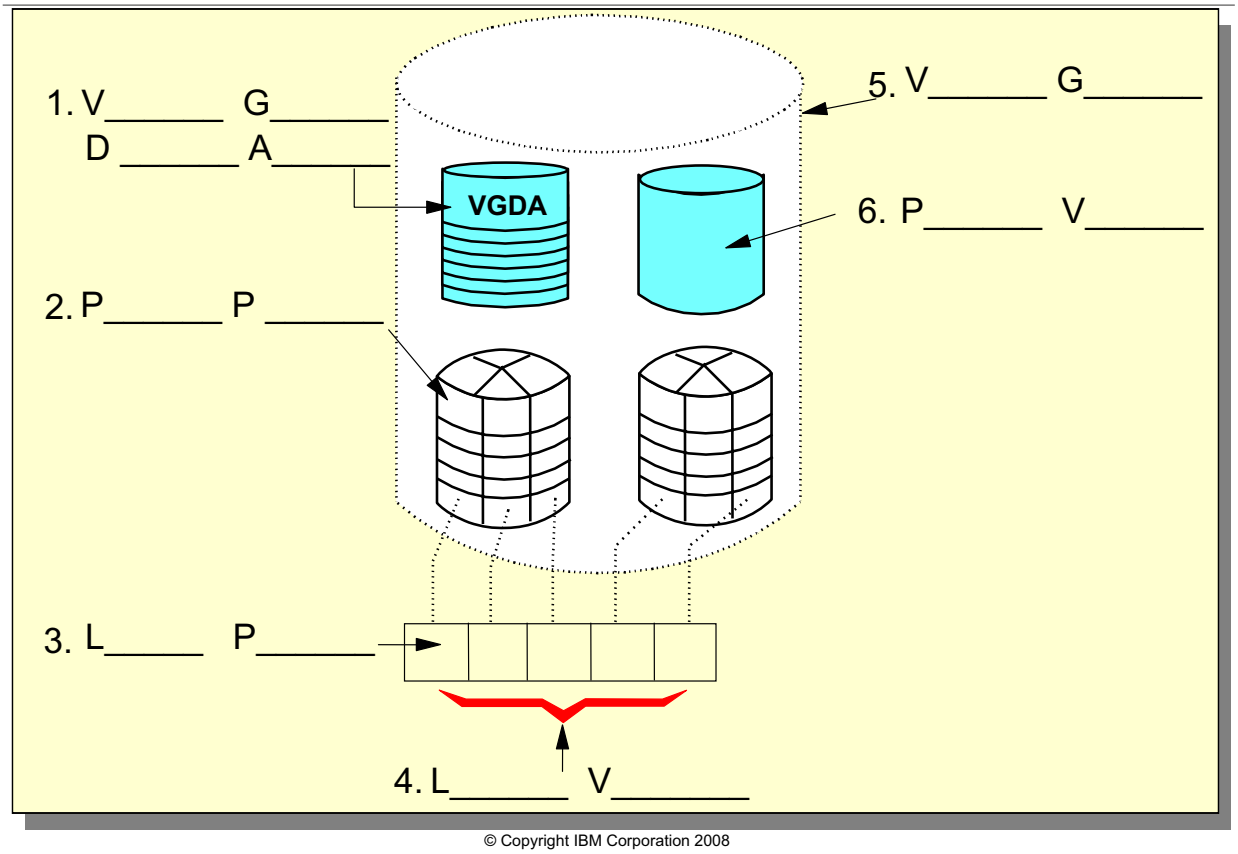


Figure 8-14. Let's review AU1412.0

Notes:

Instructions

Label the items shown in the picture above.

/etc/filesystems

```
/:
  dev         = /dev/hd4
  vol         = root
  mount       = automatic
  check       = false
  vfs         = jfs2
  log         = /dev/hd8
  type        = bootfs

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

/home/team01:
  dev         = /dev/fslv00
  vfs         = jfs2
  log         = /dev/loglv00
  mount       = true
  options     = rw
  account     = false
```


Figure 8-15. /**etc/filesystems** AU1412.0

Notes:

What is /etc/filesystems?

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.

File system attributes

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

`dev` For local mounts, identifies either the block special file where the file system resides, or the file or directory to be mounted.

`vol` Used by the **`mkfs`** command when initiating the label on a new file system.

`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 is not automatically mounted.

`check` Used by the **`fsck`** command to determine the default file systems to be checked. `True` enables checking.

`vfs` Specifies the type of mount. For example, `vfs=jfs2`.

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

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

`account` Used to determine the file systems to be processed by the accounting system.

`free` A component of traditional UNIX systems. Its is totally ignored by all AIX commands.

`quote` Allows system administrator to control the number of files and data blocks that can be allocated to a user or group.

Mount

- **`mount`** is 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

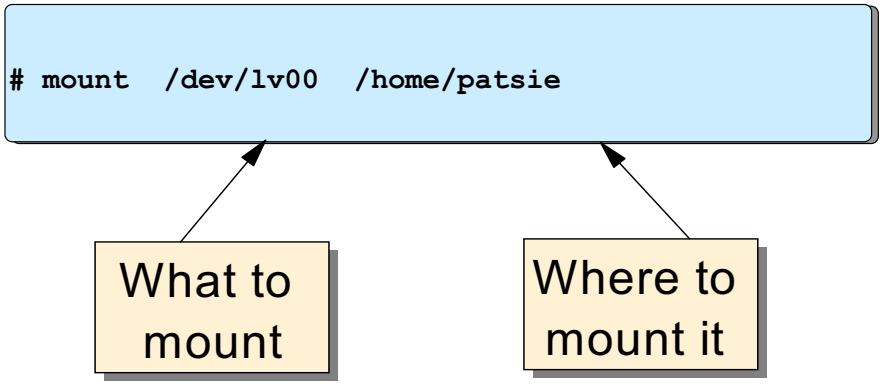


Figure 8-16. Mount AU1412.0

Notes:

Mounting a file system

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 umounted 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.

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.

Mount points

Full path names must be used when specifying the mount point. If SMIT is used to create the file system, the mount point is created automatically.

Mounting over an empty directory

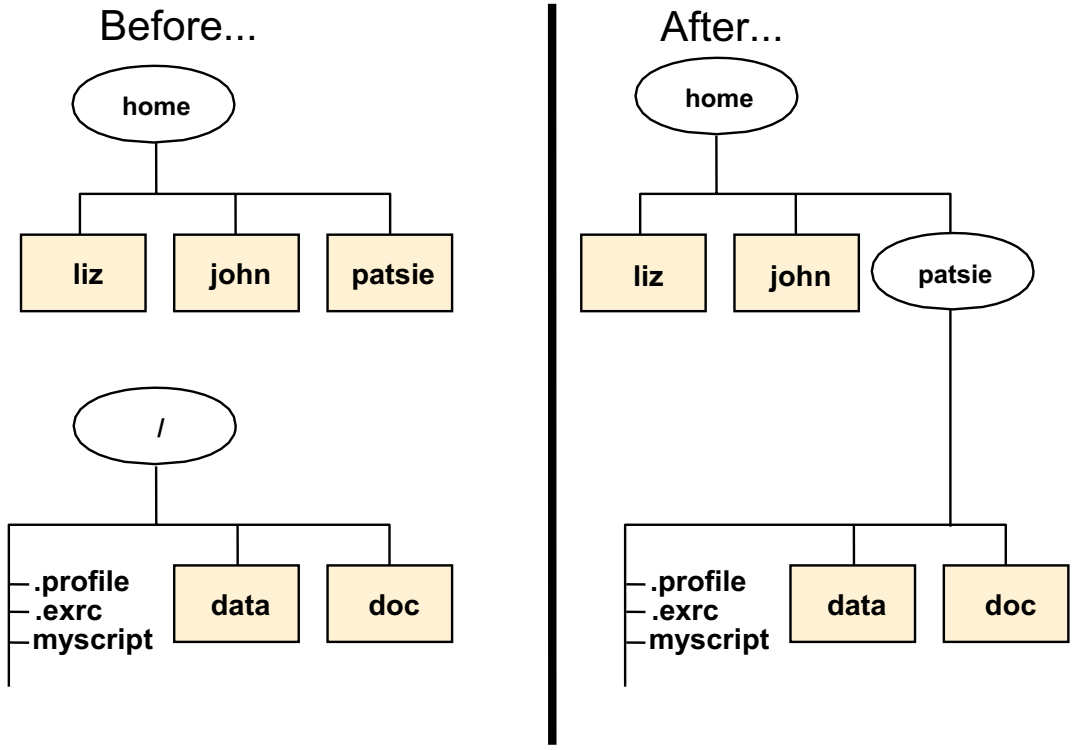


Figure 8-17. Mounting over an empty directory AU1412.0

Notes:

Accessing data in a file system

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 user's perspective, there is no way to tell where one file system ends and another begins.

Mounting over files

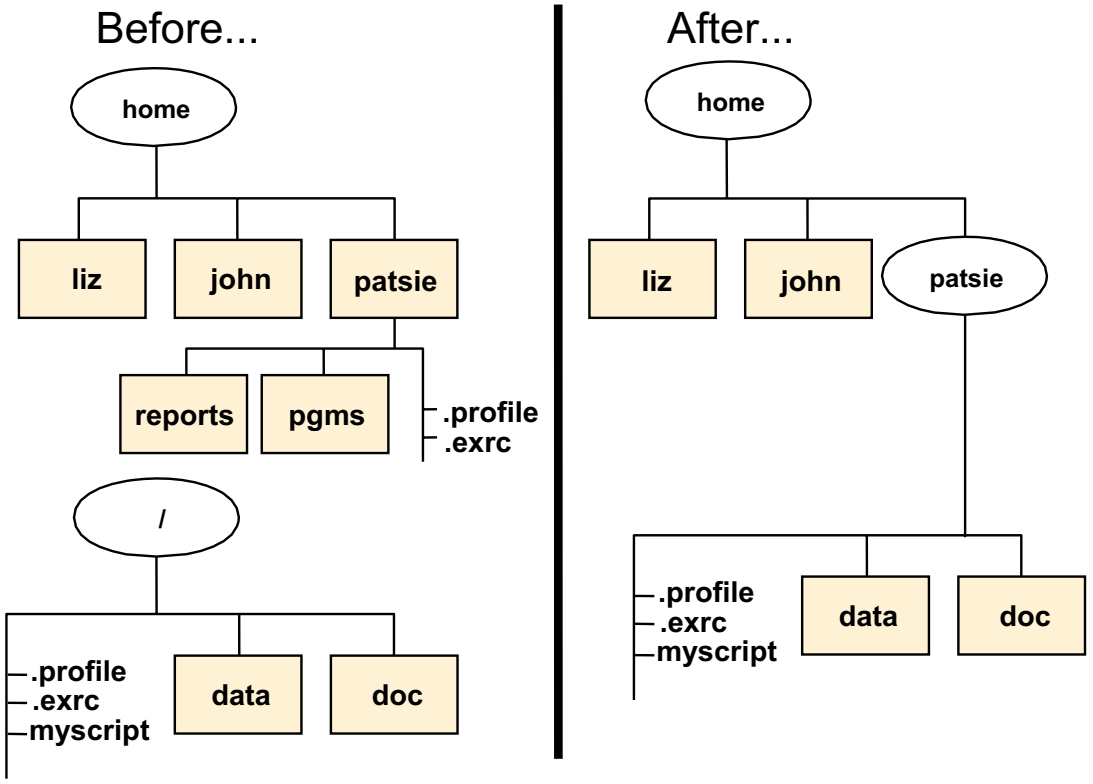


Figure 8-18. Mounting over files AU1412.0

Notes:

What happens when mounting over files?

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 are again accessible when the `unmount` 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. They can do device mounts only to the default mount points mentioned in the file /**etc**/**filesystems**. **root** can mount anywhere under any set of permissions.

Listing file systems

```
# lsfs
Name          Nodename  Mount Pt    VFS    Size    Options  Auto Accounting
/dev/hd4      --        /           jfs2   294912   --       yes  no
/dev/hd1      --        /home       jfs2   32768    --       yes  no
/dev/hd2      --        /usr        jfs2   3309568  --       yes  no
/dev/hd9var   --        /var        jfs2   65536    --       yes  no
/dev/hd3      --        /tmp        jfs2   131072   --       yes  no
/dev/cd0      --        /infocd     cdrfs   --      ro       yes  no
/dev/lv00     --        /home/john  jfs2   32768    rw       yes  no
/proc         --        /proc       procfs  --      --       yes  no
/dev/hd10opt --         /opt        jfs2   65536    --       yes  no
/dev/hd11admin--        /admin      jfs2   262144   --       yes  no
```


Figure 8-19. Listing file systems AU1412.0

Notes:

The `lsfs` command

You can list the various file systems that are defined using the **`lsfs`** command. This command displays information from /**etc/filesystems** and from the logical volumes in a more readable format. The **`lsfs`** command also displays information about CD-ROM file systems and remote NFS file systems.

The SMIT fastpath to get to the screen which accomplishes the same task as the **`lsfs`** command is: **`smit fs`**.

The syntax for the **`lsfs`** command is:

```
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.

Listing logical volume information

List all logical volumes for a volume group

```
# lsvg -l rootvg
LVNAME    TYPE    LPs    PPs    PVs   LV  STATE      MOUNT POINT
hd6       paging  32     32     1     open/syncd     N/A
hd5       boot    2      2      1     closed/syncd   N/A
hd8       jfslog  1      1      1     open/syncd     N/A
hd4       jfs2    9      9      1     open/syncd     /
hd2       jfs2    101    101    1     open/syncd     /usr
hd9var    jfs2    2      2      1     open/syncd     /var
hd3       jfs2    4      4      1     open/syncd     /tmp
hd1       jfs2    1      1      1     open/syncd     /home
hd10opt   jfs2    5      5      1     open/syncd     /opt
hd11adminjfs2     8      8      1     open/syncd     /admin
```


Figure 8-20. Listing logical volume information AU1412.0

Notes:

Viewing logical volume information

`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 physical partition (PP) sizes can be set within a single VG? _____
2. By default, how big are PPs?
   _____
   _____
3. How many volume groups (VGs) can a physical volume (PV) belong to?
   a) Depends on what you specify through SMIT
   b) Only one
   c) As many VGs as exist on the system
4. True or False? All VGDA information on your system is identical, regardless of how many volume groups (VGs) exist.

Figure 8-21. Checkpoint (1 of 3) AU1412.0

Notes:

Checkpoint (2 of 3)

Use the following output to answer the questions below:

```
# lsfs
Name        Nodename  Mount Pt   VFS    Size   Options  Auto   Accounting
/dev/hd4     --      /           jfs2   294912   --     yes       no
/dev/hd1     --      /home       jfs2   32768    --     yes       no
/dev/hd2     --      /usr        jfs2   3309568  --     yes       no
/dev/hd9var  --      /var        jfs2   65536    --     yes       no
/dev/hd3     --      /tmp        jfs2   131072   --     yes       no
/dev/hd10opt --      /opt        jfs2   163840   --     yes       no
/dev/cd0     --      /infocd     cdrfs            ro     yes       no
/dev/lv00    --      /home/john  jfs2   32768    rw     yes       no
/dev/hd11admin--     /admin      jfs2   262144   --     yes       no
```

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 8-22. Checkpoint (2 of 3) AU1412.0

Notes:

Checkpoint (3 of 3)

Use the following output to answer the question below:

```
# 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   jfs2     1       1       1      open/syncd     /var
hd3      jfs2     2       2       1      open/syncd     /tmp
lv00     jfs2     1       1       1      closed/syncd   /home/john
```

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

_____

Figure 8-23. Checkpoint (3 of 3) AU1412.0

Notes:

Exercise 8: LVM commands

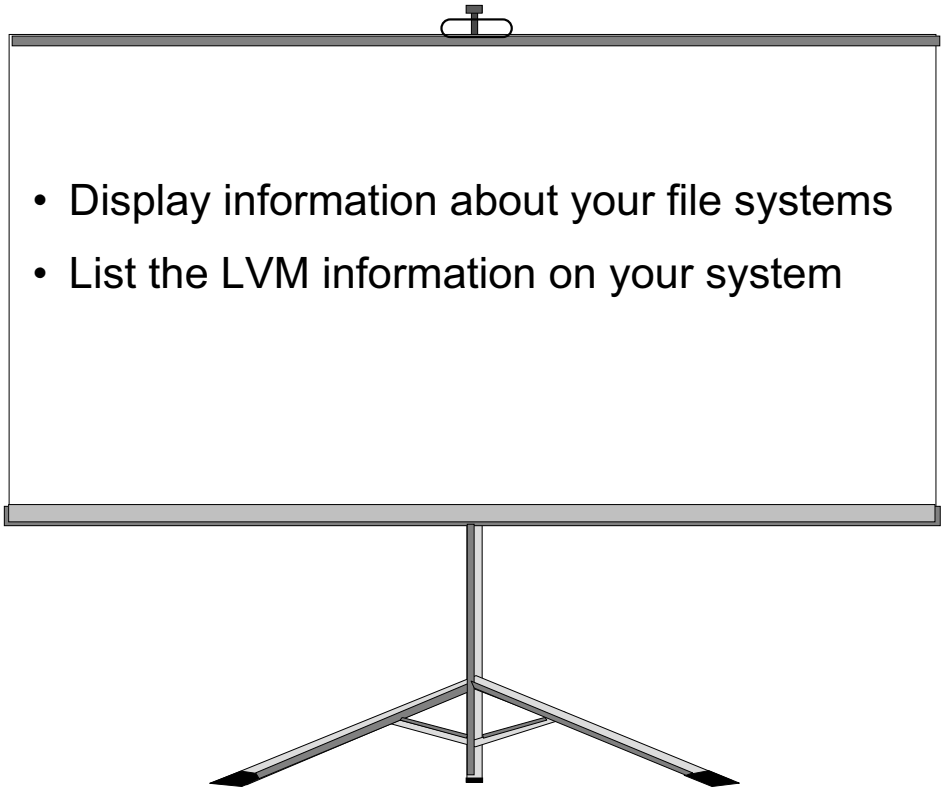


Figure 8-24. Exercise 8: LVM commands AU1412.0.

Notes:

Introduction

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

Unit summary

- The LVM is organized as follows:
  - A volume group consists of one or more physical volumes
  - Each physical volume is divided into physical partitions
  - A logical volume is made up of logical partitions
  - Logical partitions are mapped to physical partitions
- Logical volumes are used to contain:
  - JFS or JFS2 file systems
  - Journal log
  - Paging space
  - Dump space
  - Boot logical volume
  - Raw space
- The most common use of logical volumes is to contain JFS or JFS2 file systems

Figure 8-25. Unit summary AU1412.0

Notes:

Unit 9. 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, and delete:
  - Volume groups
  - Logical volumes
  - Physical volumes
- Describe mirroring
- Describe striping

How you will check your progress

Accountability:

- Checkpoint questions
- Exercise

References

SG24-7559	*AIX Version 6.1 Differences Guide*
Online	AIX 6.1 Operating system and device management
Online	AIX 6.1 Command References

Note: References listed as "Online" above are available at the following address:

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

Unit objectives

After completing this unit, you should be able to:

- Add, change, and delete:
  - Volume groups
  - Logical volumes
  - Physical volumes
- Describe mirroring
- Describe striping

Figure 9-1. Unit objectives AU1412.0

Notes:

Logical Volume Manager

```
# smit lvm
```

```
                    Logical Volume Manager

 Move cursor to desired item and press Enter.

   Volume Groups
   Logical Volumes
   Physical Volumes
   Paging Space





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


Figure 9-2. Logical Volume Manager AU1412.0

Notes:

Introduction

The SMIT **Logical Volume Manager** menu is used to manage many aspects of the system's storage. The Web-based System Manager can also be used to manage the Logical Volume Manager.

Volume groups

The SMIT **Volume Groups** menu provides facilities to manipulate the volume groups in the system.

Logical volumes

The SMIT **Logical Volumes** 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 does not contain an AIX file system, is being manipulated.

Physical volumes

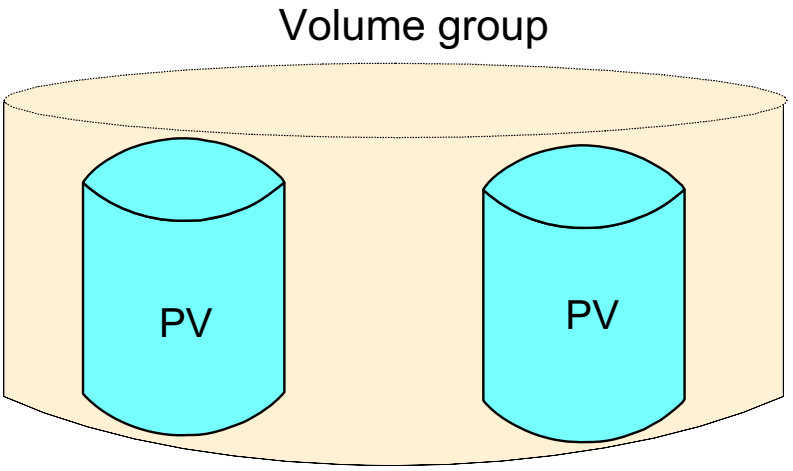
The SMIT **Physical Volumes** menu allows the user to configure the physical volumes (fixed disks) in the system. This menu duplicates options on the **Fixed Disks** menu of **Devices**.

Paging space

The SMIT **Page Space** menu allows a user to add, delete, activate, and list the paging spaces available.

9.1. Volume groups

Volume groups



- Physical Volume (PV)
  - Hard disk
- Volume Group (VG)
  - Collection of related disks (PVs)

Figure 9-3. Volume groups AU1412.0

Notes:

Physical volume

A physical volume is an actual disk or hard disk. There is a limit of 128 physical volumes per volume group unless it is a scalable volume group which can have up to 1024 physical volumes. A physical volume that supports removable media should be assigned to a volume group containing itself and no other members.

Volume group

A volume group is a collection of related physical volumes on a processor that:

- Are not members of another volume group
- 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 volume groups per system with a 32-bit kernel. The maximum volume groups per a 64-bit kernel is 4096.

SMIT Volume Groups menu

```
# smit vg
```

```
                              Volume Groups

      Move cursor to desired item and press Enter.

        List All Volume Groups
        Add a Volume Group
        Set Characteristics of a Volume Group
        List Contents of a Volume Group
        Remove a Volume Group
        Activate a Volume Group
        Deactivate a Volume Group
        Import a Volume Group
        Export a Volume Group
        Mirror a Volume Group
        Unmirror a Volume Group
        Synchronize LVM Mirrors
        Back Up a Volume Group
        Remake a Volume Group
        Preview Information about a Backup
        Verify the Readability of a Backup (Tape only)
        View the Backup Log
        List Files in a Volume Group Backup
        Restore Files in a Volume Group Backup

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


Figure 9-4. SMIT volume groups menu AU1412.0

Notes:

Volume group configuration

The visual shows the SMIT screen that allows for the configuration of volume groups. To get to this menu, use the SMIT fastpath, **`smit vg`**.

Let's describe these items throughout the course.

List all volume groups

```
# lsvg
rootvg
payrollvg

# lsvg -o
rootvg
```


Figure 9-5. List all volume groups AU1412.0

Notes:

The `lsvg` command

The **`lsvg`** command can be 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**).

List volume group contents

```
# lsvg rootvg

VOLUME GROUP:       rootvg           VG IDENTIFIER:  000bc6fd00004c00000000e10fdd7f52
VG STATE:           active           PP SIZE:        16 megabyte(s)
VG PERMISSION:      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 VG:     32512
MAX PPs per PV:     1016             MAX PVs:        32
LTG size (Dynamic): 256 kilobyte(s)  AUTO SYNC:      no
HOT SPARE:          no               BB POLICY:      relocatable
```


Figure 9-6. List volume group contents AU1412.0

Notes:

List volume group information

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:

- Volume group state (`VG STATE` - active or inactive/complete if all physical volumes are active)
- Physical partition size (`PP SIZE` - 4 MB by default)
- Total number of physical partitions (`TOTAL PPs`)
- Number of free physical partitions (`FREE PPs`)

List volume group information (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 9-7. List volume group information (physical volumes) AU1412.0

Notes:

The `lsvg -p` command

The **`lsvg -p`** ***`Volumegroup`*** command gives information about all of the physical volumes within the volume group. The information given is:

- Physical volume name (`PV_NAME`)
- Physical volume state (`PV STATE` - active or inactive)
- Total number of physical partitions (`TOTAL PPs`)
- Number of free physical partitions (`FREE PPs`)
- How the free space is distributed across the disk (`FREE DISTRIBUTION`)

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.

List volume group information (logical volumes)

```
# lsvg -l rootvg

rootvg:
LVNAME   TYPE    LPs   PPs  PVs   LV STATE      MOUNT POINT

hd6      paging  32    32   1     open/syncd    N/A
hd5      boot    2     2    1     closed/syncd  N/A
hd8      jfslog  1     1    1     open/syncd    N/A
hd9var   jfs2    1     1    1     open/syncd    /var
hd4      jfs2    9     9    1     open/syncd    /
hd2      jfs2    101   101  1     open/syncd    /usr
hd3      jfs2    4     4    1     open/syncd    /tmp
hd1      jfs2    1     1    1     open/syncd    /home
hd10opt  jfs2    5     5    1     open/syncd    /opt
hd11admin jfs2   8     8    1     open/syncd    /admin
lv00     jfs2    1     2    2     open/syncd    /home/john
lv01     jfs2    4     4    2     open/syncd    /home/fred
```


Figure 9-8. List volume group information (logical volumes) AU1412.0

Notes:

The `lsvg -l` command

The **`lsvg -l`** ***`Volumegroup`*** command gives information about all of the logical volumes within the volume group. The details given are:

- Logical volume name (`LVNAME`)
- Type of logical volume (`TYPE`, for example, file system, paging)
- Number of LPs (`LPs`)
- Number of physical partitions (`PPs`)
- Number of physical volumes (`PVs`)
- Logical volume state (`LV STATE`)
- Mount point (`MOUNT POINT`), if the logical volume contains a journaled file system

Add a Volume Group

```
# smit mkvg
```

```
                    Add a Volume Group

Move cursor to desired item and press Enter.

  Add an Original Volume Group
  Add a Big Volume Group
  Add a Scalable Volume Group
```

```
                  Add an Original 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                          +
* PHYSICAL VOLUME names                        []               +
  FORCE the creation of volume group?          no               +
  Activate volume group AUTOMATICALLY          yes              +
    at system restart?
  Volume group MAJOR NUMBER                    []               +#
  Create VG Concurrent Capable?                no               +
```


Figure 9-9. Add a Volume Group AU1412.0

Notes:

The `mkvg` command

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 the smallest physical partition size consistent with the maximum PP/PV and the largest physical volume in the volume group.

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.

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

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

Using SMIT

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 Systems (HANFS) and High Availability Cluster Multi-Processing (HACMP) applications.

The item on the SMIT dialog screen referring to concurrent mode operation have no meaning on systems without HACMP installed. This item is valid on AIX V4.2 and later.

There is a separate SMIT panel for adding a big volume group which is identical to this panel.

Add a Scalable Volume Group

```
# smit mkvg
```

```
                    Add a Scalable 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                          +
  * PHYSICAL VOLUME names                        []                +
    FORCE the creation of volume group?          no                +
    Activate volume group AUTOMATICALLY          yes               +
      at system restart?
    Volume group MAJOR NUMBER                    []                +#
    Create VG Concurrent Capable?                no                +
    Max PPs per VG in units of 1024              32                +
    Max Logical Volumes                          256               +


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


Figure 9-10. Add a Scalable Volume Group AU1412.0

Notes:

Additional options for scalable volume groups

There is a separate SMIT panel for adding scalable volume groups. Besides creating a different format VGDA, the administrator has the option to set the **Maximum PPs per VG** and the **Max Logical Volumes** for the volume group.

With non-scalable volume groups, LVM allows tuning of the number of physical partitions for each physical volume via the -t factor. In scalable volume groups, the physical partitions are managed on a volume group wide basis.

The maximum number of logical volumes was fixed depending upon the type of volume group. Now, in scalable volume groups the maximum is tunable.

Set Characteristics of a Volume Group

```
# smit vgsc
```

```
          Set Characteristics of a Volume Group

   Move cursor to desired item and press Enter.

     Change a Volume Group
     Add a Physical Volume to a Volume Group
     Remove a Physical Volume from a Volume Group
     Reorganize a Volume Group




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


Figure 9-11. Set Characteristics of a Volume Group AU1412.0

Notes:

Volume group operations

Once the volume group has been created you can do four operations on the volume group:

- Modify the attributes of the volume group
- Increase the size of the volume group by adding physical volumes
- Decrease the size of the volume group by removing physical volumes
- Reorganize the volume group

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                                    rootvg
* Activate volume group AUTOMATICALLY                  yes           +
    at system restart?
* A QUORUM of disks required to keep the volume        yes           +
   group on-line ?
  Convert this VG to Concurrent Capable?               no            +
  Change to big VG format?                             no            +
  Change to scalable VG format?                        no            +
  LTG Size in kbytes                                   128           +
  Set hotspare characteristics                         n             +
  Set synchronization characteristics of stale         n             +
   partitions
  Max PPs per VG in units of 1024                      32            +
  Max Logical Volumes                                  256           +

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


Figure 9-12. Change a Volume Group AU1412.0

Notes:

Activate volume group automatically

The SMIT option **Activate volume group AUTOMATICALLY at system restart** calls the **chvg** command 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**.

Quorum of disks

The SMIT option **A QUORUM of disks required to keep the volume group on-line?** determines if the volume group is automatically varied off (deactivated) after losing its quorum of physical volumes. Selecting **no** means that the volume group stays active until it loses all of its physical volumes. However, if this option is set to **no** then you are in danger of having backdated VGDAs. To activate a non-quorum user-defined volume group, all of the physical volumes within the volume group must be accessible or the activation fails.

Scalable volume group options

If the volume group is a scalable volume group, then you can change the **Max PPs per VG** and the **Max Logical Volumes**. Note that the screen says the units are in kilobytes. This is misleading text. The value is simply a count in units of 1024 physical partitions. For example, a value of 2 would indicate 2048 physical partitions per volume group. The default is 32 (32,768 physical partitions). Allowable values are powers of 2 up to 2048 (2,097,152 physical partitions).

Logical track group (LTG) size

- LTG is the maximum transfer size of a logical volume
- Prior to AIX 5L V5.3:
  - Default LTG size is 128 KB
  - LTG size can be changed by the **-L** flag on the **chvg** or **mkvg** command
- AIX 5L V5.3 and V6.1:
  - AIX dynamically sets the LTG size (calculated at each volume group activation)
  - LTG size can be changed with the command:
    varyonvg -M <*LTGsize*>
  - The **mkvg -L** flag is no longer supported
  - The **chvg -L** flag has no effect on volume groups created in AIX 5L V5.3 or later
  - Enable variable LTG on old volume groups using **chvg -L 0**
- To display the LTG size of a disk, use the command:
  # /usr/sbin/lquerypv -M <*hdisk#*>

Figure 9-13. Logical track group (LTG) size AU1412.0

Notes:

Logical track group (LTG) size

When LVM receives a request for an I/O, it breaks the I/O down into logical track group (LTG) sizes before it passes the request down to the device driver of the underlying disks. The LTG is the maximum transfer size of a logical volume and is common to all the logical volumes in the volume group since it is a volume group attribute.

Prior to AIX 5L V5.1, the only supported LTG was 128 KB. In AIX 5L V5.1 and V5.2, LVM accepted LTG values of 128 KB, 256 KB, 512 KB, and 1024 KB. However, many disks now support transfer sizes larger than 1 MB. To take advantage of these larger transfer sizes and get better disk I/O performance, AIX now accepts values of 128 KB, 256 KB, 512 KB, 1 MB, 2 MB, 4 MB, 8 MB, and 16 MB for the LTG size.

The default LTG size prior to AIX 5L V5.3 is 128 KB. In AIX now, LVM dynamically discovers the optimal LTG size each time the volume group is varied on.

Setting the LTG size

In AIX 5L V5.1 and V5.2, the LTG size is set by the **-L** flag on the **chvg** or **mkvg** command.

In AIX now, if you want to specify the LTG size, you can set using the **varyonvg -M** flag. The following command sets the LTG size of the **myvg** volume group at 512 KB:

```
# varyonvg -M512K myvg
```

The **chvg -L** flag has no effect on volume groups created in AIX 6.1, but allows the LTG size to change on standard and big volume groups which were created on AIX releases prior to AIX 5L V5.3.

The following command enables variable LTG for a **tmpvg** volume group created prior to AIX 5L V5.3 at the next **varyonvg** and sets the logical track group size to the common maximum transfer size of the disks (because of the **-L 0**):

```
# chvg -L 0 tmpvg
```

If this command is not executed, a volume group created prior to AIX 5L V5.3 has the old LTG size in AIX 5L V5.3 and AIX 6.1, and this LTG size behaves the same way it did in the prior release.

You can specify 128, 256, 512, or 1024 instead of 0 on the command line to indicate 128 KB, 256 KB, 512 KB, or 1024 KB LTG size, respectively. The value should be less than or equal to the maximum transfer size of all disks in the volume group. The default size is 128 kilobytes.

Displaying the LTG size

To obtain what the maximum supported LTG size of your hard disk is, you can use the **lquerypv** command with the **-M** flag. The output gives the LTG size in KB. For example:

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

In AIX 6.1, this information is automatically included in the **lspv** command output that is discussed later.

Hot spare

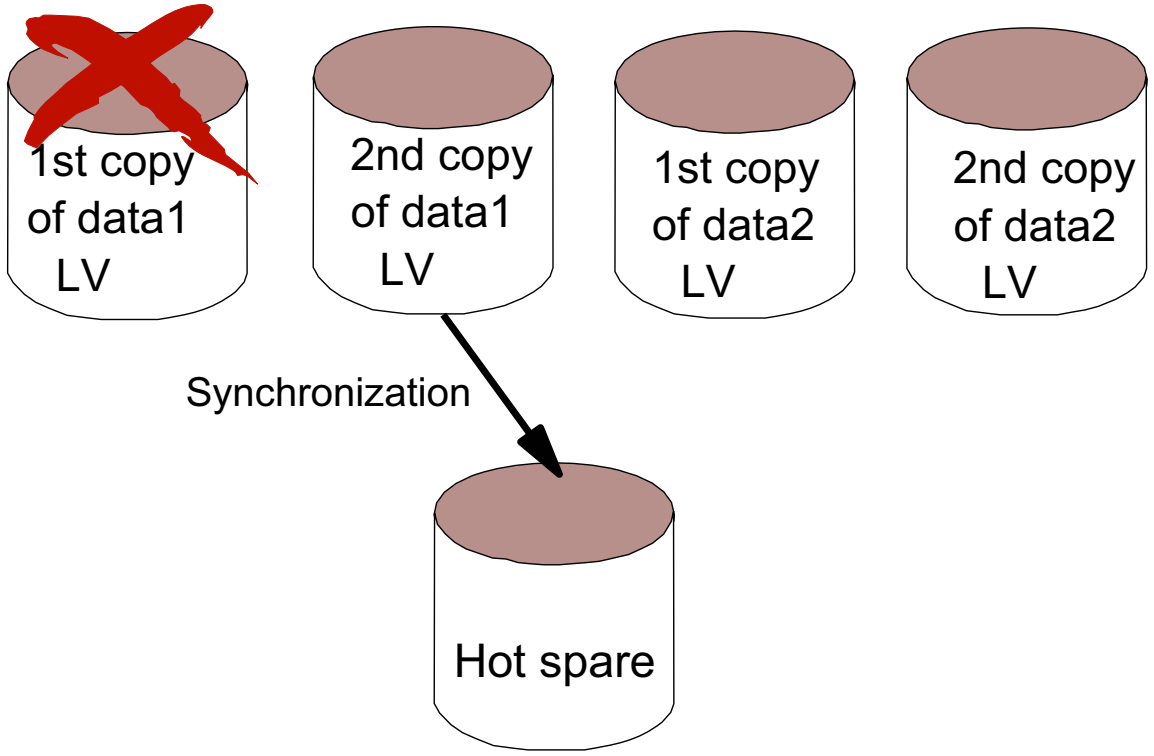


Figure 9-14. Hot spare AU1412.0

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.

Designate a hot spare disk

To designate a disk as a hot spare disk, use the **chpv -h** command.

For example, to mark **hdisk1** as a hot spare disk, use the command:

```
# chpv -hy hdisk1
```

To remove **hdisk1** from the hot spare pool, use the command:

```
# chpv -hn hdisk1
```

Set the hot spare migration policy for the volume group

The **chvg -h** command sets the migration policy for the volume group. The following table shows the arguments that can be used with the **chvg -h** command:

chvg -h argument	Description
y (lower case)	Permits one for one migration of partitions from one failed 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 is used.
Y (upper case)	Migrates partitions from a failing disk to one or more hot spare disks; might use the complete pool of hot spare disks.
n	No automatic migration takes place. (Default)
r	Removes all disks from the pool of hot spare disks for this volume group.

For example, to set an automatic migration policy which uses the *one* smallest hot spare disk that is large enough to replace the failing disk in the **datavg** volume group, use the command:

```
# chvg -hy datavg
```

Set the hot spare synchronization policy for the volume group

The **chvg -s** command is used to specify the synchronization characteristics. The following two values are valid for the synchronization argument:

chvg -s argument	Description
y	Automatically attempts to synchronize stale partitions.
n	Does not automatically attempt to synchronize stale partitions. (Default)

For example, to automatically synchronize stale partitions for the **datavg** volume group, use the command: **`chvg -sy datavg`**

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	**`extendvg`**	Add the hot spares to the volume groups which they are to protect
4	-	Decide which hot spare policy is most effective for your volume groups
5	**`chpv`**	Designate the selected disks as hot spares
6	**`chvg`**	Decide which synchronization policy meets the business needs, and set the policy
7	-	Sleep well at night!

Instead of using the command line interface, you can use the Web-based System Manager to make the changes on the hot spare information.

Extending and reducing volume groups

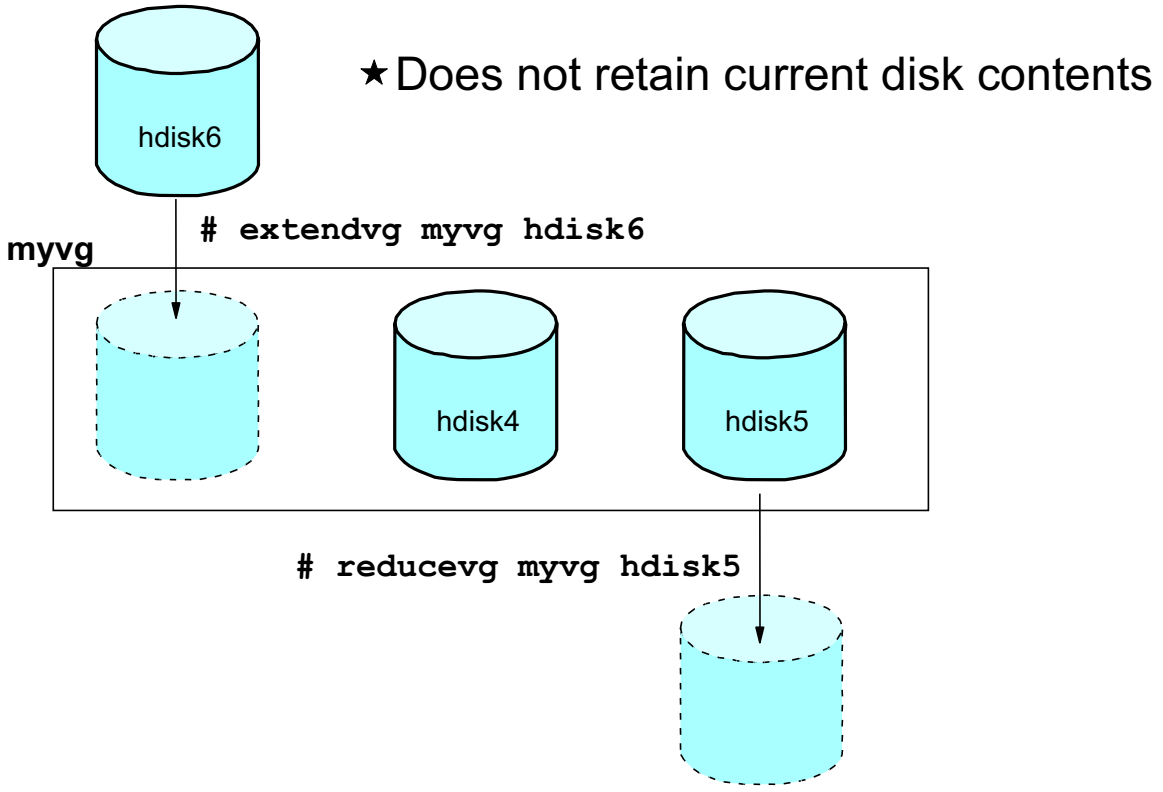


Figure 9-15. Extending and reducing volume groups AU1412.0

Notes:

Add a Physical Volume to a Volume Group

To add a disk to an existing volume group, use the **extendvg** command or SMIT fastpath **smit extendvg**. The disk must be installed in the system or connected to it externally, and must be powered on.

extendvg formats the disk into physical partitions and then add these to the physical partition mapping maintained in the VGDA for the volume group. The space on the new disk is now available to be allocated to logical volumes in the is volume group. If the existing data in the VGDA 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 contains data which is no longer needed.

The syntax for the **extendvg** command is:

extendvg [-f] ***Volumegroup*** **hdiskn**

Remove a Physical Volume from a Volume Group

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

To remove a disk from the volume group, first be sure to free up all the storage on the disk by either deleting the logical volumes or migrating them to some other disk in the volume group. Once there are no logical volumes on the disk, you can remove that disk from the volume group by using the **reducevg** command or the SMIT fastpath **smit reducevg**.

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

```
reducevg [-d] [-f] Volumegroup hdiskn
```

The **-d** option deallocates the existing logical volume partitions and then deletes resultant empty logical volumes from the specified physical volumes. User confirmation is required unless the **-f** flag is added.

Remove a Volume Group

```
# smit reducevg2
```

```
                        Remove a Volume Group

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

                                                      [Entry Fields]
  * VOLUME GROUP name                                 []      +




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


Figure 9-16. Remove a Volume Group AU1412.0

Notes:

How to remove a volume group

You can use the **smit reducevg2** fastpath to remove a volume group. It runs a script which identifies what physical volumes are in the volume group and then runs the **reducevg** command to remove each physical volume until there are no more physical volumes in the volume group.

The **Remove a Volume Group** menu 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 (which calls the **reducevg** command). This removes the volume group when you remove the last physical volume within it.

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

reducevg [-d] [-f] ***VolumeGroup PhysicalVolume***

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 9-17. Activate/Deactivate a volume group AU1412.0

Notes:

The `varyonvg` command

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. It allows a volume group to be made active that does not currently have a quorum of available disks. Any disk that cannot be brought to an active state is put in a removed state. At least one disk must be available for use in the volume group.

The `varyoffvg` command

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.

In AIX 6.1 the option **-M** allows the specification of a logical track group size for the volume group, instead of allowing LVM to determine it dynamically.

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                   []            +#





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


Figure 9-18. Import/Export a Volume Group AU1412.0

Notes:

Exporting a volume group

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 all information about the volume group from the system. To export a volume group it must be inactive.

Importing a volume group

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 **rootvg**.

Advanced RAID support

- Checks all disks in a volume group if they have grown in size:

  chvg -g *Volumegroup*

  ```
  # chvg -g datavg
  ```

- Turns on bad block relocation policy of a volume group:

  chvg -b [y/n] *Volumegroup*

  ```
  # chvg -b y datavg
  ```

- Turns off bad block relocation policy of a volume group:

  ```
  # chvg -b n datavg
  ```


Figure 9-19. Advanced RAID support AU1412.0

Notes:

Has the disk grown in size?

Modern storage subsystems, such as RAID arrays, have the ability to increase the size of what looks like a disk to LVM. The command **chvg -g *vgname***, examines all the disks in the volume group to see if they have grown in size. If any disks have grown in size it attempts to *dynamically* add additional physical partitions to the physical volumes. If necessary, the proper -t factor is applied or the volume group is converted to a big volume group.

Bad block relocation

The command **chvg -b y *vgname***, turns on the bad block relocation policy of a volume group. The command **chvg -b n *vgname***, turns off the bad block relocation policy of a volume group.

Bad block relocation policy should be turned off for RAID devices and storage subsystems unless the manufacturer tells you otherwise.

Exercise 9: Working with LVM (parts 1-2)

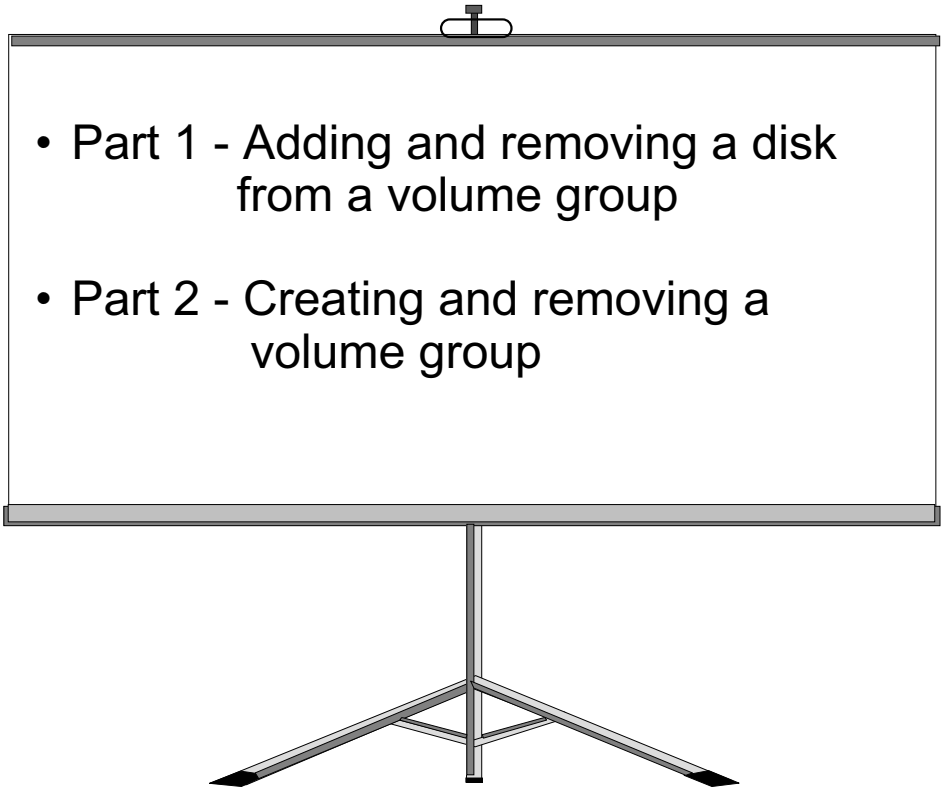


Figure 9-20. Exercise 9: Working with LVM (parts 1 and 2) AU1412.0

Notes:

Introduction

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

9.2. Logical volumes

Logical storage

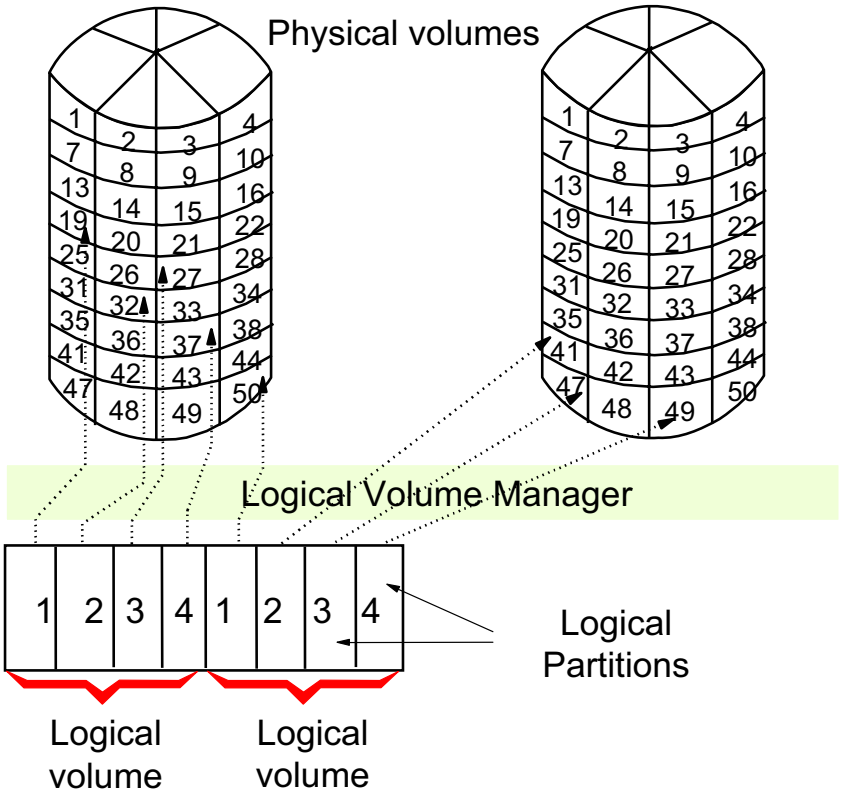


Figure 9-21. Logical storage AU1412.0

Notes:

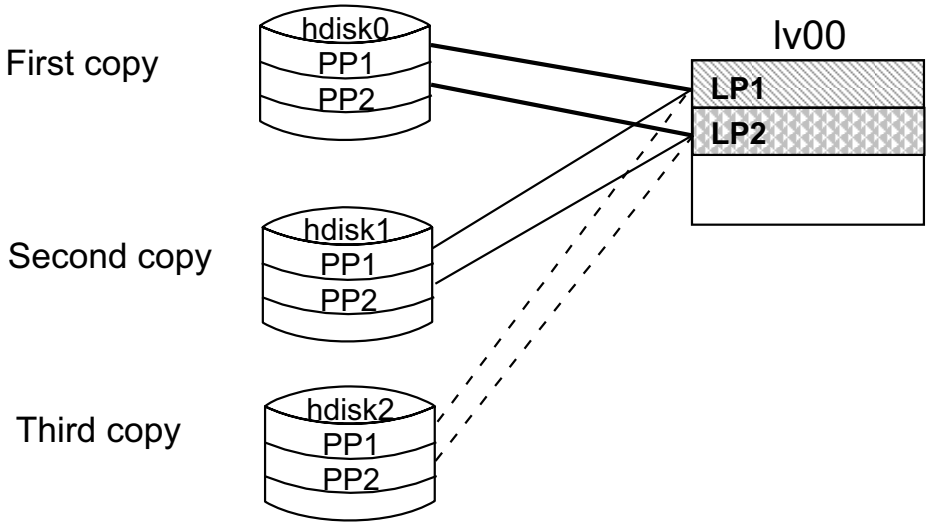
Logical volumes

A logical volume is a group of logical partitions which may span physical volumes (as long as the physical volumes are in the same volume group). A file system resides on top of a logical volume (LV). A logical volume can be dynamically extended.

Logical partitions

Logical partitions are mapped one-to-one to physical partitions unless they are being mirrored.

Mirroring



- Mirroring is when a logical partition maps to more than one physical partition of the same volume group
- Scheduling policy:
  Parallel: Physical partitions written simultaneously
  Sequential: Physical partitions written in sequence

Figure 9-22. Mirroring AU1412.0

Notes:

Mirroring concept

Mirroring of data over multiple drives protects against a potential hardware failure. The structure of LVM enables mirroring by manipulating the relationship between the physical partition and the logical partition.

The AIX mirror function does not apply to a physical disk, only to logical volumes. This is the most important principle to understand for the AIX LVM mirroring function.

In a normal operating environment each physical partition is mapped to a logical partition. When you mirror data, the ratio becomes one logical partition to two physical partitions for a two-way mirror. Or, one logical partition to three physical partitions for a three-way mirror.

Scheduling policies

The scheduling policy determines how reads and writes are conducted to a mirrored logical volume. The following table describes the four possible scheduling policies.LVM offers several scheduling policies for mirrored volumes to control how data is written and read from the copies. The following table shows the scheduling policies and the **mklv** or **chlv** arguments to set the policies.

Policy	Write Operation	Read Operation	mklv or chlv argument
Sequential	Sequential	Sequential	**-d s**
Parallel	Parallel	Parallel	**-d p**
Parallel/sequential	Parallel	Sequential	**-d ps**
Parallel/round-robin	Parallel	Round-robin	**-d pr**

Sequential write

Sequential mirroring performs writes to multiple copies or mirrors in order. The multiple physical partitions representing the mirrored copies of a single logical partition are designated primary, secondary, and tertiary. In sequential scheduling, the physical partitions are written to in sequence. The system waits for the write operation for one physical partition to complete before starting the write operation for the next one. When all write operations have been completed for all mirrors, the write operation is complete.

Parallel write

Parallel mirroring simultaneously starts the write operation for all the physical partitions in a logical partition. When the write operation to the physical partition that takes the longest to complete finishes, the write operation is completed.

Sequential read

When a sequential read is specified, the primary copy of the read is always read first. If that read operation is unsuccessful, the next copy is read. During the read retry operation on the next copy, the failed primary copy is corrected by LVM with a hardware relocation. This patches the bad block for future access.

Parallel read

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 and then the tertiary. If those are also busy, the read is initiated in the copy with the least number of outstanding I/Os.

Round-robin read

Round-robin reads alternate between copies. This results in equal utilization for reads even when there is more than one I/O outstanding.

Which is right for me?

Each of the scheduling policies provide benefits, as well as drawbacks. When deciding on a method of mirroring, you need to take into consideration how critical the data is, as well as performance. The trade off is performance versus availability.

In general, a mirrored logical volume is slower than an unmirrored logical volume, because you have to write the data in two or three places. The exception can be a mirrored LV in a high-read environment. If your application does mostly reads, and you are using parallel or parallel/round robin scheduling, reads may complete faster because the I/Os are spread across multiple disks, which can occur simultaneously if the disks are on separate controllers.

One of the parallel scheduling policies usually provides the best performance in a write intensive environment, because writes can proceed in parallel. However, there is some additional overhead, and mirrored logical volumes are usually slower than comparable unmirrored logical volumes in a write intensive environment.

Sequential scheduling provides the worst performance, but provides the best chance of recovering data in the event of a system crash in the middle of a write operation. Sequential scheduling makes it more likely that you have at least one good copy (the primary copy) of a logical partition after a crash. This is discussed further in the next visual.

Synchronizing

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 **mklvcopy** 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 is marked stale.

Mirror write consistency

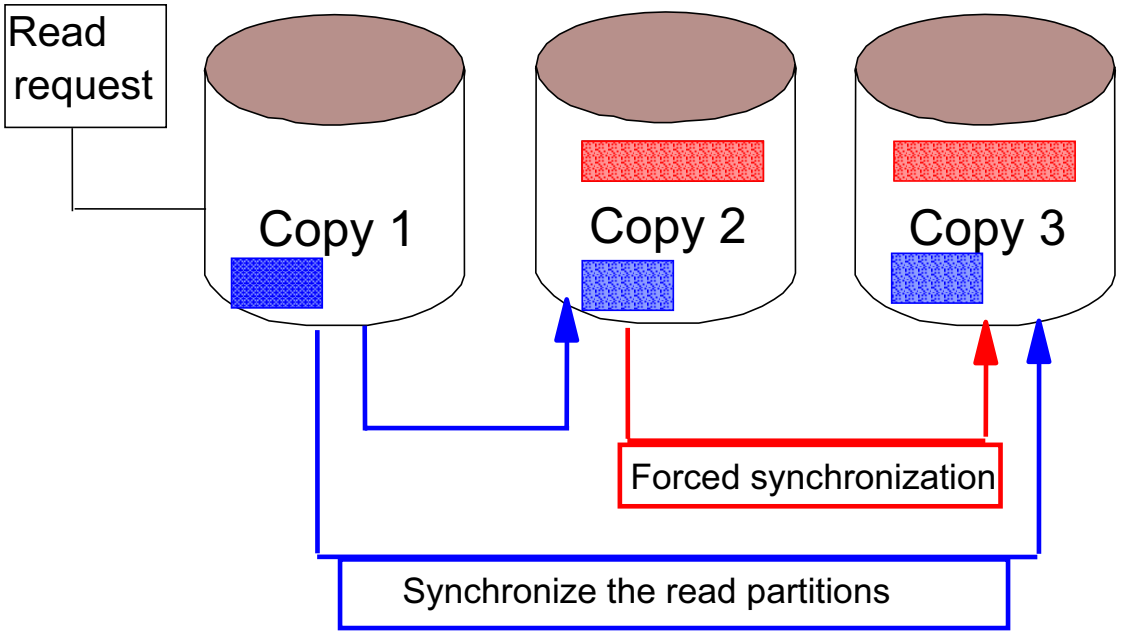


Figure 9-23. Mirror write consistency AU1412.0

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. Records of the last 62 distinct logical transfer groups (LTG) written to disk are kept in memory and also written to a separate checkpoint area on disk (MWC log). This results in a performance degradation during random writes.

With AIX 5L V5.1 and later, there are two ways of handling MWC:

- Active, the existing method (prior to AIX 5L V5.1)
- Passive, the new method (beginning with AIX 5L V5.1)

The purpose of the MWC

The purpose of the MWC is to guarantee the consistency of the mirrored logical volumes in case of a crash. The consistency of the file systems is guaranteed by the JFS logs.

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 LVM starts a forced synchronization of the entire logical volume when the system restarts. Data consistency exists for reads that occur during the synchronization so that applications can start using the data as soon as the volume group is varied on.

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.

MWC settings

The following syntax is used with either the **mklv** or **chlv** command to set MWC options:

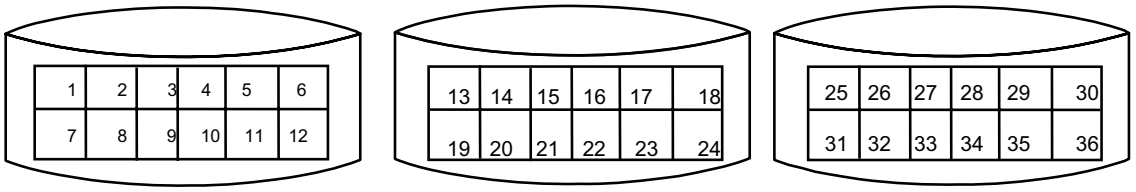
```
mklv -w y|a|p|n
chlv -w y|a|p|n
```

Following is a description of the MWC arguments:

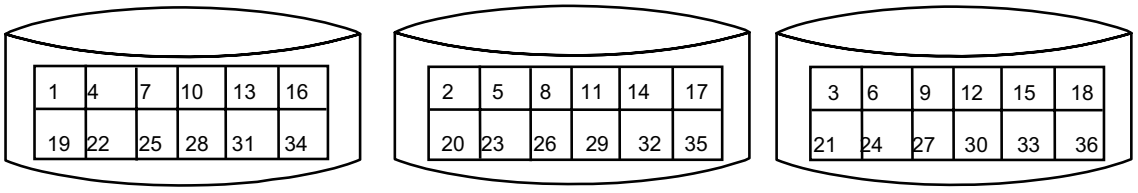
Argument	Meaning	Description
y or **a**	Yes or Active	Each write is logged to the MWC log. When the volume group is varied back online, the log is used to make logical partitions consistent. This is the default for mirrored logical volumes.
p	Passive	The volume group logs that the logical volume has 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.

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 stripe units among disks
- Stripe unit size is specified at creation time

Figure 9-24. Striping AU1412.0

Notes:

Overview

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.

Striping is designed to increase the read/write performance of frequently accessed, large sequential files. Striping can also be used to simply distribute data evenly across a set of disks so that random I/O can be scattered across many drives simultaneously.

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.

Stripe size

The size of the stripe unit is specified at creation time.

Prior to AIX 5L V5.3, the stripe size could range from 4 KB -128 KB in powers of 2 (4 KB, 8 KB, 16 KB, 32 KB, 64 KB, and 128 KB). AIX 5L V5.3 supports the following stripe sizes: 4 KB, 8 KB, 16 KB, 32 KB, 64 KB, 128 KB, 1 MB, 2 MB, 4 MB, 8 MB, 16 MB, 32 MB, 64 MB, and 128 MB.

Constraints

There are some constraints imposed by implementing striping:

- 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

Performance considerations

There are some considerations in configuring striping for performance:

- Use as many adapters as possible. For example, if multiple disks in the stripe width are on the same SCSI adapter, a read/write of a stripe is not able to read/write the stripe units in parallel.
- Design to avoid contention with other uses of the disks used by the striped logical volume.
- Create on a volume group dedicated to striped logical volumes.

It is not a good idea to mix striped and non-striped logical volumes in the same physical volume. Physical volumes should be the same size (ideally) within the set used for a striped logical volume.

Just because a logical volume is striped does not mean that the file's data blocks are going to be perfectly aligned with the stripe units. Therefore, if a file block crosses a stripe boundary, the block gets split up into multiple LVM I/Os.

Striped columns

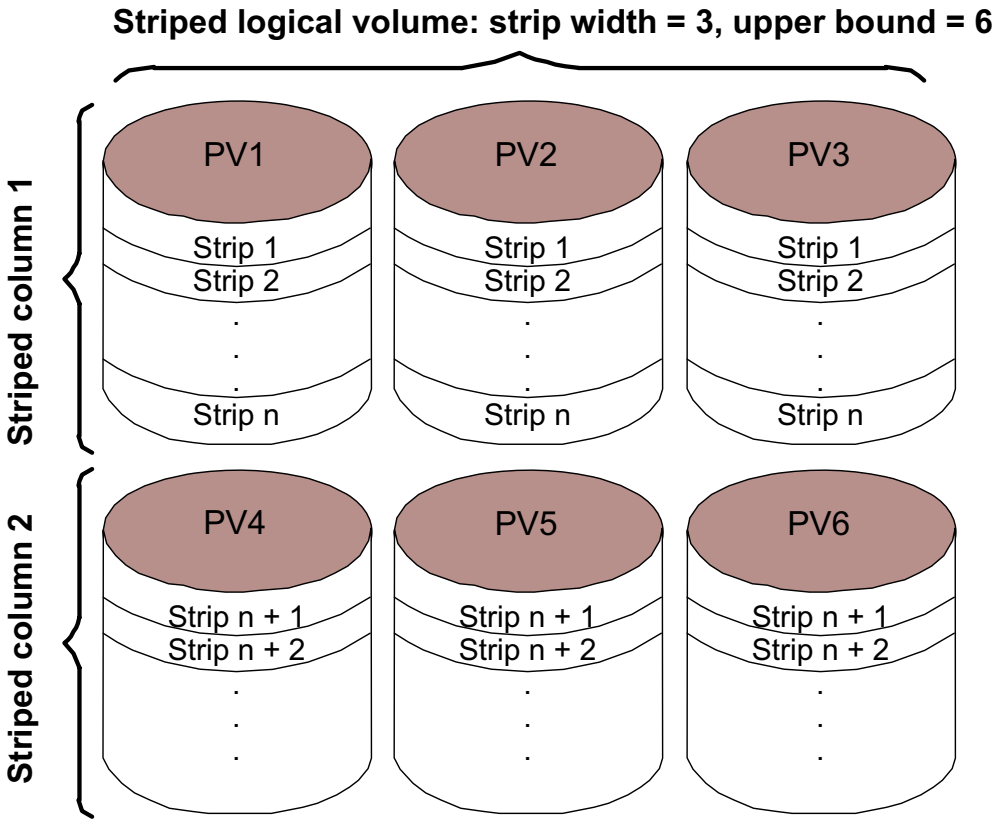


Figure 9-25. Striped columns AU1412.0

Notes:

Striped column support

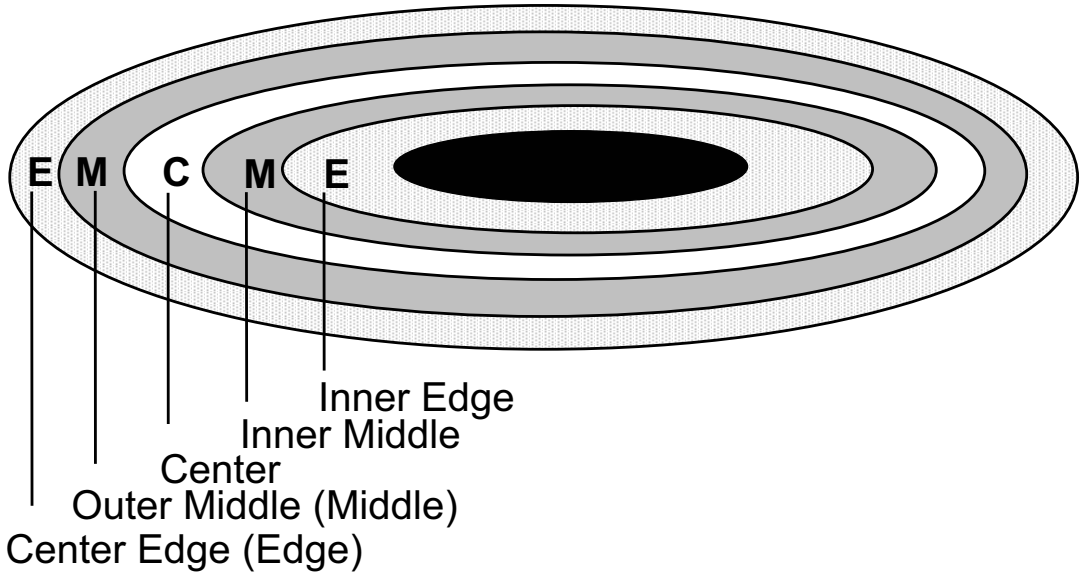
AIX 6.1 provides *striped columns* support for logical volumes. This new feature allows a striped logical volume to be extended even if one of the physical volumes in the disk array becomes full.

Prior to AIX 5L V5.3, if you had a striped logical volume that completely filled the capacity of the disks that formed its stripe width and you needed more room to grow the logical volume, it was not easy. The work-around required you to backup and delete the striped logical volume and then to re-create the logical volume with a larger stripe width followed by a restore operation of the logical volume data.

Prior to AIX 5L V5.3, you could not configure a striped logical volume with an upper bound larger than the stripe width. In AIX 5L V5.3 and later, the upper bound can be a multiple of the stripe width. One set of disks, as determined by the stripe width, can be considered as one *striped column*. There is no need to back up, redefine, and then restore the data. It is done dynamically.

Logical volume policies

Intra-physical volume allocation policy:



Inter-physical volume allocation policy:

- Maximum number of physical volumes to use
- Range of physical volumes to use

Figure 9-26. Logical volume policies AU1412.0

Notes:

Introduction

When creating or changing a logical volume you can define the way the Logical Volume Manager decides on which physical partitions to allocate to the logical volume. This affects the performance of the logical volume.

Intra-physical volume allocation policy

The intra-physical volume allocation policy indicates where on the physical volume partitions are 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 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.

Inter-physical volume allocation policy

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
- Maximum
  Allocate partitions across all physical volumes up to the maximum number of physical volumes

SMIT Logical Volumes menu

```
# smit lv
```

```
                        Logical Volumes

  Move cursor to desired item and press Enter.

    List All Logical Volumes by Volume Group
    Add a Logical Volume
    Set Characteristic of a Logical Volume
    Show Characteristics of a Logical Volume
    Remove a Logical Volume
    Copy a Logical Volume





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


Figure 9-27. SMIT Logical Volumes menu AU1412.0

Notes:

SMIT Logical Volumes screen

This is the top-level SMIT menu for logical volumes. The next few pages discuss these items.

Show logical volume characteristics

- Physical volume map:

```
# lslv -l lv00

  lv00:/home/john
  PV        COPIES          IN BAND      DISTRIBUTION
  hdisk0    010:000:000     70%          000:000:007:003:000
```

- Logical partition map:

```
# lslv -m lv00

  lv00:/home/john
  LP         PP1    PV1         PP2    PV2         PP3    PV3
  00010134     hdisk0
  0002         0135    hdisk0
  00030136     hdisk0
```


Figure 9-28. Show logical volume characteristics AU1412.0

Notes:

`lslv -l` *`lvname`*

The `lslv -l` *`lvname`* command 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

The `COPIES` field of the **`lslv -l`** ***`lvname`*** command has the following three fields:

- The number of logical partitions containing at least one physical partition (no copies) on the physical volume
- The number of logical partitions containing at least two physical partitions (one copy) on the physical volume
- The number of logical partitions containing three physical partitions (two copies) on the physical volume

The example in the visual, `COPIES (010:000:000)`, can be interpreted as:

- 010 - Information regarding first copy, that is, 10 physical partitions
- 000:000 - These two fields hold information regarding the second and the third copies, mirrored copies on the physical volume. By looking at this output, you can tell if the logical volume is mirrored and if the mirrored copies are on the same physical volume. If they are all on separate physical volumes, which is the default, the last two fields still show 000.

IN BAND

The `IN BAND` field displays the percentage of physical partitions on the physical volume that belong to the logical volume and were allocated within the physical volume region specified by Intra-physical allocation policy.

DISTRIBUTION

The **`DISTRIBUTION`** field displays the number of physical partitions allocated within each section of the physical volume: outer edge, outer middle, center, inner middle, and inner edge.

The example in the visual, `DISTRIBUTION (000:000:007:003:000)`, shows of the 10 physical partitions, 7 physical partitions are located in the center area and 3 physical partitions in the inner-middle area of the disk. There is a relationship between the distribution 000:000:007:003:000 and the copies 010:000:000 whereby the 007:003 numbers indicate the distribution of the 010.

`lslv -m` ***`lvname`***

The **`lslv -m`** ***`lvname`*** command 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                  outer_middle             +
    RANGE of physical volumes                    minimum                  +
    MAXIMUM NUMBER of PHYSICAL VOLUMES           []                        #
      to use for allocation
    Number of COPIES of each logical             1                        +
      partition
    Mirror Write Consistency?                    active                   +
    Allocate each logical partition copy         yes                      +
  [MORE...11]

  F1=Help       F2=Refresh       F3=Cancel        F4=List
  F5=Reset      F6=Command       F7=Edit          F8=Image
  F9=Shell      F0=Exit          Enter=Do
```


Figure 9-29. Add a Logical Volume AU1412.0

Notes:

The `mklv` command

The **`mklv`** command creates a logical volume. The name of the logical volume can be specified or alternatively a system-generated name is used. The volume group the logical volume belongs to and the size (in logical partitions) must be specified. Other characteristics that can be set are the allocation policy, copies (mirroring), scheduling policy and striping. Using **`mklv`** from the command line, you can now specify blocks (**`b`**,**`B`**), KB (**`k`**,**`K`**), MB (**`m`**,**`M`**) and GB (**`g`**,**`G`**) rather than number of partitions. Examples are:

```
# mklv -y newlv1 datavg 1
# mklv -y newlv2 datavg 1b
# mklv -y newlv3 datavg 1k
# mklv -y newlv4 datavg 1m
# mklv -y newlv5 datavg 1g
```

The system rounds to the physical partition size of the volume group.

Remove a Logical Volume

```
# smit rmlv
```

```
                    Remove a Logical Volume

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

[TOP]                                              [Entry Fields]
  LOGICAL VOLUME name                               []        +




F1=Help       F2=Refresh       F3=Cancel        F4=List
F5=Reset      F6=Command       F7=Edit          F8=Image
F9=Shell      F0=Exit          Enter=Do
```


Figure 9-30. Remove a Logical Volume AU1412.0

Notes:

The `rmlv` command

The **`rmlv`** command removes a logical volume. The **`-f`** option prevents the command from prompting for confirmation.

Do not use **`rmlv`** to remove journaled file systems or paging space volumes. These high-level structures have information relating to them saved in the ODM database and in files such as the **/etc/filesystems** file. This information is not removed by the **`rmlv`** command. You should use the appropriate command for that type of data structure.

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 9-31. Set Characteristics of a Logical Volume AU1412.0

Notes:

The `chlv` command

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.

Changing the size of the logical volume

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 now specify blocks, KB, MB and GB rather than number of partitions. 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.

List all logical volumes by volume group

```
# lsvg -o | lsvg -i -l

rootvg:
LVNAME     TYPE     LPs    PPs        PVs    LV STATE MOUNT POINT

hd6        paging   32     32         1      open/syncd       N/A
hd5        boot     2      2          1      closed/syncd     N/A
hd8        jfslog   1      1          1      open/syncd       N/A
hd9var     jfs2     2      2          1      open/syncd       /var
hd4        jfs2     9      9          1      open/syncd       /
hd2        jfs2     101    101        1      open/syncd       /usr
hd3        jfs2     4      1          1      open/syncd       /tmp
hd1        jfs2     1      1          1      open/syncd       /home
hd10opt    jfs2     5      5          1      open/syncd       /opt
hd11admin  jfs2     8      8          1      open/syncd       /admin
lv00       jfs2     1      2          2      open/syncd       /home/john
lv01       jfs2     4      4          2      open/syncd       /home/fred
```


Figure 9-32. List all logical volumes by volume group AU1412.0

Notes:

Listing logical volumes

From the **`smit lv`** fastpath, the **List all Logical Volumes by Volume Group** option 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.

Show logical volume characteristics

```
# lslv  lv02

 LOGICAL VOLUME:    lv02                  VOLUME GROUP:  course
 LV IDENTIFIER:     0000000000004c00000000e5cf75106f.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/ACTIVE
 EACH LP COPY ON A SEPARATE PV ?:  yes
 Serialize IO ?    NO
```


Figure 9-33. Show logical volume characteristics AU1412.0

Notes:

Specific logical volume characteristics

The following characteristics are specific to logical volumes:

- `WRITE VERIFY`

  Specifies whether to verify all writes to the logical volume with a follow-up read.

- `BB POLICY`

  Indicates whether the LVM should try to relocate a bad block if one is encountered.

- `UPPER BOUND`

  Specifies the maximum number of disks that this logical volume can span.

Add Copies to a Logical Volume

```
# 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         2            +
   copies
  PHYSICAL VOLUME names                         []           +
  POSITION on physical volume                   middle       +
  RANGE of physical volumes                     minimum      +
  MAXIMUM NUMBER of PHYSICAL VOLUMES            [32]          #
    to use for allocation
  Allocate each logical partition copy          yes          +
    on a SEPARATE physical volume?
  File containing ALLOCATION MAP                []
  SYNCHRONIZE the data in the new               no           +
    logical partition copies?


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


Figure 9-34. Add Copies to a Logical Volume AU1412.0

Notes:

Adding a copy of a logical volume

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 and the desired total number of copies. This only succeeds 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).

Once a logical volume has been created, striping cannot be imposed or removed.

Synchronizing a mirrored logical volume

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.

Removing a copy of a logical volume

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.

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 VOLUMES                      [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 9-35. Reorganize a Volume Group AU1412.0

Notes:

Reorganizing a volume group

If the intra-physical volume allocation policy (location on disk: center, middle, edge, inner edge, and inner middle) is changed after the logical volume is created, the physical partition does 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 syntax

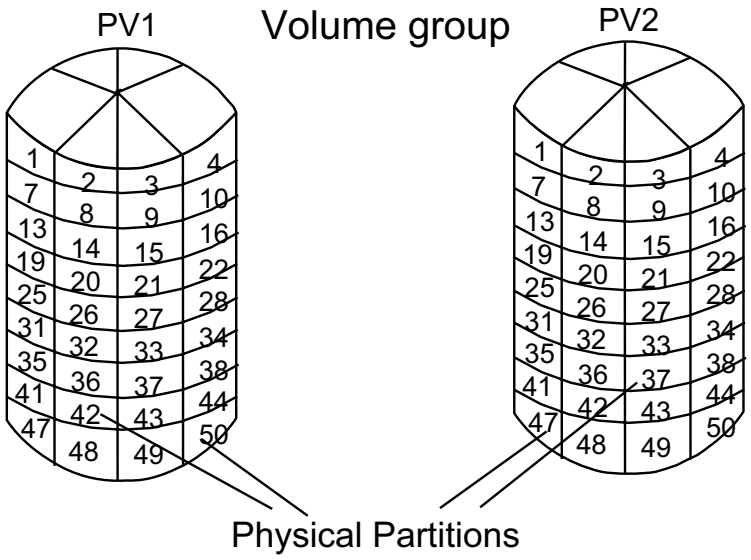
The syntax is: **reorgvg** ***volumegroup*** **[*lvname*]**

For example: **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.

9.3. Physical volumes

Physical volumes



- Physical volume (PV)
  - Hard disk
- Physical partition (PP)
  - Smallest assignable unit of allocation on a physical disk

Figure 9-36. Physical volumes AU1412.0

Notes:

Overview

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 volume group. However, there may be multiple volume groups on a single system, each having a different PP size.

The limitations for each type of volume group (normal, big, and scalable) such as the number of physical volumes and size of the physical partitions, was given in the last unit, *System Storage Overview*.

SMIT Physical Volumes menu

```
# smit pv
```

```
                    Physical Volumes

  Move cursor to desired item and press Enter.

    List All Physical Volumes in System
    Add a Disk
    Change Characteristics of a Physical Volume
    List Contents of a Physical Volume
    Move Contents of a Physical Volume




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


Figure 9-37. SMIT Physical Volumes menu AU1412.0

Notes:

SMIT Physical Volumes menu

This is the top-level menu for physical volume. Each of these items is discussed in the following pages.

List physical volume information

- List all physical volumes in system:

```
# lspv

hdisk0  da1c923411d52ec91cd600802eda72c9     rootvg    active
hdisk1  bebc80000000000000000802evg79c9      rootvg    active
```

- List the 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 megabytes)      VG DESCRIPTORS:   2
FREE PPs:          3 (12 megabytes)        HOT SPARE:        no
USED PPs:          92 (368 megabytes)      MAX REQUEST       256 KB
FREE DISTRIBUTION: 00..03..00..00..00
USED DISTRIBUTION: 19..16..19..19..19
```


Figure 9-38. List physical volume information AU1412.0

Notes:

Listing physical volume information

From the **`smit pv`** fastpath, the **List all Physical Volumes in System** option 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, physical volume 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 physical partition copies that are stale (are not up to date with other copies), total number of physical partitions, number of free physical partitions, and distribution of free space on the physical volume.

List logical volumes on a physical volume

```
# lspv -l hdisk0
hdisk0:
LV NAME    LPs    PPs    DISTRIBUTION          MOUNT POINT
hd1        1      1      00..00..00..12..00    /home
hd3        4      4      00..03..00..00..00    /tmp
hd2        101    101    00..00..17..12..00    /usr
hd4        9      9      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        2      2      01..00..00..00..00    N/A
hd9var     2      2      00..00..02..00..00    /var
hd10opt    5      5      00..00..02..00..00    /opt
hd11admin  8      8      00..00..02..00..00    /admin
```


Figure 9-39. List logical volumes on a physical volume AU1412.0

Notes:

Listing logical volumes

The **`lspv -l`** ***`pvname`*** command lists all the logical volumes on a physical volume including number of logical partitions, physical partitions and distributions on the disk.

List a physical volume partition map

```
# lspv -p hdisk0
hdisk0:
PP RANGE   STATE    REGION          LV NAME     TYPE      MOUNT POINT
1-2        used     outer edge      hd5         boot      N/A
3-154      free     outer edge
155-186    used     outer middle    hd6         paging    N/A
187-307    free     outer middle
308-308    used     center          hd8         jfslog    N/A
309-309    used     center          hd4         jfs2      /
310-313    used     center          hd2         jfs2      /usr
314-314    used     center          hd9var      jfs2      /var
315-317    used     center          hd3         jfs2      /tmp
318-318    used     center          hd1         jfs2      /home
319-319    used     center          hd10opt     jf22      /opt
320-360    used     center          hd2         jfs2      /usr
361-363    used     center          hd10opt     jfs2      /opt
364-364    used     center          hd3         jfs2      /tmp
365-372    used     center          hd4         jfs2      /
373-380    used     center          hd11admin   jfs2      /admin
381-423    used     center          hd2         jfs2      /usr
424-424    used     center          hd9var      jfs2      /var
425-425    used     center          hd10opt     jfs2      /opt
426-438    used     center          hd2         jfs2      /usr
439-460    free     center
461-613    free     inner middle
614-767    free     inner edge
```


Figure 9-40. List a physical volume partition map AU1412.0

Notes:

Showing partition map

The **`lspv -p`** ***`pvname`*** command 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, center, middle, edge, inner edge, and inner middle.

Add or move contents of physical volumes

- A disk can be either added:
  - Through SMIT
  - 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 9-41. Add or move contents of physical volumes AU1412.0

Notes:

Adding a physical device

To add a physical volume to the system using SMIT, the path is SMIT -> **Devices -> Add a Disk -> Fixed Disk**. 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 SMIT **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** is invoked, which picks 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 multiplied by 1016.

Preparation to remove a physical device

The **`migratepv`** command can be used to move all partitions (or partitions from a selected logical volume) 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.

Documenting the disk storage setup

- List of the disks on the system (PVID and volume group):

```
# lspv
```

- List the volume groups:

```
# lsvg
```

- List what logical volumes are contained in each volume group:

```
# lsvg -l vgname
```

- List the logical volumes on each disk:

```
# lspv -l pvname
```


Figure 9-42. Documenting the disk storage setup AU1412.0

Notes:

What to document

It's important to have your storage information readily available in case you have a problem with your system, or in the very worst case, a totally crashed system. The commands in the visual help you to get this information.

Checkpoint

1. True or False? A logical volume can span more than one physical volume.
2. True or False? A logical volume can span more than one volume group.
3. True or False? The contents of a physical volume can be divided between two volume groups.
4. True or False? If mirroring logical volumes, 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 9-43. Checkpoint AU1412.0

Notes:

Exercise 9: Working with LVM (parts 3-5)

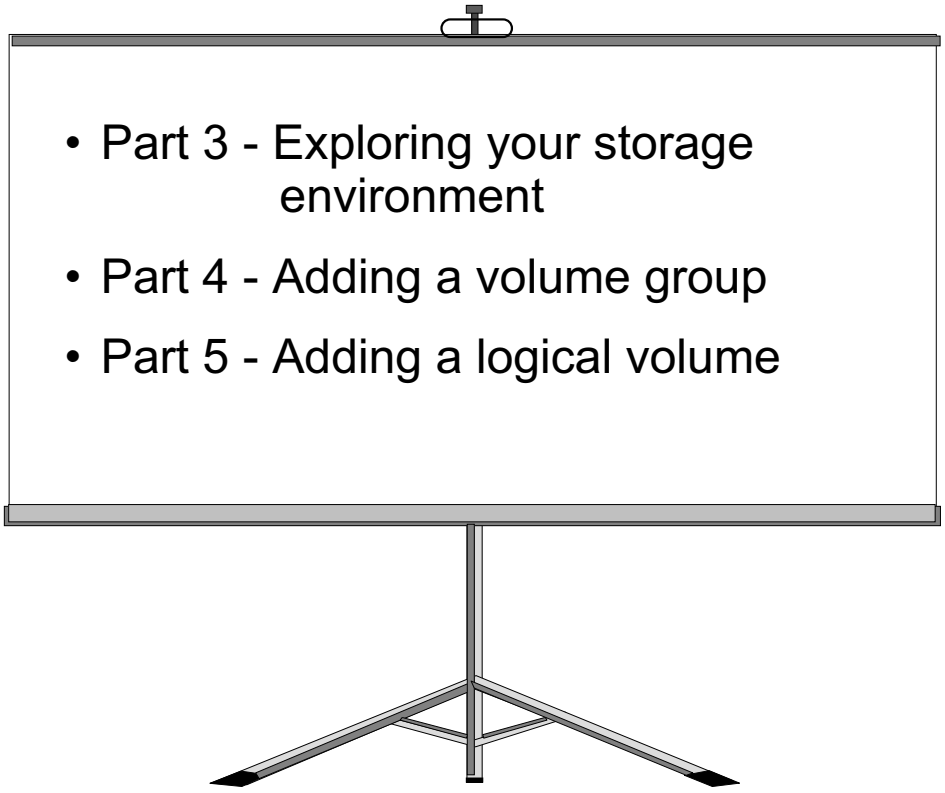


Figure 9-44. Exercise 10: Working with LVM (parts 3-5) AU1412.0

Notes:

Introduction

This lab has you set up a new volume group and a new logical volume. You use this volume group and logical volumes in future exercises.

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

Unit summary

- SMIT or high-level commands can be used to add, change, or 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.

Figure 9-45. Unit summary AU1412.0

Notes:

Unit 10. Working with file systems

What this unit is about

This unit covers important concepts and procedures related to 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 system
- Add an enhanced journaled file system
- Change characteristics of a file system
- Add a RAM file system
- Add a UDF file system on a DVD-RAM

How you will check your progress

Accountability:

- Checkpoint questions
- Exercise
- Activity

References

GG24-4484	*AIX Storage Management*
SG24-7559	*AIX Version 6.1 Differences Guide*
Online	*AIX 6.1 Operating system and device management*
Online	AIX 6.1 File Reference

Note: References listed as "Online" above are available at the following address:

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

Unit objectives

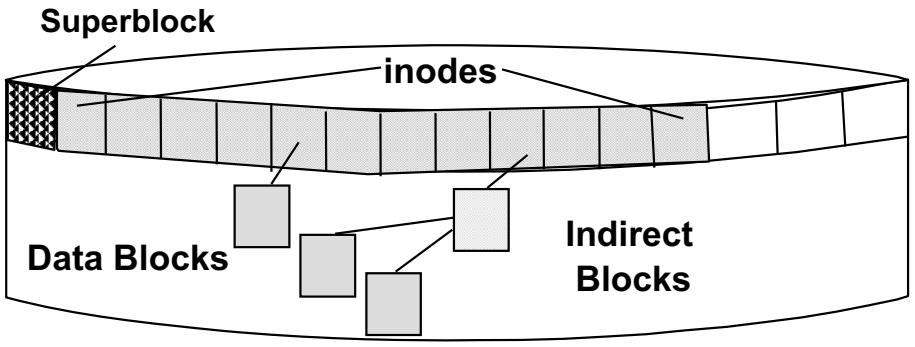
After completing this unit, you should be able to:

- Identify the components of an AIX file system
- Add an enhanced journaled file system
- Change characteristics of a file system
- Add a RAM file system
- Add a UDF file system on a DVD-RAM

Figure 10-1. Unit objectives AU1412.0

Notes:

Structure of a journaled 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 10-2. Structure of a journaled file system AU1412.0

Notes:

File systems and logical volumes

AIX journaled file systems are built within logical volumes. Because journaled file systems exist within logical volumes, the size of the file system is always a multiple of the logical partition size for that logical volume (for example, 4 MB).

Unit of allocation

An individual file within a file system, by default, has units allocated to it in blocks of 4096 bytes. (This may change if you have implemented fragmentation or large files - both of which are 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.

Superblock

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, and date/time of creation.

The superblock is critical to the file system and, if corrupted, prevents the file system from mounting. For this reason a backup copy of the superblock is always written in block 31.

Inodes

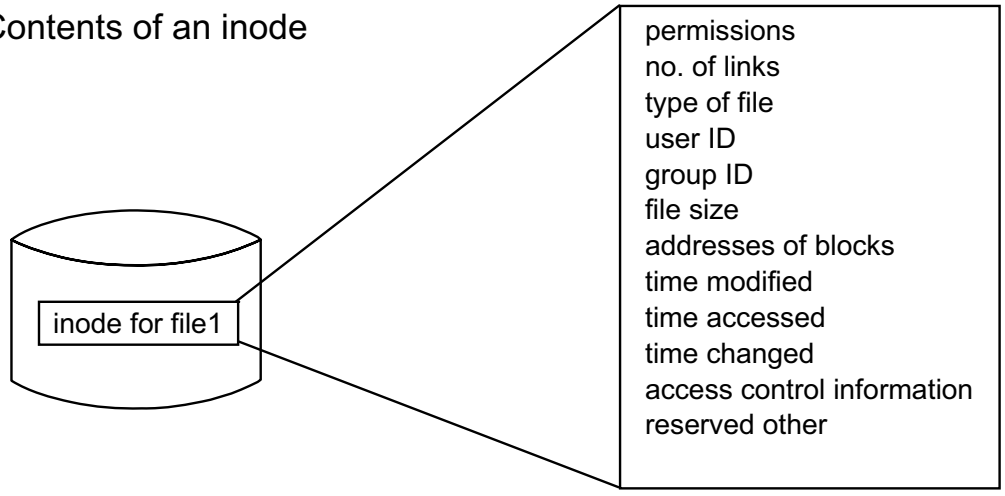
Immediately following the superblock are inodes which contain identifying information for files, such as the file type, size, permissions, user/group/owner, and create/modification and last access dates. They also contain pointers to the data blocks for fragment addresses which hold the data.

Indirect blocks

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 10-3. Structure of an inode AU1412.0

Notes:

Function and contents of an inode

Each file is represented by a single inode.

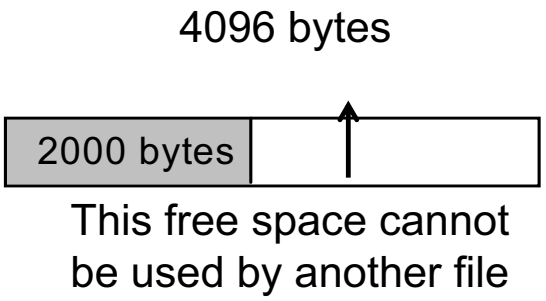
The inode contains information about that file such as the following:

- 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

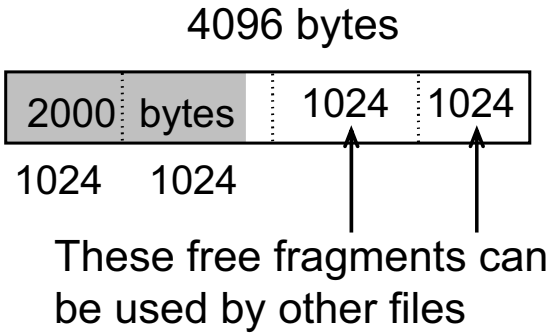


Figure 10-4. File system fragmentation AU1412.0

Notes:

Use of fragmentation

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. JFS fragment support provides a view of the file system as a contiguous series of fragments rather than logical disk blocks.

Fragment support applies to the last direct block of small user files and directories and long symbolic links.

Fragment size

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.

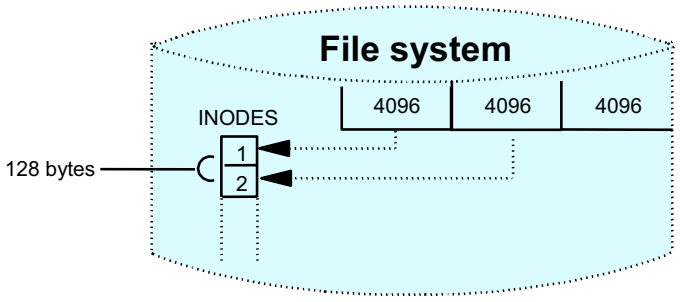
Balancing competing goals

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 is created for every 4096 bytes of file system.



Using the value nbpi = 1024 an inode is created for every 1024 bytes of file system.

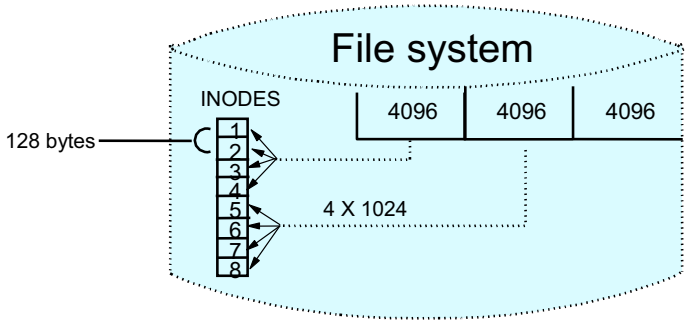


Figure 10-5. Variable number of inodes AU1412.0

Notes:

Use of inodes

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 to describe 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.

Fixed number of inodes

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.

Specifying the number of inodes

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.

Decisions regarding 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.

Dynamic allocation of inodes by JFS2

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

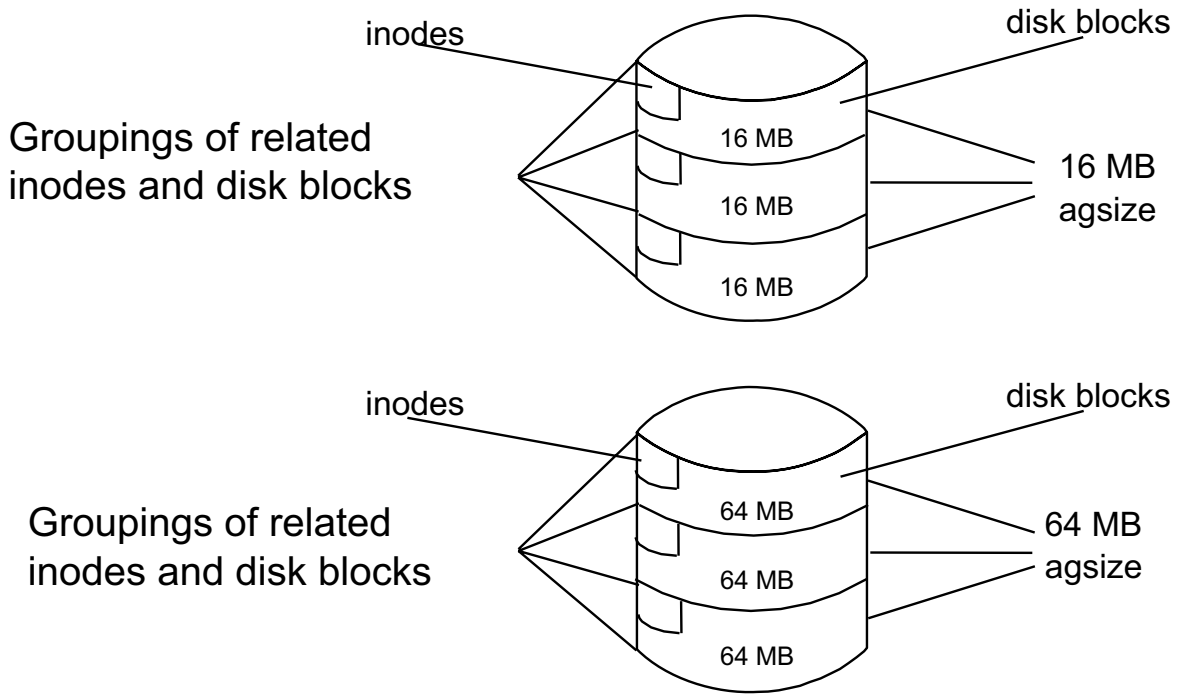


Figure 10-6. Allocation group size AU1412.0

Notes:

Importance of allocation groups

The ability to specify *allocation group size* is supported by AIX V4.2 and later AIX versions. This capability is used to increase the efficiency of the file system. The inodes are further grouped with the corresponding data blocks 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.

Specifying allocation group size

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.

Relationship between allowable NBPI values and allocation group size

The allowable NBPI values are dependent on the allocation group size (agsize). For example, for an agsize value of 8 MB, 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.

Refer to the table below 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

Compressed file systems

compression = LZ (yes)
fragment size = 1024

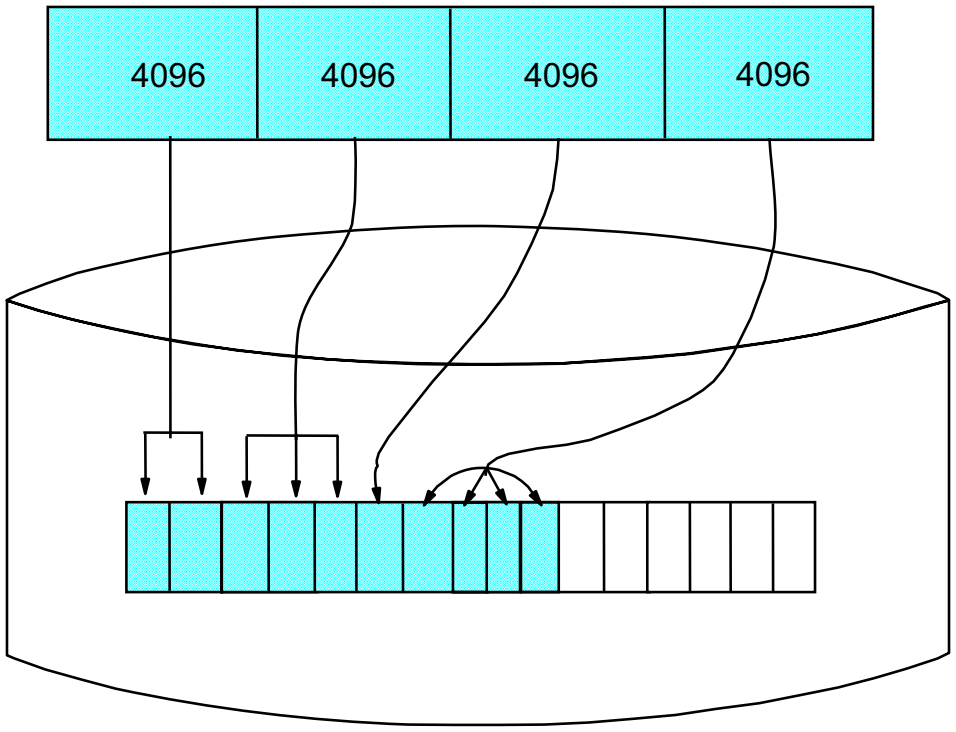


Figure 10-7. Compressed file systems AU1412.0

Notes:

Use of fragmented and compressed file systems

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 blocks 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.

Fragmentation concerns

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.

The `defragfs` utility

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.

Special considerations regarding the / and /usr file systems

Warning: The **root** (/) file system *must* not be compressed.

Compression of the /**usr** file system is *not recommended*.

Additional considerations

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, are allocated 4096 bytes of disk space, and this space is subsequently 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 are required 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

(1024 * 4 KB blocks) + (1024 * 128 KB blocks) = 132 MB
4 MB + 128 MB = 132 MB

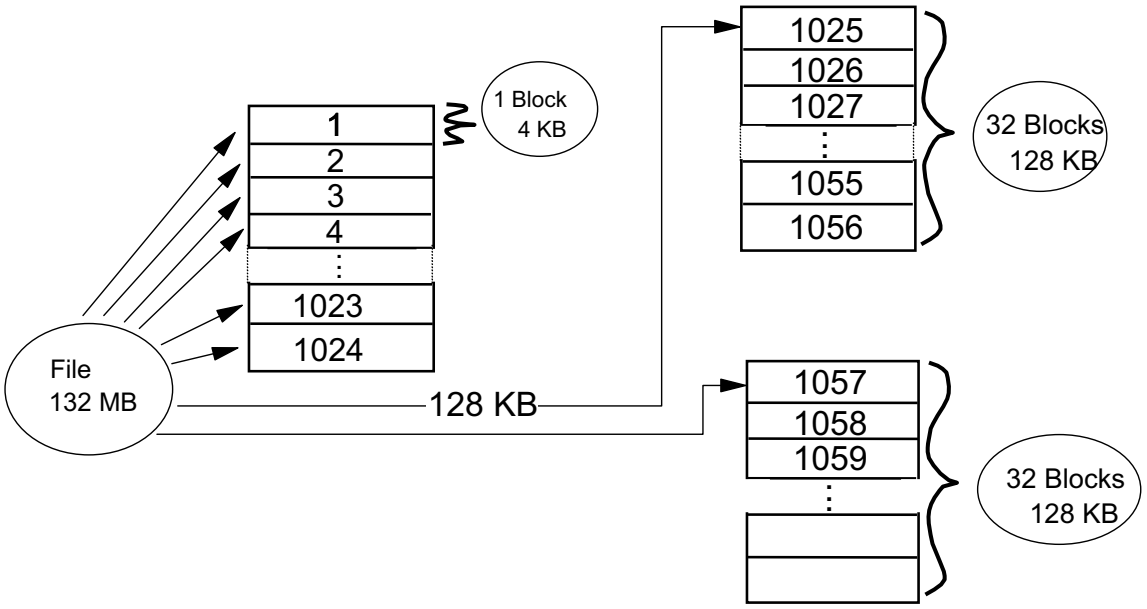


Figure 10-8. Large file enabled file systems AU1412.0

Notes:

Support of file sizes greater than 2 GB

On AIX V4.2 and later versions of AIX, JFS supports large file enabled file systems. Only file systems enabled for large files can support files with a size greater than 2 GB.

Use of 128 KB blocks

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.

Fewer indirect blocks required with large file geometry

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.

JFS2 support of large files

It is not necessary to use large enabled file systems in JFS2, since large file and file system support is built in by default.

Exercise 10: Working with file systems (part 1)

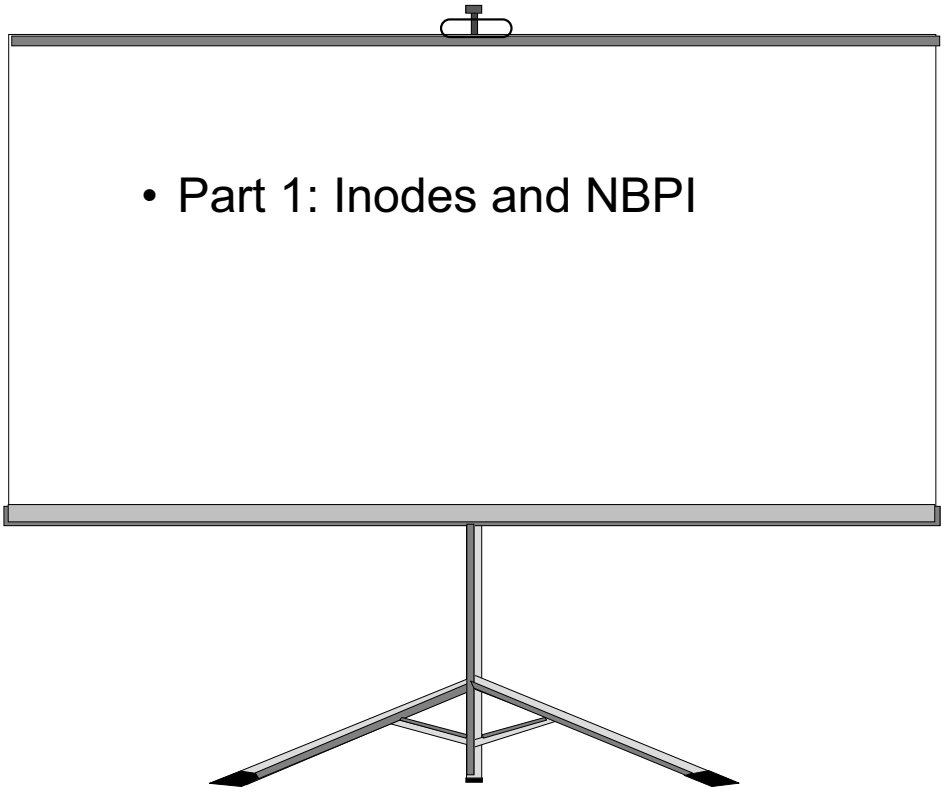


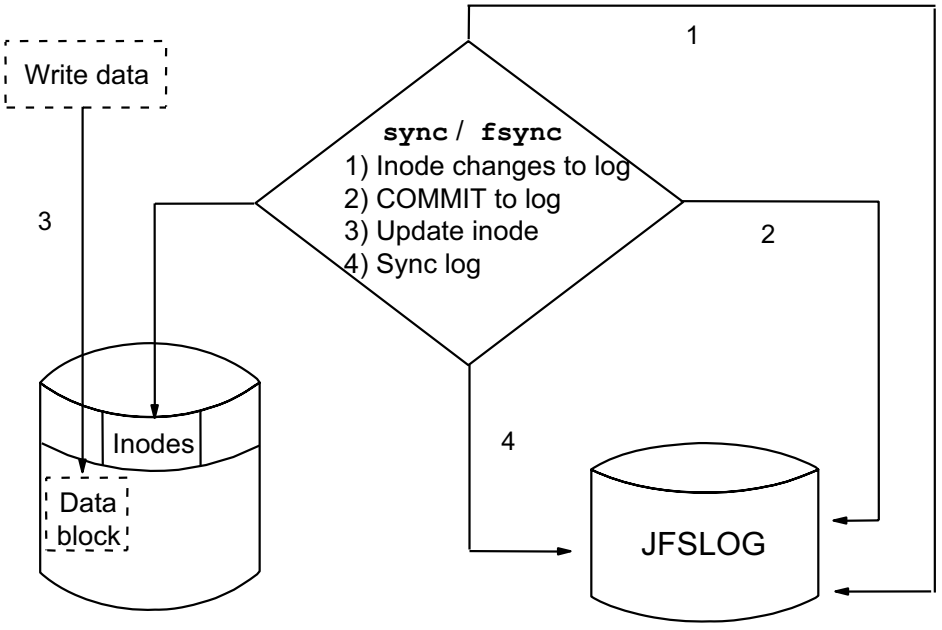
Figure 10-9. Exercise 10: Working with file systems (part 1) AU1412.0

Introduction

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

Only do *Part 1 - Inodes and NBPI*

Journal log



- No journaling of data blocks - only journals inode information (and indirect block information).

Figure 10-10. Journal log AU1412.0

Notes:

Writes to files done first in memory

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

The *jfslog* for each volume group (such as the **rootvg** /**dev**/**hd8**) is a circular log. A jfslog 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 file system structure itself, such as changes to the inodes and the free list.

Indication of sync points

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.

Journal log logical volumes

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 is 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.

Inline logs

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 (in AIX 5L V5.1).

The following table lists the default inline log size in AIX 5L V5.2 and later.

LVsize	Inline log size
<32 MB	256 KB
> 32 MB up to 64 MB	512 KB
>64 MB up to 128 MB	1 MB
128 MB	2 MB
128 MB to 1 GB	1/128th of size
1 GB to 2 GB	8 MB
2 GB to 128 GB	1/256th of size
128 GB up to 512 GB	512 MB
512 GB	1/1024th of size

Available logging options

The following table lists the three logging options and indicates which file system types support these options.

Option	JFS	JFS2
Default volume group log	Yes	Yes
Specific user-created log	Yes	Yes
Log directly to the file system	No	Yes

JFS versus JFS2 file systems

	JFS	JFS2
Maximum File Size Architectural / Tested	64 Gigabytes / 64 Gigabytes	1 Petabyte / 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
Directory File Access	Sequential	B-tree
Journal Log support	External JFSlog only	Inline or External JFS2log
Compression	Yes	No
Quotas	Yes	AIX 5L V5.3 and later

JFS2 uses extent based allocation for high performance and large file size.

Figure 10-11. JFS versus JFS2 file systems AU1412.0

Notes:

Introduction to JFS2

Enhanced **J**ournaled **F**ile **S**ystem (JFS2) is a new file system type that was introduced in AIX 5L V5.1. It is based on JFS.

Reference information

The following reference information may be useful when you are reviewing the table on the visual:

1 Petabyte (PB) = 1024 Terabytes (TB) = (2^{50}) bytes

1 Terabyte (TB) = 1024 Gigabytes (GB) = (2^{40}) bytes

1 Gigabyte (GB) = 1024 Megabytes (MB) = (2^{30}) bytes

1 Megabyte (MB) = 1024 Kilobytes (KB) = (2^{20}) bytes

1 Kilobyte (KB) = 1024 Bytes = (2^{10}) bytes

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.

Directory file b-tree

In JFS, the directory files are accessed sequentially. For large directory files, this is inefficient. In JFS2, the directories files are accessed via a b-tree index. For very large directories, applications doing large numbers of adds and deletes to a JFS2 directory can see as much as a 40 fold improvement in performance.

Inline journal logs

Normally, multiple file systems use the same journal log. The associated contention can impact performance. Creating a separate journal log for each file system takes special planning and requires an excessive amount of disk storage. JFS2 allows the definition of inline logs where each file system has its own log allocated out of the file system's logical volume. The space used by the inline log can be as small as 256 KB (for a file system < 32 MB). For details, see the notes on the visual covering the role of a journal log.

JFS2 disk quota system

Prior to AIX 5L V5.3, JFS2 did not support a disk quota system, though the Berkeley Disk Quota System was supported under JFS.

JFS2 quotas may be set for individual users or groups on a per file system basis. The quota system issues a warning to the user when a particular quota is exceeded, but allows some extra space for current work. Remaining over quota beyond a specified grace period results in further allocation attempts being denied until the total usage is reduced below the user's or group's quota.

The administration is similar to administration of the BSD Disk Quota System (refer to *http://www.openbsd.org* for details) except that AIX added a new method for mapping the users to the quotas. The quotas are assigned to a *Limits class* and then the users are assigned to the class. This greatly simplifies the quota administration. AIX 6.1 has a command to administer *Limits classes* - **`j2edlimit`**.

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 are necessary:

- Backup JFS file system data
- Create new JFS2 file systems
- Restore JFS file system data to new JFS2 file systems

Extended attributes (EA)

- Extensions to regular attributes
- Two versions
  - AIX 5L V5.2 or earlier supported only EAv1
  - EAv1 used for local file permission ACLs
  - EAv2 improved (more and larger attributes)
  - JFS2 under AIX 5L V5.3 and later support both versions
- NFS V4 ACLs stored in JFS2 with EAv2
- User defined information may be in EAv2

```
$ getea  HenryVIII
EAName:  Author
EAValue: Shakespeare
```


Figure 10-12. Extended attributes (EA) AU1412.0

Notes:

What are extended attributes?

Extended attributes are an extension of the normal attributes of a file (such as size and mode). They are (name, value) pairs associated with a file or directory. The name of an attribute is a null-terminated string. The value is arbitrary data of any length.

Types of extended attributes

There are two types of extended attribute: extended attribute version 1 (EAv1) and extended attribute version 2 (EAv2). For many years, AIX has supported extended attributes for Access Control Lists (ACL), which provide for more granular control of file access. That support was in EAv1 format. Starting with AIX 5L V5.3, EAv2 with JFS2 is now available.

EAv1 had restrictions of only eight attributes, 4 KB per attribute, 16-bit encoded names and no support for user defined attributes. EAv2 effectively eliminates these restrictions.

The primary use for EAv2, currently, is the support for the NFS V4 ACL capability. The discussion of NFS V4 ACLs is outside the scope of this class.

Managing attributes

AIX 6.1 provides line commands to manage the user defined attributes. To set an attribute value, you would use the **`setea`** command. To view a user attribute, you would use the **`getea`** command.

EAv2 compatibility issues

The major concern for the system administrator regarding EAv2 is the lack of backwards compatibility with earlier versions of AIX. AIX 6.1 continues to support EAv1 as the default format, and provides an option to create a file system with EAv2 and a runtime command to convert dynamically from EAv1 to EAv2 to create or access named attributes and advanced ACL. However, once a file system is created with EAv2 or conversion has been initiated, AIX 5L V5.2 cannot access the file system and attempting to mount results in an EFORMAT error.

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
Create and backup a snapshot
```


Figure 10-13. File Systems AU1412.0

Notes:

Using SMIT to manage file systems

As shown on the visual, SMIT can be used to complete numerous file system management tasks.

Using the Web-based System Manager to manage file systems

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        __         /          jfs2    294912   __             yes
/dev/hd1        __         /home      jfs2    32768    __             yes
/dev/hd2        __         /usr       jfs2    3309568  __             yes
/dev/hd9var     __         /var       jfs2    65536    __             yes
/dev/hd3        __         /tmp       jfs2    131072   __             yes
/proc           __         /proc      procfs  ________ ro             yes
/dev/hd10opt    __         /opt       jfs2    163840   __             yes
/dev/hd11admin  __         /admin     jfs2    262144   __             yes
/budget         sys4       /reports   nfs2    __       bg,hard,intr
/dev/cd0        __         /cdrom     cdrfs   __       ro             no
```


Figure 10-14. Listing file systems AU1412.0

Notes:

Function of `lsfs` command

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

The **`lsfs`** command also displays information about CD-ROM file systems and remote NFS file systems.

Syntax and options

The general syntax for the **`lsfs`** command is as follows:

`lsfs [-q] [-c | -l ][ -v` ***`vfstype`*** **`| -u`** ***`mountgrp`*** **`|`** ***`filesystem`*** **`]`**

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.

SMIT fastpath

The SMIT fastpath to get to the screen which accomplishes the same task as the **lsfs** command is **smit fs**. This takes you to the **File Systems** SMIT menu. Select the **List All File Systems** menu item. This selection does not have a dialog panel; it just runs the **lsfs** command.

List all mounted file systems

```
# mount

node   mounted        mounted over    vfs      date           options
       /dev/hd4       /               jfs2     Jul 11 20:14   rw,log=/dev/hd8
       /dev/hd2       /usr            jfs2     Jul 11 20:15   rw,log=/dev/hd8
       /dev/hd9var    /var            jfs2     Jul 11 20:15   rw,log=/dev/hd8
       /dev/hd3       /tmp            jfs2     Jul 11 20:15   rw,log=/dev/hd8
       /dev/hd1       /home           jfs2     Jul 11 20:16   rw,log=/dev/loglv00
       /proc          /proc           procfs   Jul 11 20:16   rw
       /dev/hd10opt   /opt            jfs2     Jul 11 20:16   rw,log=/dev/hd8
       /dev/hd11admin /admin          jfs2    Jul 11 20:16   rw,log=/dev/hd8
sys4   /budget        /reports        nfs      Jul 11 20:16   rw,hard,bg,intr
       /dev/ramdisk   /ramdisk        jfs      Jul 11 20:17   rw,nointegrity
       /dev/project   /project        jfs2     Jul 11 20:18   rw,log=INLINE
       /dev/cd0       /cdrom          cdrfs    Jul 11 20:19   ro
```


Figure 10-15. List all mounted file systems AU1412.0

Notes:

Listing currently mounted file systems

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.

Performing the mount operation

The **mount** command, when used with a number of parameters, is also used to perform the mount operation.

System-created and user-created file systems

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 devices and file systems

Standard device and file system names include the following:

- **hd4** /
- **hd1** **/home**
- **hd2** **/usr**
- **hd3** **/tmp**
- **hd9var** **/var**
- **proc** **/proc**
- **hd10opt** **/opt**
- **hd11admin** **/admin**

Using SMIT to list mounted file systems

SMIT can also be used to obtain this information. From SMIT, you want to select **List all Mounted File Systems** under **File Systems.**

Add/Change/Show/Delete File Systems

```
# smit manfs
```

```
     Add / Change / Show / Delete File Systems

Move cursor to desired item and press Enter

Enhanced Journaled File Systems
Journaled File Systems
CDROM File Systems
Network File Systems (NFS)
```


Figure 10-16. Add/Change/Show/Delete File Systems AU1412.0

Notes:

Selecting the file system type

In AIX 6.1 and AIX 5L, when you use the command **`smit manfs`**, SMIT presents a menu which prompts you for the *type* of file system, be it JFS, Enhanced JFS, CDROM file system, or NFS.

Managing JFS file systems

You can use the fastpath **`smit jfs`** if you wish to work with a JFS file system.

Managing JFS2 file systems

You can use the fastpath **`smit jfs2`** if you wish to work with a JFS2 file system.

Working with journaled file systems in SMIT

```
                        Journaled File Systems
Move cursor to desired item and press Enter.

Add a Journaled File System
Add a Journaled File System on a Previously Defined Logical Volume
Change / Show Characteristics of a Journaled File System
Remove a Journaled File System
Defragment a Journaled File System
```

```
Add a Journaled File System on a Previously Defined Logical Volume
Move cursor to desired item and press Enter.

Add a Standard Journaled File System
Add a Compressed Journaled File System
Add a Large File Enabled Journaled File System
```


Figure 10-17. Working with journaled files systems in SMIT AU1412.0

Notes:

Managing JFS file systems

The visual shows the SMIT menu displayed when the **`smit jfs`** fastpath is used.

Two ways of adding a JFS file system

When choosing to add a JFS file system, there are two options:

- If you choose to **Add a Journaled File System,** SMIT uses defaults to create the logical volume in which the file system sits.
- If you choose to **Add a Journaled 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 is the size of the logical volume.

In AIX V4.2 and later, the second SMIT menu shown on the visual is displayed no matter which option is chosen for adding a JFS file system.

Add a standard journaled file system on a previously defined logical volume

```
                 Add a Standard 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                                      [ ]             +
   Start Disk Accounting ?                             no             +
   Fragment Size (bytes)                               4096           +
   Number of bytes per inode                           4096           +
   Allocation Group Size (MBytes)                      8              +
   Logical Volume for Log                             [ ]             +


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


Figure 10-18. Add a standard journaled file system on a previously defined logical volume AU1412.0

Notes:

Using a logical volume

When a logical volume is created it is simply an empty container waiting to be formatted for use. The journaled 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 resides on disk.

SMIT fastpath for reaching this screen

The SMIT fastpath for this screen is **`smit crjfslvstd`**.

Specifying which logical volume to use for the JFS log

AIX 6.1 has a line on this panel: **Logical Volume for Log.** Prior to AIX 5L V5.3, you needed to edit /**etc**/**filesystems** after creating the file if you wanted to use anything other than the default /**dev**/**hd8** logical volume for the log. With AIX 5L V5.3 and later, you can identify what log to use in the initial definition. Note that the jfslog itself has to be previously defined and formatted.

Add a Standard Journaled File System

```
                    Add a Standard 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
        Unit Size                                       Megabytes     +
*       Number of units                                 [ ]            #
*  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             +
   Logical Volume for Log                               [ ]           +


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


Figure 10-19. Add a Standard Journaled File System AU1412.0

Notes:

SMIT fastpath for reaching this screen

Use the SMIT fastpath **`smit crjfsstd`** to access this screen.

The `crfs` command

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. The **`crfs`** command does a lot more; it creates the logical volume if necessary using **`mklv`**, builds the file system structure on that logical

volume using **mkfs**, and then makes all appropriate changes to the ODM and /**etc**/**filesystems** for that logical volume and file system.

JFS file system parameters

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***), that is, 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 unit size (**Megabytes | Gigabytes | 512bytes**) specifies the selected unit.
- The size (**-a size=*number of units***), that is, the number of units of unit size. The size of the file system is rounded up to the nearest logical partition boundary.
- The mount point (**-m** ***mntpt***). The name of a directory within the overall file system on which the new file system is normally 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 places the `mount=true` line in the /**etc**/**filesystems** file and causes 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 is used for the file system if it is mounted automatically, or if it is mounted without providing overriding permissions.

  The permissions setting for a mounted file system may not be bypassed, 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, the 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.

Working with enhanced journaled file systems (JFS2) in SMIT

```
                    Enhanced Journaled File Systems

Move cursor to desired item and press Enter.

Add an Enhanced Journaled File System
Add an Enhanced Journaled File System on a Previously Defined Logical Volume
Change / Show Characteristics of an Enhanced Journaled File System
Remove an Enhanced Journaled File System
Manage Quotas for an Enhanced Journaled File System
Defragment an Enhanced Journaled File System
List Snapshots for an Enhanced Journaled File System
Create Snapshot for an Enhanced Journaled File System
Mount Snapshot for an Enhanced Journaled File System
Remove Snapshot for an Enhanced Journaled File System
Unmount Snapshot for an Enhanced Journaled File System
Change Snapshot for an Enhanced Journaled File System
Rollback an Enhanced Journaled File System to a Snapshot



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


Figure 10-20. Working with enhanced journaled file systems (JFS2) in SMIT AU1412.0

Notes:

Managing JFS2 file systems

The visual shows the SMIT menu displayed if the **`smit jfs2`** fastpath is used.

Two ways of adding a JFS file system

When choosing to add a JFS2 file system, there are two options:

- If you choose to **Add an Enhanced Journaled File System**, SMIT uses defaults to create the logical volume in which the file system resides.
- If you choose to **Add an Enhanced Journaled 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 is the size of the logical volume.

Add an enhanced journaled file system (JFS2) 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          +
   Logical Volume for Log                            [ ]            +
   Inline Log size (MBytes)                          [ ]             #
   Extended Attribute Format                          Version 1     +
   Enable Quota Management                            no            +
   Enable EFS?                                        no           +
   Allow internal snapshots?                          no            +

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


Figure 10-21. Add an enhanced journaled file system (JFS2) on a previously defined logical volume AU1412.0

Notes:

SMIT fastpath for reaching this screen

The SMIT fastpath for reaching this screen is **`crjfs2lvstd`**.

The `Block Size` parameter

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.

The `Logical Volume for Log` parameter

Logical Volume for Log provides a choice between using either an existing jfs2log logical volume (the first jfs2log for this volume group is the default) or an inline log. If you use the inline log, then you have the option to override the default log size.

AIX Enhancements

With AIX 5L V5.3 and later, there are two attributes on this panel:

- **Extended Attribute Format** allows you to choose between the default EAv1 or EAv2.
- **Enable Quota Management** does what it says for this particular file system. You should be sure to plan and build the user quota definitions before enabling disk quotas for a file system.

With AIX 6.1, there are two attributes on this panel:

- **Encrypted File System (EFS)** specifies whether the file system is an Encrypted File System. This is only available for JFS2.
- **Isnapshot** specifies whether the file system supports internal snapshots.

Add an Enhanced Journaled File System (JFS2)

```
                    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
        Unit Size                                Megabytes        +
*       Number of units                          [ ]               #
*  MOUNT POINT                                   [ ]
   Mount AUTOMATICALLY at system restart?         no              +
   PERMISSIONS                                    read/write      +
   Mount OPTIONS                                 [ ]              +
   Block size (bytes)                             4096            +
   Logical Volume for Log                        [ ]              +
   Inline Log size (MBytes)                      [ ]               #
   Extended Attribute Format                      Version 1       +
   Enable Quota Management                        no              +
  [MORE...2]
```


Figure 10-22. Add an Enhanced Journaled File System (JFS2) AU1412.0

Notes:

SMIT fastpath for reaching this screen

Use the SMIT fastpath **`smit crjfs2std`** to access this panel.

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        no          +
  in this file system?

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


Figure 10-23. Mount a File System AU1412.0

Notes:

Mounting a file system makes it accessible

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.

Automatic mounting of file systems

File systems defined with the `mount=true` or `mount=automatic` attribute in the **/etc/filesystems** file are mounted automatically during system startup.

Use of the `mount` and `unmount` commands

The **mount** command is used to mount a file system, that is, make it available for use.

We illustrate basic use of the **mount** command here, but the full syntax for this command is quite complex, and there are numerous options that can be used with the command. Refer to the entry for **mount** in the *AIX 6.1 Commands Reference* online (or the corresponding **man** page) for the full syntax of this command and a detailed discussion of its various options.

The following simple examples illustrate basic use of the **mount** command:

- Example 1:

  ```
  # mount /home/george/myfs
  ```

  In this example, the only parameter specified is the name of a directory to be used as a mount point. Successful use of this simplified syntax requires predefinition of the file system in /**etc**/**filesystems**. In this case, the **mount** command checks /**etc**/**filesystems** to determine the device containing the associated file system and mounts the file system at the specified mount point.

- Example 2:

  ```
  # mount /dev/lv03 /myfs3
  ```

  In this case, the directory to be used as a mount point and the device containing the file system are both specified. The **mount** command mounts the specified file system at the specified mount point. Note that the device containing the file system is specified before the mount point.

The **unmount** command is used to unmount a previously mounted file system. (Another name for the **unmount** command is **umount**. Either name can be used.)

The full syntax for the **unmount** command (like that of the **mount** command) is quite complex. Refer to the entry for **umount** in the *AIX 6.1 Commands Reference* online (or the corresponding **man** page) for the full syntax of this command and a detailed discussion of its various options.

The following simple example illustrates basic use of the **unmount** command:

```
# unmount /myfs3
```

In this case, the file system previously mounted at the mount point /**myfs3** is unmounted.

Mounting and unmounting of file systems

File systems are usually mounted at startup and are unmounted as part of the **shutdown** procedure.

However, the **root** user may issue **mount** commands at any time. The **root** user is not affected by permissions on the mount point or root directory of the file system to be

mounted. (A normal user belonging to the **system** group can also mount file systems at any time, provided the user has write permission to the mount point and read permission on the root directory of the file system to be mounted.)

Options of the `mount` and `unmount` commands

The `mount` command has many options which may be specified by the user. Default values for these options are set by the system or indicated in /**etc**/**filesystems**.

The `unmount` command also has many options.

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)
        Unit Size                               512bytes       +
*       Number of units                         [65536]         #
   Mount GROUP                                 [bootfs]
   Mount AUTOMATICALLY at system restart ?      yes            +
   PERMISSIONS                                  read/write     +
   MOUNT OPTIONS                               [ ]            +
   Start Disk Accounting ?                      no             +
   Fragment Size (bytes)                        4096
   Number of bytes per inode                    4096
   Compression algorithm                        no
   Large File Enabled                           true
   Allocation Group Size (MBytes)               16
```


Figure 10-24. Change/Show Characteristics of a Journaled File System AU1412.0

Notes:

Changing JFS file system characteristics

A JFS 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 or SMIT. Many characteristics may be changed. The most important of these are described below.

Changing the mount point

The mount point can be changed using the **-m** flag of **chfs** (**-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.

Changing file system size

The file system size can be specified in Megabytes, Gigabytes or 512-byte blocks. The size of a Journaled File System may be *increased* while it is in use (**-a size=*number of units***). The size of a JFS 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 is rounded up to the next logical partition boundary. If you extend the logical volume directly, the partitions are added, but the file system is not changed. Extending the file system uses those added partitions.

Changing the mount group

The mount group of a file system may be changed (**-u *mntgrp***), and the change is effective the next time the new mount group is referenced.

Mount automatically a system restart?

The administrator can specify whether a file system is automatically mounted at system startup (**-A yes|no**). If a specification change is made, it is effective at the next startup.

Changing permissions

The permissions associated with the file system may be changed (**-p ro|rw**), and the change is 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
        Unit Size                                512bytes          +
*       Number of units                         [32768]             #
   Mount GROUP                                  [ ]
   Mount AUTOMATICALLY at system restart ?       yes               +
   PERMISSIONS                                   read/write        +
   MOUNT OPTIONS                                [ ]                +
   Start Disk Accounting?                        no                +
   Block size (bytes)                            4096
   Inline Log?                                   no
   Inline Log size (MBytes)                     [ ]
   Extended Attribute Format                    [v1]               +
   Enable Quota Management                       no                +
   Allow Small Inode Extents                     no                +
   Enable EFS?                                   no                +
```


Figure 10-25. Change/Show Characteristics of an Enhanced Journaled File System AU1412.0

Notes:

Changing JFS2 file system characteristics

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 or SMIT.

JFS and JFS2 similarities

Most of the statements in the notes associated with the last visual (regarding changing the characteristics of JFS file system) also apply to JFS2 file systems. An important difference is that it is possible to decrease the size of a JFS2 file system.

Dynamically shrinking a JFS2 file system

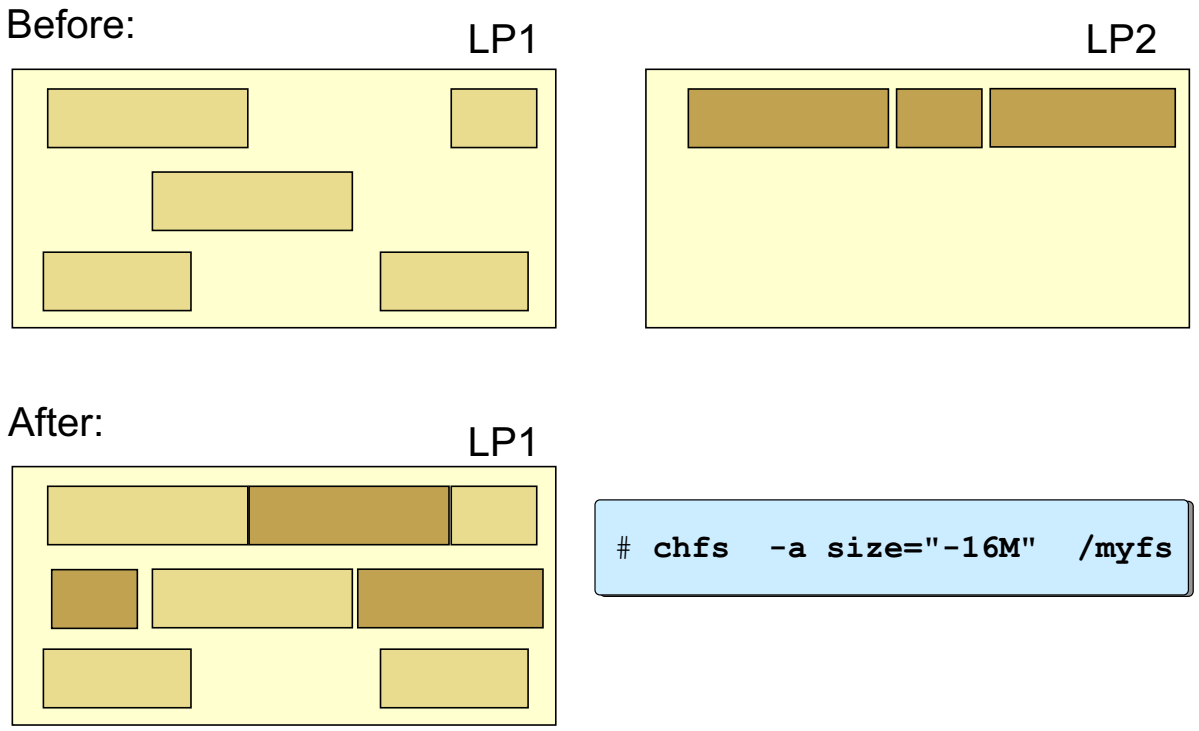


Figure 10-26. Dynamically shrinking a JFS2 file system AU1412.0

Notes:

Dynamic file system shrink capability (with JFS2)

On versions of AIX prior to AIX 5L V5.3, there is no way to shrink a file system dynamically while you are using it, although you can easily extend as needed. The procedure to shrink a file system was to create a new smaller version, copy the data, take the old version offline, then delete the old version. In AIX 5L v5.3 and later, dynamic file system shrink is available with Enhanced Journaled File System (JFS2).

Changes in `chfs` and SMIT

The **`chfs`** command (and corresponding SMIT panel) support for the size attribute has been changed to support either a final size which is smaller than the current size or a decrement (value preceded with the minus sign). The requested difference is translated into a *whole number of physical partitions with any remaining amount beyond being ignored*. Thus, asking to decrease by 1 MB would have no effect (minimum PPsize for JFS2 is 16 MB).

Restrictions

There must be enough free space in the remaining physical partitions of the file system to stored the file data and metadata structures being moved from the freed physical partitions.

Effect on inline log

If there is an inline log, that log is also proportionally adjusted in size.

Remove a Journaled File System

```
                 Remove a Journaled 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               +





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


Figure 10-27. Remove a Journaled File System AU1412.0

Notes:

Ways to remove a file system

The **rmfs** command or SMIT can be used to remove a file system. JFS and JFS2 file system removal work the same way.

Restrictions

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.

Effects of using rmfs command

The **rmfs** command removes 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.

Syntax

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

rmfs [-r] [-i] FileSystem

- **-r** Removes the mountpoint of the file system
- **-i** Displays warning and prompts the user before removing the file system

Example:

```
# rmfs -r /home/george/myfs
```

Add a RAM file system

- Create a RAM disk of 4 MB

```
# mkramdisk 4M
/dev/rramdisk0
```

- Create a JFS file system on this RAM disk

```
# mkfs -V jfs /dev/ramdisk0
mkfs: destroy /dev/ramdisk0 (yes)? y
```

- Create mount point

```
# mkdir /ramdisk
```

- Mount RAM file system

```
# mount -V jfs -o nointegrity /dev/ramdisk0 /ramdisk
```


Figure 10-28. Add a RAM file system AU1412.0

Notes:

Adding a RAM file system

The visual shows a series of steps that can be used to add a RAM file system.

Use of the `mkramdisk` command

The purpose of the **`mkramdisk`** command is to enable creation of file systems directly in memory. This is useful for applications that make many temporary files.

Limitations

Use ramdisk only for data that can be lost. After each reboot, the ramdisk file system is destroyed and must be rebuilt.

Add a UDF file system on a DVD-RAM

- Create a UDF file system

```
# udfcreate -d /dev/cd0
```

- Change the label on a UDF file system

```
# udflabel -d /dev/cd0 -l testdvd
```

- Create a mount point

```
# mkdir /dvddisk
```

- Mount a UDF file system

```
# mount -V udfs -o rw /dev/cd0 /dvddisk
```

- Check a UDF file system

```
# udfcheck -d /dev/cd0
```


Figure 10-29. Add a UDF file system on a DVD-RAM AU1412.0

Notes:

Adding a UDF file system

The visual shows a series of steps that can be used to add a UDF file system on a DVD-RAM.

Using a UDF

Once you have created a UDF on a DVD-RAM, you can just treat it like a normal hard disk. It enables you to read, write, delete, copy, move, mount, unmount and edit a file within the DVD directory.

The Universal Disk Format Specification (UDFS)

The Universal Disk Format Specification (UDFS) is based on the Micro Design International (MDI) UDF implementation. It supports UDFS 1.50, 2.00, and 2.01. (The implementation is based on UDFS 2.01, but backward compatible to 2.00 and 1.50.) It is now possible to read and write to a DVD media in 32/64 bit mode.

System storage review

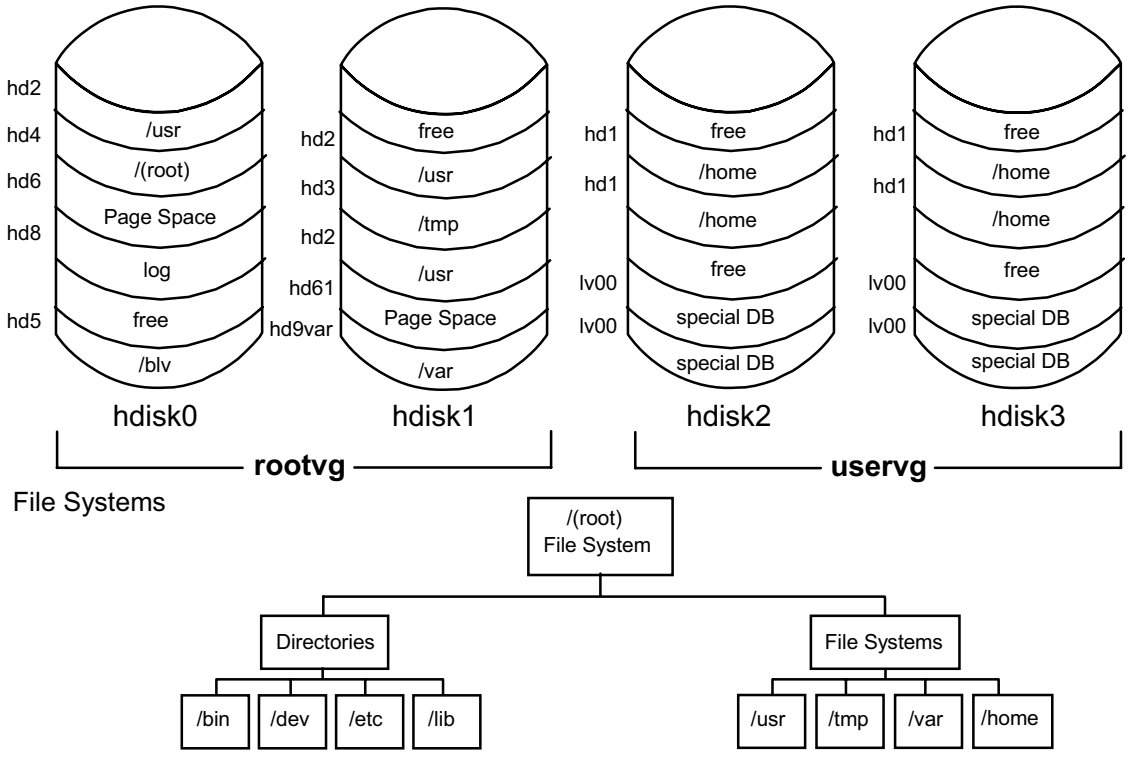


Figure 10-30. System storage review AU1412.0

Notes:

Difference between file system and simple directory

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.

File systems on the visual

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.

Simple directories

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. There is no separate logical volume associated with these directories.

Checkpoint

1. Does 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, does the size of the logical volume increase when the size of the file system that is residing on it increases? _____

3. If you remove a logical volume, is the file system that is residing on it removed as well?

   _____

   _____

Figure 10-31. Checkpoint AU1412.0

Notes:

Exercise 10: Working with file systems (parts 2-6)

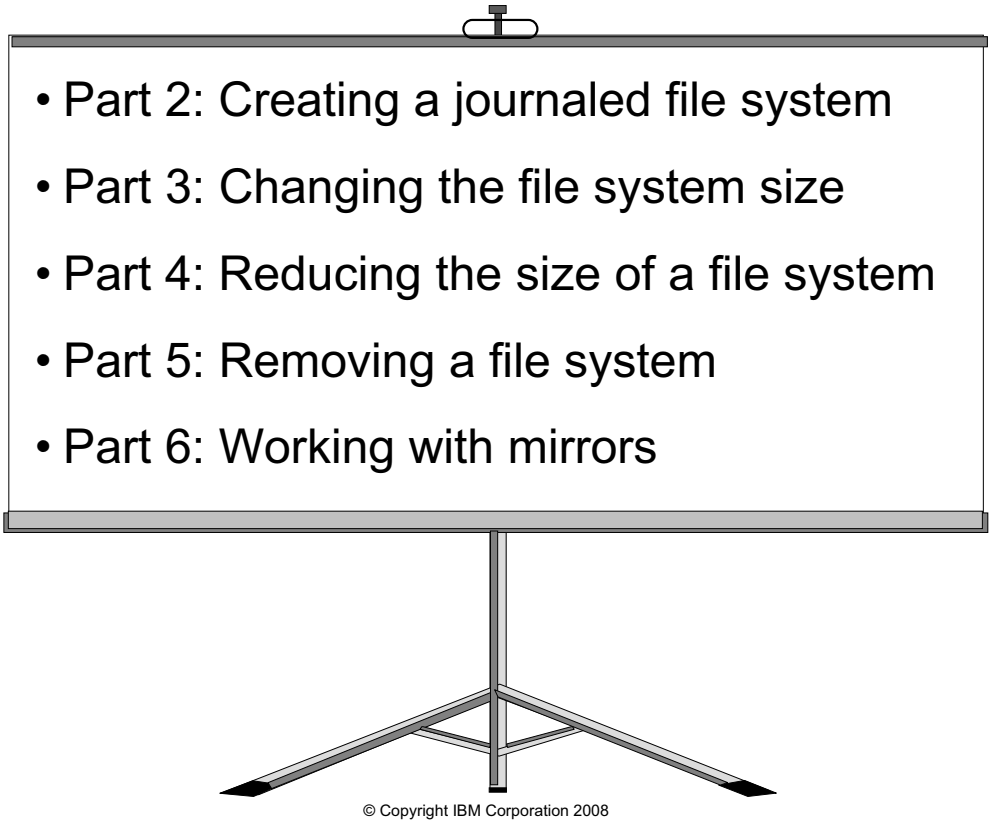


Figure 10-32. Exercise 10: Working with file systems (parts 2-6) AU1412.0

Notes:

Introduction

This lab has you build on the logical volume you created in the last exercise. It also gives 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 *Student Exercise Guide*.

Unit summary

- The components of a JFS 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.

Figure 10-33. Unit summary AU1412.0

Notes:

Unit 11. 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

SG24-7559	*AIX Version 6.1 Differences Guide*
Online	*AIX 6.1 Operating System and device management*

Note: References listed as "Online" above are available at the following address:

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

Unit objectives

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

Figure 11-1. Unit objectives AU1412.0

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 11-2. Space management AU1412.0

Notes:

The need to monitor file system growth

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 file system growth and expand file systems as required *before* they get full. If a file system becomes 100% full, then the users receive out of space messages when they try to extend files.

Regular use of the `df` command

One useful technique is to run the **`df`** command via **`cron`** (the job scheduler) to perform a regular check of the space available in the file system and produce a report. **`cron`** is covered in a later unit.

The Resource Monitoring and Control (RMC) subsystem

You can also use the Resource Monitoring and Control (RMC) subsystem that is based on the Reliable Scalable Cluster Technology (RSCT) on the IBM SP platform. Use the Web-based System Manager to configure RMC. You have 84 conditions and 8 responses to predefine. The **ctrmc** subsystem is started in the **/etc/inittab**.

Further discussion of RMC is not within the scope of this course, but there is a good description of this facility in the *AIX 5L Differences Guide Version 5.2 Edition* (SG24-5765-02).

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    Iused   %lused  Mounted on
/dev/hd4       294912       228088   23%     1925     7%     /
/dev/hd2       3309568      339408   90%     36788   47%     /usr
/dev/hd9var    65536        37600    43%     479     11%     /var
/dev/hd3       131072       129352    2%     54       1%     /tmp
/dev/hd1       32768        32064     3%     5        1%     /home
/proc                         -        -        -      -    /proc
/dev/hd10opt   163840       20760    88%     1617    36%     /opt
/dev/hd11admin 262144       261416    1%     5        1%     /admin
/dev/ramdisk0  8192         7848      5%     17       2%     /ramdisk
```


Figure 11-3. Listing free disk space AU1412.0

Notes:

Importance of the `df` command

The **df** command lists the free space on all mounted file systems.

This is an important command to be aware of and to use frequently. If you run out of space in a file system (especially / or /**tmp**), system corruption could occur.

Useful `df` command flags

A number of flags (options) can be used with the **`df`** command. Some of the most useful of these flags are shown in the following table:

Flag (Option)	Brief Description of Output
`df -I`	Displays information on the total number of blocks, the used space, the free space, the percentage of used space, and the mount point for the file system
`df -k`	Information displayed is like that shown on the visual, but statistics in first two columns are given in units of 1024-byte blocks
`df -m`	Information displayed is like that shown on the visual, but statistics in first two columns are given in units of 1 MB blocks
`df -g`	Information displayed is like that shown on the visual, but statistics in first two columns are given in units of 1 GB blocks

The **`-m`** and **`-g`** flags were introduced in AIX 5L V5.2.

Control growing files

- /var/adm/wtmp
- /etc/security/failedlogin
- /var/adm/sulog
- /var/spool/*/*
- $HOME/smit.log
- $HOME/smit.script
- $HOME/websm.log
- $HOME/websm.script

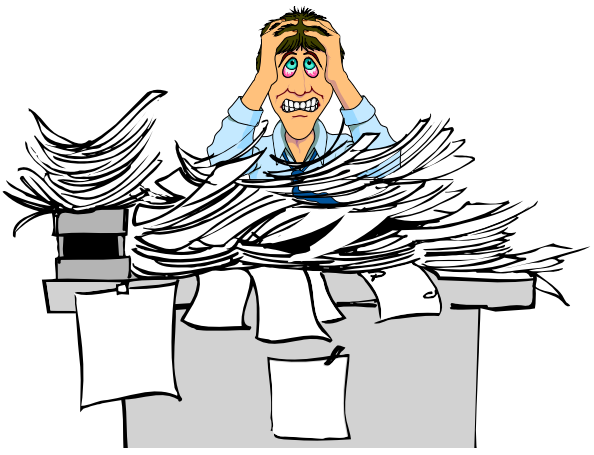


Figure 11-4. Control growing files AU1412.0

Notes:

Managing files that grow

Growing files should be monitored and cleaned out periodically. Some of the files that grow are listed on the visual.

Records of login activity

The files /**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 turned on, /**var**/**adm**/**wtmp** is kept to a reasonable size. 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 log file can be purged by overwriting it with a null string. Two ways of overwriting a log file in this way are illustrated in the following examples:

- Example 1:

  # **cat /dev/null > /var/adm/wtmp**

- Example 2:

  # **> /etc/security/failedlogin**

The file /**var**/**adm**/**sulog** can be edited directly.

The /var/spool directory

The directory /**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 with the printer files, 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**

Records of SMIT and Web-based System Manager activity

Files such as **smit.log** and **websm.log** in the home directory of the **root** user and other system administration accounts can also become quite large. These files need to be monitored regularly and managed appropriately.

The `skulker` command

- The **`skulker`** command cleans up file systems by removing unwanted or obsolete files
- Candidate files include:
  - Files older than a selected age
  - Files in the **/tmp** directory
  - **a.out** files
  - **core** files
  - **ed.hup** files
- **`skulker`** is normally invoked daily by the **`cron`** command as part of the **crontab** file of the **root** user
- Modify the **`skulker`** shell script to suit local needs for the removal of files

Figure 11-5. The `skulker` command AU1412.0

Notes:

Function of the `skulker` command

The shell script **`/usr/sbin/skulker`** includes a series of entries containing commands that remove unwanted or obsolete files of various types. (To analyze the commands that are executed by each entry, print out or view the contents of the /**usr**/**sbin**/**skulker** file.) Currently, items removed by the **`skulker`** script include the following:

- 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

Concerns related to `skulker`

A particular version of **`skulker`** is suited to the operating system and level with which it was distributed. If the operating system has been upgraded or modified, it may be *inadvisable to use an old version* of **`skulker`**.

In addition, the **`skulker`** shell script is moderately complex. When making modifications, you should make a copy of the shell script first - just in case!

Note that if **`skulker`** is modified, or if it is used on the incorrect version of the operating system, it ceases to be a supported component of AIX.

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 11-6. Listing disk usage AU1412.0

Notes:

Use of the du command

There may be a number of files or users that are causing the increased use of space in a particular file system. The **du** command helps to determine which files and/or users are causing the problem.

Specifying the units du should use

By default, **du** gives size information in 512-byte blocks. Use the **-k** option to display sizes in 1 KB units, use the **-m** option to display sizes in 1 MB units, or use the **-g** option to display sizes in 1 GB units. The options **-m** and **-g** were introduced in AIX 5L V5.2.

Specifying output by file

By default, **du** gives information by directory. With the **-a** option, output is displayed by file rather than by directory.

Using `du` in conjunction with `sort`

If the output of **`du`** is sorted numerically and in descending order (using the **`-n`** and **`-r`** flags of the **`sort`** command) by the value in the first column, this output 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` flag

The **`-x`** flag/option is also very useful. When you use **`du -ax`**, the report only shows information from the specified file system. This is the best way to determine what file is filling a particular file system.

Using the `find` command to locate large files

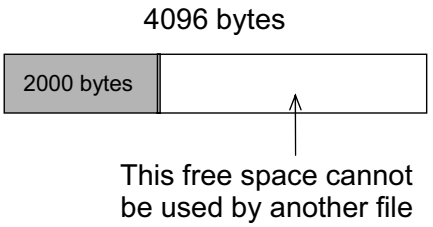
The **`find`** command is useful for locating files that are over a certain size. For example, to find all files that contain more than 1,000,000 characters and then list them use the following command:

```
# find  . -size +1000000c -exec ls -l {} \;
```

Fragmentation considerations

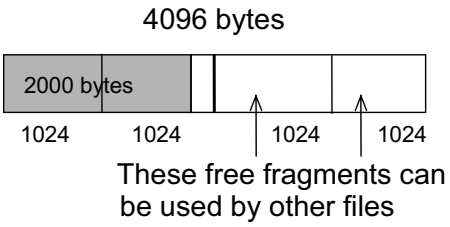
Without fragmentation

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 11-7. Fragmentation considerations AU1412.0

Notes:

Benefits of a small fragment size

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 are 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.

Adverse effects of a small fragment size

Although small fragment sizes can be beneficial in reducing wasted disk space, they 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 are 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 sizes 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.

JFS2 block size

In JFS2, the block size has a function similar to that of the JFS fragment size. The default block size is 4096, but it 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

```
defragfs [-q | -r | -s]  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**, **-r** or **-s**

-s Gives short report regarding the current state of the file system

Figure 11-8. Defragmenting a file system AU1412.0

Notes:

Information returned by `defragfs` command (JFS file systems)

Information that is returned by the **defragfs** command for a JFS file system includes the following:

- `Number of Fragments Moved`: Displays the total number of fragments that have been moved
- `Number of Logical Blocks Moved`: Displays how many logical blocks were relocated
- `Number of Allocation Attempts`: The required number of calls to the allocation routine to defragment the file system
- `Number of Exact Matches`: The number of times the fragments moved fit exactly in some free space

Note: Sometimes the estimates for these items provided when running **defragfs** with the **-q** or **-r** option indicate results different from what is actually done when **defragfs** is run without any options.

Using defrags with JFS2 file systems

The **defragfs** command can also be used with a JFS2 file system. The information returned by **defragfs** in this case is quite different from that returned when using the command with a JFS file system. Refer to the entry for **defragfs** in the *AIX 6.1 Commands Reference online* for additional information regarding this topic.

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 checks all file systems which have the **check=true** attribute set in the **/etc/filesystems**
- Orphan files are placed in the **lost+found** directory
- Unmount the file system before running **fsck**

Figure 11-9. Verify a file system AU1412.0

Notes:

Function of the fsck command

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

The **-p** (preen) option is used to check a file system and make only minor changes, without bothering the user. When **fsck** is run under SMIT, this option is used.

Error reporting by `fsck`

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 a *no* answer to all questions ask by **`fsck`**. The yes option is typically used to recover a badly damaged file system. Using the **`-y`** option allows **`fsck`** to discard some badly damaged files. Note, however, that mounted file systems are not repaired.

The lost+found directory

If any files are found that are not allocated to a directory anywhere in the file system being checked, then **`fsck`** creates an entry for that data in the **lost**+**found** directory in the root 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`**.

Running of `fsck` during system startup

The **`fsck`** command executes each time the system boots up (from the /**etc**/**rc** file).

Unmounting the file system

Unmount the file system before using **`fsck`**. If a file system is in use, the **`fsck`** utility can get confused and return inaccurate messages. If **`fsck`** were to try repairing a file system that is in use, it could actually *create* problems. Because of this, **`fsck`** refuses make corrections to a mounted file system.

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

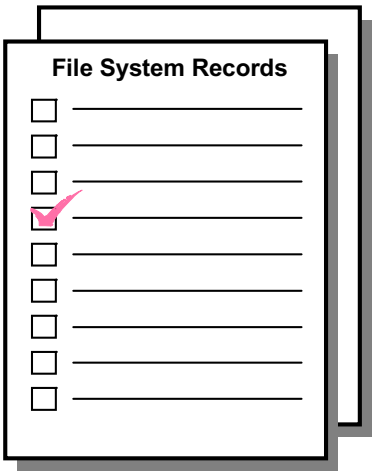


Figure 11-10. Documenting file system setup AU1412.0

Notes:

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 or False? It is good practice to run **`fsck -y`** on all file systems, even if they are mounted.

Figure 11-11. Checkpoint AU1412.0

Notes:

Exercise 11: Managing file systems

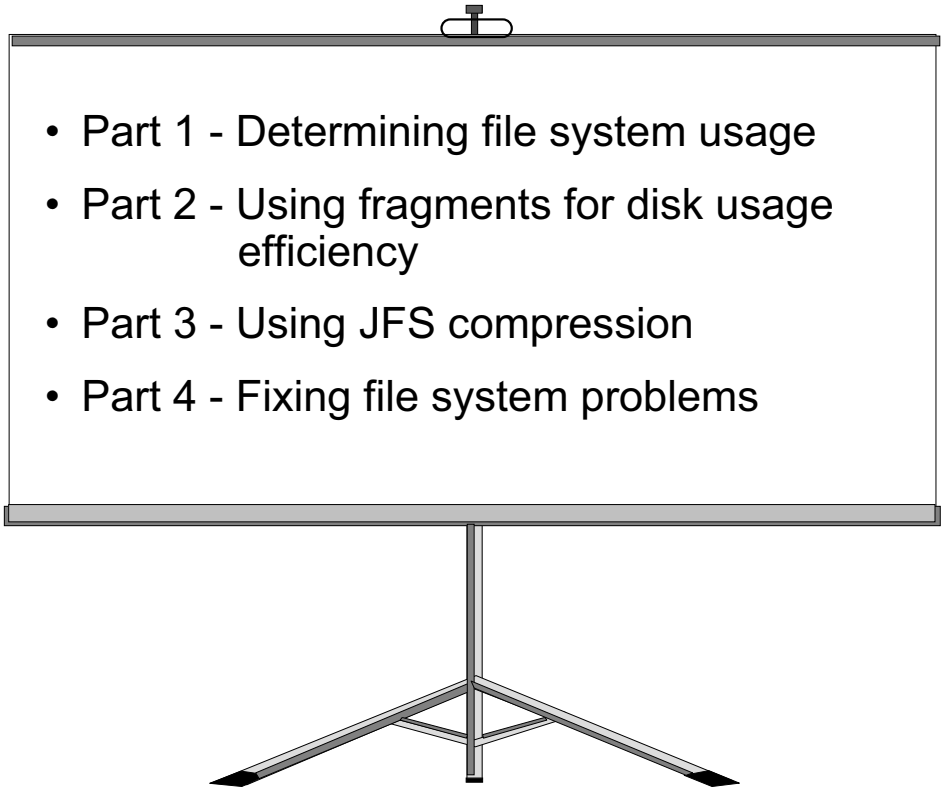


Figure 11-12. Exercise 11: Managing file systems AU1412.0

Notes:

Introduction

The lab allows 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 *Student Exercises Guide*

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 11-13. Unit summary AU1412.0

Notes:

Unit 12. Paging space

What this unit is about

This unit presents the key concepts related to 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

SG24-7559	*AIX Version 6.1 Differences Guide*
Online	*AIX 6.1 Operating system and device management*
Online	AIX 6.1 Installation and migration

Note: References listed as "Online" above are available at the following address:

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

Unit objectives

After completing this unit, you should be able to:

- 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 12-1. Unit objectives AU1412.0

Notes:

What is paging space?

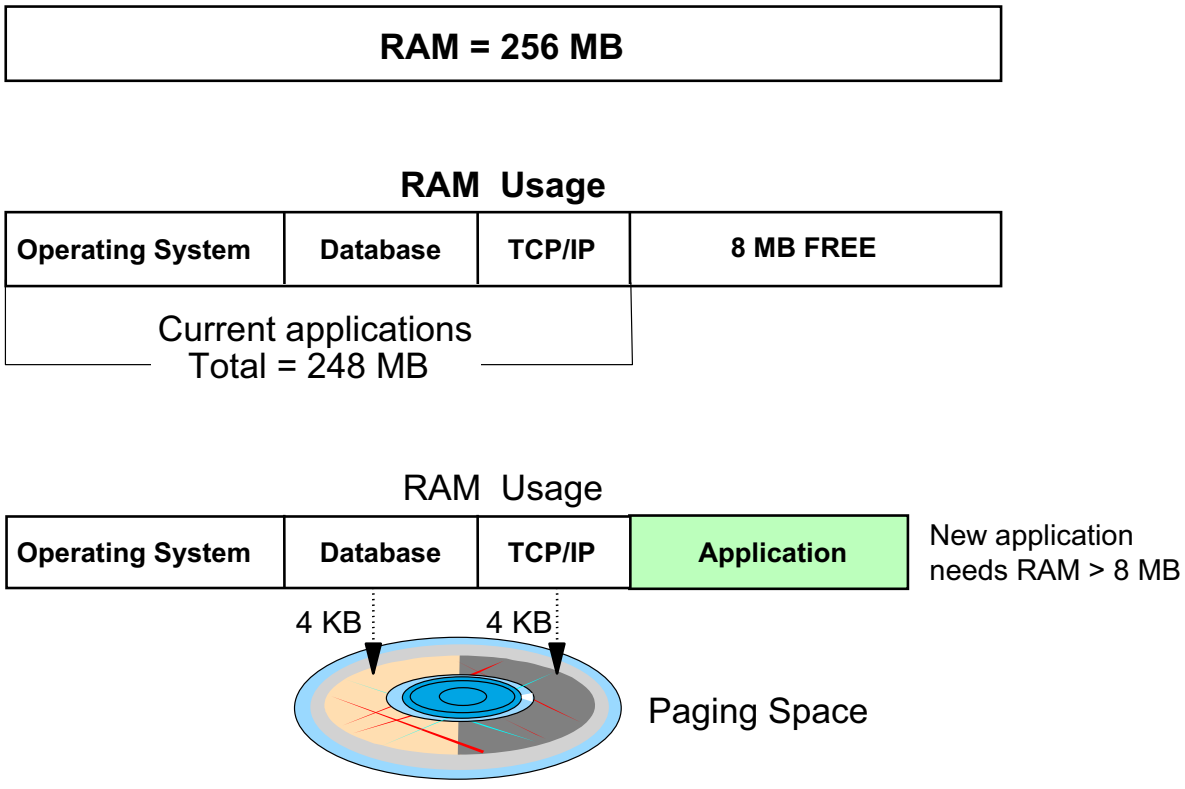


Figure 12-2. What is paging space? AU1412.0

Notes:

Use of paging space

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 (and the data it contains cannot be moved to a “permanent home” in file system space), a 4 KB block or page of disk space is allocated and the data from the page frame is moved to a special area on disk. This area on disk is called *paging space*.

The paging space is a reserved area on disk that can contain information that resided in memory but was inactive and was moved to make room for information that is currently being used. If paged-out information is needed in memory again, the page is retrieved and brought back into memory or *paged-in*.

Role of the Virtual Memory Manager (VMM)

In the AIX environment, paging and virtual storage are managed by the Virtual Memory Manager (VMM).

Paging space

- Is a secondary storage area for over-committed memory
- Holds inactive 4 KB pages on disk
- Is not a substitute for real memory

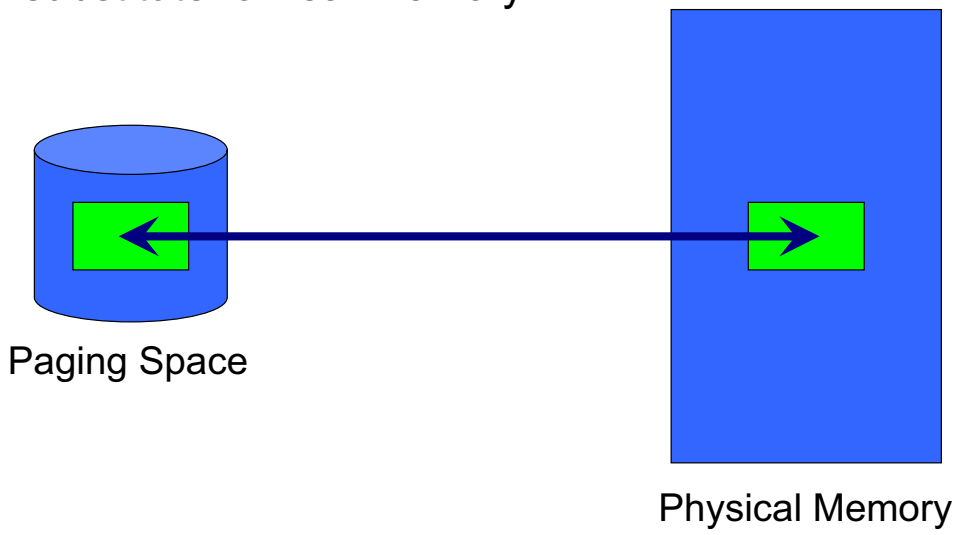


Figure 12-3. Paging space AU1412.0

Notes:

A secondary storage area

Paging space is disk storage for 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.

A temporary holding area for inactive pages

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 requires 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 reaches 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 are wasted performing only paging activity, and no real work gets done.

Thrashing indicates a need for additional memory

Increasing the amount of paging space when your machine is thrashing does 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 12-4. Sizing paging space AU1412.0

Notes:

Creation of paging space

Paging space is created during AIX installation.

The initial size is dependent on various factors, particularly the amount of RAM in your system. Currently, the initial paging space size is determined according to the following standards:

- Paging space can use no less than 16 MB, except for **hd6**, which can use no less than 64 MB in AIX V4.3 and later versions
- Paging space can use no more than 20% of total disk space
- If RAM is greater than or equal to 256 MB, paging space is 512 MB
- If RAM is less than 256 MB, paging space is twice the size of RAM

Adjusting the amount of paging space

The initial size of paging space 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 dictates 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 utilization of your paging space.

Monitoring paging space

Monitoring the utilization of the paging space is done with the command **`lsps -a`**. This command and its output are covered shortly.

Results of low paging space

If your system runs low on paging space, a message is sent to the console and sometimes to users as well. At this point, the system is unable to start any new processes until some running processes are terminated or release allocated memory. This situation should obviously be avoided. A low paging space condition may be indicated by the appearance of one or more of the following messages on the console or in response to a command on any terminal:

```
"INIT: Paging space is low"

"ksh: cannot fork no swap space"

"Not enough memory"

"Fork function failed"

"fork () system call failed"

"unable to fork, too many processes"

"Fork failure - not enough memory available"

"Fork function not allowed.  Not enough memory available."

"Cannot fork: Not enough space"
```

The situation can get worse. If paging space continues to fill, non-system processes are 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 physical volumes
- Use multiple disk controllers

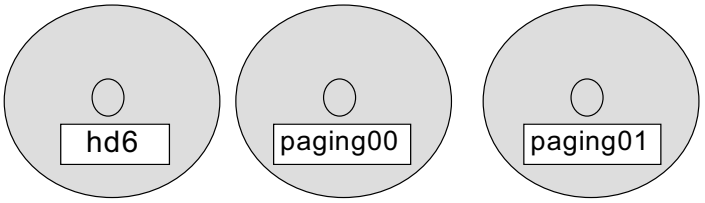


Figure 12-5. Paging space placement AU1412.0

Notes:

Introduction

Placement and size of your paging space does impact its performance. The following material contains tips regarding placement and size of paging areas.

Configure only one paging space per disk

Do not have more that one paging space per disk. The paging space is allocated in a round-robin manner, and uses all paging areas equally. If you have two paging areas on one disk, then you are no longer spreading the activity across several disks.

Use disks with low levels of activity

Paging space performs best when it is not competing with other activity on the disk. Use disks that do not have much activity.

Create paging spaces of roughly the same size

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 is not balanced. Smaller paging areas fill faster.

Do not span multiple physical volumes

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 disk, the round-robin technique treats the paging area as one single paging area. Therefore, the activity is not evenly spread across the disks.

Use multiple disk controllers

Use disks on different controller. If the disks are attached to different controllers you get better throughput when reading and writing to the disk. That improves your performance.

Checking paging space

- Check paging activity:

```
# lsps -a
Page Space   Physical Volume   Volume Group   Size    %Used   Active   Auto   Type    chksum
hd6          hdisk0            rootvg         64MB    43      yes      yes    lv      0
paging00     hdisk2            rootvg         64MB    20      yes      yes    lv      0
```

- Check total RAM:

```
# lsattr -El sys0 -a realmem
realmem 262144 Amount of usable physical memory in KB False
```

- Check paging space activated at startup:

```
# cat /etc/swapspaces
...
hd6:
   dev=/dev/hd6
...
paging00:
   dev=/dev/paging00
```


Figure 12-6. Checking paging space AU1412.0

Notes:

The `lsps` command

The **`lsps`** command lists detailed information regarding 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 are named **paging00**, **paging01**, and so on.

The /etc/swapspaces file

The file **/etc/swapspaces** contains a list of the paging space areas that are 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.

                                                  [Entry Fields]
  Volume group name                                   rootvg
  SIZE of paging space (in logical partitions)       [4]          #
  PHYSICAL VOLUME name                                hdisk2     +
  Start using this paging space NOW?                  no         +
  Use this paging space each time the system is       no         +
          RESTARTED?



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


Figure 12-7. Adding paging space AU1412.0

Notes:

Ways of adding extra paging space

To add extra paging space volumes to the system, you can use SMIT (as illustrated on the visual), the **mkps** command, or the Web-based System Manager.

Using the mkps command

When using the **mkps** command, the syntax and options are:

mkps [-a] [-n] [-t *Type*] -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 (adds it to **/etc/swapspaces**)

-n	Activate the paging space immediately
-t *Type*	Specifies the type of paging space (lv or nfs)

When a paging space is created, the /**etc**/**swapspaces** file is also updated, if needed.

The following example illustrates use of the **mkps** command:

```
# 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 the system is     yes       +
          RESTARTED?



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


Figure 12-8. Change paging space AU1412.0

Notes:

Characteristics that can be changed

A 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**).

These changes can be made via SMIT (as illustrated on the visual) or by using the **chps** command.

Decreasing paging space

The ability to decrease paging space was introduced in AIX 5L 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 are not allowed to run out of paging space. The script checks paging space actually in use and adds a paging space warning threshold buffer. The SMIT fastpath is **smit chps**.

The process **chps** goes through to decrease an active paging space is as follows:

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 32 MB.

When you reduce the primary paging space, a temporary boot image and a temporary /**sbin**/**rc.boot** pointing to this temporary primary paging space are created to make sure the system is always in a state where it can be safely rebooted.

These command enhancements are also available through the Web-based System Manager, starting in AIX 5L V5.2.

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**/**swapspaces**. This command is run in /**etc**/**rc** at system startup.

Examples of chps command use

The following examples illustrate use of the **chps** command:

- Example 1: Delete 1 logical partition from the **paging00** paging space

  # **chps -d 1 paging00**

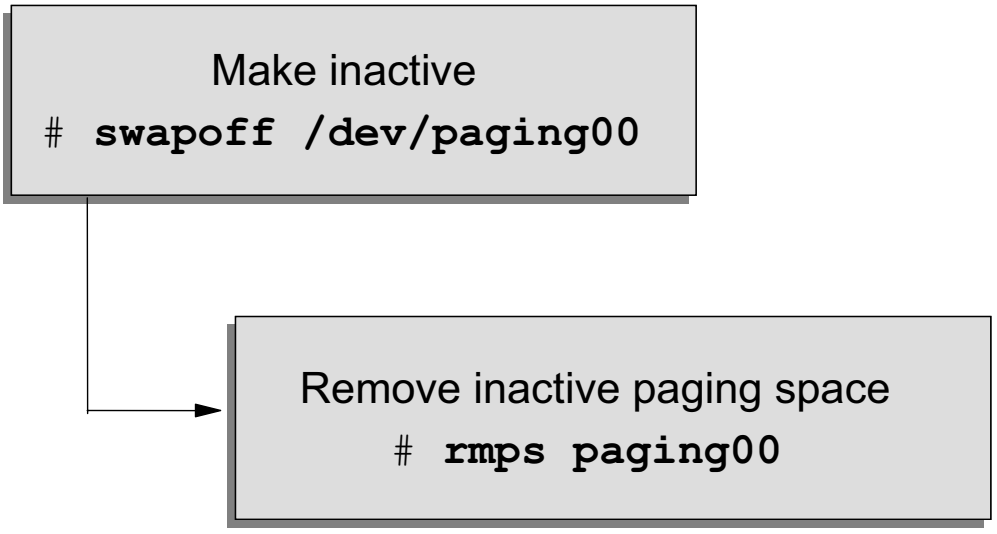
- Example 2: Add 1 logical partition to the **paging00** paging space

  # **chps -s 1 paging00**

Refer to the entry for **chps** in the online *AIX 6.1 Commands Reference* (or the corresponding **man** page) for more information regarding the **chps** command.

Remove paging space

To remove an active paging space:



NOTE: **/dev/hd6** cannot be removed using this process

Figure 12-9. Remove paging space AU1412.0

Notes:

Deletion of surplus paging space

As we've discussed, paging space can be added to the system if necessary. Similarly, surplus paging space can be *deleted* to free up the disk space for other logical volumes.

Deactivation of paging space

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 5L V5.1, active paging spaces can be deactivated while the system is running using the **`swapoff`** command or with the SMIT fastpath **`swapoff`**.

Reasons the `swapoff` command may fail

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 fails.
- I/O errors.

Problems with paging space

- Paging space too small:
  Dynamically increase the size by allocating more partitions
  chps -s *LogicalPartitions PagingSpace*

  Example:
  # **chps -s 1 paging00**

- Paging space too large:
  Dynamically decrease the size by deallocating partitions
  chps -d *LogicalPartitions PagingSpace*

  Example:
  # **chps -d 1 paging00**

Figure 12-10. Problems with paging space AU1412.0

Notes:

Overview

All ideas on this visual have already been considered. The visual simply pulls together all the ideas discussed so far.

Documenting paging space setup

- Run the `lsps` command
- Have a hardcopy of the **/etc/swapspaces** file

Figure 12-11. Documenting paging space setup AU1412.0

Notes:

Running `lsps`

Run `lsps` to monitor paging space activity. Keep good documentation so that you know what is normal for that system.

A copy of /etc/swapspaces

Keep a copy of /**etc**/**swapspaces** so that you know what paging spaces are defined to started at boot.

Checkpoint

1. What conclusions regarding potential paging space problems can you reach based on the following listing?

```
Page      Physical Volume Size  %Used Active Auto Type chksum
Space     Volume   Group

hd6       hdisk0   rootvg 64 MB 43%  yes    yes   lv    0
paging00  hdisk1   rootvg 64 MB  7%  yes    yes   lv    0
paging01  hdisk1   rootvg 16 MB 89%  yes    yes   lv    0
```

_____
_____
_____
_____
_____
_____
_____
_____
_____

2. True or False? The size of **paging00** (in the above example) can be dynamically decreased.

Figure 12-12. Checkpoint AU1412.0

Notes:

Exercise 12: Paging space

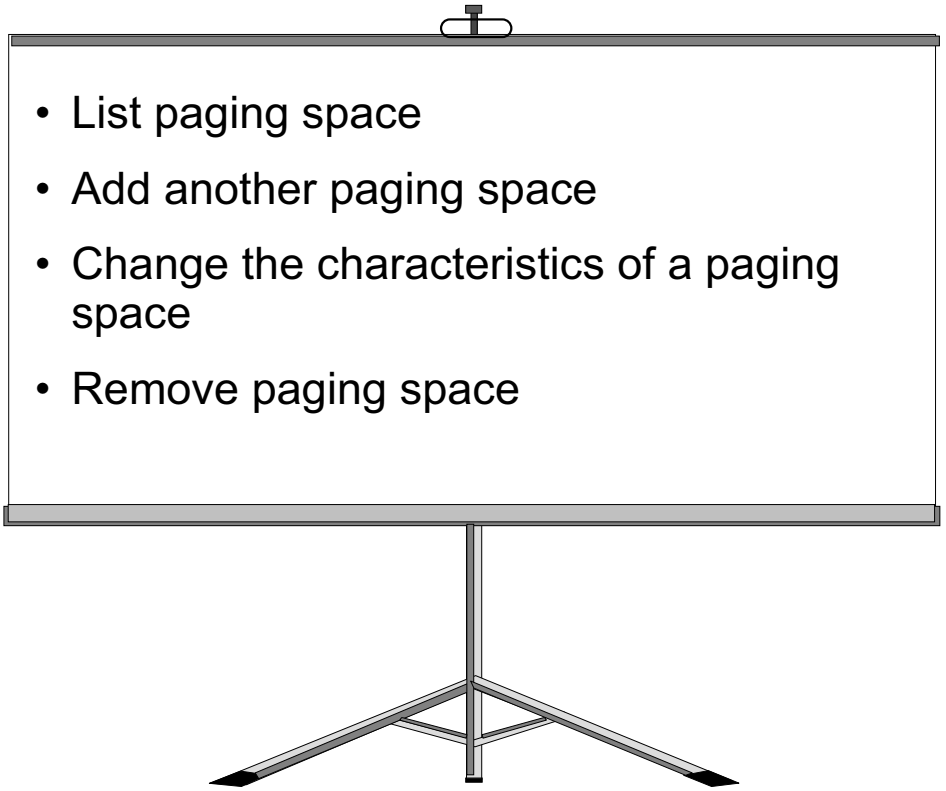


Figure 12-13. Exercise 12: Paging space AU1412.0

Notes:

Introduction

This lab allows you to add, decrease, monitor, and remove paging space.

The exercise can be found in your *Student Exercises Guide*.

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 **`lsps` command** can be used to do this.
- Paging space can be **inactivated** and the **size** can be increased or decreased dynamically.

Figure 12-14. Unit summary AU1412.0

Notes:

Unit 13. 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 that 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

SG24-7559	*AIX Version 6.1 Differences Guide*
Online	*AIX 6.1 Operating system and device management*
Online	*AIX 6.1 Installation and migration*

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

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

Unit objectives

After completing this unit, you should be able to:

- 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 13-1. Unit objectives AU1412.0

Notes:

Why backup?

- Data is very important:
  - Expensive to re-create
  - Can it be re-created?
- 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 and after upgrade)
- Long term archive

Figure 13-2. Why backup? AU1412.0

Notes:

Reason to have backups

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.

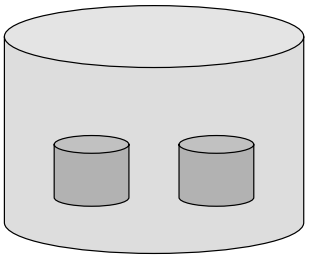
When reorganizing the file systems on the disk you need to backup file systems so that they can be deleted and moved to another location.

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.

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



rootvg

Figure 13-3. Types of backup AU1412.0

Notes:

System backup

The system backup (system image backup) records and preserves all system data, including logical volume inter/intra allocation policies, striping, file system setup (nbpi, compression, fragment size) and paging setup.

Full backup

A full backup is similar to a system backup, but it is for user data.

Incremental backup

There are two types of incremental backups:

- The first method is to do a full backup. For example, do a full backup 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 may 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 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 is similar to a full backup. The restoration procedure does not depend on the tape from the previous day.

Backup strategy

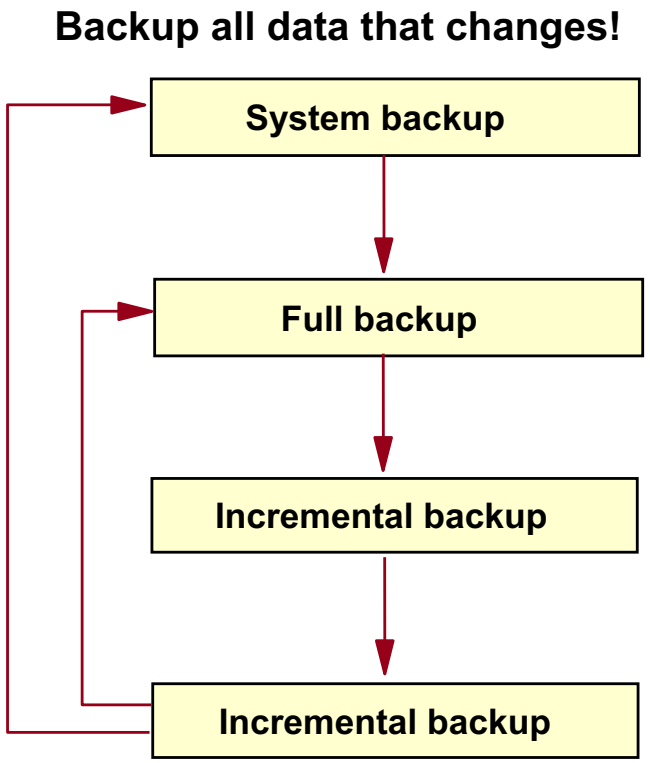


Figure 13-4. Backup strategy AU1412.0

Notes:

Elements of a backup strategy

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.

Data Consistency

Some storage systems provide a mean for holding all I/O to a set of volumes to produce a consistent image for the held volumes. This function is referred to as ***"consistency groups"***. Consistency groups are used when backing up data to guarantee that the restored data functions correctly. Set up by the administrator, it is possible to create a

consistency group for a point-in-time image of a JFS2 file system and log. The **JFS2 freeze/thaw** commands are used for this function.

The recommended scenario would include operations listed below:

1. Set the application to on-line backup mode, if possible
2. Issue a sync command
3. Issue the file system freeze command

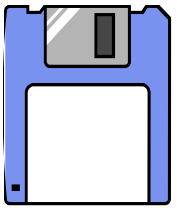
   chfs -a freeze=,timeout in second> /fsname
4. Backup the JFS2 file system data
5. Issue the file system thaw command

   chfs -a freeze=off /fsname
6. Set the application back to normal mode

Backup devices - diskette

/dev/fd0 - Built in 3 1/2-inch diskette drive

/dev/fd1 - Second diskette drive



Drive

	3 1/2-inch (1.44)	3 1/2-inch (2.88)
/dev/fdxl	720 KB	720 KB
/dev/fdxh	1.44 MB	2.88 MB
/dev/fdx.9	720 KB	720 KB
/dev/fdx.18	1.44 MB	1.44 MB
/dev/fdx.36	-	2.88 MB

Figure 13-5. Backup devices - diskette AU1412.0

Notes:

Introduction

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 most likely has 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.

Commands

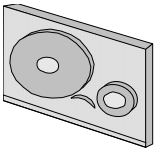
To format a diskette, use the **`format`** command. There is a **`-l`** options if you want to format at low density.

The **`flcopy`** 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
- VXA
- QIC

	Low Capacity	Retension 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 13-6. Backup devices - tape AU1412.0

Notes:

Tape technologies

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.
- Quarter Inch Cartridge (QIC)

  Can hold up to 4 GB with a data transfer rate of 380 KB/sec.

- Digital Linear Tape (DLT)

  Can hold up to 70 GB at a transfer rate of 10 MB/sec.
- Magstar

  Can hold up to 420 GB per cartridge with a transfer rate of 15 MB/sec.
- VXA Tape Data Cartridge

  Can hold up to 160 GB with a data transfer rate of 12 MB/sec.
- 8 mm Data Cartridge with smart clean technology

  Can hold up to 150 GB with a data transfer rate of 30 MB/sec.

For large scale backups, tape subsystems and tape libraries would the sensible choice. For details on all tape devices supported on AIX systems, go to: *www.ibm.com/storage/tape*.

Device names

The tape devices use the logical device name of **rmt*x*** (raw magnetic tape). In the chart, you see the seven 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 **rmt*x*** and **rmt*x*.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 the `tctl` or `mt` commands. If there two tape devices, `tcopy` allows 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
- For CD:
  - OEM CD-RW drive
  - Third-party CD burn software
    (AIX Toolbox for Linux Applications)
- For DVD:
  - Need 7210 DVD-RAM drive
  - No additional software needed for UDF format

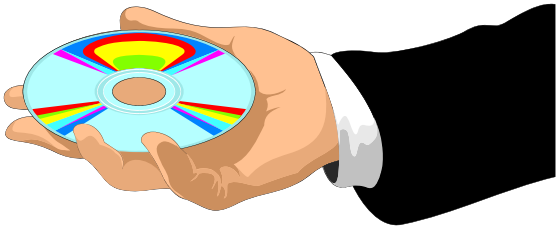


Figure 13-7. Backup device - read/write optical drive AU1412.0

Notes:

Introduction

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 is read only.

The optical drive volume group must be wholly contained on the single optical disk. It cannot span beyond one optical drive.

CD-ROM file system

To use the information on the read/write 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 using one of the following commands:
   - **smit crcdrfs**
   - **crfs -v cdrfs -p ro -d *DeviceName***
2. Mount the file system (**mount *mount_point***)

JFS file system

To use the read/write optical drive as a read/write device, you must create a volume group using the same commands that are used with a hard drive.

1. Make the volume group with one of the following commands:
   - **smit mkvg**
   - **mkvg -f -y *VGName* -d 1 *DeviceName***
2. Create a file system with one of the following commands:
   - **smit crfs**
   - **crfs -v jfs -g *VGName* -a size=*FSsize* -m *Mountpt* -A [yes|no] -p rw**
     The **-A** option designates whether to automatically mount at system start.
3. Mount the file system (**mount *mount_point***)

CD burner

To burn a backup image onto a CD (IS09660), you must install an OEM drive and software that is capable of CD writes.

To find out what CD writers are supported examine: **/usr/lpp/bos.sysmgt/README.oem_cdwriters**.

Two of the CD burner software packages that have been tested with AIX and are provided on the AIX Toolbox for Linux Applications CD are **mkisofs** and **cdrecord**. You may alternatively download the software from:

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

Whatever software package is installed you need to link their executables to the AIX standard command names of **/usr/sbin/mkrr_fs** and **/usr/sbin/burn_cd**. For more details refer to: **/usr/lpp/bos.sysmgt/mkcd.README**.

Backing up to DVD is only supported with the IBM 7210 (see next visual) and there is no need to install special software in order to write using the standard UDF format.

In order to boot from a **mksysb** CD or DVD, you need to be sure that your hardware is at the latest firmware level. Procedures for updating pSeries firmware is covered in the *AU16 AIX System Administration II: Problem Determination* course.

Backup device – 7210 external DVD-RAM drive

- Writes DVD-RAM media
- Reads DVD media in 2.6 GB, 4.7 GB, 5.2 GB, and 9.4 GB
- Supports CD-ROM media in Modes 1 or 2, XA, and CDDA and audio formats
- Reads multi-session disks, CD-R, CD-ROM, and CD-RW disks
- Loading tray accommodates 8 cm and12 cm media
- SCSI attachable

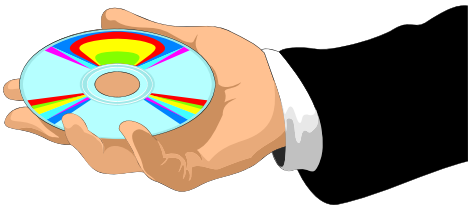


Figure 13-8. Backup device - 7210 external DVD-RAM drive AU1412.0

Notes:

Description

The IBM 7210 External DVD-RAM Drive Model 025 is a DVD-RAM drive designed to provide a high performance storage solution. This self-powered stand-alone drive is designed for the open systems environment, which includes the IBM iSeries, pSeries, AS/400, and RS/6000 servers.

SMIT backup menus

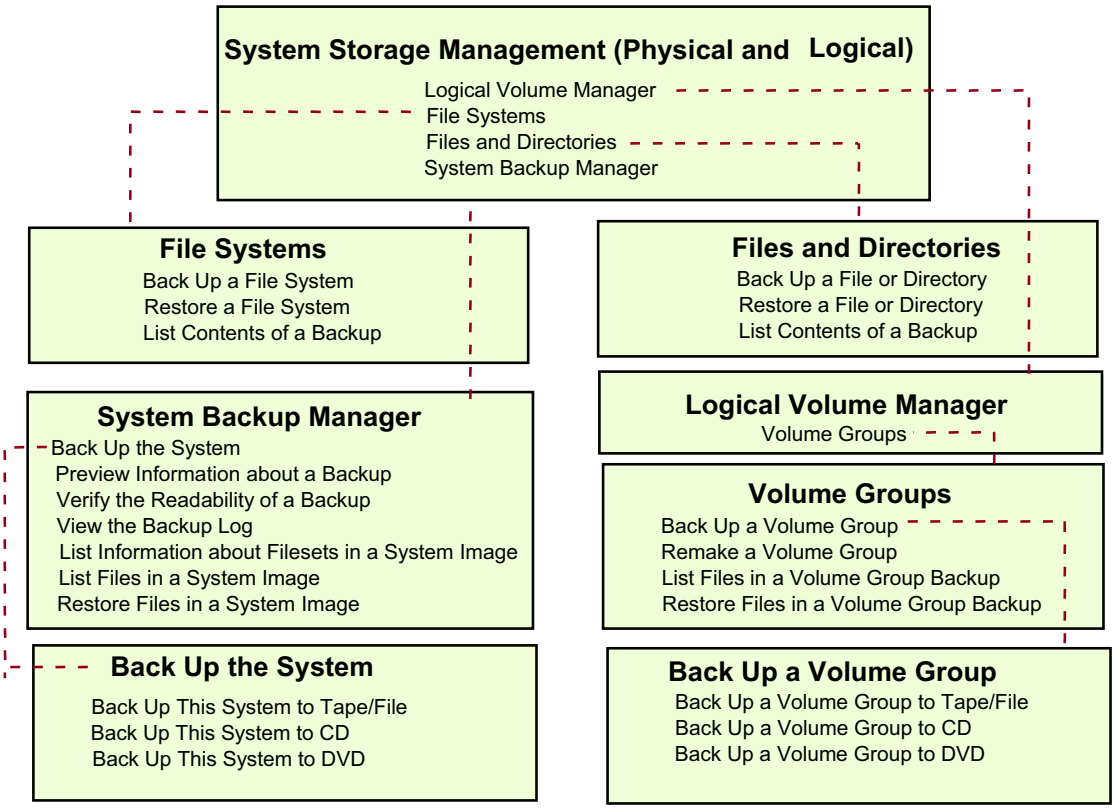


Figure 13-9. SMIT backup menus AU1412.0

Notes:

Backup menus

The visual shows the SMIT menus that have backup options. Note that the menus in the visual do not show all the options for a given menu, only those that pertain to backups.

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 13-10. **rootvg backup process -mksysb** AU1412.0

Notes:

What does `mksysb` do?

The **`mksysb`** utility 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 installation time
- Saves the file system block size and number of bytes per inode
- 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.

System backup or clone?

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 does not install correctly on a multiprocessor machine because they use different AIX filesets. However, this is an easy problem to resolve. In additional to the **mksysb** tape, you 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 is automatically selected and loaded from the CD.

Non-interactive installation

If a system backup is being made to install another system or to reinstall the existing system, a customer can predefine installation 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 installation through taking a snapshot of the current system and its customized state.

System backup components

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 installation routines.

/image.data file for rootvg

```
image data:
                IMAGE_TYPE= bff
                DATE_TIME= Fri Nov 29 10:23:36 NFT 2007
                UNAME_INFO= AIX ibm150 2 5 00428DFB4C00
                PRODUCT_TAPE= no
                USERVG_LIST=
                PLATFORM= chrp
                OSLEVEL= 6.1.0.0
                CPU_ID= 00428DFB4C00
  logical_volume_policy:
                SHRINK= no
                EXACT_FIT= no
  ils_data:
                LANG= en_US
  #Command used for vg_data, /usr/sbin/lsvg
  vg_data:
                VGNAME= rootvg
                PPSIZE= 16
                VARYON= yes
                VG_SOURCE_DISK_LIST= hdisk0
                BIGVG= no
                TFACTOR= 1
  #Command used for source_disk_data: /usr/sbin/bootinfo
  source_disk_data: (stanza is repeated for each disk in rootvg)
                PVID=(physical volume id)
                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 13-11. **/image.data file for rootvg** AU1412.0

Notes:

/image.data file creation

The **/image.data** file has information used by the BOS installation 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 you edit the **image.data** file, then you 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, **lsfs** 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.

You can provide additional processing (if required) after the BOS installation by using the BOSINST_FILE= field in the post_install_data stanza or through your own program. The BOSINST_FILE and SHRINK= fields must be edited by the user before calling **mksysb** if changes are desired.

logical_volume_policy stanza

The logical_volume_policy stanza contains information to be used at reinstallation 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 stanza

The vg_data stanza contains information about the volume group. The VG_SOURCE_DISK_LIST= field specifies the disks that BOS installation uses on a best effort basis to place the volume group. If the EXACT_FIT= field is set to YES, the user is warned before installation begins.

lv_data stanza

The lv_data stanza 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 are placed in this stanza in AIX V4.2 and later.

/bosinst.data file for rootvg

```
control_flow:
          CONSOLE = Default
          INSTALL_METHOD = overwrite
          PROMPT = yes
          EXISTING_SYSTEM_OVERWRITE = yes
          INSTALL_X_IF_ADAPTER = yes
          RUN_STARTUP = yes
          RM_INST_ROOTS = no
          ERROR_EXIT =
          CUSTOMIZATION_FILE =
          TCB = no
          INSTALL_TYPE =
          BUNDLES =
          RECOVER_DEVICES = Default
          BOSINST_DEBUG = no
          ACCEPT_LICENSES =
          DESKTOP = CDE
          INSTALL_DEVICES_AND_UPDATES = yes
          IMPORT_USER_VGS =
          ENABLE_64BIT_KERNEL = no
          CREATE_JFS2_FS = no
          ALL_DEVICES_KERNELS = yes
          (some bundles ....)
target_disk_data:
          LOCATION =
          SIZE_MB =
          HDISKNAME =
locale:
          BOSINST_LANG =
          CULTURAL_CONVENTION =
          MESSAGES =
          KEYBOARD =
```


Figure 13-12. **/bosinst.data file for rootvg** AU1412.0

Notes:

/bosinst.data creation

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 in the visual has been condensed to highlight key areas. The actual file is well documented with comments contained within the file.

control_flow stanza

The control_flow stanza contains variables that control the way the installation program works.

The field definitions are:

- 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 installation program overwrites 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), personal (personal workstation configuration), and CC_EVAL (enables CAPP and EAL4+ technology). 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.
- DESKTOP specifies the desktop to be installed. Choices include CDE (the default), NONE, GNOME, and KDE. If you choose GNOME or KDE, you install open-source software.
- INSTALL_DEVICES_AND_UPDATES does additional installations. When installing a **mksysb** image to a system with a different hardware configuration, boot from product media to get any missing device drivers installed. In addition, if the product media is a later level of AIX than the **mksysb**, software in the **mksysb** image is updated. To prevent either of these additional installations from occurring, set this field to no. The default is yes.
- IMPORT_USER_VGS specifies whether you want any user volume groups to be automatically imported after the system has been installed.
- ENABLE_64BIT_KERNEL specifies whether you want to enable the 64-bit kernel.
- CREATE_JFS2_FS specifies whether you want to create enhanced journaled file systems.
- ALL_DEVICES_KERNELS specifies whether to install all device and kernel filesets.

target_disk_data stanza

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.

Multiple target_disk_data stanzas can exist. They define the disks that are to contain the **root** volume group. Only one field (PVID, PHYSICAL_LOCATION, SAN_DISKID, CONNECTION, LOCATION, SIZE_MB, HDISKNAME) must be non-null for BOS installation to choose a disk. The order of precedence is PVID, PHYSICAL_LOCATION, SAN_DISKID, then CONNECTION, then LOCATION, then SIZE_MB, and then HDISKNAME.

The field definitions are:

- 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.
- PVID specifies the physical volume identifier for the disk.
- PHYSICAL_ LOCATION provides a way to identify fibre channel disks during BOS Install. The information in the PHYSICAL_LOCATION field supersedes the information in the SAN_DISKID field.
- SAN_DISKID specifies the World Wide Port Name and a Logical Unit ID for fibre channel-attached disks.

- `CONNECTION` specifies the combination of the parent attribute and the connwhere attribute associated with a disk.

`locale` stanza

- The `locale` stanza contains variables for the primary language the installed machine uses
- `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

Customizing the bosinst.data file

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 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, you must also have on your diskette a file called **signature**. The file **signature** must contain the word data.

Unattended reinstallation

With both the /**image.data** and the /**bosinst.data** files created, the reinstallation of AIX V4 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 the command:

   `ls ./bosinst.data ./signature | backup -iqv`

   This assumes there is already a preformatted diskette in the drive.

4. Run the command **`mksysb /dev/rmt0.1`**

   This assumes there is a tape in the first tape drive and that it is 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 installation media, the BOS installation program uses the diskette file rather than the default /**bosinst.data** file shipped with the installation 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 installation by typing **`000 <Enter>`** when you see the startup symbols **`\ | /`** on the display.

rootvg - Back Up the System

```
# smit sysbackup
```

```
                    Back Up the System

Move cursor to desired item and press Enter.

Back Up This System to Tape/File
Back Up This System to CD
Create a Generic Backup CD or DVD
Back Up This System to DVD




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


Figure 13-13. **rootvg - Back Up the System** AU1412.0

Notes:

Location of the backup

In AIX 5L and later, you can use **`smit sysbackup`** to preselect if you want to back up the system (**rootvg**) to Tape/File, CD or DVD. On the following visuals you see:

- Back up the System to Tape/File
- Back up a Volume Group to Tape/File
- Restore the System from Tape
- Restore a Volume Group from Tape
- Back up the System to CD
- Back up the System to ISO9660 DVD
- Back up the System to UDF DVD
- Back up a Volume Group to CD
- Back up a Volume Group to ISO9660 DVD
- Back up a Volume Group to UDF DVD

rootvg - Back Up This System to Tape/File

```
# smit mksysb
```

```
                 Back Up This System to Tape/File

Type or select values in entry fields.
Press Enter AFTER making all desired changes.
                                                      [Entry Fields]
    WARNING:  Execution of the mksysb command will
              result in the loss of all material
              previously stored on the selected
              output medium. This command backs
              up only rootvg volume group.

* Backup DEVICE or FILE                              []           +/
  Create MAP files?                                   no          +
  EXCLUDE files?                                      no          +
  List files as they are backed up?                   no          +
  Verify readability if tape device?                  no          +
  Generate new /image.data file?                      yes         +
  EXPAND /tmp if needed?                              no          +
  Disable software packing of backup?                 no          +
  Backup extended attributes?                         yes         +
  Number of BLOCKS to write in a single output        []            #
     (Leave blank to use a system default
  Location of existing mksysb image                   []            /
  File system to use for temporary work space         []            /
     (If blank, /tmp will be used.)
  Backup encrypted files?                             yes           +
  Back up DMAPI filesystem files?                     yes           +
[BOTTOM]
```


Figure 13-14. **rootvg - Back Up This System to Tape/File** AU1412.0

Notes:

Introduction

The SMIT screen shown in the visual, **Back Up the System**, only backs up mounted file systems in **rootvg**. Use one of the other **backup** commands to backup other volume groups.

Create MAP files?

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 excludes 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 see a percentage-completed progress message while the backup is created. This option is supported at AIX V4.2 and later.

Verify readability if tape device?

Verifies the file header of each file on the backup tape and report any read errors as they occur.

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.

Backup extended attributes?

By default, the **`mksysb`**, **`savevg`** and **`backup`** utilities save any extended attributes. If you plan to restore to a back-level system which does not understand the format with extended attributes, then this option allows you to override that default behavior.

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.

Location of existing mksysb image

Specifies the full path name to the location of a previously-created mksysb image that can be used to create a bootable tape backup.

File system to be used for temporary work space

Specifies the full path name to the location of a directory or file system to be used as temporary space to create a bootable tape backup. The file system used must have at least 100MB of available free disk space for the creation of the bootable image. If this field is left blank, the /tmp file system is used.

Backup encrypted files?

Specifies if encrypted files should be backed up. AIX 6.1 introduces the ability to encrypt files on a per file basis without the need of third party tools.

Backup DMAPI file system files?

Specifies if DMAPI file system files are to be backed up.

mksysb image

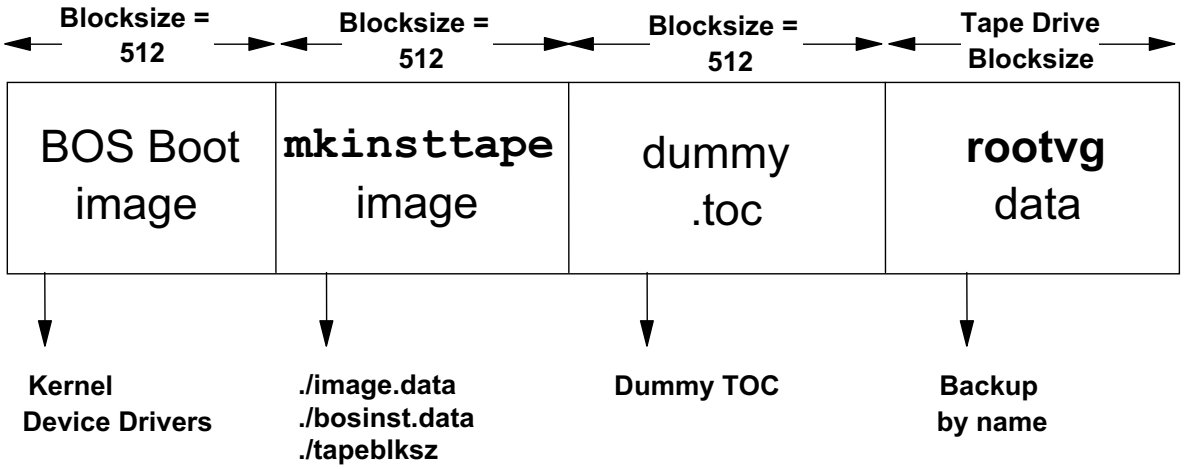


Figure 13-15. mksysb image AU1412.0

Notes:

Introduction

This visual shows the tape layout of a **mksysb** image.

BOS boot image

The BOS boot image contains a copy of the system's kernel and device drivers needed to boot from the tape.

mkinsttape image

The **mkinsttape** image contains two files:

- **./image.data** holds the information needed to re-create the root volume group and its logical volumes and file systems.

- **./bosinst.data** contains the customizable installation procedures and dictates how the BOS installation program behaves. This file allows for the non-interactive installations.
- **./tapeblksz** contains the block size setting of the tape drive used during the backup.

Dummy TOC

The dummy TOC is used to make **mksysb** tapes have the same number of files as the BOS installation tapes.

rootvg data

The **rootvg** data contains all the data from the backup. This data is saved using the **backup** command which is discussed shortly.

Back Up a Volume Group

```
# smit vgbackup
```

```
                    Back Up a Volume Group

Move cursor to desired item and press Enter.

  Back Up a Volume Group to Tape/File
  Back Up a Volume Group to CD
  Back Up a Volume Group to DVD


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


Figure 13-16. Back Up a Volume Group AU1412.0

Notes:

Back Up a Volume Group SMIT screen

In AIX 5L and later, you can use **`smit vgbackup`** to preselect if you want to back up a **non-rootvg** volume group to Tape/File, CD or DVD.

Back Up a Volume Group to Tape/File

```
# 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      +
  Backup extended attributes?                           yes     +
  Number of BLOCKS to write in a single output          []       #
     (Leave blank to use a system default)
  Verify readability if tape device                     no      +
   Backup Volume Group information files only?          no      +
  Backup encrypted files?                               yes     +
  Back up DMAPI filesystem files?                       yes     +
[BOTTOM]
```


Figure 13-17. Back Up a Volume Group to Tape/File AU1412.0

Notes:

Backing up rootvg versus a non-rootvg volume group

The **Back Up a Volume Group to Tape/File** SMIT screen looks very similar to the **Back Up the System** SMIT screen. This is because they are both performing a volume group backup except the **Back Up the System** SMIT screen is using the **mksysb** command to create bootable images. The **Back Up the System** SMIT screen is using the **savevg**. command.

Some of the differences between the **Back Up the System SMIT** screen and the **Back Up a Volume Group to Tape/File** SMIT screen are:

- **VOLUME GROUP to back up**

  Enter the name of the volume you want to back up.

  A new **vg.data** file is generated. This file is equivalent to the **image.data** file for **rootvg**. Unless you have a customized file that you want to use, let SMIT (using **savevg**) create this file for you. The file is called

/**tmp**/**vgdata**/***vg_name***/***vg_name***.**data**. This file can also be created by running the **mkvgdata** ***vg_name*** command.

- **EXCLUDE files?**

This option allows you exclude files (during the backup) located in mounted file systems within the volume group. It creates a file called /**etc**/**exclude.*vg_name*** and add the list of filenames that are not wanted.

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.

Backup extended attributes?

By default, the **mksysb**, **savevg** and **backup** utilities saves any extended attributes. If you plan to restore to a back-level system which does not understand the format with extended attributes, then this option allows you to override that default behavior.

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.

Verify readability if tape device?

To attempt to read backup image from the tape and report any read errors if they occur, “select yes”.

Back up Volume Group information files only?

This backs up files such as /**tmp**/**vgdata**/**vgname**/**vgname.data** and map files if any exist. This does not back up user data files.

Backup encrypted files?

Specifies if encrypted files should be backed up. AIX 6.1 introduces the ability to encrypt files on a per file basis without the need of third party tools.

Backup DMAPI file system files?

Specifies if DMAPI file system files are to be backed up.

Restoring a `mksysb` (1 of 2)

- Boot the system in install/maintenance mode:

```
                 Welcome to Base Operating System
                 Installation and Maintenance

        1        Start Install Now With Default Settings
        2        Change/Show Installation Settings and Install
>>      3        Start Maintenance Mode for System Recovery
        4        Configure Network Disks (iSCSI)
```

```
                 Maintenance

        1        Access A Root Volume Group
        2        Copy a System Dump to Removable Media
        3        Access Advanced Maintenance Functions
        4        Erase Disks                 ....
>>      6        Install from a System Backup
```

```
                 Choose Tape Drive

                 Tape Drive                          Path Name
>>      1        tape/scsi/4mm/2GB          /dev/rmt0
```


Figure 13-18. Restoring a `mksysb` (1 of 2) AU1412.0

Notes:

Start a `mksysb` restoration

To restore a **`mksysb`** image, boot the machine just as if you were performing an installation. Be sure your 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 boots from the tape and prompts you to define the console and select a language for installation. Once you have answered those questions, then the **Installation and Maintenance** menu is presented.

You can also boot from an installation CD. The CD presents the same screens. Just be sure to put the **`mksysb`** tape in the tape drive before answering the last question.

Select **Start Maintenance Mode for System Recovery**, then **Install from a System Backup** and select the tape drive that contains the **`mksysb`** tape.

Restoring a `mksysb` (2 of 2)

Welcome to Base Operating System
Installation and Maintenance

```
Type the number of your choice and press Enter.  Choice is indicated by >>.
        1          Start Install Now With Default Settings
>>      2          Change/Show Installation Settings and Install
        3          Start Maintenance Mode for System Recovery
        4          Configure Network Disks (iSCSI)
```

```
                 System Backup Installation and Settings

Type the number of your choice and press Enter.

        1          Disk(s) where you want to install        hdisk0
        2          Use Maps                                 No
        3          Shrink Filesystems                       No
        0          Install with the settings listed above
```


Figure 13-19. Restoring a `mksysb` (2 of 2) AU1412.0

Notes:

Changing installation settings

After selecting the tape drive (and a language, which is not shown on the visuals), you return to the **Installation and Maintenance** menu. Now select option **2, Change/Show Installation Settings and Install**.

The options from the **System Backup and Installation and Settings** menu are:

- **1 Disk(s) where you want to install**

  Select all disks where you want to install. If your **rootvg** was mirrored, you need to select both disks.

- **2 Use Maps**

  The option **Use Maps** lets you use the map file created (if you created one) during the backup process of the `mksysb` tape. The default is **no**.

- **3 Shrink Filesystems**

  The option **Shrink Filesystems** installs the file systems using the minimum required space. The default is **no**. If **yes**, all file systems in **rootvg** are shrunk. So remember after the restore, evaluate the current file system sizes. You might need to increase their sizes.

- **0 Install with the settings listed above**

  At the end, select option **0** which installs using the settings selected. Your **`mksysb`** image is restored.

The system then reboots.

Note: The total restore time varies from system to system. A good rule of thumb is twice the amount of time it took to create the **`mksysb`**.

Remake/Restore a non-rootvg volume group

```
# smit restvg
                         Remake a Volume Group

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


                                                [Entry Fields]
* Restore DEVICE or FILE                        [/dev/rmt0] +/
  SHRINK the filesystems?                        no         +
  Recreate logical volumes and filesystems only  no         +
  PHYSICAL VOLUME names                         []          +
     (Leave blank to use the PHYSICAL VOLUMES listed
      in the vgname.data file in the backup image)
  Use existing MAP files?                        yes        +
  Physical partition SIZE in megabytes          []          +#
     (Leave blank to have the SIZE determined
      based on disk size)
  Number of BLOCKS to read in a single input    []           #
     (Leave blank to use a system default)
  Alternate vg.data file                        []           /
    (Leave blank to use vg.data stored in
     backup image)


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


Figure 13-20. Remake/Restore a non-rootvg volume group AU1412.0

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 the **PHYSICAL VOLUME names** field is left blank, the volume group goes 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 already exist, they are used by default during recovery. If you don't want to use them, set this selection to **no**.

Physical partition SIZE in megabytes

The physical partition size is determined based on disk size. This characteristic makes it easy to resize the partitions in a volume group. If you want to move the volume group to a larger disk, the physical partition adjusts automatically during the restore.

AIX properly sizes the physical partitions for the disk it is using. If you prefer to have a larger physical partition size than the standard, you can set it here. If, for example, you have a 4.5 GB drive, the partition size is 8 MB. If you want it to be 16 MB, you can set it here.

mksysb - ISO9660 burn image

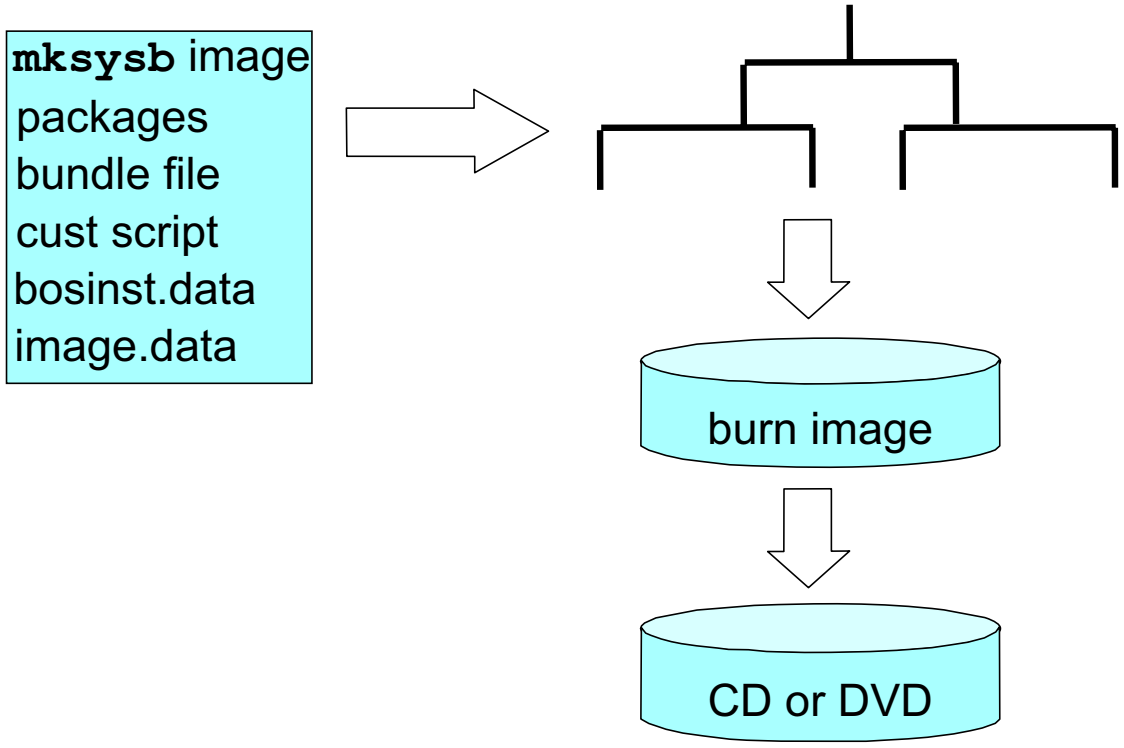


Figure 13-21. mksysb - ISO9660 burn image AU1412.0

Notes:

What is stored?

When creating a system backup on CD or DVD, it is actually creating a file system on the disk. Within the file system, many things are stored.

The **mksysb** image file itself is stored (in backup format).

The files that would normally be placed in the second record of a **mksysb** tape also need to be stored: **bosinst.data** and **image.data**.

If you want to be able to install additional software during the restore (such as device drivers) you can place them in this file system as packages or additionally defined as bundles.

Finally, you may want to run a customization script after the image restore to do additional configuration.

When burning the file system onto a CD or DVD, using the ISO9660 standard, you need to first build a burn image on the hard drive. Then, you need to actually burn that to the disk.

When using ISO9660, you need to identify:

- Where to store the **mksysb** image
- Where to build the file structure
- Where to build the burn image

mksysb - UDF DVD

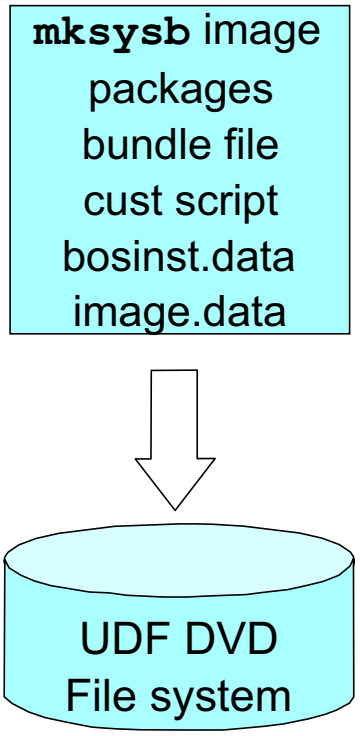


Figure 13-22. mksysb - UDF DVD AU1412.0

Notes:

Advantages of a UDF DVD file system

The Universal Disk Format (UDF) file system on a DVD allows you to write to the DVD as a mounted file system thus avoiding the need to first build a burn image on your hard drive.

While you are still storing the same kind of information in a file structure, the directory tree is built directly on the DVD.

As a result, you do not need to identify any file systems on the hard drives.

The only item that needs to be pre-built before it is written to the DVD is the **mksysb** image file itself.

rootvg - Back Up This System to CD (ISO9660)

```
# smit mkcd
```

```
                              Back Up This System to CD

Type or select values in entry fields.
Press Enter AFTER making all desired changes.
                                                      [Entry Fields]
  CD-R Device                                        []         +
  mkysyb creation options:
     Create map files?                                no        +
     Exclude files?                                   no        +
  Disable software packing of backup?                 no        +
  Backup extended attributes?                         yes       +
  File system to store mksysb image                  []              /
  File system to store CD file structure             []              /
  File system to store final CD images               []              /
  If file systems are being created:
    Volume Group for created file systems            [rootvg]   +
  Advanced Customization Options:
  Do you want the CD to be bootable?                  yes       +
  Remove final images after creating CD?              yes       +
  Create the CD now?                                  yes       +
  Install bundle file                                []              /
  File with list of packages to copy to CD           []          /
  Location of packages to copy to CD                 []         +/
  Customization script                               []              /
  User supplied bosinst.data file                    []              /
  Debug output?                                       no        +
  User supplied image.data file             []            /
  Backup encrypted files?                             yes
  Back up DMAPI filesystem files?                     Yes
[BOTTOM]
```


Figure 13-23. **rootvg - Back Up This System to CD (ISO9660)** AU1412.0

Notes:

Specifics for backups in ISO9660 format

Backup volume groups in ISO9660 format on CD or DVD-RAM require a significant amount of space. When you use the **`smit mkcd`** fastpath (which uses the **`mkcd`** command), it allows you to specify where you want to create the various structures and images needed to:

- Create backup image
- Create CD file system and copy backup to it
- Create CD image on hard disk
- Burn to media

Be sure you have sufficient space in the selected file systems to hold the pre-burn data.

rootvg - Back Up This System to ISO9660 DVD

 # smit mkdvd -> Select 1 ISO9660 (CD format)

```
                          Back Up This System to ISO9660 DVD

Type or select values in entry fields.
Press Enter AFTER making all desired changes.
                                                          [Entry Fields]
  DVD-R or DVD-RAM Device                                 []          +
  mkysyb creation options:
    Create map files?                                      no         +
    Exclude files?                                         no         +
  Disable software packing of backup?                      no         +
  Backup extended attributes?                              yes        +
  File system to store mksysb image                       []       /
  File system to store DVD file structure                 []       /
  File system to store final DVD images                   []       /
  If file systems are being created:
    Volume Group for created file systems                 [rootvg]    +
  Advanced Customization Options:
  Do you want the DVD to be bootable?                      yes        +
  Remove final images after creating DVD?                  yes        +
  Create the DVD now?                                      yes        +
  Install bundle file                                     []          /
  File with list of packages to copy to DVD               []          /
  Location of packages to copy to DVD                     []          +/
  Customization script                                    []          /
  User supplied bosinst.data file                         []          /
  Debug output?                                            no         +
  User supplied image.data file                           []          /
  Backup encrypted files?                                 yes        +
  Back up DMAPI filesystem files?                         yes        +
[BOTTOM]
```


Figure 13-24. **rootvg - Back Up This System to ISO9660 DVD** AU1412.0

Notes:

Overview

The smit fastpath for the panel shown in the visual is **smit mkdvd**. When prompted, choose the ISO9660 option.

The types of information to write to media and the mechanisms are about the same for CD or DVD when using ISO9660 to first build the image and then burn it to the media.

rootvg - Back Up This System to UDF DVD

 # smit mkdvd -> Select 2 UDF (Universal Disk Format)

```
                                Back Up This System to UDF DVD

Type or select values in entry fields.
Press Enter AFTER making all desired changes.
                                                        [Entry Fields]

  DVD-RAM Device                                          []            +
  mkysyb creation options:
    Create map files?                                   no     +
    Exclude files?                                      no     +
  Disable software packing of backup?                   no     +
  Backup extended attributes?                           yes    +
  File system to store mksysb image                    []            /
     (If blank, the file system
       will be created for you.)

  If file system is being created:
    Volume Group for created file system               [rootvg]     +

  Advanced Customization Options:
  Do you want the DVD to be bootable?                       yes         +
  Install bundle file                                      []            /
  File with list of packages to copy to DVD                []            /
  Location of packages to copy to DVD                      []           +/
  Customization script                                     []            /
  User supplied bosinst.data file                          []            /
  Debug output?                                             no           +
  User supplied image.data file                    []            /
  Backup encrypted files?                              Yes         +
  Back up DMAPI filesystem files?                      yes         +
[BOTTOM]
```


Figure 13-25. **rootvg - Back Up This System to UDF DVD** AU1412.0

Notes:

The smit fastpath for the panel shown in the visual is: **smit mkdvd**. When prompted, choose the UDF option.

Backup volume groups in UDF (Universal Disk Format) format on DVD-RAM requires only the space for the backup image:

1. Create backup image
2. Burn to media

This allows modification of files such as **bosinst.data**, **image.data**, and **vgname.data**.

Back Up a Volume Group to CD

```
# smit savevgcd
```

```
                          Back Up a Volume Group to CD

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

                                                [Entry Fields]
  CD-R Device                                          []        +
* Volume Group to back up                              []        +

  savevg creation options:
    Create map files?                                   no       +
    Exclude files?                                      no       +
    Disable software packing of backup?                 no       +
    Backup extended attributes?                         yes      +

  File system to store savevg image                    []          /
  File system to store CD file structure               []          /
  File system to store final CD images                 []          /
  If file systems are being created:
    Volume Group for created file systems              [rootvg]  +

  Advanced Customization Options:
  Remove final images after creating CD?                yes      +
  Create the CD now?                                    yes      +
  Debug output?                                         no       +
  Backup Volume Group information files only?           no       +
  Backup encrypted files?                              Yes         +
  Back up DMAPI filesystem files?                      yes         +
[BOTTOM]

   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. Back Up a Volume Group to CD AU1412.0

Notes:

Overview

You don't have the multiple types of information backed up for a **non-rootvg** volume group as you did for the system backups, but the mechanisms are very similar when using ISO9660.

Back Up a Volume Group to ISO9660 DVD

```
# smit savevgdvd
```

```
                        Back Up a Volume Group to ISO9660 DVD

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

                                                [Entry Fields]

  DVD-R or DVD-RAM Device                                []        +
* Volume Group to back up                                []        +

  savevg creation options:
    Create map files?                                    no        +
    Exclude files?                                        no       +
    Disable software packing of backup?                  no        +
    Backup extended attributes?                          yes       +

  File system to store savevg image                      []         /
  File system to store DVD file structure                []         /
  File system to store final DVD images                  []         /
  If file systems are being created:
    Volume Group for created file systems                [rootvg] +

  Advanced Customization Options:
  Remove final images after creating DVD?                yes       +
  Create the DVD now?                                    yes       +
  Debug output?                                          no        +
  Backup encrypted files?                                 yes      +
  Back up DMAPI filesystem files?                         yes      +
[BOTTOM]

   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. Back Up a Volume Group to ISO9660 DVD AU1412.0

Notes:

Overview

Backing up a **non-rootvg** volume group to ISO9660 DVD is similar to backing up a **rootvg** volume group.

Back Up a Volume Group to UDF DVD

```
# smit savevgdvd
```

```
                        Back Up a Volume Group to ISO9660 DVD

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

                                                [Entry Fields]

  DVD-RAM Device                                          []        +
* Volume Group to back up                                 []        +

  savevg creation options:
    Create map files?                                     no        +
    Exclude files?                                        no        +
    Disable software packing of backup?                   no        +
    Backup extended attributes?                           yes       +

  File system to store savevg image                       []         /
          (If blank, the file system
          will be created for you.)

  If file systems are being created:
    Volume Group for created file systems                 [rootvg]  +

  Advanced Customization Options:
  Debug output?                                           no         +
  Backup Volume Group information files only?             no         +
  Backup encrypted files?                                 yes        +
  Back up DMAPI filesystem files?                         yes        +
[BOTTOM]

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


Figure 13-28. Back Up a Volume Group to UDF DVD AU1412.0

Notes:

Overview

Backing up a **non-rootvg** volume group to UDF DVD is similar to backing up a **rootvg** volume group.

Exercise 13: Using `backup` and `restore` (part 1)

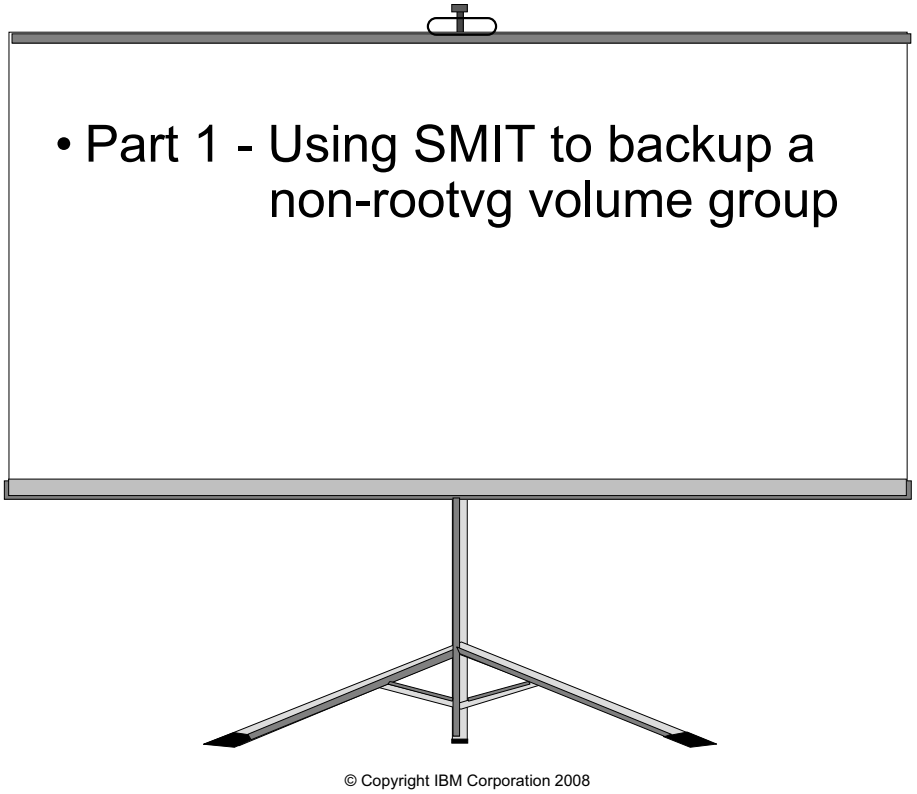


Figure 13-29. Exercise 13: Using `backup` and `restore` (part 1) AU1412.0

Notes:

Introduction

In this exercise, you create a backup of **datavg** and save it to the a file in **rootvg**.

The exercise can be found in your *Student Exercises Guide*.

Back up by filename

```
backup -i [-q] [-v] [-p] [-U] [-Z][-f device]
```

- **-q** Media is ready
- **-v** Verbose - display filenames during backup
- **-p** Pack files which are less than 2 GB
- **-U** Specifies to backup any ACLs
- **-Z** Backs up the Encrypted File System (EFS)

Filenames are read from standard input

Figure 13-30. Back up by filename AU1412.0

Notes:

The `backup` command

The **backup** command is the preferred command for making backups of AIX files and directories. **backup** supports two different methods:

- Backup by filename
- Backup by inode (also call a file system backup)

When performing a backup by filename, the files must be in a mounted file system to be backed up. Backups by inode backup file systems when they are unmounted.

Syntax for the `backup` command by filename

The visual shows the syntax for the **`backup`** command by filename.

- The **`-i`** option is used to indicate a backup by filename.
- 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 -e option allows you to identify a regular expression which used to identify files which you do not want to have packed when using the -p option.
- The **`-v`** option displays the files and pathnames to standard out as they are backed up.
- The -U specifies to backup any ACLs or named extended attributes. Without this option the image will include only AIXC ACLs and PCLs in the archive along with the other regular file data. For files containing NFS4 ACLs, conversion to AIXC happens by default during archival.
- The -Z backs up the Encrypted File System (EFS) information for all of the files, directories, and file systems. The EFS information is extracted by default. Archives created with **-Z** option can be restored only on AIX 6.1 or later releases.

Back up by filename examples

- Example 1: Read input from a file

```
# cat listfile
/home/roy/file1
/home/roy/file2
/home/roy/file3
# backup -iqvf /dev/rmt0  < listfile
```

- Example 2: Use **find** to generate list

```
# find /home/roy | backup -iqvf  /dev/rmt0
# cd  /home/roy
# find . | backup -iqvf /dev/rmt0
```

Relative versus full filenames will impact location of files on recovery!

Figure 13-31. Back up by filename examples AU1412.0

Notes:

Introduction

The list of files **backup** uses can be supplied by a file or by commands. The visual provides a sample of each.

Example 1

In the first example, the file **listfile** contains the files you want to back up. That is fed into the **backup** command by using a redirection (**<**).

Example 2

In the second example, there are two examples that can be used to back up the same data using the **find** command to generate the file list. Both commands back up the files stored in /**home**/**roy**. Even though both **find** examples save the same data, the filenames will be stored differently.

Types of pathnames

There are two types of filenames:

- Relative
- Full (or absolute)

The difference is that 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 is important when you try to recover the data.

Full pathname backups restore to the same location in the directory structure since their position is referenced from the **root** directory. But, a relative pathname file is 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 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            +
  Backup extended attributes?                  yes           +
  Back up EFS Attributes?                      Yes           +


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


Figure 13-32. Backup a File or Directory AU1412.0

Notes:

FILE or DIRECTORY to backup

This is a parameter for the **find** command that runs 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 are from the local system only.

Back up a file system by inode

Syntax:
backup [-u] [-*level*] [-f *device*] *filesystem*

- Levels provide incremental backups:
  - **-0** Full file system back up
  - **-1**,**-2**, etc Backup changes since *level* -1
  - **-u** Updates **/etc/dumpdates**
    (**/etc/dumpdates** contains a backup history)

```
# backup  -u -1 -f /dev/rmt0  /home
```


Figure 13-33. Back up a file system by inode AU1412.0

Notes:

How to backup by inode

If you do not specify the **-i** option, the **backup** command performs a file system backup by inode.

To insure integrity of the backup, you must unmount the file system before backing up by inode. If the file system is mounted, the utility gives a warning, though you can choose to backup anyway. Unmounting the file system is strongly recommended for user-created file systems and system file systems (other than /), otherwise errors in mapping when restoring 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** / is 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.

`-u` option

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.

`-level` option

The `-level` option allows you to perform incremental backups. The `-0` level backs up all files in the file system. The `-1` level backs up all files changed since the last `-0` backup, and so on. (If you do not specify a level, `-9` is assumed.)

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 13-34. Incremental backup example AU1412.0

Notes:

Defining the level of backup

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.

Calendar example

The visual shows a calendar that describes how different levels of backups 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 backs up all files modified since the level 0 backup on the previous Friday. Each level 6 backup Tuesday through Thursday backs up 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 backs up all files modified since the level 0 backup. The remaining level 3 backups backs up all those files modified since the last level 0 backup.
- During the second full week, the level 6 backups on Monday through Thursday back up all files that were modified since the last level 3 backup. The level 3 backups on Fridays backs up everything since the last level 0.

Back up a file system by inode using SMIT

```
# 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           +
* Backup extended attributes?              yes          +
* Backup EFS attributes?                   yes          +



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


Figure 13-35. Back up a file system by inode using SMIT AU1412.0

Notes:

Using SMIT to backup by inode

SMIT provides a front-end for file system backups as well.

Each line represents the options from the command line.

`restore` command (1 of 2)

- 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 13-36. `restore command (1 of 2)` AU1412.0

Notes:

Listing files on the backup

The **`restore`** command is used to restore data backed up with the **`backup`** command. **`restore -T`** shows the contents of the media and display the list of files.

Restoring individual files

`restore -x` can be used to restore selective files from the backup. The file names and paths on the backup are preserved on the restore. If the backup was created with relative path names, then the files are 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 file system backup.

Restoring a complete file system

`restore -r` works with backups by inode. It ensures 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 are 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 goes 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` command (2 of 2)

- Restores the file attributes without restoring the file contents:

```
restore -Pstring  [-q] [-v] [-f device] [file1 file2 ...]
```

string can be:

A	Restore all attributes
a	Restore only the permissions of the file
o	Restore only the ownership of the file
t	Restore only the timestamp of the file
c	Restore only the ACL attributes of the file

- To restore only the permissions of the file **/etc/passwd** from the archive:

```
# restore -Pa -vf /dev/rmt0 ./etc/passwd
```

- To display only the permissions of the file **/etc/passwd** on the archive:

```
# restore -Ta -vf /dev/rmt0 ./etc/passwd
```


Figure 13-37. `restore command (2 of 2)` AU1412.0

Notes:

Restoring file attributes

The **`restore -Pstring`** option restores only the file attributes. It does not restore the file contents. This flag restores file attributes selectively depending on the flags specified in the ***`string`*** parameter. The ***`string`*** parameter can be a combination of the following characters:

A	Restore all attributes
a	Restore only the permissions of the files
o	Restore only the ownership of the files
t	Restore only the timestamp of the files
c	Restore only the ACL attributes of the files

The -***`Pstring`*** option can be used with both file name and file system archives.

These options are only available on AIX 5L V5.2 and later.

Restore a File or 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
  Restore Extended Attributes?                     yes         +


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


Figure 13-38. Restore a File or Directory AU1412.0

Notes:

Restore SMIT screens

To restore a file or directory, you can use the SMIT screen shown in the visual. The fastpath is **`smit restfile`**.

There is another SMIT screen (fastpath **`smit restfilesys`**), which 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 13: Using `backup` and `restore` (parts 2, 3 and 4)

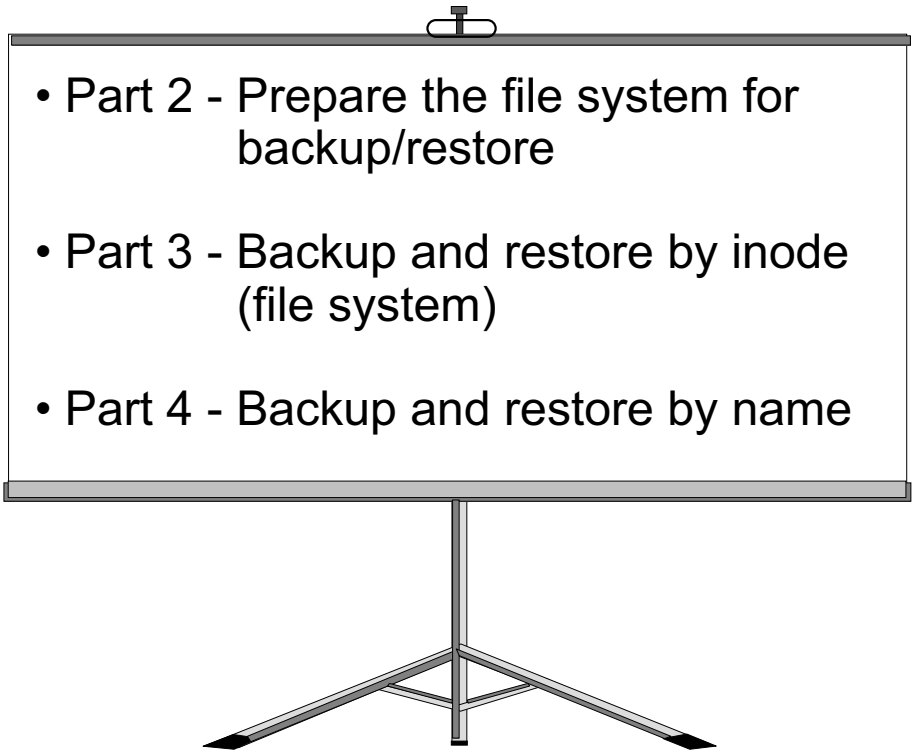


Figure 13-39. Exercise 13: Using `backup` and `restore` (parts 2, 3, and 4) AU1412.0

Notes:

Introduction

This lab allows you to perform backups and recoveries using the AIX tools, **`backup`** and **`restore`**. It gives you an opportunity to perform a backup by name and a backup by inode.

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

Other UNIX backup commands

- **`tar`** (tape archive)
  - Widely available
  - Good for transfer of data between platforms
- **`cpio`** (copy input to output)
  - Widely available
  - Difficulties can occur with many symbolic links
- **`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 13-40. Other UNIX backup commands AU1412.0

Notes:

Why need backup commands?

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 by default. The tar and cpio commands must use the “-U” option to backup and restore ACLs. If you are using ACLs, a good practice is to use **`backup`** or this element of security may get lost during the backup or restore.

AIX does support other generic UNIX backup tools. For backups that are recovered on another UNIX operating system, these tools would need to be used since only AIX supports **`backup`** and **`restore`**.

The `tar` and `cpio` commands

The `tar` command is widely used throughout UNIX and is supported on AIX as well. The `cpio` command is also widely used and is also in AIX. Both commands must use the “-U” option to support ACLs. Also, `cpio` has difficulties following symbolic links. It may not have enough memory to follow the link and the link is lost in the backup.

The `dd` command

The `dd` command is used to copy and convert data byte-by-byte.

The `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 13-41. The `tar` command AU1412.0

Notes:

Introduction

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
-U allows archival and extraction of Extended Attributes.

Generating a backup using `tar`

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 does recursive backups. In the

example, /**home** is the starting point for the **tar** command. It backs up all of /**home** and its subdirectories, and so on.

Restoring a backup using tar

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 in the visual shows the recovery of the /**home**/**team01**/**mydir** directory. If no file is named, then the entire **tar** image is restored.

The **tar** command has been modified to exit now with error when trying to extract a file that is not part of the **tar** archive. For example:

```
# tar -xvf /dev/rmt0 aaa bbb ccc
File aaa not present in the archive.
File bbb not present in the archive.
File ccc not present in the archive.

# echo $?
3
```

Listing the contents of a tar backup

To verify the **tar** image, use **-t**. In the example, the content of **rmt0** is displayed. With **-t**, no files are actually recovered.

AIX enhancements

AIX has provided some very nice enhancements to the **tar** utility.

Here is a list of the new options:

-D	Suppress recursive processing (only current directory)
-R	Use recursive processing (default)
-L <*filename*>	Input list of filenames to process
-x	Exclude list of files or directories to not be copied
-U	Allows archival and extraction of Extended Attributes

The cpio command

- Generate a **cpio** backup:

```
# find   /home | cpio -ov> /dev/rmt0
```

- Restore from a **cpio** image:

```
# cpio -idv </dev/rmt0
```

- List (verify) the contents of a **cpio** image:

```
# cpio -itv < /dev/rmt0
```


Figure 13-42. The cpio command AU1412.0

Notes:

Introduction

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.
- **-U** Performs archival and extraction of ACL and Extended attributes.

Generate a `cpio` backup

`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 (**>**).

To create the `cpio` image in the example in the visual, the `find` command recursively lists all files in the /**home** directory. `cpio` then creates its output, `-o`, on /**dev**/**rmt0**.

Restore from a `cpio` backup

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.

List the contents of a `cpio` backup

To verify or read the content of the `cpio` image, use the `-t` option.

The dd command

- The **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 13-43. The dd command AU1412.0

Notes:

Introduction

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=	Designates the conversion to be done

Examples

In the first example in the visual, the file **/etc/inittab** is copied to the floppy diskette.

In the second example in the visual, the file **text.ascii** is converted into EBCDIC and is written to a file called **ebcdic.text**.

In the last example in the visual, 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.

Copying specific blocks

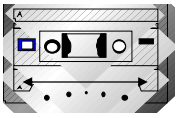
The **dd** command is also useful when you need to copy specific blocks of data. For example, if a file systems superblock (stored in the first block of the file system) is corrupt, a copy is kept at the 31st block. The **dd** command can copy that 31st block back to the first to repair the file system. 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 13-44. Controlling the tape AU1412.0

Notes:

The `tctl` command

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 positions. To ensure you position it correctly, be sure to document the content and order of the data on the tape.

Some of the **`tctl`** options are:

- The **`rewind`** option is generally the first place to start. This ensures you start from the beginning.
- The **`fsf`** option moves the tape forward. It counts end-of-file markers. In the example in the visual, **`fsf 3`** positions the tape to the beginning of the fourth file.
- The **`offline`** and **`rewoffl`** options eject the tape.

The `restore` command

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 in the visual, the fourth file on the tape is read and the **./etc/inittab** file is restored.

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.

The `tcopy` command

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!

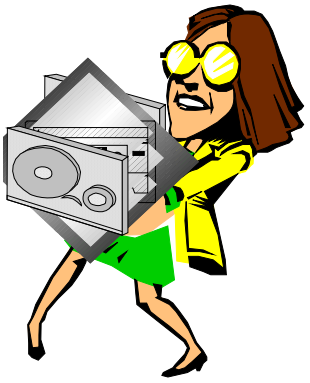


Figure 13-45. Good practices AU1412.0

Notes:

Verify your backups

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 the tape device

The **`tapechk`** command can be used to check a number of files on a tape. If no argument is specified, then the first block on the tape is checked. If a number is specified, that number of files are checked. You can also position the tape before **`tapechk`** is run by specifying a second number. For example, **`tapechk 2.1`** reads two files after skipping past the first file.The **`tapechk`** command can be used to detect malfunctioning hardware.

Keep old backups

Keep old backups in case something goes wrong with the new ones.

Offsite secure storage

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 recovery procedures

Test your recovery procedure before you have to. Know that you can recover before you have to recover.

Checkpoint

1. What is the difference between the following two commands?
   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?

   _____
   _____

3. Which command could you use to restore these files?

   _____

4. True or False? **smit mksysb** backs up all file systems, provided they are mounted. _____
   _____

Figure 13-46. Checkpoint AU1412.0

Notes:

Exercise 14: (optional) Using `tar` and `cpio`

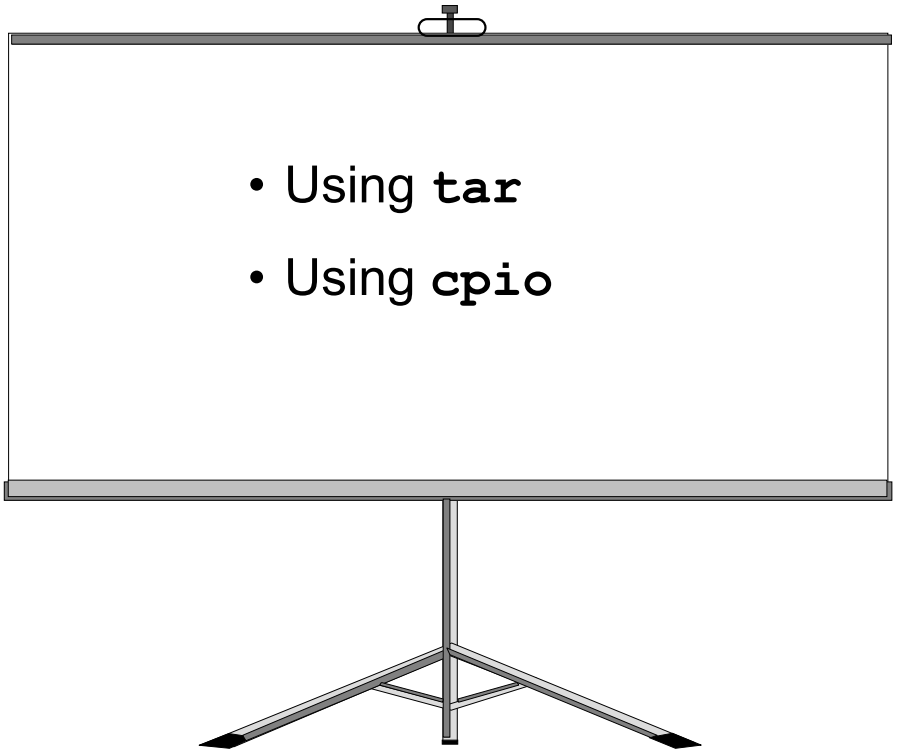


Figure 13-47. Exercise 14: (optional) Using `tar` and `cpio` . AU1412.0

Notes:

Introduction

This is an optional exercise. The instructor determines 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 *Student Exercise Guide*.

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 13-48. Unit summary AU1412.0

Notes:

Unit 14. Security and user administration

What this unit is about

This unit explains key concepts related to AIX users and groups, and also describes 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 explain how and when these should be allocated on the system
- Describe ways of controlling **root** access on the system
- Explain the uses of SUID, SGID, and SVTX permission bits
- Administer user accounts and groups
- Identify the data files associated with users and security

How you will check your progress

Accountability:

- Checkpoint questions
- Exercise

References

SG24-7424	*AIX 6.1 Advanced Security Features Introduction and Configuration*
Online	*AIX 6.1 Security Guide*

Note: References listed as "Online" above are available at the following address:

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

Unit objectives

After completing this unit, you should be able to:

- Define the concepts of users and groups, and explain how and when these should be allocated on the system
- Describe ways of controlling root access on the system
- Explain the uses of SUID, SGID, and SVTX permission bits
- Administer user accounts and groups
- Identify the data files associated with users and security

Figure 14-1. Unit objectives AU1412.0

Notes:

14.1. 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** — Superuser
  - **adm**, **sys**, **bin**, ... — IDs that own system files but cannot be used for login

Figure 14-2. User accounts AU1412.0

Notes:

Importance of user accounts

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.

File ownership

When a file is created, the UID associated with the process that created the file is assigned ownership of the file. Only the owner or **root** can change the access permissions.

Automatically created user accounts

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 a file if any group in the user's groupset provides access. To list the groupset, use the **`groups`** command.
- The user's real group ID is used for file ownership on creation. To change the real group ID, use the **`newgrp`** command.
- Default groups:
  - System administrators: **system**
  - Ordinary users: **staff**

Figure 14-3. Groups AU1412.0

Notes:

Function of groups

Users that require shared access to a set of files are placed in groups. Each group has a unique name and Group ID (GID). The GID, like the UID, is assigned to a file when it is created.

A user can belong to multiple groups.

Predefined groups

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.

Planning and administering groups

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.

Types of groups

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.

Use of the `newgrp` command

A user's real group identification is used to determine the group ownership of a file created by that user. The `newgrp` command changes a user's real group identification. If you provide a group name as a parameter to the `newgrp` command, the system changes the name of your real group to the group name specified (if the group name specified is part of your groupset). If no group name is provided as a parameter, the `newgrp` command changes your real group to the group specified as your primary group in the **/etc/passwd** file.

Group hierarchy

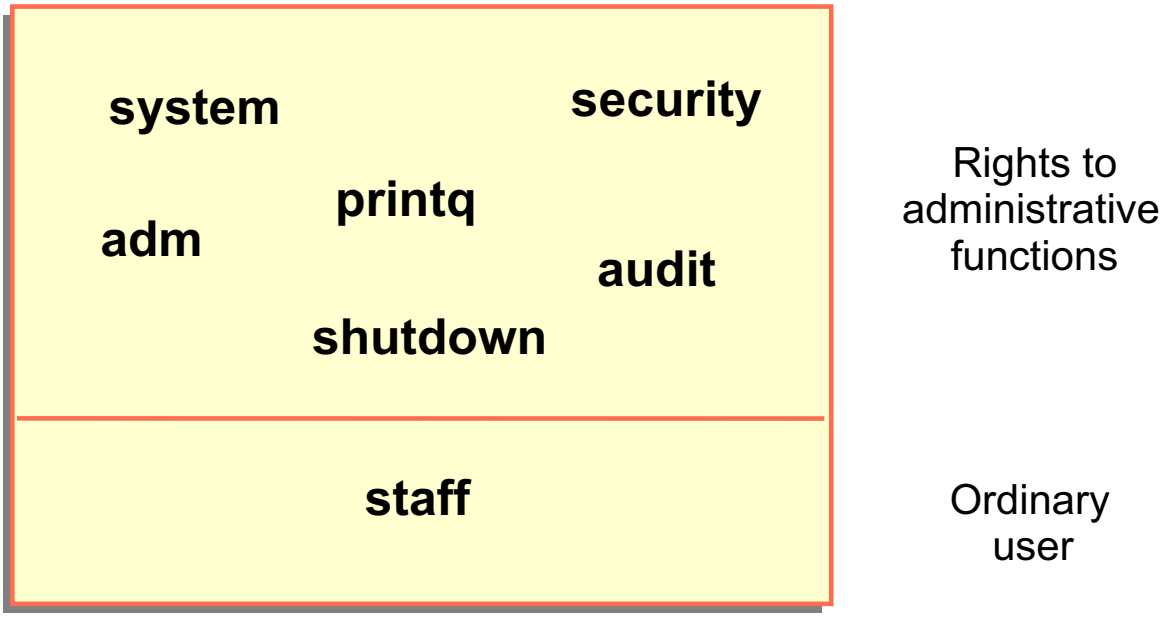


Figure 14-4. Group hierarchy AU1412.0

Notes:

Rights to administrative functions

As indicated on the visual, membership in some groups confers rights to the use of certain administrative functions. Membership in the **staff** group does not provide rights to the use of administrative functions.

Common groups

Common groups on the system (and their intended uses) are as follows:

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 and groups from members of the **security** group, AIX has **admin users** and **admin groups**
- Only **root** can add, remove, or 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

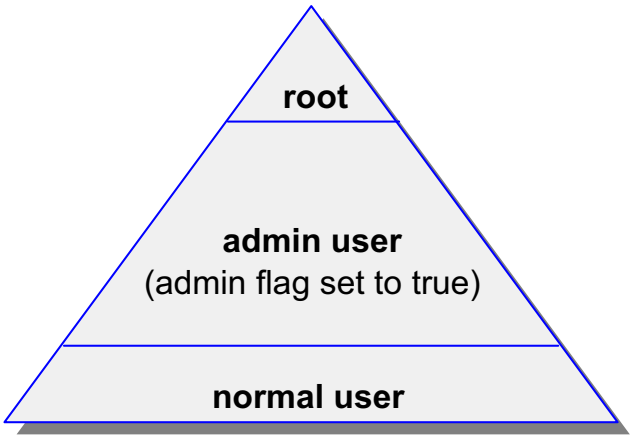


Figure 14-5. User hierarchy AU1412.0

Notes:

Capabilities of members of certain groups

The ability to perform certain system tasks (like creating users) depends upon the standard AIX file permissions. Most system administration tasks can be performed by users other than **root** if those users are assigned 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.

Purpose of user hierarchy

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, or change an admin user or admin group. Therefore, you can define a user that has a high level of access, but who is protected from users in the **security** group.

Controlling access to the root account

- 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 14-6. Controlling access to the root account AU1412.0

Notes:

Guidelines for root account password

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**.

Use of the `su` command

Attempts to become **root** through `su` can be investigated. Successful and unsuccessful attempts might be logged by the audit system.

PATH variable for root account

Do not include unsecured directories in the value of PATH for the **root** account. Note that **root**'s PATH is used by many implicit system functions, not just by a user logged in as **root**.

Security logs

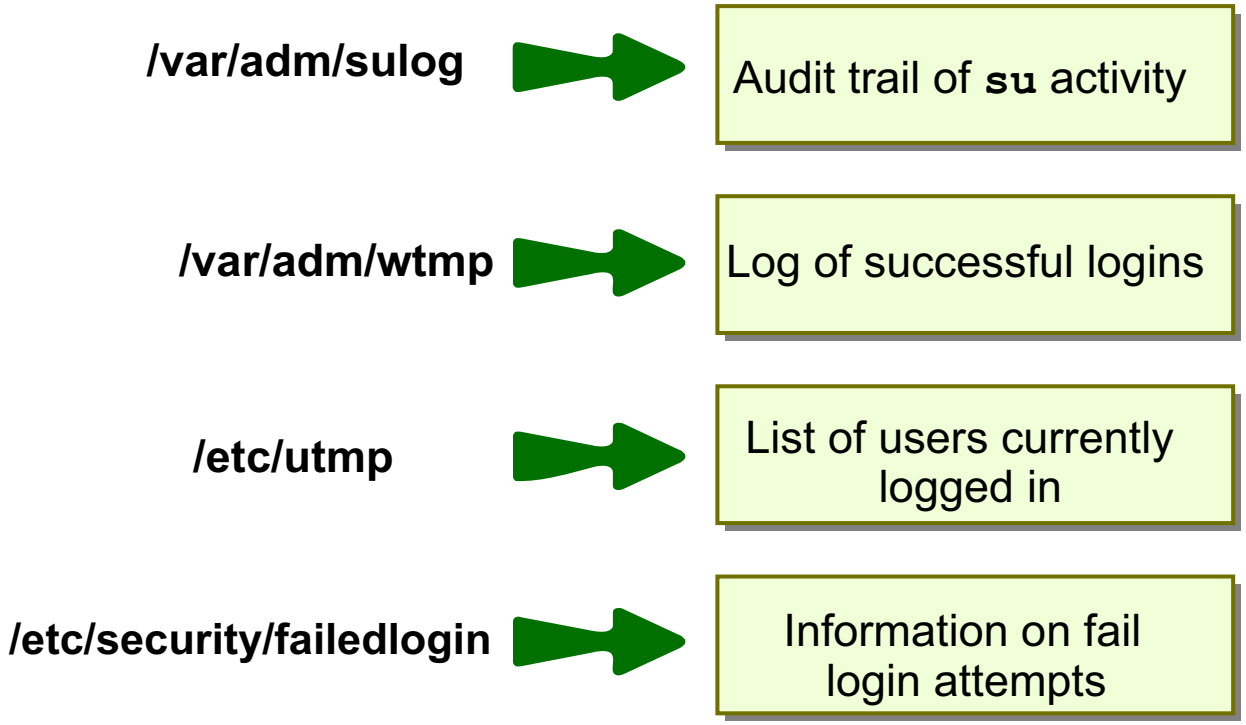


Figure 14-7. Security logs AU1412.0

Notes:

The sulog file

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 the login attempt was successful (and indicates a success by a + and a failed login by a –).

The utmp and wtmp files

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** file, but you can specify either one of the files just mentioned as an argument to the command.

The `last` 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

The `utmpd` daemon

AIX 5L V5.2 introduced a new daemon called **`utmpd`** to manage the entries in the **/etc/utmp** file. This daemon monitors the validity of the user process entries at regular intervals. The default interval time would be 300 seconds.

The syntax of the command is:

```
/usr/sbin/utmpd [ Interval ]
```

To start **`utmpd`** from the **/etc/inittab**, add the following entry to the file:

```
utmpd:2:respawn:/usr/sbin/utmpd
```

The failedlogin file

The **/etc/security/failedlogin** file maintains a record of unsuccessful login attempts. The file can be displays using the **`who`** command with the file as an argument.

File/Directory permissions

File	Perm. Bit	Directory
Read content of file	r	List content of directory
Modify content of file	w	Create and remove files in directory
Use file name to execute as a command	x	Give 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 files to delete files from directory

Figure 14-8. File/Directory permissions AU1412.0

Notes:

Permission bits

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).

The SUID 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.

The SGID bit

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 or directory created within the directory will have the same group ownership as the directory rather than the real group ID or primary group of the user.

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.

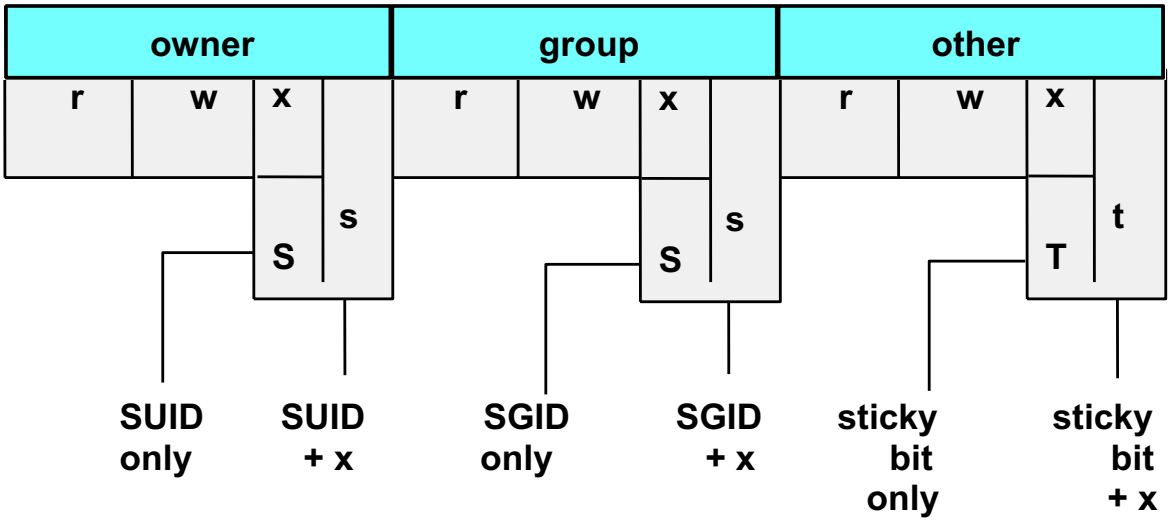
The SVTX bit

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.

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 14-9. Reading permissions AU1412.0

Notes:

How SUID, SGID, and SVTX settings are indicated

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 is on/off, the uppercase **S** or **T** is used to indicate that the execute permission is off. The lowercase **s** or **t** indicates the execute permission is on.

Discussion of examples on visual

Three examples of files that use these additional permissions are shown on the visual:

- The **passwd** command allows users to change their passwords even though passwords are stored in a restricted area
- The **crontab** command allows users to create a **crontab** file even though access to the directory where **crontab** files reside is restricted for ordinary users
- Permission bit settings for /**tmp** allow everyone to write to the directory, but only the owner of a file can remove a file from the /**tmp** directory

Changing permissions

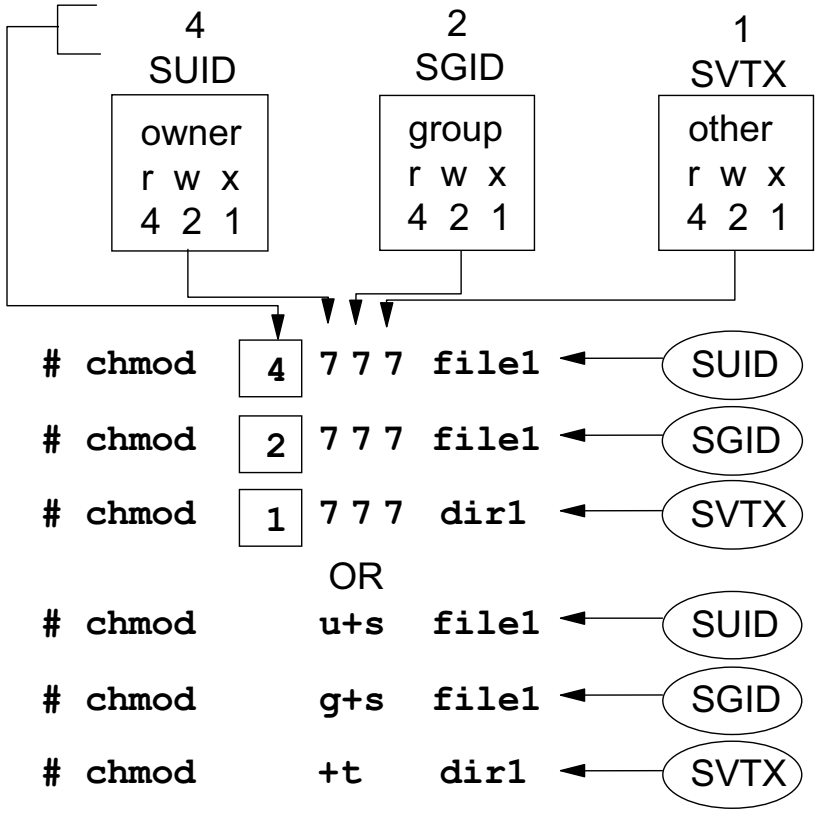


Figure 14-10. Changing permissions AU1412.0

Notes:

Setting the additional permission bits

To set the additional permission bits, you use the same command (**chmod**) as you do to set the regular permission bits.

Using octal notation to set the additional permission bits

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 numeric values of 4, 2, and 1 to set **r**, **w** and **x**. That remains the same. To set the additional bits, you are affecting the **x** position in either the user, group or other area. If you assign numeric values to user (4), group (2), and other (1), these are the values that you 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
- SGID is indicated in the group area; therefore use a **2** in the fourth position
- SVTX is indicated in the others area; therefore use a **1** in the fourth position

Using the symbolic method to set the additional permission bits

You can also use the symbolic method to set the additional permission bits. The visual 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 inherit the following permissions:
  - Ordinary file: `rw-r--r--`
  - Directory: `rwxr-xr-x`
- **/etc/security/user** specifies default and individual user **umask** values

Figure 14-11. **umask** AU1412.0

Notes:

Function of umask

The **umask** specifies what permission bits are set on a new file when it is created. It is an octal number that specifies which of the permission bits are not set.

Default value of umask

If no **umask** was used, then files would be created with permissions of 666 and directories would be created with permissions of 777. The system default **umask** is 022 (indicating removal of the 2 bit or *write* from the group and others area). Therefore, removing *write* from group and other results in an initial permission for files of 644 and, for directories, 755. Execute permission is never set initially on a file.

Changing the umask to enhance security

The default setting of the **umask** is 022. For tighter security you should make the **umask** 027, or even 077.

The `umask` command

To view or change the value of the **umask** for the current session use the `umask` command.

Values stored in /etc/security/user file

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

The **chown** command:

```
# chown fred file1
```

The **chgrp** command:

```
# chgrp staff file1
```

Changing both user and group ownership:

```
# chown fred:staff file1
# chown fred.staff file1
```


Figure 14-12. Changing ownership AU1412.0

Notes:

Using chown to change ownership

As illustrated on the visual, the **chown** command can be used by **root** to change the ownership on a file.

Using chgrp to change group ownership

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. The **root** user can change the group ownership to any group on the system.

Changing both ownership and group ownership

The **chown** command can be used by **root** to set both the ownership and group ownership of a file. As illustrated on the visual, this can be done two different ways:

- Method 1:

  # **chown fred:staff file1**

- Method 2:

  # **chown fred.staff file1**

Role based access control (RBAC)

- Fine grained delegation of authority
  - Roles assigned as an attribute of the user or group
- Legacy RBAC (AIX V4.2+):
  - User space implementation
  - Role assignment alone was insufficient
- Enhanced RBAC (AIX 6.1):
  - Covers user and kernel space
  - Effective role assignment without additional configuration
  - AIX 6.1 SP1 provides 10 predefined roles
- User can activate/inactivate roles as needed
  - Create subshell with role in effect:

```
$ swrole  SysBoot
```


Figure 14-13. Role based access control (RBAC) AU1412.0

Notes:

Why do we need RBAC?

The difficulty with permission (or even access control list) base access control is that you must secure the needed resource rather than the command. It was often difficult to know which resources were the ones needed. In some cases we are dealing with kernel resources. In addition, a given resource may have multiple uses and a single group access to it may not work. Allowing a program to be root with suid allowed one to bypass the resource permissions, but suid itself was a potential exposure.

With Enhanced Resource Based Access Control (RBAC), resource access is controlled via privileged commands and then only users with the proper authorization are allowed execute the privileged command. The authorization and privileges are fine grained.

Legacy RBAC

Starting with AIX 4.2, a form of RBAC was provided but was difficult to work with. Even though a user was assign a role, that user was often still unable to execute the associated tasks until a requisite command was converted to a set uid executable and the user was made a member of the associated command. In addition, the legacy framework was implemented without involvement of the kernel.

Enhanced RBAC

Starting with AIX 6.1, an enhanced form of RBAC is provided. The enhanced RBAC framework involves the kernel and thus is more secure. The new framework is also more granular and extensive than the legacy RBAC. Once a role is assigned to a user, they have the authorization to do the related tasks without having to play with file permissions or group membership. While the framework supports user defined privileged commands, authorizations and roles, AIX 6.1 SP1 provides 10 predefined roles that can be used without additional RBAC configuration.

The details of the RBAC framework is outside the scope of this course.

Activation and deactivation of user roles

Under enhanced RBAC, users can control when they are using their assigned roles. To activate a role, the user would run the **`swrole`** command with one or more comma delimited roles as an argument. The designated role must either be one assigned as an attribute of the user or as an attribute of a group the user belongs to. This creates a subshell where the specified roles are in effect. When the user exits the subshell the activated roles are then effectively deactivated.

If a user attempts a task without activating the related role, the normal authorization mechanisms are in effect (permission settings, hard coded identify checks in the executable, and so forth).

Predefined enhanced RBAC roles

- isso - Information System Security Officer
- sa - System Administrator
- so – System Operator
- AccountAdmin - User and Group Account Administration
- BackupRestore -Backup and Restore Administration
- DomainAdmin - Remote Domain Administration
- FSAdmin - File System Administration
- SecPolicy - Security Policy Administration
- SysBoot - System Boot Administration
- SysConfig - System Configuration

Figure 14-14. Predefined enhanced RBAC roles AU1412.0

Notes:

Overview

AIX 6.1 SP1 provides 10 predefined roles. The first three in the list provide authorization for broad task areas. The ones after that provide the ability to delegate smaller and more focused task areas which are a subset of what the first three provide.

The following are only summaries of authorization. The complete and detail description can only be determined through researching the RBAC databases on your system.

role: isso - Information system security officer

The ISSO role is responsible for creating and assigning roles and is thus the most powerful user-defined role on the system. Some of the ISSO responsibilities include:

- Establishing and maintaining security policy
- Setting passwords for users

- Network configuration
- Device administration

role: sa - System administrator

The SA role provides the functionality for daily administration and is responsible for:

- User administration (except password setting)
- File system administration
- Software installation update
- Network daemon management
- Device allocation

role: so - System operator

The SO role provides the functionality for day to day operations and is responsible for:

- System shutdown and reboot
- File system backup, restore, and quotas
- System error logging, trace, and statistics
- Workload administration

role: AccountAdmin - User and group account administrator

The AccountAdmin role provides the functionality for users and group definitions and is responsible for:

- Define, modify and remove users
- Define modify and remove groups

role: BackupRestore - Backup and restore administrator

The BackupRestore role provides the functionality for backup and restore operations for file systems using various commands such as:

- cpio, pax, tar, backup and restore

role: DomainAdmin - Remote domain administrator

The DomainAdmin role provides the functionality for managing network security mechanisms such as:

- kerberos, ldap, NIS, and PKI

role: FSAdmin - File system administrator

The FSAdmin role provides the functionality for managing file systems and has the ability to:

- Create, modify and remove file systems
- Mount and unmount file systems
- Defrag file systems
- Format file system logs
- Manage file system user quotas
- Create and manage JFS2 snapshots

role: SecPolicy - Security policy administrator

The SecPolicy role provides the functionality for security administration and is responsible for most of what the ISSO covers, except for:

- Domain Administration
- System Configuration

role: SysBoot - System boot administrator

The SysBoot role provides the functionality system shutdown and booting via the facilities for:

- halt, shutdown, and reboot

role: SysConfig - System configuration

The SysConfig role provides the functionality for system configuration and is responsible for such components as:

- inittab
- System console
- Kernel extensions
- uname
- Resource sets
- Date and time zone
- Software license management
- Performance tunables
- Diagnostics

Exercise 15: Security files

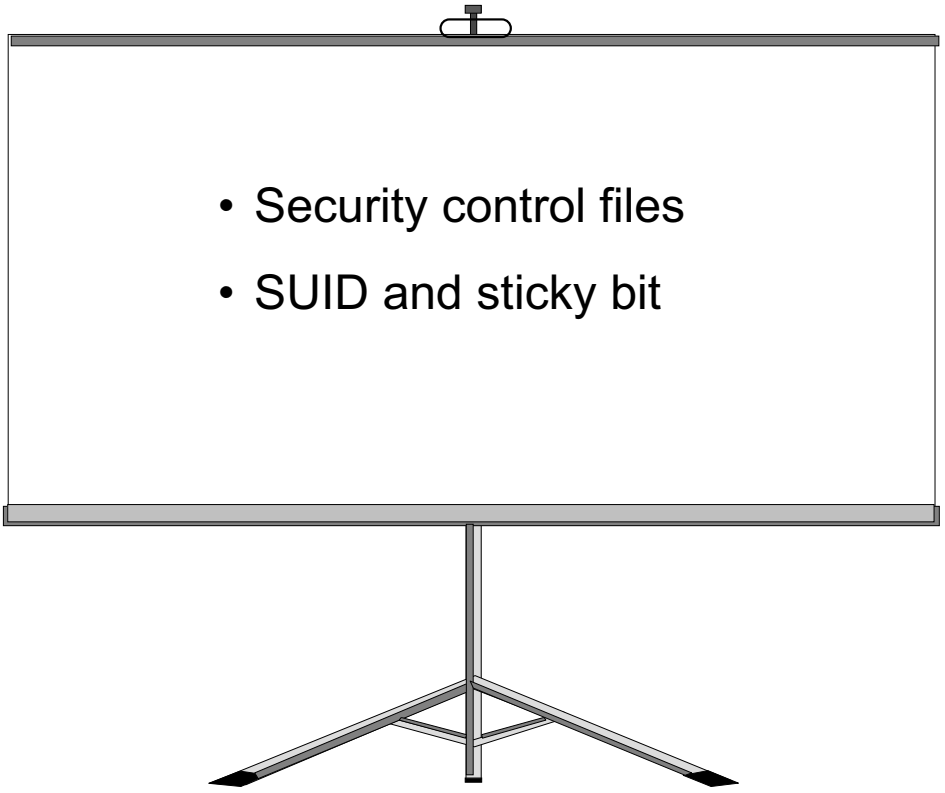


Figure 14-15. Exercise 15: Security files AU1412.0

Notes:

Introduction

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.

The exercise can be found in your *Student Exercises Guide.*

14.2. User administration

Login sequence

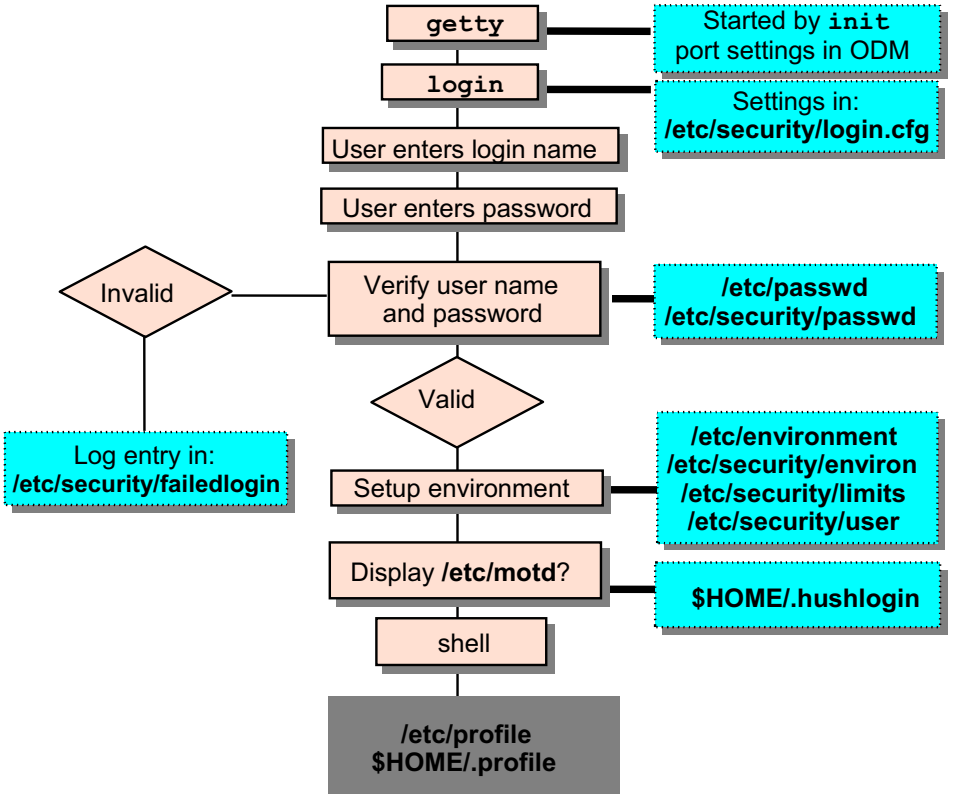


Figure 14-16. Login sequence AU1412.0

Notes:

Introduction

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 section provides an overview of the checks performed during the login process.

The `getty` process

Ports set up for login are listed in the **/etc/inittab**. When `init` runs, a `getty` process is 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 the message is displayed, the `getty` process waits for a user to make a login attempt.

Entry of username and password

When a user is ready to log in, 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.

Validation

When the user enters the password, it is checked. If the password is incorrect or if an invalid user name was given, then the login fails, and an entry is made in the file **/etc/security/failedlogin**. (Use the command **`who /etc/security/failedlogin`** to view this file.) 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 of concurrent logins for a user account. If that number is exceeded, the login is denied and a message is displayed to the user.

Setup of user's environment

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.

Passing of control to shell

Finally, control is passed to the login shell (as defined in **/etc/passwd**) which will read **/etc/environment** and run **/etc/profile** and **$HOME/.profile** when using Korn or Bourne shells.

Results of a user logging out

When a user logs out, the shell terminates and a new **`getty`** process is spawned for that port.

User initialization process

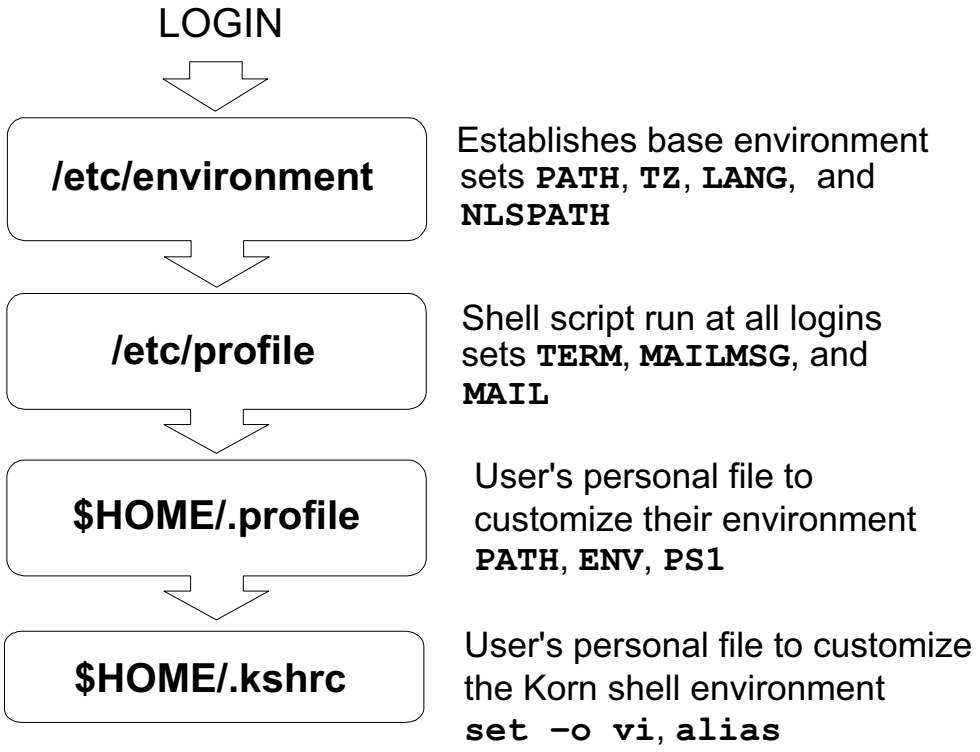


Figure 14-17. User initialization process AU1412.0

Notes:

The /etc/environment file

/etc/environment is used to set variables. No commands should be placed in this file. Only **root** can change this file.

The /etc/profile file

/etc/profile will be read and executed during every login. Like the **/etc/environment** file, this file can be changed only by **root**.

The $HOME/.profile and $HOME/.kshrc files

$HOME/.profile and **$HOME/.kshrc** can be customized by the user. The user can overwrite any variable set in **/etc/environment** and **/etc/profile**.

Common Desktop Environment (CDE) considerations

If you are using CDE, **.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
```

```
                    Security & Users

Move cursor to desired item and press Enter.
Users
Groups
Passwords
Login Controls
Roles
PKI
LDAP
Role Based Access Control (RBAC)
Trusted Execution

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


Figure 14-18. Security and users AU1412.0

Notes:

The Security & Users menu

The **Security & Users** menu is used to manage user and group IDs on the system. The menu consists of the seven options described below.

Users

This option is used to add users to the system, delete existing users and change the characteristics of existing users.

Groups

This option is used to add groups to the system, delete groups and change the characteristics of existing groups.

Passwords

This option is used to change the password for a user. It is also required when setting up a new user or when a user has forgotten their password.

Login Controls

This option provides functions to restrict access for a user account or on a particular terminal.

PKI

PKI stands for X.509 Public Key Infrastructure certificates. This option is used to authenticate users using certificates and to associate certificates with processes as proof of a user's identity.

LDAP

LDAP stands for Light Directory Access Protocol. It provides a way to centrally administer common configuration information for many platforms in a networked environment. A common use of LDAP is the central administration of user authentication. The SMIT option here allows us to configure this platform as either an LDAP client or an LDAP server.

Roles Based Access Control (RBAC)

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.

Trusted Execution

Trusted Execution (TE) refers to a collection of features that are used to verify the integrity of the system and implement advanced security policies, which together can to used enhance the trust level of the complete system.

Using the Web-based System Manager

The Web-based System Manager can also be used to manage users and groups.

SMIT users

```
# smit users
```

```
                    Users

Move cursor to desired item and press Enter.

Add a User
Change a User's Password
Change / Show Characteristics of a User
Lock / Unlock a User's Account
Reset User's Failed Login Count
Remove a User
List All Users


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


Figure 14-19. SMIT users AU1412.0

Notes:

Add a User

Add user accounts.

Change a User's Password

Make 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 for a reasonably long period of time, as 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

The **lsuser** command:

lsuser [-c | -f] [-a attribute …] {ALL | username …}

Example:

```
# 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 14-20. List all users AU1412.0

Notes:

Function of the lsuser command

The **lsuser** command is used to list the attributes of all users (**ALL**) or individual users on the system.

Using SMIT to list users

When the **List All Users** option in SMIT is used, the user name, ID and home directory are listed.

Commonly used lsuser flags

When the **lsuser** 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.

Sources of information listed

The information reported by `lsuser` 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

  ...
                                          [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 ...37]
  ...
```


Figure 14-21. Add a user to the system AU1412.0

Notes:

Ways of adding a user

The **mkuser** command or SMIT can be used to add a user. User attributes can be specified to override the default values.

User name

The only value that must be specified is the user name. Traditionally, this name was restricted to 8 characters in length. Beginning with AIX 5L V5.3, this limit can be changed to allow names as long as 255 characters. The limit is modified in the **Change/Show Attributes of the Operating System** panel (**smit chsys**).

Resources involved in user creation process

The following resources are involved in the user creation process:

- Default ID numbers stored in **etc/security/.ids**
- The **usr/lib/security/mkuser.sys** shell script used to set up a user ID
- Default values for characteristics stored in **usr/lib/security/mkuser.default**
- Default values for characteristics stored in **/etc/security/user**
- The default **.profile** stored in **etc/security/.profile**

Some of these resources are discussed further in the material that follows.

The /usr/lib/security/mkuser.default file

The **/usr/lib/security/mkuser.default** file contains the defaults for the **mkuser** command. 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.

The /etc/security/.ids file

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 are given IDs starting from 200.

The /usr/lib/security/mkuser.sys shell script

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.

List of user characteristics

The full list of user characteristics contains 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`**).

Setting a password

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 either the **`passwd`** or **`pwdadm`** command.

Change / Show Characteristics of a User

```
# smit chuser
```

```
                Change / Show Characteristics of a User

...
                                          [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 ...37]
...
```


Figure 14-22. Change / Show Characteristics of a User AU1412.0

Notes:

Changing user characteristics

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 characteristic

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, and so forth. Some programs (such as the **`finger`** program) use this information when reporting on user activity. Users can change their own user information.

The Initial PROGRAM characteristic

The initial program is the shell which the user logs into. It is usually set to one of:

- **`/usr/bin/bsh`** (Bourne shell)
- **`/usr/bin/csh`** (C shell)
- **`/usr/bin/ksh`** (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 login shell (as well as their own *full name*).

The `chuser` command

The following command can be used to change 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
```

- When you remove a user, that user's home directory is not deleted. Therefore, you must remember to manually *clean up* the directories of users you remove. (Remember to backup important files first!)

```
# rm -r /home/team01
```


Figure 14-23. Remove a user from the system AU1412.0

Notes:

Ways to remove a user

The **Remove a User from the System** option in SMIT 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 of rmuser

The **-p** option removes 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.

Removing the user's files

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 making password changes:

```
# passwd [username]
```

```
# pwdadm username
```

- SMIT invokes the **passwd** command
- An ordinary user can use the **passwd** command to change own password
- Only **root** or member of **security** group can change password of another user

Figure 14-24. Passwords AU1412.0

Notes:

Setting an initial password

When a user ID is created with SMIT or with the **mkuser** command, the user ID is disabled. (An asterisk (*) 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.

Entry of passwords (things to be aware of)

When passwords are entered, they are not displayed. When changing a password, the new password is requested a second time for verification.

The ADMCHG flag

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.

Recovering from a forgotten password

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.

Invocation of `passwd` command by SMIT

The option **Passwords on the Users** menu of SMIT uses the `passwd` command.

Using the `passwd` command

Ordinary users who use `passwd` to change their passwords are first prompted for the old password, and then they are asked twice for a new password. When **root** uses `passwd` to set a user's password, `passwd` only prompts twice for the new password.

Using the `pwdadm` command

Members of the **security** group can use `pwdadm` to change the passwords of non-administrative accounts. Members of the **security** group are first prompted to enter their own password, then prompted twice to enter the user's new password. The **root** user is only prompted twice for the new password.

Users with `ADMIN` flag set

Only **root** can change the password for a user who has the `ADMIN` flag set in **/etc/security/passwd**.

Regaining root's password

1. Boot from CD-ROM, NIM, or a bootable tape
2. Select option 3: **Start Maintenance Mode for System Recovery** from the **Installation and Maintenance** menu
3. Follow the options to activate the **root** volume group and obtain a shell
4. Once a shell is available, execute the **`passwd`** command to change **root**'s password
5. Enter the following command:
   `# sync ; sync`
6. Reboot the system

Figure 14-25. Regaining **root's** password AU1412.0

Notes:

Introduction

A series of steps that can be used to recover if you forget the **root** password are given on this visual.

Step 1

First, you must boot your machine from media other than its normal hard drive. Either an installation CD, a NIM server or a **`mksysb`** tape works just fine. Remember to invoke the service boot list, usually by pressing **`F5`** while your machine is booting. Booting in maintenance mode is covered in AU16 course.

Step 2

You will need to define your system console and select a language. Then the **Installation and Maintenance** menu is displayed. Be certain to select Option 3, **Start Maintenance Mode for System Recovery**. If you select Option 1 or 2, you are reinstalling your operating system.

Step 3

Select the options required to activate the **root** volume group and start a shell. This gets you access to **rootvg** without any passwords.

Step 4

Once you get the # prompt, use the **passwd** command as you normally would to create a new **root** password.

Step 5

Enter the command **# sync ; sync**. This ensures that the memory buffer is written to disk. In other words, it ensures that the new **root** password is saved to disk.

Step 6

Reboot your system. The command **shutdown -Fr** is a good way to accomplish this.

SMIT groups

```
# smit groups
```

```
                    Groups

Move cursor to desired item and press Enter.

List All Groups
Add a Group
Change / Show Characteristics of a Group
Remove a Group




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


Figure 14-26. SMIT groups AU1412.0

Notes:

Purpose of groups

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.

Group management restrictions

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.

Predefined groups

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

The **lsgroup** command:

lsgroup [-c | -f] [-a attribute …] {ALL | groupname …}

Example:

```
# lsgroup ALL
system id=0 admin=true users=root,test2 registry=compat
staff id=1 admin=false users=ipsec,team01,team02,team03,
team04,team05,test1,daemon registry=compat
bin id=2 admin=true users=root,bin registry=compat
sys id=3 admin=true users=root,bin,sys registry=compat
adm id=4 admin=true users=bin,adm registry=compat
uucp id=5 admin=true users=uucp,nuucp registry=compat
...
ipsec id=200 admin=false users= registry=compat
```


Figure 14-27. List all groups AU1412.0

Notes:

The lsgroup command

The **lsgroup** command is used to list all groups or selected groups on the system. The data is presented in line format by default or in colon format (**-c**) or in stanza format (**-f**).

Commonly used options of the lsgroup command

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 a Group

```
# 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]          +
   Projects                              [ ]             +
   Initial Keystore Mode                 [ ]             +
   Keystore Encryption Algorithm         [ ]             +
   Keystore Access                       [ ]             +

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


Figure 14-28. Add a Group AU1412.0

Notes:

The `mkgroup` command

The **`mkgroup`** command is the command used to create a new group. The group name, traditionally, must be a unique string of eight or fewer characters. With AIX 5L V5.3 and later, the maximum name length can be modified to be as large as 255 characters.

Limit on group membership

A user may belong to no more than 32 groups.

The `-a` option

The **`mkgroup -a`** option 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

The `-A` option makes the invoker of the `mkgroup` command the group administrator.

ADMINISTRATOR list and USER list

In the SMIT screen shown on the visual, **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.

The `Projects` field

Starting with AIX 5L V5.3, the SMIT **Add a Group** screen has a new field, **Projects**, for tracking resource usage in the Advanced Accounting subsystem provided in AIX 5L V5.3 and later.

Initial Keystore Mode

The efs_initalks_mode of admin allows for root or other security privileged system users to reset the user's key store password. Otherwise, if the user forgets their key store password, they will not be able to access their Encrypted File System files. If the guard mode is selected, then root cannot reset the user's key store password.

Keystore Encryption Algorithm

This option specifies the algorithm for the user's key within the key store. This key protects the encrypting key of files the user creates within the Encrypted File System.

Keystore Access

The key store will allow the user to utilize files in Encrypted File System. The selection of file will create a key store file associated with this user. It is recommended that file is selected. Select none for no key store to be created. All other EFS (efs_*) attributes will not have any effect.

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]                     +
  Projects                              [ ]                        +
  Initial Keystore Mode                 [ ]                        +
  Keystore Encryption Algorithm         [ ]                        +
  Keystore Access                       [ ]                        +

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


Figure 14-29. Change / remove groups AU1412.0

Notes:

The `chgroup` command

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.

Group attributes

The group attributes are:

- **Group ID** (`id=groupid`). It is not advisable to change the group ID, but it is occasionally done immediately after a group has been created to match the ID of a previously deleted group, or a specific group ID needed 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.
- **Projects** (`projects=projectnames`). As previously mentioned, this attribute was added to support the Advanced Accounting subsystem.

The `chgrpmem` command

The `chgrpmem` command can be used by any user to change either the administrators or the members of a group for which the user running the command is a group administrator.

The `rmgroup` command

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 is 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 are not displayed to that user

Figure 14-30. Message of the day AU1412.0

Notes:

Using the /etc/motd file

The *message of the day* (*motd*) is a convenient way to communicate information, such as installed software version numbers or current system news, to all users. The message of the day is contained in the **/etc/motd** file. To change the message of the day, simply edit this file.

Other ways to communicate with the user community

Many other commands exist to provide ways to communicate with the user community. Several of these commands, such as **`write`**, **`wall`**, **`mail`** and **`talk`**, are covered in the *AIX 6.1 Basics* course.

Exercise 16: User administration (parts 1-5)

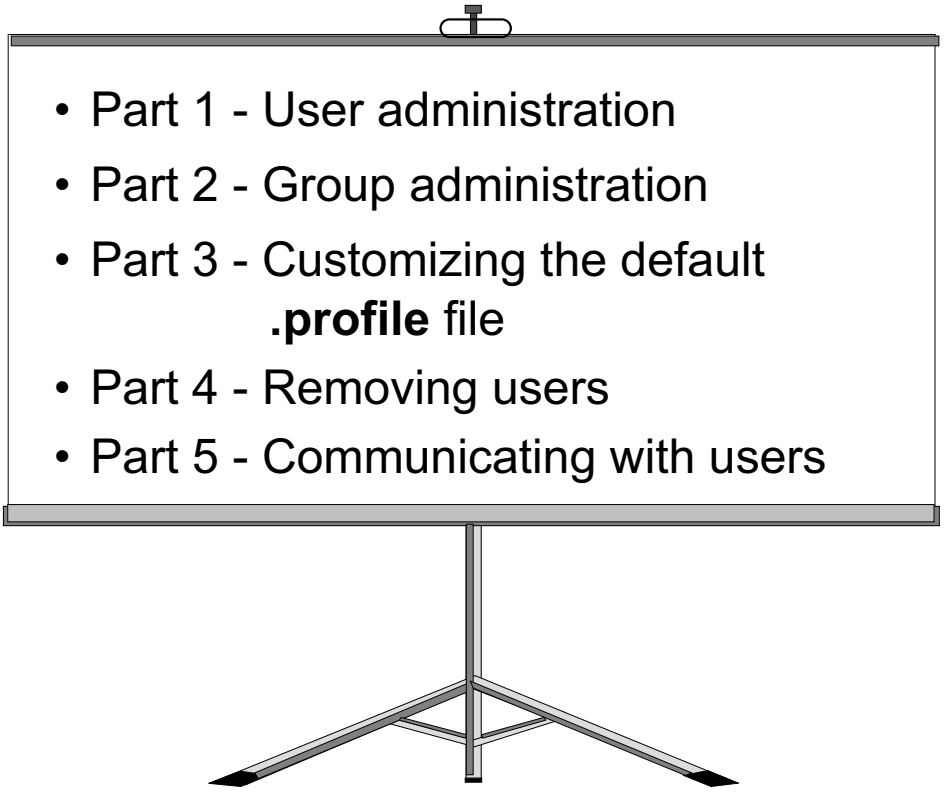


Figure 14-31. Exercise 16: User administration (parts 1-5) AU1412.0

Notes:

Introduction

This lab gives you an opportunity to expand your knowledge of user administration. You add users and groups and review many of the user characteristics.

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

Be sure to only do Parts 1-5. You will be doing Parts 6-7 at the end of this unit.

14.3. 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 14-32. Security files AU1412.0

Notes:

Introduction

The security on the system is controlled by a number of ASCII files. Key files are listed on the visual and briefly described below.

/etc/passwd

The /**etc**/**passwd** file lists the valid users, and the user ID, primary group, home directory, and default login shell for each of these users.

/etc/group

The /**etc**/**group** file lists the valid groups, their group IDs, and members.

The /etc/security directory

The **/etc/passwd** and **/etc/group** files have global read access to all users. A number of other files control the attributes of users. These files are in the **/etc/security** directory, which can only be accessed by **root** or the **security** group.

/etc/security/passwd

/etc/security/passwd contains the encrypted password and update information for users.

/etc/security/user

/etc/security/user contains extended user attributes.

/etc/security/group

/etc/security/group contains extended group attributes.

/etc/security/limits

/etc/security/limits contains process resource limits for users.

/etc/security/environ

/etc/security/environ contains environment variables for users. This file is not often used.

/etc/security/login.cfg

/etc/security/login.cfg is a configuration file for the `login` program. This file 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::/:/usr/bin/ksh
daemon:!:1:1::/etc:
bin:!:2:2::/bin:
sys:!:3:3::/usr/sys:
adm:!:4:4::/var/adm:
uucp:!:5:5::/usr/lib/uucp:
guest:!:100:100::/home/guest:
nobody:!:4294967294:4294967294::/:
lpd:!:9:4294967294::/:
lp:*:11:11::/var/spool/lp:/bin/false
invscout:*:6:12::/var/adm/invscout:/usr/bin/ksh
snapp:*:200:13:snapp login user:/usr/sbin/snapp:/usr/sbin/snappd
nuucp:*:7:5:uucp login user:/var/spool/uucppublic:/usr/sbin/uucp/uucico
ipsec:*:201:1::/etc/ipsec:/usr/bin/ksh
esaadmin:*:811:0::/home/esaadmin:/usr/bin/ksh
john:!:200:0:x7560 5th floor:/home/john:/usr/bin/ksh
bill:*:201:1::/home/bill:/usr/bin/ksh
```


Figure 14-33. **/etc/passwd** file AU1412.0

Notes:

Role of the /etc/passwd file

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.

Fields in the /etc/passwd file

The fields in the **/etc/passwd** file are:

- User name - Up to eight alphanumeric characters (not all uppercase).
- Password - On older UNIX systems, this contained the encrypted password. Beginning with AIX 5L, it cannot contain the encrypted password on AIX systems and should contain a `!` to refer to the **/etc/security/passwd** file. Other common values are an `*`, which means the ID is invalid, and no value, which 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 the initial program or shell that is executed after a user invokes the **`login`** command or **`su`** command.

Using index files for better login performance

In AIX, additional files can be created to be used as index files 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.mzJBjlfbY
      lastupdate = 885485990
      flags =

daemon:
      password = *

bin:
      password = *
...

john:
      password = q/gD6q.ss21x.
      lastupdate = 884801337
      flags = ADMCHG,ADMIN,NOCHECK
```


Figure 14-34. **/etc/security/passwd** file AU1412.0

Notes:

Role of the /etc/security/passwd file

The **/etc/security/passwd** file contains the encrypted user passwords and 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.

Index files

As previously mentioned, in AIX, additional files can be created to be used as index files for **/etc/security/passwd** and some related files. These index files provide for better performance during the login process. These indexes are created using the **`mkpasswd`** command.

Entries in /etc/security/passwd

Valid entries in **/etc/security/passwd** 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.)

/etc/security/user file (1 of 2)

```
# 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 14-35. **/etc/security/user** file (1 of 2) AU1412.0

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 separates by commas, `ALL` or `*`

auth1

Defines the primary authentication method for a user. 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 the 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 two digits of the years 1939 to 2038. If 0101000070 then the account is disabled.

/etc/security/user file (2 of 2)

```
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 14-36. /**etc**/**security**/**user** file (2 of 2) AU1412.0

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` Allows only local users access to the system
- `compat` The normal login procedure and therefore allows local and NIS users access to the system
- `DCE` 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 the day of the week (0=Sunday, 6=Saturday).

pwdwarntime

The number of days before a forced password change that a warning is 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
...
usr:!:100:guest
accounts:!:200:john
...
```

```
# more /etc/security/group
```

```
system:
            admin=true
staff:
            admin=false
accounts:
            admin=false
            adms=john
            projects=system
```


Figure 14-37. Group files AU1412.0

Notes:

The /etc/group file

The fields in the **/etc/group** file are:

- Group - Up to eight alphanumeric characters (not all uppercase)
- Password - This field is *not used in AIX* and should contain a !
- ID - The group ID
- Members - A comma-separated list of the users who belong to this group

The /etc/security/group file

The **/etc/security/group** file is a stanza file with one stanza for each group. The valid entries are:

- `admin` Defines whether the group is an administrative group; values are true or false.
- `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.
- `projects` A list of project names to be associated with the group.

/etc/security/login.cfg file

```
default:
  herald =“Authorized use only.\n\rlogin:"
  logintimes =
  logindisable = 0
  logininterval = 0
  loginreenable = 0
  logindelay = 0
  pwdprompt = "Password: "
  usernameecho = false
```


Figure 14-38. **/etc/security/login.cfg** file AU1412.0

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 obtained 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 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 is two seconds, then four seconds, then six seconds and so forth.

pwdprompt

Defines the password prompt message printed when requesting password input. The value is a character string.

usernameecho

Defines whether the user name should be echoed on a port. If true (this is the default) the user name echo is enabled. If false, user name echo is disabled. The user name is not echoed at the login prompt and is masked out of security-related messages.

The `chsec` command

Changes to the **/etc/security/login.cfg** file can be done by the command **`chsec`**:

```
# chsec -f /etc/security/login.cfg -s default -a pwdprompt="Password:"
```

To reset to the default value:

```
# chsec -f /etc/security/login.cfg -s default -a pwdprompt=
```

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 {-l|-b|-n|-p|-t|-y} {ALL | *username*}**
  - Checks each user name in **/etc/passwd**, **/etc/security/user**, **/etc/security/limits** and **/etc/security/passwd**
  - 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 14-39. Validating the user environment AU1412.0

Notes:

Use of validation commands

The commands listed on the visual 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

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.

Options for `pwdck`, `usrck`, and `grpck` commands

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

Additional options for **`usrck`**, are as follows:

- **`-b`** Reports users who are not able to access the system and the reasons, with the reasons displayed in a bit-mask format.
- **`-l`** Scans all users or the users specified by the User parameter to determine if the users can access the system.

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

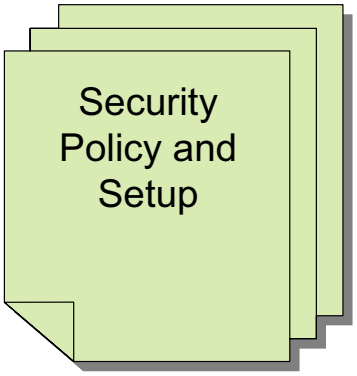


Figure 14-40. Documenting security policy and setup AU1412.0

Notes:

Planning user and group administration

Plan and organize your user and group administration. Every user does not need their own group. Good planning up front reduces any reorganizing of users and groups later on.

Use of the sticky bit

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.

Figure 14-41. Checkpoint (1 of 2) AU1412.0

Notes:

Checkpoint (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?
   a. **/etc/passwd**
   b. **/etc/security/passwd**
   c. **/etc/security/restrictions**
   d. **/etc/security/user**

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 (except **root**)
   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

Figure 14-42. Checkpoint (2 of 2) AU1412.0

Notes:

Exercise 16: User administration (parts 6-7)

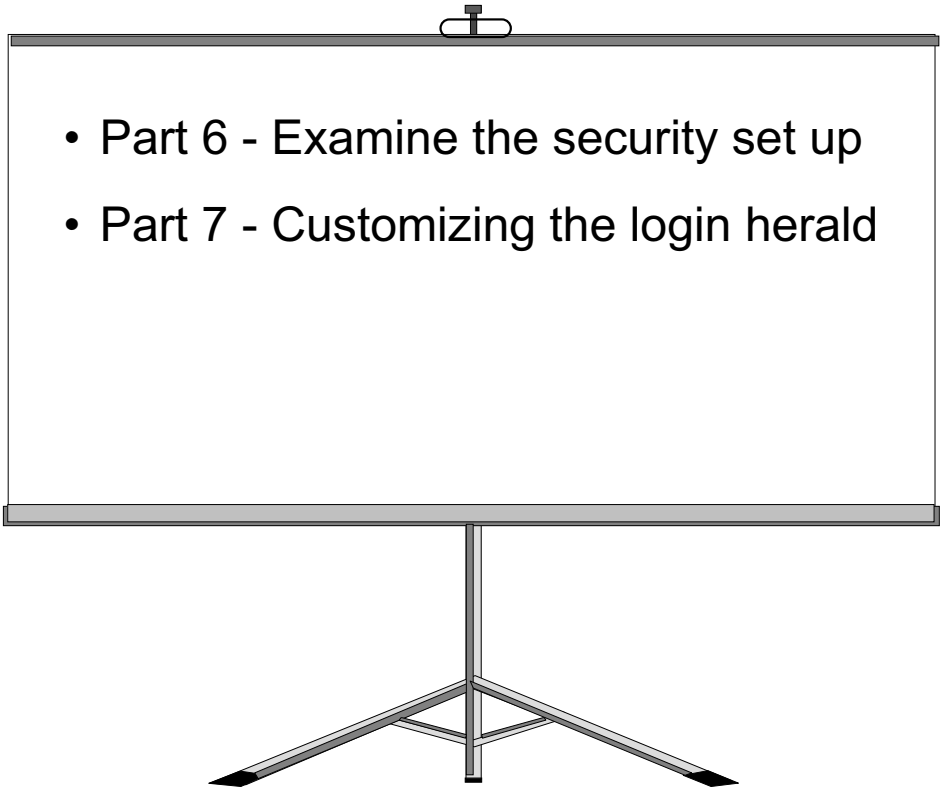


Figure 14-43. Exercise 16: User administration (parts 6-7) AU1412.0

Notes:

Introduction

This lab gives you an opportunity to expand your knowledge of user administration. You will examine the security set up and customize the login herald.

The exercise can be found in your *Student Exercises Guide*.

Unit summary

- User and groups can be added and deleted from the system by using SMIT or by using high level commands.
- Passwords must be set for all users using either **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 the **/etc** and **/etc/security** directories.

Figure 14-44. Unit summary AU1412.0

Notes:

Unit 15. 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

References

Online	*AIX 6.1 Commands Reference*
Online	*AIX 6.1 Files Reference*

Unit objectives

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

Figure 15-1. Unit objectives AU1412.0

Notes:

The `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)

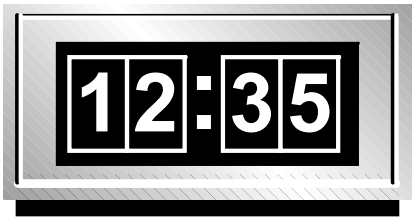


Figure 15-2. The cron daemon AU1412.0

Notes:

Function of the `cron` daemon

The system process that allows batch jobs to be executed on a timed basis is the **`cron`** daemon. Many people rely on **`cron`** to execute jobs. Jobs are submitted to the **`cron`** daemon in 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

Starting of `cron`

The **`cron`** process is usually started at system startup by **/etc/inittab**. It runs constantly as a daemon. If killed, it is automatically restarted.

Changing how `cron` event types are handled

The **/var/adm/cron/queuedefs** file defines how the system handles different **`cron`** daemon event types. The file specifies the maximum number of processes per event type to schedule at one time, the nice value of the event type, and how long to wait before retrying to execute a process. This file is empty as shipped, but can be modified to change how the **`cron`** daemon handles each event type.

For example, by default, **crontab** events are inspected every 60 seconds, run at a nice value of 2 higher 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 **crontab** jobs were to run at a nice value of 10 higher than the default with files inspected every two minutes and with 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 **crontab** privileges of users:
  - **/var/adm/cron/cron.deny** lists users who cannot use **crontab**
  - **/var/adm/cron/cron.allow** lists users who can use **crontab**
- An empty **cron.deny** exists by default

Figure 15-3. crontab files AU1412.0

Notes:

Scheduling a job

The `cron` daemon starts processes at specified times. It can be used to run 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.

The /var/adm/cron/cron.deny file

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, exists and is empty. As the administrator, you can restrict access to `cron` by adding user names to this text file.

The /var/adm/cron/cron.allow file

Another file that also restricts users' privileges is **/var/adm/cron/cron.allow**. To use this file, you should remove the **cron.deny** file and create the **cron.allow** file to list the users that are allowed to use `cron`. If **cron.allow** exists and is empty, ***NO*** user is 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** nor **cron.deny** exists, then only **root** can use `cron`.

Format of a crontab file

To view current **crontab:**

```
# crontab -l
```

```
...
#0  3 * * * /usr/sbin/skulker
#45 2 * * 0 /usr/lib/spell/compress
...
0 11  * * * /usr/bin/errclear -d S,O 30
0 12  * * * /usr/bin/errclear -d H 90
0 15  * * * /usr/lib/ras/dumpcheck >/dev/null 2>&1
...
```

Format of entries:

```
minute hour date-of-month month day-of-week command
```


Figure 15-4. Format of a crontab file AU1412.0

Notes:

Viewing a crontab file

Each user can view their **crontab** file by using the command `crontab -l`.

The user's **crontab** file contains the schedule of jobs to be run on behalf of that user. There is a separate **crontab** file for each user of the `crontab` facility. This file is located in **/var/spool/cron/crontab/$USER**.

Format of crontab file entries

The format for the lines in this file is as follows:

```
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, and so forth)
command
```

Fields are separated by spaces or tabs. 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 (-).

Examples of crontab entries

Here are some examples of **crontab** entries:

- To start the **backup** command at midnight, Monday through Friday:

  ```
  0  0  *  *  1-5  /usr/sbin/backup  -0  -u  -q -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 a crontab file

- One way to edit a **crontab** file:

```
# crontab -e
```

- A safer method:

```
# crontab -l > /tmp/crontmp
# vi /tmp/crontmp
# crontab /tmp/crontmp
```


Figure 15-5. Editing a crontab file AU1412.0

Notes:

Creating or updating a crontab file

To schedule a job, you must create a **crontab** file. The `cron` daemon keeps the **crontab** files in memory, so you cannot update the **crontab** entries by just modifying the file on disk.

Using `crontab -e` to edit the crontab file

To edit the **crontab** file, one method is to use `crontab -e`. This opens 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.

Another method of updating your crontab file

The `crontab -l` command always shows the crontab file that `cron` is using on your behalf. Another method to update the file is to use the command `crontab -l > mycronfile`. This creates a copy of the current **crontab** file and allows you to safely edit the **mycronfile** file without affecting the current **crontab** file. To submit your changes, use the command: `crontab mycronfile`. The content of the **mycronfile** file replaces the content of your file in the **crontab** directory and refreshes the `cron` daemon, all at once. Now, you also have a backup of the **crontab** file in **mycronfile**.

Removing your crontab file

Use the command `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 to be run by **cron** at a specified time:

```
# at now +2 mins
banner hello > /dev/pts/0
<ctrl-d>
job user.time.a will be run at date
```

- The **batch** command submits a job to be run when the processor load is sufficiently low:

```
# batch
banner hello > /dev/pts/0
<ctrl-d>
```


Figure 15-6. The `at` and `batch` commands AU1412.0

Notes:

Use of the `at` command

The **at** command submits a job for **cron** to run once (rather than on a recurring basis) at a specified time. It reads the commands to execute from standard input. The **at** command mails you all output from standard output and standard error for the scheduled commands, unless you redirect that output.

Examples of 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**.

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 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** sends you mail to say that the job is complete.

Controlling use of `at`

The **`at`** command can only be used by **root** unless one of the following files 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 in it can use **`at`** (**root** included).

Use of the `batch` command

The **`batch`** command submits a job to be run when the processor load is sufficiently low.

Like the **`at`** command, the **`batch`** command reads the commands to be run from standard input and mails you all output from standard output and standard error for the scheduled commands, unless you redirect that output.

Controlling at jobs

- To list at jobs:

```
at -l [user]
atq [user]
```

```
# at -l
root.1118077769.a       Mon Jun  6 10:09:29 2007
root.1118078393.a       Mon Jun  6 10:19:53 2007
test2.1118079063.a      Mon Jun  6 10:31:03 2007
```

- To cancel an **at** job:

```
at -r job
atrm [job | user]
```

```
# at -r test2.1118079063.a
at file: test2.1118079063.a deleted
```

- To cancel all your **at** jobs:

```
atrm -
```


Figure 15-7. Controlling at jobs AU1412.0

Notes:

Listing at jobs

To list **at** jobs use the **at -l** command or the **atq** command. The **root** user can look at another user's **at** jobs by using the command **atq <*user*>**.

Removing at jobs

To cancel an **at** job use **at -r** or **atrm** followed by the job number. Use the command **atrm -** (placing nothing after the **-** character) to cancel all of your jobs. The **root** user can cancel all jobs for another user using **atrm <*user*>**.

Documenting scheduling

- Have a copy of each user's **crontab** file
- Have a copy of the **/etc/inittab** file

Figure 15-8. Documenting scheduling AU1412.0

Notes:

Overview

It is important to have correct up to date information regarding your system, in case of an unexpected system failure.

Maintain as much documentation as possible about all aspects of the system by following the recommendations we have given throughout the course.

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. Give a **crontab** entry that would specify that a job should run every Thursday at 10 past and 30 minutes past every hour.

   _____

3. How would you schedule a script named **myscript**, to run 10 minutes from now?

   _____

   _____

   _____

   _____

Figure 15-9. Checkpoint AU1412.0

Notes:

Exercise 17: Scheduling

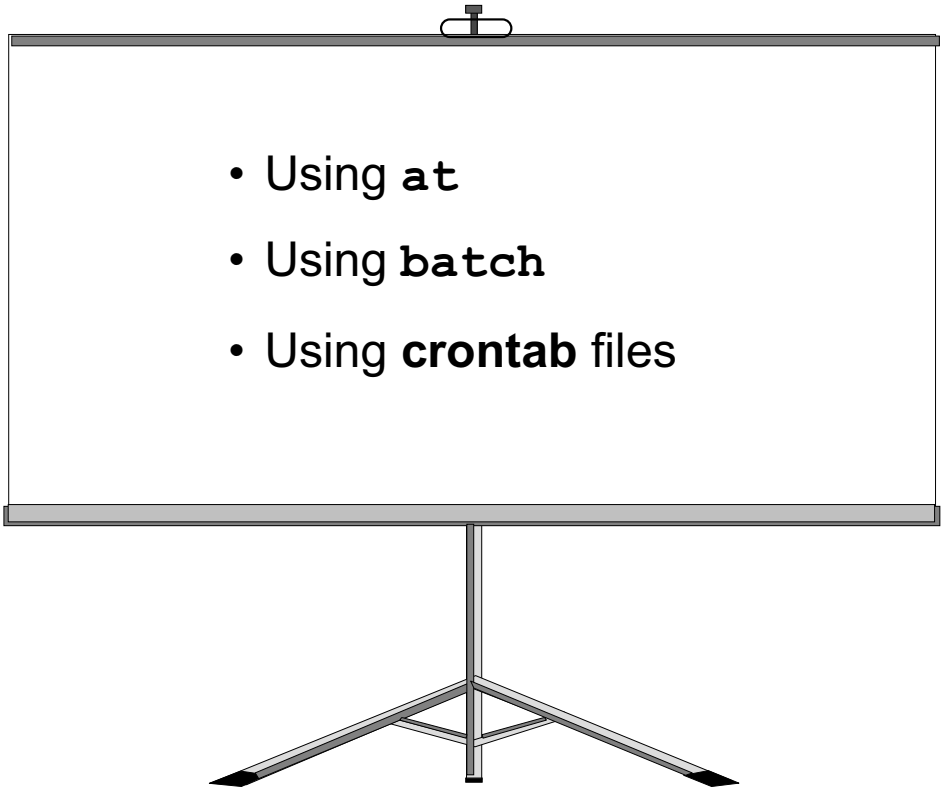


Figure 15-10. Exercise 17: Scheduling AU1412.0

Notes:

Introduction

This lab gives you the opportunity to schedule jobs using both **`at`** and **`crontab`**.

The exercise can be found in your *Student Exercises Guide*.

Unit summary

- The **cron** daemon is responsible for running scheduled jobs.
- The **crontab** files are used to schedule recurring jobs.
- The **at** command is used to schedule a command for one time only execution.
- The **batch** command is used to submit a job to be run when the processor load is sufficiently low.

Figure 15-11. Unit summary AU1412.0

Notes:

Unit 16. Printers and queues

What this unit is about

This unit describes the concepts behind the AIX print spooling mechanisms in AIX 6.1.

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

Online	*AIX 6.1 System Management Guide*
Online	*AIX 5L Version 5.3 Guide to Printers and Printing*

Unit objectives

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
- Submit jobs for printing
- Manage jobs in the queue

Figure 16-1. Unit objectives AU1412.0

Notes:

AIX 6.1 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 16-2. AIX 5L printing environments AU1412.0

Notes:

Introduction

The visual gives an overview of the different approaches that can be taken to printing under AIX 5L and later. In the next two visuals, System V printing is compared 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/lp0
```

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, **`lpsched`**, 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 backend 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/internet/wwsites.nsf/vwwebpublished/ipmaix_ww

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
- Customizable spooling subsystem

Figure 16-3. AIX print subsystem: Advantages AU1412.0

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 the Web-based System Manager or SMIT, as well as the command line. 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 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.

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 16-4. System V print subsystem: Advantages AU1412.0

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 rewritten. This has led to a limited number of printers supported under AIX. With the support of System V printing in AIX 6.1, it is 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 backend program must be customized to restrict user access.

Support for forms

If you are printing to preprinted 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 are added to the System V print subsystem in later releases, while the AIX print subsystem is supported, but not enhanced in future releases.

Concepts of queues

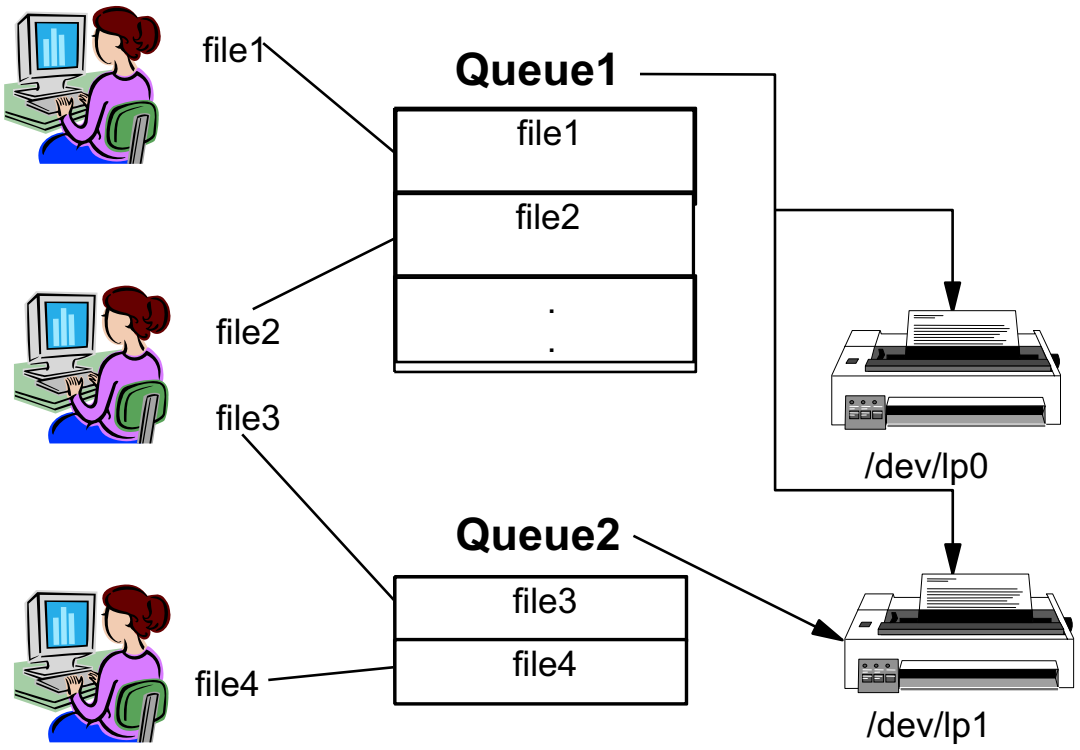


Figure 16-5. Concepts of queues AU1412.0

Notes:

Purpose for queues

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 6.1 queuing system performs this function.

Benefits of queues

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

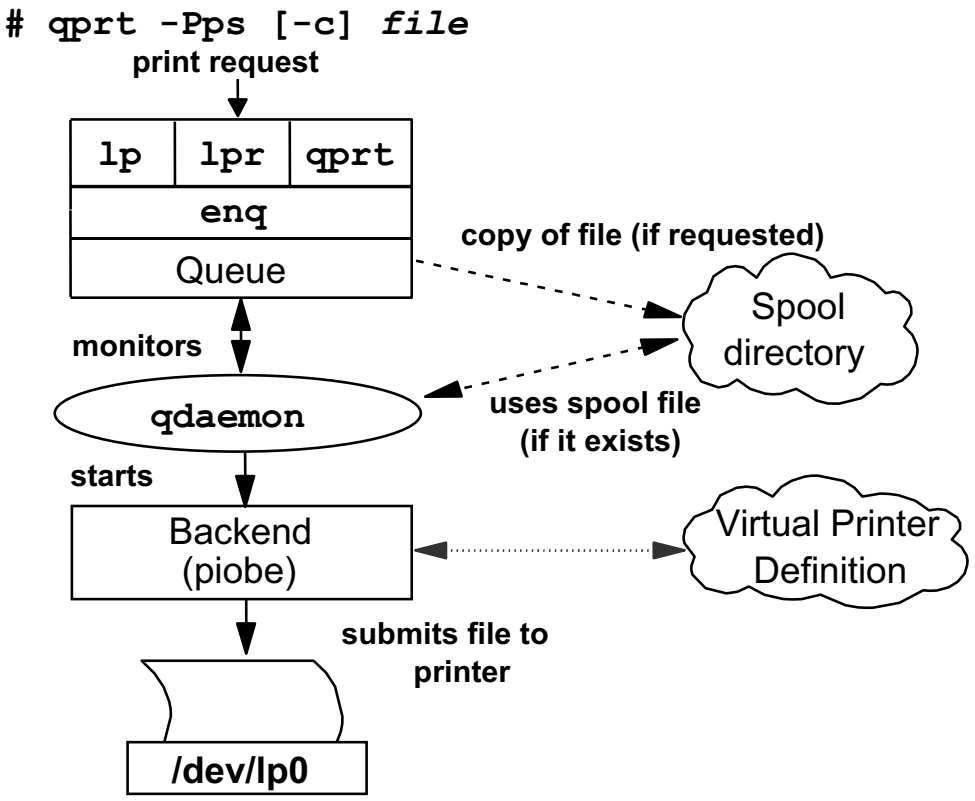


Figure 16-6. Printer data flow AU1412.0

Notes:

Print request

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

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** tries to process the job if the destination device is available, otherwise the job remains in the queue and **qdaemon** tries again later.

Queueing system process

The flow of the queuing system shown in the visual:

- 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 back end 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.

What happens when a file is spooled?

When a file is spooled, a copy of that file is sent to the print spool directory, /**var**/**spool**/**qdaemon**. The copy remains 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 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 is 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 16-7. System files associated with printing AU1412.0

Notes:

Print related files and directories

The system files and directories used for printing include:

- 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

- Manages queues
- Started in the **/etc/inittab** file
- Invokes the backend programs
- Optionally records accounting data

Figure 16-8. `qdaemon` . AU1412.0

Notes:

`qdaemon` introduction

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

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 AIX system. 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 also prints header and trailer pages if they are enabled.

The /etc/qconfig file

```
lp0:                                          * 1 queue pointing to 1 device
            device = lp0dev
            up = TRUE
            discipline = fcfs
lp0dev:
            file = /dev/lp0
            backend = /usr/lib/lpd/piobe
            header = group
            trailer = never
            feed = never
lpq:                                          * 1 queue pointing to 2 devices
            device = lpqdev1,lpqdev2
lpqdev1:
            file = /dev/lp1
            backend = /usr/lib/lpd/piobe
lpqdev2:
            file = /dev/lp2
            backend = /usr/lib/lpd/piobe
ps:                                           * 2 queues pointing to 1 device
            device = psdev
psdev:
            file = /dev/lp3
            backend = /usr/lib/lpd/piobe
asc:
            device = ascdev
ascdev:
            file = /dev/lp3
            backend = /usr/lib/lpd/piobe
```


Figure 16-9. The /etc/qconfig file AU1412.0

Notes:

Introduction

The /**etc**/**qconfig** file is an attribute file. Some stanzas in this file describe queues, and other stanzas describe devices. Every queue stanza requires that one or more device stanzas immediately follow it in the file.

This file is the key to customizing the queues. Although the file can be edited directly, it is recommended that it be changed through high-level commands or via SMIT.

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.

Some of the attributes that can be found in the queue stanza include:

Attribute	Definition	Default	Other
device	Identifies the symbolic name that refers to the device stanza		
discipline	Defines the queue serving algorithm	fcfs	sjn
acctfile	Identifies the file used to save print accounting information	false	*filename*
up	Defines the state of the queue	TRUE	FALSE

Device stanza

The name of a device stanza is arbitrary and can be from 1 to 20 characters long. The name is followed by a colon.

The attributes that can be found in the device stanza include:

Attribute	Description	Default	Other
file	Identifies the special file where the output of backend is to be redirected. FALSE indicates no redirection and that the file name is /**dev**/**null**.	FALSE	
backend	Specifies the full path name of the backend, optionally followed by the flags and parameters to be passed to it.		
access	Specifies the type of access the backend has to the file specified by the file field. This field is ignored if the file field has the value FALSE.	write	both (used for modems or backends needing read capability)
header	Specifies whether a header page prints before each job or group of jobs.	never	always group
trailer	Specifies whether a trailer page prints after each job or group of jobs.	never	always group
feed	Specifies either the number of separator pages to print when the device becomes idle or the value never, which indicates that the backend is not to print separator pages.	never	integer
align	Specifies whether the backend sends a form-feed control before starting the job if the printer was idle.	FALSE	TRUE

The device stanza must contain an attribute that designates the backend program. 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 set up 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_choice

                        Print Spooling

Move cursor to desired item and press Enter.

  AIX Print Spooling
  System V Print Spooling





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


Figure 16-10. Printer menu AU1412.0

Notes:

Interface to manage spooling

AIX print spooling as well as System V print spooling are supported by SMIT in AIX 6.1. The Web-based System Manager also supports both print spooling systems.

AIX printer menu

```
# smit spooler

                          AIX Print Spooling
Move cursor to desired item and press Enter.

  Start a Print Job
  Manage Print Jobs
  List All Print Queues
  Manage Print Queues
  Add a Print Queue
  Add an Additional Printer to an Existing Print Queue
  Change / Show Print Queue Characteristics
  Change / Show Printer Connection Characteristics
  Remove a Print Queue
  Manage Print Server
  Programming Tools

  Change / Show Current Print Subsystem

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


Figure 16-11. AIX printer menu AU1412.0

Notes:

SMIT AIX Printer menu

The SMIT fastpath to this menu is **`smit spooler`**. Printers and print queues can also be managed using the Web-based System Manager.

The options on this menu are:

- **Start a Print Job**
  This option starts a print job by submitting the job to a print queue.
- **Manage Print Jobs**
  This option puts you into a submenu which allows 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 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 are 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.
- **Change/Show Current Print Subsystem**
  Only one of the two print subsystems at the same time can be active. Per default after installation the AIX printer subsystem is active.

Other commands

To show current print subsystem: **`# switch.prt -d`**

To change current print subsystem, you can use either:

```
-# switch.prt -s AIX
-# switch.prt -d SystemV
```

To check if binaries are correct linked, you can use either:

```
-/usr/bin/lpstat --> /usr/aix/bin/lpstat
-/usr/bin/lpstat --> /usr/sysv/bin/lpstat
```

Configuring a printer with a queue

```
                          AIX Print Spooling

  Move cursor to desired item and press Enter.

                          Add a Print Queue

Move cursor to desired item and press Enter.Use arrow keys to scroll.
#ATTACHMENT TYPE              DESCRIPTION
  local                       Printer Attached to Local Host
  remote                      Printer Attached to Remote Host
  xstation                    Printer Attached to Xstation
  ascii                       Printer Attached to ASCII Terminal
  hpJetDirect                 Network Printer (HP JetDirect)
  file                        File (in /dev directory)
  ibmNetPrinter               IBM Network Printer
  ibmNetColor                 IBM Network Color Printer
  other                       User Defined Backend


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


Figure 16-12. Configuring a printer with a queue AU1412.0

Notes:

Adding a local print queue

In our example, assume that the printer is directly attached to our AIX 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 (1 of 2)

```
                          AIX Print Spooling

  Move cursor to desired item and press Enter.

                              Printer Type
Move cursor to desired item and press Enter.

Bull
Canon
Dataproducts
Hewlett-Packard
IBM
Lexmark
OKI
Printronix
QMS
Texas Instruments
Other (select this if your printer is not listed above)

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


Figure 16-13. Selecting a printer type (1 of 2) AU1412.0

Notes:

Specify the printer manufacturer

The next selection that has to be made is the printer type. Notice that IBM is only one of the choices and many other manufacturers are supported as well. Note also that there is an **Other** option which will be selected if the printer type is not supported; that is, not part of the list.

Selecting a printer type (2 of 2)

```
                    AIX Print Spooling

                        Printer Type

  Move cursor to desired item and press Enter.

  [MORE...8]
    ibm2391-2    IBM 2391 Plus printer (Model 2)
    ibm3112      IBM 3112 Page Printer
    ibm3116      IBM 3116 Page Printer
    ibm3130      IBM 3130 LaserPrinter
    ibm3812-2    IBM 3812 Model 2 Page Printer
    ibm3816      IBM 3816 Page Printer
    ibm4019      IBM 4019 LaserPrinter
    ibm4029      IBM 4029 LaserPrinter
    ibm4037      IBM 4037 LP printer
    ibm4039      IBM 4039 LaserPrinter
  [MORE...49]

  F1=Help                 F2=Refresh              F3=Cancel
  Esc+8=Image             Esc+0=Exit              Enter=Do
  /=Find                  n=Find Next
```


Figure 16-14. Selecting a printer type (2 of 2) AU1412.0

Notes:

Select the manufacturer's supported printer

If you do not have the software installed for your printer, you are prompted to insert the media to install the software first before configuring the device and the queue.

The choice of printer determines 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 guide you through the creation of up to four separate queues which submit to the same printer.

Printer attachment

```
                    Printer Interface
Move cursor to desired item and press Enter.

 parallel
 rs232
 rs422
```

```
                     Parent Adapter
Move cursor to desired item and press Enter.

 ppa0  Available     01-G0   Standard Parallel Port Adapter
```


Figure 16-15. Printer attachment AU1412.0

Notes:

Selecting the printer attachment

After selecting a printer type, a pop-up window is 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 is 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]

Description                               IBM 4029 LaserPrinter
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 16-16. Add the print queues AU1412.0

Notes:

Create the print queues

This menu varies depending on the characteristics of the physical printer. If the printer is capable of two or three different modes or emulations the system prompts 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

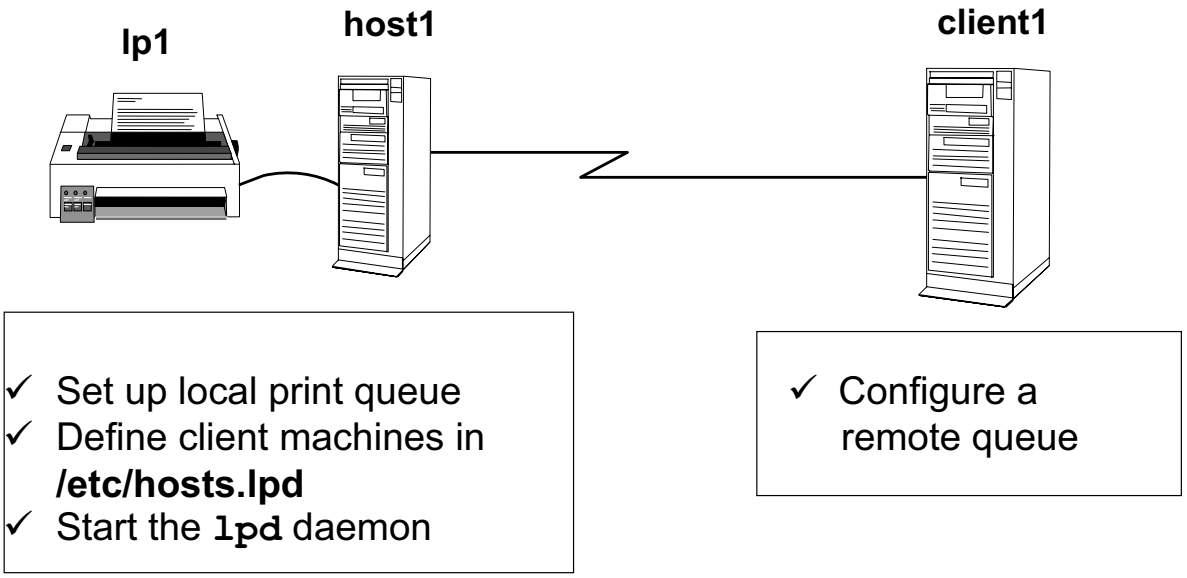


Figure 16-17. Remote printing AU1412.0

Notes:

Overview of print server setup

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 the **/etc/hosts.lpd** file 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 mkhostslpd**.

The **lpd** daemon is controlled by SRC. You should use SMIT to start it however, because SMIT also adds entries to **/etc/inittab** to ensure that it is started on reboot. The fastpath for this screen is **smit mkitab_lpd**.

Client authorization

```
# smit mkhostslpd

                 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 16-18. Client authorization AU1412.0

Notes:

Set up client authorization

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 AIX system is 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 16-19. Start lpd AU1412.0

Notes:

Starting the `lpd` daemon

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

```
                          AIX Print Spooling

  Move cursor to desired item and press Enter.

                          Add a Print Queue

Move cursor to desired item and press Enter.Use arrow keys to scroll.
#ATTACHMENT TYPE              DESCRIPTION
  local                       Printer Attached to Local Host
  remote                      Printer Attached to Remote Host
  xstation                    Printer Attached to Xstation
  ascii                       Printer Attached to ASCII Terminal
  hpJetDirect                 Network Printer (HP JetDirect)
  file                        File (in /dev directory)
  ibmNetPrinter               IBM Network Printer
  ibmNetColor                 IBM Network Color Printer
  other                       User Defined Backend


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


Figure 16-20. Add a remote print queue AU1412.0

Notes:

Adding a remote queue on the client

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 are 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                  [host1]
 *Name of QUEUE on remote server             [lp1]
  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 16-21. Define the print server on the client AU1412.0

Notes:

Required input

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

1. True or false? 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?

   _____

   _____

   _____

Figure 16-22. Let's review AU1412.0

Notes:

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 16-23. Submitting print jobs AU1412.0

Notes:

Introduction

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.

Submitting a print job

To submit a print job to a queue, use either **lp**, **lpr**, or **qprt**. All jobs 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**.

Spooling

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**.

Multiple copies

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.

The enq command

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
```

Requesting a specific printer

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 16-24. Listing jobs in a queue AU1412.0

Notes:

Checking status with the `qchk` command

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`, and so forth)
`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 viewing commands

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 and later.

Change characteristics of a queue

```
# smit chpq
```

```
                    Print Queue to Change / Show
Type or select values in entry fields.
Press Enter AFTER making all desired changes.


                                    [Entry Fields]

PRINT QUEUE name                    [ps]          +

                 Characteristics to Change / Show
Move the cursor to the desired item and press Enter.

1. Printer Setup
2. Default Print Job Attributes
3. Accounting File
4. Queuing Discipline
```


Figure 16-25. Change characteristics of a queue AU1412.0

Notes:

Attributes for Printer Setup option

After selecting **1. Printer Setup**, the following attributes can be changed or shown:

- 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

Attributes for Default Print Job option

After selecting **2. Default Print Job Attributes**, the following attributes can be changed or shown:

- 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

Attributes for Accounting File option

After selecting **3. Accounting File**, the following attribute can be changed or shown:

- Accounting file name

Attributes for Queuing Disciple option

After selecting **4. Queueing Disciple**, the following attribute can be changed or shown:

- 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 16-26. Removing a queue AU1412.0

Notes:

Removing a queue with SMIT

It is not possible to remove a queue containing jobs. The jobs would have to be removed first.

The last option on the screen asks whether the printer device definition should be kept. This option will only appear if the queue being removed is the only queue defined for a printer. Note that by default, it will be removed.

Managing queues

```
# smit pqmanage
```

```
                    Manage Print Queues

Move the cursor to the desired item and press Enter.

  Show Status of Print Queues
  Stop a Print Queue
  Start a Print Queue
  Set the System's Default Print Queue





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


Figure 16-27. Managing queues AU1412.0

Notes:

SMIT Managing Queues options

The following actions can be done:

- **Show Status of Print Queue** gives 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 16-28. Understanding queue status AU1412.0

Notes:

Introduction

The status of the queues and jobs can be displayed with **`qchk`**, **`lpstat`** or **`lpq`**. There are a number of different status states 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 16-29. Bringing queues up and down AU1412.0

Notes:

Enabling a queue

Occasionally, problems with printers can bring a queue down. Once the problem has been fixed it can be brought back up with:

```
# enable <queuename>
```

Disabling a queue

Sometimes, you may wish to bring a queue down. This is recommended if any maintenance is going to be performed on the printer. You can do this with either of the commands:

- `# disable <queuename>`
- `# enq -D -P <queuename>`

Manage Print Jobs

```
# smit jobs
```

```
                    Manage Print Jobs

Move the cursor to the desired item and press Enter.

  Cancel a Print Job
  Show the Status of Print Jobs
  Prioritize a Print Job
  Hold / Release a Print Job
  Move a Job between Print Queues



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


Figure 16-30. Managing Print Jobs AU1412.0

Notes:

Who can manage print jobs?

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.

Cancel a Print Job

```
# smit qcan
```

```
                    Cancel a Print Job

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


                                        [Entry Fields]
PRINT QUEUE containing job              [ ]   +
(required for remote jobs)
* Print JOB NUMBER                      [ ]   +#



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


Figure 16-31. Cancel a Print Job AU1412.0

Notes:

Introduction

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.

Commands to cancel print jobs

To cancel a job you can either use the **smit qcan** fastpath, or use one of the following commands:

- **cancel** (System V)
- **lprm** (BSD)
- **qcan** (AIX)

Examples

To cancel job number 127 on whatever queue the job is on, you can use either of the following two commands:

- `# qcan -x 127`
- `# cancel 127`

To cancel all jobs queued on printer **lp0**, you can use either of the following two commands:

- `# qcan -X -Plp0`
- `# cancel lp0`

Job priority example

```
# qchk -L
Queue   Dev     Status  Job             Name   From           To
_______ ___     _______ Submitted        Rnk    Pri     Blks   Cp PP %
ps      lp0     DOWN
                QUEUED  569 /etc/qconfig root   root
                1/07/03  09:39:25
                                          1     15      2      1
                            /etc/qconfig

                QUEUED  570 /etc/motd    root   root
                1/07/03 09:40:15          2     15      1      1
                            /etc/motd
```

```
# qpri -#570 -a 25
# qchk -L
Queue   Dev     Status  Job             Name   From           To
_______ ___     ______  Submitted        Rnk    Pri     Blks   Cp PP %
ps      lp0     DOWN
                QUEUED  570 /etc/motd    root   root
                1/07/03 09:40:15          1      25     1      1
                            /etc/motd

                QUEUED  569 /etc/qconfig root   root
                        1/07/03 09:39:25   2     15      2      1
                                /etc/qconfig
```


Figure 16-32. Job priority example AU1412.0

Notes:

Processing order

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.

Changing print job priority

Each print job also has a priority that can be changed via SMIT (**`smit qpri`**) 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 **`qprt -R`** command can also be used to set job priority.

Example

The example in the visual 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.

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 16-33. Holding a job in a queue AU1412.0

Notes:

Holding and releasing a print job

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 visual 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

Queue Dev Status    Job  Files User   PP%  Blks Cp  Rnk
asc   lp0  DOWN
           QUEUE    11   /etc/qconfigroot  2    1   1
ps    lp0  READY
```

```
# qmov -mps -#11
# qchk -A

Queue Dev Status    Job  Files User   PP%  Blks Cp  Rnk
asc   lp0  DOWN
ps    lp0  RUNNING 11  /etc/qconfig root  2    1   1
```


Figure 16-34. Moving a job between queues AU1412.0

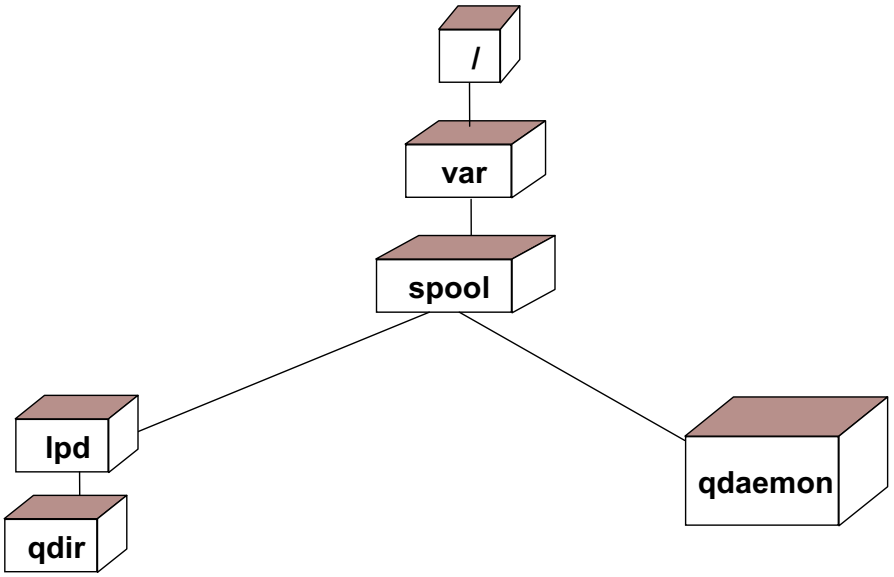
Notes:

Moving print jobs

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



- Contains queue requests (job description files)
- Temporary copies of enqueued files if spooling

Figure 16-35. Printing-related directories to monitor AU1412.0

Notes:

Why directories may fill up

The directories shown in the visual 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 the **/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.

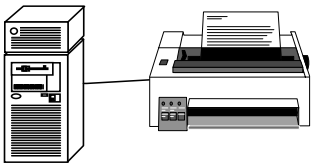
Printing problem checklist

```
# cat file  > /dev/lp0
```

Any output?

NO

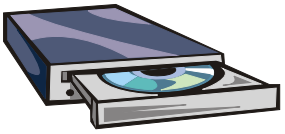
Check hardware



- ✓ Check physical cables
- ✓ Printer online and ready
- ✓ No paper jams
- ✓ Not out of paper

YES

Check software



- ✓ `qdaemon` running
- ✓ Check **/etc/qconfig**
- ✓ Queue enabled
- ✓ **/var** and **/tmp** not full

Figure 16-36. Printing problem checklist AU1412.0

Notes:

First step

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.

Check hardware

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.

Potential software problems

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 is enabled. If not, enable it.

```
# lpstat
```
or
```
# qprt -A
# enable queuename
```

Check to make /**tmp** and /**var** are not full with the command: **df**

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
```

   _____

Figure 16-37. Checkpoint (1 of 2) AU1412.0

Notes:

Checkpoint (2 of 2)

5. What three methods can be used to find out what the system default queue is?
   -
   -
   -

6. What users can bring print queues down?

   _____

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?

   _____

   _____

Figure 16-38. Checkpoint (2 of 2) AU1412.0

Notes:

Exercise 18: Printers and queues

- Add a printer and a queue
- Install printer support software (if needed)
- Check the queue
- Change the characteristics of a queue
- Manage jobs in queues
- Troubleshooting printer problems (optional)

Figure 16-39. Exercise 18: Printers and queues AU1412.0

Notes:

Introduction

This exercise gives you an opportunity to work with the AIX queuing system. If your classroom does not have locally attached printers, your instructor needs to supply you with local modification for this lab.

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

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 16-40. Unit summary AU1412.0

Notes:

Unit 17. 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*

Unit objectives

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

Figure 17-1. Unit objectives AU1412.0

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 17-2. What is TCP/IP? AU1412.0

Notes:

Networking architecture

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.

Protocols

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.

What does TCP/IP stand for?

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.

When and how was TCP/IP started?

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)

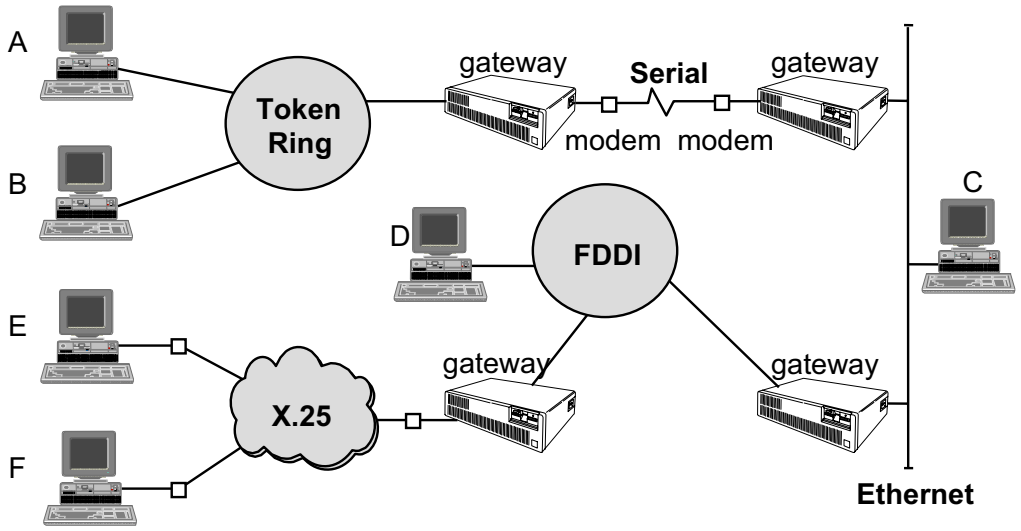
Most TCP/IP development is initiated by an organization called the Internet Architecture Board (IAB) which oversees development of the Internet network and the TCP/IP software it uses. Other TCP/IP development is performed by vendor organizations who write protocols which may become Internet standards.

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 17-3. An Internet AU1412.0

Notes:

Introduction

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).

Local Area Network (LAN)

Local Area Networks (LAN) are networks in a close geographical area. They often provide high-speeds over short distances:

- Token-Ring (4 or 16 Mb per second)
- Ethernet (10, 100, or 1000 Mb per second)
- FDDI (100 Mb per second)

Computers must connect directly to network media (via a transceiver or tap).

Wide Area Network (WAN)

Wide Area Networks (WAN) are networks that provide data communications capability throughout geographic areas larger than those serviced by LANs, such as across a country or across continents.

Computers often connected indirectly (for example, modems, public telephone networks) and generally have slower speeds than LANs.

Internet

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.

Host

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.

Gateway

A host which has interfaces on multiple networks and the ability to route traffic from one to another is called an IP Gateway or an IP Router. The ability to route between the attached networks is an important factor. A server which has interfaces on multiple networks for the purpose of improved availability or performance, is called a multihomed host or multihomed server. It is not recommended to configure a multihomed server to be router because the extra load of the routing will impact the server performance.

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 has one or more TCP/IP addresses:
  - For example: **10.0.0.3**
- If you know the address, but not the name, you can use some TCP/IP facilities with the address

Figure 17-4. Names and addresses AU1412.0

Notes:

Names and addresses

All network utilities need to know a computer system identification. This can be achieved by assigning each system a unique number.

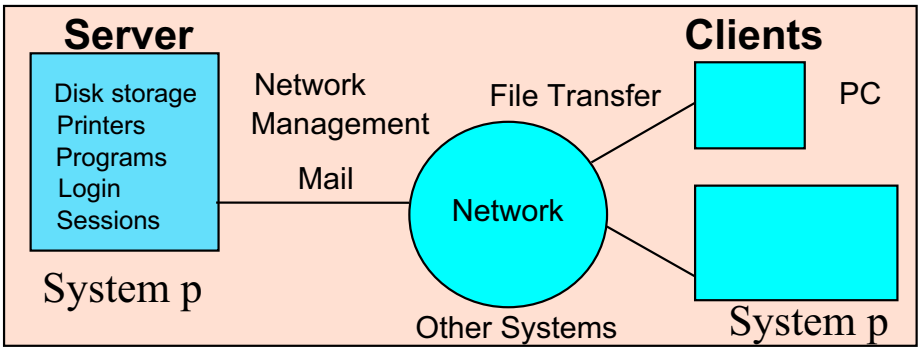
This appears in either a numerical format, for example, 234.56.78.91 or in a textual format (symbolic name), for example, **spud.maff.uk.gov**.

The symbolic name is translated into the numeric IP address by name resolution services, such as Domain Name Server (DNS). The above symbolic name example is what is called a fully qualified name which includes the names services domain name (**maff.uk.gov**).

The name services can often obtain a translation for an abbreviated form of the name (typically called the short name) when defined as an alias name or when it is in your default DNS domain. For example, the short name, **spud**, would translate to the same IP address as would **spud.maff.uk.gov**.

Some network utilities work with both the notations, while others (**rsh**, **rlogin**, **rcp**) must know of the textual form in order to work.

TCP/IP network facilities



- Standard TCP/IP facilities include: mail, file transfer, remote login, remote execution, and 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
  - Tivoli Netview for AIX

Figure 17-5. TCP/IP network facilities AU1412.0

Notes:

Network applications

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 are 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 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.
- The Web-based System Manager allows system management from a remote client over a TCP/IP Internet.

Information needed to configure TCP/IP

- Address:
  - Each adapter is given a unique TCP/IP address and often a subnet mask
  - These are usually assigned by your network administrator
- Name:
  - Each machine has a unique hostname
  - Each machine must have access to a table of name to address translations, which 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 17-6. Information needed to configure TCP/IP AU1412.0

Notes:

Introduction

Each system in a TCP/IP network must have a unique TCP/IP address and hostname. Your network administrator centrally manages tables of names and addresses, and assigns 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.

Address

AIX V4.3 and later 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 configuring IPV6.

Name resolution

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.

Two common mechanisms for name resolution are:

- 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 Server)

  Hosts are grouped together into domains which form a hierarchy (similar to the file directory structure). One (or more) hosts in a domain (called name servers) have a record of the name and address of all hosts. Client hosts request name to address translations from a name server. 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 name servers. This mechanism is more suitable for large networks because administration is centralized on a few machines.

Gateway

If your network is just part of a larger network then you need to know about the gateway machines which link your network to others. Most network designs only have one gateway, called the default gateway. You 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)                  [10.0.0.1]
  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)     [10.0.0.192]
       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
    Esc+5=Reset         Esc+6=Command       Esc+7=Edit          Esc+8=Image
    Esc+9=Shell         Esc+0=Exit          Enter=Do
```


Figure 17-7. Configuring TCP/IP AU1412.0

Notes:

Methods to configure TCP/IP

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 Web-based System Manager can also be used to configure the network.

Minimum configuration

The SMIT **Minimum Configuration & Startup** option (fastpath: **mktcpip**) or the **mktcpip** command can be used to quickly configure TCP/IP on AIX systems. This initializes TCP/IP (for client services) but further customization is 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.

Further configuration

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 and/or tabs. A '#'
# indicates the beginning of a comment; characters up to the end
# of the line are not interpreted by routines which search this
# file. Blank lines are allowed.

#Internet Address    Hostname             #Comments
127.0.0.1            loopback             localhost
10.0.0.1             sys1                 timeserver
10.0.0.2             sys2
10.0.0.3             sys3
10.0.0.4             sys4
```


Figure 17-8. Flat name resolution AU1412.0

Notes:

The /etc/hosts file

Host names and their Internet addresses are mapped by entries in the **/etc/hosts** file. In a flat network, there should be 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, except for the loopback address and the local machine's own assigned IP address(es).

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:
  - Example:

```
# hostname
sys3
```

- **host** command:
  - Examples:

```
# host  sys3
sys3 is 10.0.0.3,  Aliases: sys3.washington.ibm.com

# host  10.0.0.3
sys3 is 10.0.0.3,  Aliases: sys3.washington.ibm.com
```


Figure 17-9. Identifying the hostname AU1412.0

Notes:

Commands to determine the hostname or IP address

Two useful commands are **hostname** and **host**:

- **hostname** is used to determine the name of the machine.
- **host** determines the IP address, when used with the hostname. When used with the IP address, it determines the hostname.

Basic TCP/IP user functions

- The following commands work with any TCP/IP system (not just UNIX/AIX):
  - Test connectivity: **`ping`**
  - Remote execution: **`rexec`**
  - File transfer: **`ftp`**
  - Remote login: **`telnet`**

Figure 17-10. Basic TCP/IP user functions AU1412.0

Notes:

Introduction

The ARPA commands for testing connectivity, remote execution, file transfer, and remote login are **`ping`**, **`rexec`**, **`ftp`**, and **`telnet`** 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 always require a user name and password to be supplied when you establish a connection.

The `ping` command

The **`ping`** command tests connectivity with another system.

For example:

```
$ 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
$
```

The `rexec` command

The **`rexec`** command executes a command on another system. Some restrictions are:

- Cannot run interactive commands
- Cannot run commands that run full screen

If the command contains metacharacters for the remote system, they must be enclosed in quotes.

The command format is: **`rexec`** ***`host command`***.

For example:

```
sys1$_ rexec sys2 uname -x
Name (sys2:tom): tom
Password: tom's password
AIX sys2 526332 2 3 000003F41C00
sys1$ _
```

The `ftp` command

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` requires you to specify a user and password to establish a connection to the remote system. This user id 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 by **ftp** are:

- List, transfer, and delete local and remote files
- Change the current local and remote directory
- Create and remove directories

An 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**.

The telnet command

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 is passed correctly. Otherwise, you have to set the TERM variable after you log in.

Normally, **telnet** is executed with an argument of the hostname or IP address that you want to connect to. If you leave off the argument, you receive a **telnet>** prompt which accepts **telnet** subcommands. Use **?** or **help** to list the available subcommands. The most important **telnet** subcommands are **open**, and **quit**. If you specify a hostname or IP address after the **open** subcommand, it will establish a connection to that host. You

can terminate the **telnet** client by using the **quit** subcommand. Many of the subcommands are for managing an existing connection.

When in a **telnet** session with a remote host, you can obtain the **telnet>** prompt by issuing the escape sequence **<ctrl+]>** (press the **]** key while holding down the **Ctrl** key). The **telnet** client maintains the connection while allowing you to use **telnet** subcommands. If you then wish to obtain a shell prompt from your local client host, the telnet subcommand **z** places the **telnet** client in a suspended state and returns you to the parent shell. It tells you the job number of the suspended **telnet** process. From here, you can run other commands. If you wish to return to your **telnet** session, simply resume the suspended job with the **fg** command. The **telnet** process is resumed. If you just press **Enter**, then you are back in your interactive session with the remote host.

An example:

```
sys1$ telnet sys2
Trying . . .
Connected to sys2.
Escape character is '^]'.


AIX telnet (sys2)


login:  tom
password: tom's password
sys2$   ^]
telnet> ?
```

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? _____

Figure 17-11. Checkpoint AU1412.0

Notes:

Exercise 19: Networking

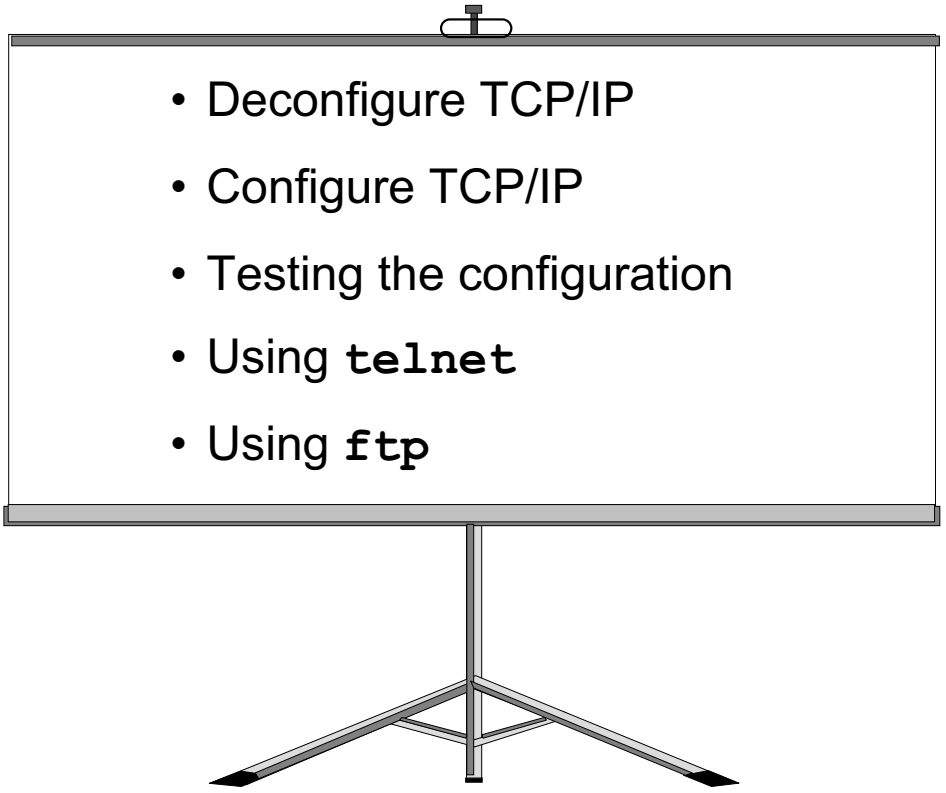


Figure 17-12. Exercise 19: Networking AU1412.0

Notes:

Introduction

This lab gives you an opportunity to configure an AIX system on a TCP/IP network. This gives you practical application of the concepts presented in this unit.

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

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.

Figure 17-13. Unit summary AU1412.0

Notes:

Appendix A. Checkpoint solutions

Unit 1: Introduction to IBM System p and AIX system administration

Checkpoint solutions

1. What type of adapter are you likely to require for communicating from a logical partition?
   a. Asynchronous
   b. Graphics
   c. Ethernet

2. True or False? The adapters seen by the AIX operating system, in an LPAR, may be either physical or virtual.
   True, with POWER5 the LPAR can have virtual SCSI and Virtual Ethernet adapters.

3. **True** or False? The `su` command allows you to get root authority even if you signed on using another user ID.
   But, you must also know the **root** password.

Unit 2: System Management Interface Tool (SMIT)

Checkpoint solutions

1. Specify 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. Specify two ways you can request the ASCII character version of SMIT from an X-windows environment command prompt:
   – `smitty`
   – `smit -C`

Unit 3: System start-up and shutdown

Checkpoint solutions

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?

   The initial process is `init`, which checks **/etc/inittab** for information regarding other processes that have to be started.

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

   The System Resource Controller (SRC)

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

Checkpoint solutions

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?
   `lslpp -l`
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)**?
   Install Software by default installs 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 5: AIX 6.1 installation

Checkpoint solutions

1. AIX V6.1 can that be installed from which of the following? (Select all are correct)
   a. 8 mm tape
   b. CD-ROM
   c. Diskette
   d. 4 mm tape
2. True or False? A Preservation Install preserves all data on the disks.

   Preserves SOME of the existing data on the disk selected for installation. Warning: This method overwrites the user (**/usr**), variable (**/var**), temporary (**/tmp**), and root (**/**) file systems. Other product (application) files and configuration data are destroyed.
3. What is the console used for during the installation process?

   The console is used to display all the system messages and interact with the installation.

Checkpoint solutions

1. True or False? The Web-based System Manager is available for client access automatically after the BOS is installed.

   False. The Web-based System Manager server must be configured and enabled for client access.

2. Which of the statements are true regarding the Web-based System Manager?

   a) An AIX V6.1 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 V6.1 system from a remote AIX V6.1 system using an ASCII terminal.

   C is false. However, with a graphics terminal it is possible to manage different systems simultaneously by adding the remote systems in the Navigation window of Web-based System Manager.

Unit 7: Devices

Checkpoint solutions

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

   No. The SCSI adapter itself uses ID 7. So, it cannot be used for other devices.

2. Use the output on the next visual (**`lsdev -C -H`**) to answer the following four questions.

   a) What happens if we attempt to add another device with the SCSI address set to 4?

   The operation fails as there is already a device (SCSI Disk Drive) configured at this location.

   b) Can the 8 mm tape drive be currently used? Why?

   No, because it is in the defined state. You have to first make it available by either using SMIT or the **`mkdev`** command.

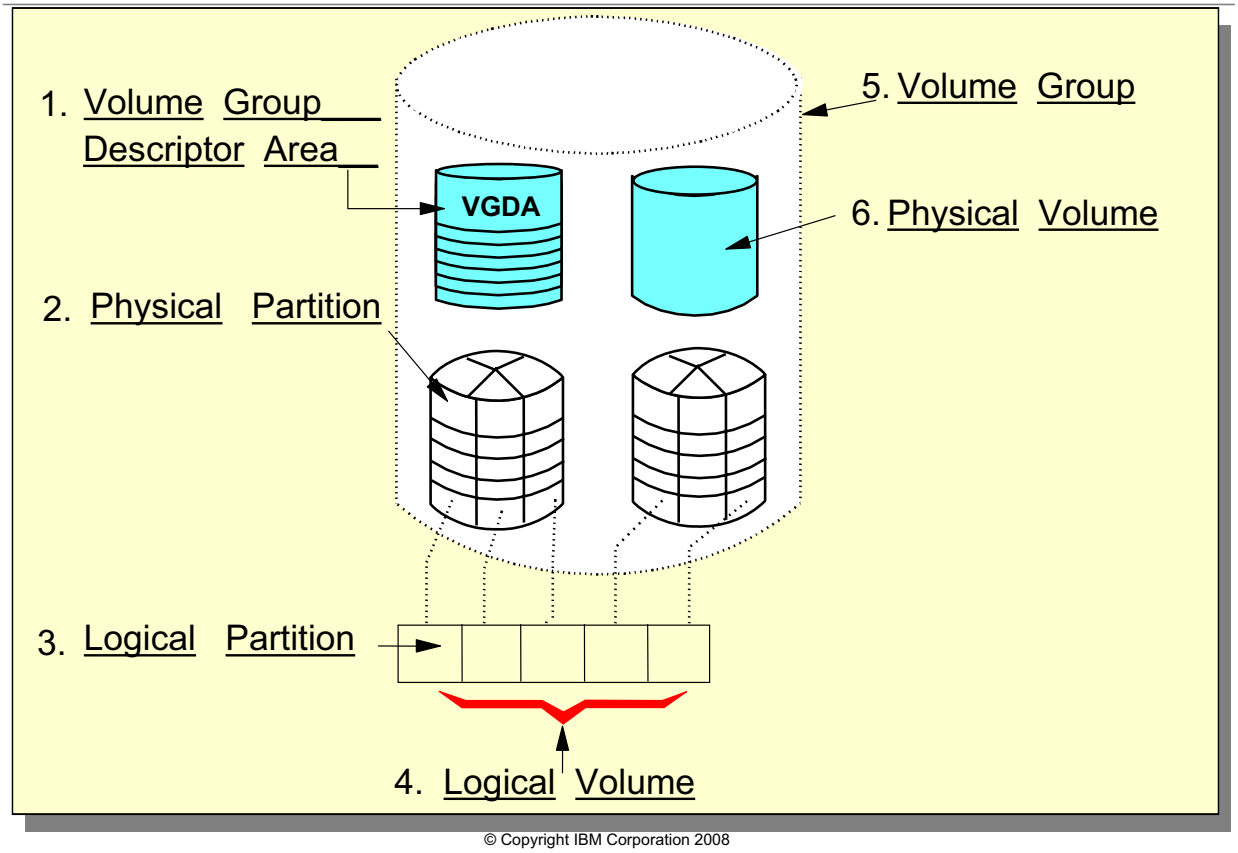
   c) Where is the printer connected? The parallel port

   d) The Ethernet adapter is installed in what slot?

   It is an integrated adapter which does not occupy a slot on the PCI bus.

Unit 8: System storage overview — Let's review

Let's review solution



Unit 8: System storage overview (1 of 3)

Checkpoint solutions (1 of 3)

1. How many different physical partition (PP) sizes can be set within a single VG? One
2. By default, how big are PPs? Traditionally 4 MB, but LVM chooses an optimal size based on the #PPs/PV and the size of largest PV in the VG.
3. How many volume groups (VGs) can a physical volume (PV) belong to?
   a) Depends on what you specify through SMIT
   b) Only one
   c) As many VGs as exist on the system
4. True or False? All VGDA information on your system is identical, regardless of how many volume groups (VGs) exist. False. All VGDAs within a VG are the same.

Checkpoint solutions (2 of 3)

Use the following output to answer the questions below:

```
# lsfs
Name        Nodename  Mount Pt   VFS    Size   Options  Auto   Accounting
/dev/hd4     --      /           jfs2   294912   --     yes       no
/dev/hd1     --      /home       jfs2   32768    --     yes       no
/dev/hd2     --      /usr        jfs2   3309568  --     yes       no
/dev/hd9var  --      /var        jfs2   65536    --     yes       no
/dev/hd3     --      /tmp        jfs2   131072   --     yes       no
/dev/hd10opt --      /opt        jfs2   163840   --     yes       no
/dev/cd0     --      /infocd     cdrfs            ro     yes       no
/dev/lv00    --      /home/john  jfs2   32768    rw     yes       no
/dev/hd11admin--     /admin      jfs2   262144   --     yes       no
```

5. With which logical volume is the **/home** file system associated? **/dev/hd1**
6. What type of file systems are being displayed? **Journaled file systems (JFS), enhanced journaled file systems (JFS2), and CD-ROM (CDRFS)**
7. What is the mount point for the file system located on the **/dev/lv00** logical volume? **/home/john**
8. Which are the system supplied logical volumes and their associated file systems? **/dev/hd4 (/), /dev/hd1 (/home), /dev/hd2 (/usr), /dev/hd9var (/var), /dev/hd3 (/tmp), /dev/hd10opt (/opt)**
9. Which file system is used primarily to hold user data and home directories? **/home**

Unit 8: System storage overview (3 of 3)

Checkpoint solutions (3 of 3)

Use the following output to answer the question below:

```
# 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   jfs2     1       1       1      open/syncd     /var
hd3      jfs2     2       2       1      open/syncd     /tmp
lv00     jfs2     1       1       1      closed/syncd   /home/john
```

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

    hd9var, hd3, and lv00

Checkpoint solutions

1. True or False? A logical volume can span more than one physical volume.
2. True or False? A logical volume can span more than one volume group.
3. True or False? The contents of a physical volume can be divided between two volume groups.
4. True or False? If mirroring logical volumes, it is not necessary to perform a backup. False. You still need to back up to external media.
5. True or False? SMIT can be used to easily increase or decrease the size of a logical volume. False. SMIT can only be used to increase a file system. Decreasing one requires backing up the file system, removing it, re-creating it, and then restoring.
6. True or False? Striping is done at a logical partition level. False. It is done at a stripe unit level.

Checkpoint solutions

1. Does the size of the file system change when the size of the logical volume it is on is increased? No

2. If a file system is the same size as the logical volume on which it sits, does the size of the logical volume increase when the size of the file system that is residing on it increases? Yes

3. If you remove a logical volume, is the file system that is residing on it removed as well?

   The contents are removed, but the information about the file system that is contained in **/etc/filesystems** is not removed.

Checkpoint solutions

1. What command can you use to determine if a file system is full? **df**

2. What two commands can be used to find the files and users that are taking the most disk space?
   - **du**
   - **ls -l**

3. True or False? It is good practice to run **fsck -y** on all file systems, even if they are mounted.

Unit 12: Paging space

Checkpoint solutions

1. What conclusions regarding potential paging space problems can you reach based on the following listing?

```
Page      Physical Volume Size  %Used Active Auto Type chksum
Space     Volume   Group

hd6       hdisk0   rootvg 64 MB 43%  yes    yes   lv    0
paging00  hdisk1   rootvg 64 MB  7%  yes    yes   lv    0
paging01  hdisk1   rootvg 16 MB 89%  yes    yes   lv    0
```

Obviously, it is difficult to come to any conclusions regarding the state of this system just by looking at a snapshot picture like the one above. However, at first glance, the following potential problems can be noticed:

- paging00 is underutilized, and it is too large. It needs to be reduced in size.
- paging01 is over utilized, and the size seems to be too small. It needs to be increased in size.
- Both user-defined paging spaces are on the same disk. It would be better if one of them were moved onto a disk which is less utilized.

2. True or False? The size of **paging00** (in the above example) can be dynamically decreased.

Unit 13: Backup and restore

Checkpoint solutions

1. What is the difference between the following two commands?
   a) find /home/fred | backup -ivf /dev/rmt0
   b) cd /home/fred; find . | backup -ivf /dev/rmt0
   (a) backs up the files using the full path names, whereas
   (b) backs up the file names using the relative path names.
   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?
   You would be at the start of the backed up images of the files, having skipped over the boot portion of the tape.
3. Which command could you use to restore these files? The files were backed up using the **backup** command so you would have to use the **restore** command.
4. True or **False**? **smit mksysb** backs up all file systems, provided they are mounted. **mksysb** only backs up **rootvg** file systems. To back up other volume groups, you must use the **savevg** command.

Unit 14: Security and user administration (1 of 2)

Checkpoint solutions (1 of 2)

1. What are the benefits of using the **su** command to switch user to **root** over logging in as **root**?

   A log (which can be monitored) of all users executing the **su** command is kept in the **sulog**.

2. Why is a **umask** of 027 recommended?

   This value removes all permission bits for the “others” category, which enhances security.

3. As a member of the **security** group, which password command would you use?
   pwdadm (This command does not prompt for the **root** password or the old password of the user whose password is being changed.)

4. Which password change command does SMIT use?
   passwd

5. True or False? When you delete a user from the system, all the user's files and directories are also deleted.

Checkpoint solutions (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? No, because the passwords are held 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**

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 (except **root**)
   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

Unit 15: Scheduling

Checkpoint solutions

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.

   False. Only one or the other of these files should be used.

2. Give a **crontab** entry that would specify that a job should run every Thursday at 10 past and 30 minutes past every hour.

   ```
   10,30 * * * 4 <job>
   ```

3. How would you schedule the script named **myscript**, to run 10 minutes from now?

   ```
   # at now + 10 minutes
   myscript
   ^d
   #
   ```

Unit 16: Printers and queues -- Let's review

Let's review solution

1. True or false? The **qdaemon** is responsible for printing jobs. False. The printer backend is responsible for printing. The qdaemon manages jobs in queue. The qdaemon hands the jobs off to the backend for printing.
2. To set up remote printing, what daemons are needed and do they run on the server, the client or both? **qdaemon** and **lpd** on the server **qdaemon** only on the client
3. What does the **up = TRUE** indicate in the **/etc/qconfig** file? It means the queue is accepting jobs. If it were FALSE, the user would be notified that the queue is not accepting jobs.
4. What does **discipline** mean in reference to the **/etc/qconfig** file? What are its possible values? **discipline** is read by **qdaemon** to determine the sorting order for jobs in the queue. The values supported are **fcfs** (first come first server) and **sjn** (shortest job next).

Unit 16: Printers and queues (1 of 2)

Checkpoint solutions (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.

   True. This can be accomplished using the **PRINTER** environment variable.

2. True or False? The **/etc/qconfig** file is read by the backend program to determine what the queue discipline is.

   False. It is read by **qdaemon**.

3. True or False? All printer software is automatically installed when you install the base operating system.

   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
```

   The **-c** flag produces a spool file.

Checkpoint solutions (2 of 2)

5. What three methods can be used to find out what the system default queue is?
   - First entry in **/etc/qconfig** file
   - The output from the **`qchk`** command with no options
   - The first queue listing from the **`lpstat`** command
6. What users can bring print queues down?

   The **root** user or members of the **printq** group.
7. True or False? Once the queue is down, no more jobs can be submitted to the printer.

   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?

   Yes, they can by only specifying a queue name and not individual job numbers.

Unit 17: Networking overview

Checkpoint solutions

1. What are the following commands used for?
   - **`ftp`** transfers files from one machine to another
   - **`rexec`** executes a command on a remote system
   - **`telnet`** logins to another system

2. What is the difference (if any) between a host and a gateway?
   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 networks together.

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? /etc/hosts

Appendix B: Configuring AIX 5L V5.2 documentation (1 of 2)

Checkpoint solutions (1 of 2)

1. True or false? 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.
2. True or false? The AIX 5L V5.2 documentation is viewed using a Web browser.
3. True of false? The Web-based System Manager is available for client access automatically after the BOS is installed.

Appendix B: Configuring AIX 5L V5.2 documentation (2 of 2)

Checkpoint solutions (2 of 2)

4. Which of the statements are true regarding the Web-based System Manager?
   a. An AIX 5L V5.2 system can be managed from a remote PC with appropriate JAVA and Web-browser code installed.
   b. In standalone mode use the **wsm** command to access the Web-based System Manager.
   c. It is possible to manage an AIX 5L V5.2 system from a remote AIX 5L V5.2 system using an ASCII terminal.
   d. The Web-based System Manager includes TaskGuides that direct the user through complex tasks.

   a, b, d are true.

   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 Web-based System Manager.

Appendix F: Serial devices

Checkpoint solutions

1. True or false? If a device, like a TTY, is left for **`cfgmgr`** to configure automatically, it picks up the default values which might not be desirable. **TTYs and other serial devices are not self-configurable and so are not detected by `cfgmgr`.**
2. True or false? If TTYs are connected via concentrator boxes, they must all be connected in sequence on the concentrator box otherwise they are not configured. **TTYs can be connected in any order on the concentrator boxes. However, the management of these is obviously more difficult.**
3. True or false? **/dev/tty0** indicates that the TTY is connected to port 0, **/dev/tty1** to port 1 and so on. **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 is always 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: The System V print subsystem — Let's review (1 of 2)

Let's review solution 1

1. What command is used to display which print subsystem is active? **`switch.prt -d`**
2. When the System V print subsystem is active, /usr/bin/cancel is linked to **/usr/sysv/bin/cancel**.
3. The **`lp`** or **`lpr`** commands can be used to submit print jobs to the System V print service.
4. **`lpsched`** is the System V print service daemon.
5. Slow filters are executed by lpsched and do NOT interact with the printer.
6. Fast filters are executed by the interface program and DO interact with the printer.
7. The printer type associates a printer with an entry in **terminfo**.

Appendix G: The System V print subsystem — Let's review (2 of 2)

Let's review solution 2

1. A class is a group of printers.
2. Use the **`accept`** command to enable a printer to begin accepting print requests.
3. Use the **`enable`** command to enable a printer to begin printing print requests.
4. AIX 5L includes two interface programs standard and PS.
5. If you want the print service to copy a file to the spool directory, you should use the **`-c`** option to the **`lp`** command.
6. Use **`lpstat -o`** to display status of outstanding print requests.
7. The System V print log file is **/var/lp/logs/requests**.

Appendix G: The System V print subsystem

Checkpoint solutions

1. List two advantages of the System V print subsystem. Compatibility, Availability of interface programs, Security, Support for Forms, Standard PostScript filters, Long term direction
2. List two advantages of the AIX print subsystem. 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? **`switch.prt -s SystemV`**
4. **`lpsched`** uses information in the printer configuration file and **terminfo** to screen print jobs.
5. The interface program uses commands in **terminfo** to initialize the printer.
6. Filters are used to convert file content.
7. Use the **`lpfilter`** command to manage filters.
8. **`lpadmin`** is used to create or modify a System V printer.
9. **`mkdev`** is used to create a printer device.

Appendix B. Configuring AIX 5L V5.2 documentation

What this unit is about

This unit introduces the tools that are available for AIX 5L V5.2 system administration.

What you should be able to do

After completing this unit, you should be able to:

- In AIX 5L V5.2:
  - Set up and use AIX Web-based documentation
  - Define the use of the Web-based System Manager

How you will check your progress

Accountability:

- Checkpoint questions
- Activities

References

Online	*AIX 5L Version 5.2 Web-based System Manager Administration Guide*
SG24-5765	*AIX 5L Differences Guide Version 5.2 Edition*

Unit objectives

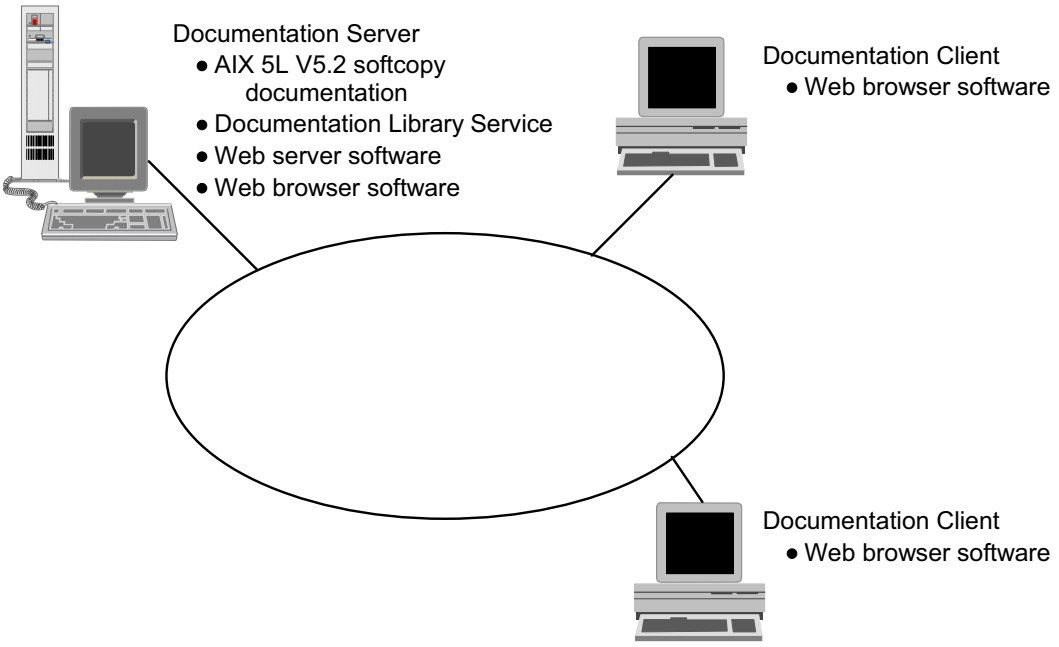
After completing this unit, you should be able to:

- In AIX 5L V5.2:
  - Set up and use AIX Web-based documentation
  - Define the use of the Web-based System Manager

Figure B-1. Unit objectives AU1412.0

Notes:

Configuring AIX 5L V5.2 documentation



View AIX documentation from anywhere with a browser

Figure B-2. Configuring AIX 5L V5.2 documentation AU1412.0

Notes:

Overview

In addition to providing SMIT to make system administration tasks easy, 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 5L V5.2 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 B-3. Configuring AIX 5L V5.2 online documentation AU1412.0

Notes:

Configuration steps

The steps outlined above are used to configure an AIX 5L V5.2 documentation server or online documentation for a stand-alone RS/6000 system.

1. Configure TCP/IP on the AIX system.
2. Install the Web server software. AIX 5L V5.2 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 5L V5.2. Actually, any browser can be used, provided it supports Java 1.3.
5. The AIX 5L V5.2 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.

Client configuration steps

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 is discussed in the AIX Installation unit.

Internet and Documentation Services

```
# smit web_configure
```

```
                Internet and Documentation Services

Move cursor to desired item and press Enter.

   Change / Show Default Browser
   Show Documentation and Search Server
   Change Documentation and Search Server
   Change/Show Default Documentation Language
   Web-based System Manager





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


Figure B-4. Internet and Documentation Services AU1412.0

Notes:

Begin configuration

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 5L V5.2 online documentation.

Change/Show Default Browser

```
                Change/Show Default Browser

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


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





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


Figure B-5. Change/Show Default Browser AU1412.0

Notes:

What browser?

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]
Documentation search server LOCATION      None - disabled     +
```

```
            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 B-6. Change Documentation and Search Server AU1412.0

Notes:

Location of the documentation server

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 stand-alone AIX 5L V5.2 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 (1 of 2)

```
          Change Local Documentation and Search Server

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

                          [Entry Fields]
  Web server SOFTWARE             IBM HTTP Server Web se>     +

                      Web server SOFTWARE

 Move cursor to desired item and press Enter.

  Lite NetQuestion web server
  IBM HTTP Server Web server in default location
  Lotus Domino Go Web server in default location
  IBM Internet Connection Server (IICS) in default location
  Other local server or one of the above in non-default location

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


Figure B-7. Change Local Documentation and Search Server (1 of 2) AU1412.0

Notes:

Define the Web server software

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

Change Local Documentation and Search Server (2 of 2)

```
                Change Local Documentation and Search Server

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


                              [Entry Fields]
    Web server SOFTWARE                        IBM HTTP Server Web se>
*   Local web server PORT number              [80]                   #
*   Local web server cgi-bin DIRECTORY        [/usr/HTTPServer/cgi-bi>
*   Local web server HTML document directory  [/usr/HTTPServer/htdocs>
*   For versions prior to 1.3.6.0, Auto-start  Yes
     server if not already running




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


Figure B-8. Change Local Documentation and Search Server (2 of 2) AU1412.0

Notes:

Changing Web servers

If using the IBM HTTP Server Web server, Lotus Domino Go Web server or the IBM Internet Connection Server, this menu is filled out automatically. Update this screen if changing the defaults or using other Web server software to access the AIX 5L V5.2 online documentation.

Note: When using the IBM HTTP Server Web server to allow other systems to use this system as a documentation server, you must configure the server name manually. Follow these steps to accomplish this:

1. Edit the file: **/usr/HTTPServer/conf/httpd.conf**
2. Change the line:

   ```
   # ServerName   new.host.name
   ```

   to

ServerName ***YourSystemName***

(Take out the comment (#) and insert the system's host name for **new.host.name)**

3. Reboot the system or run the command: **/usr/HTTPServer/bin/httpd**

AIX 5L V5.2 documentation

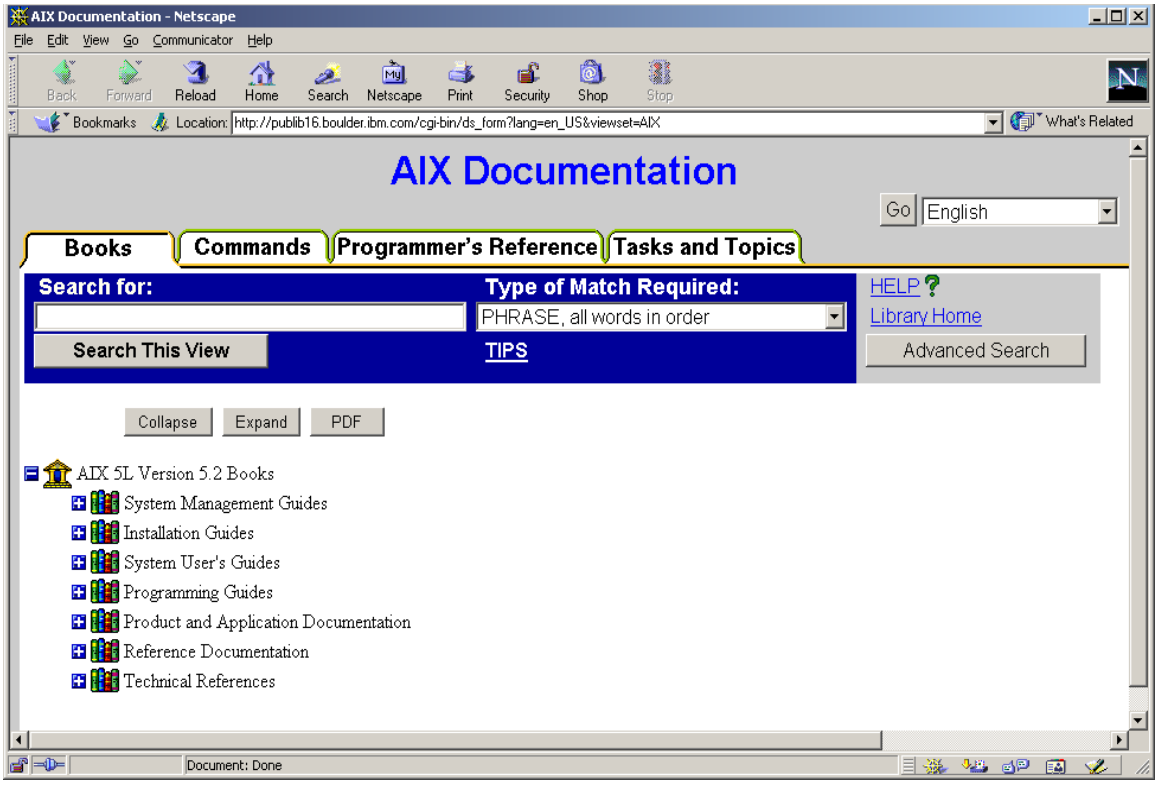


Figure B-9. AIX 5L V5.2 documentation AU1412.0

Notes:

Accessing the documentation

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 5L V5.2 documentation

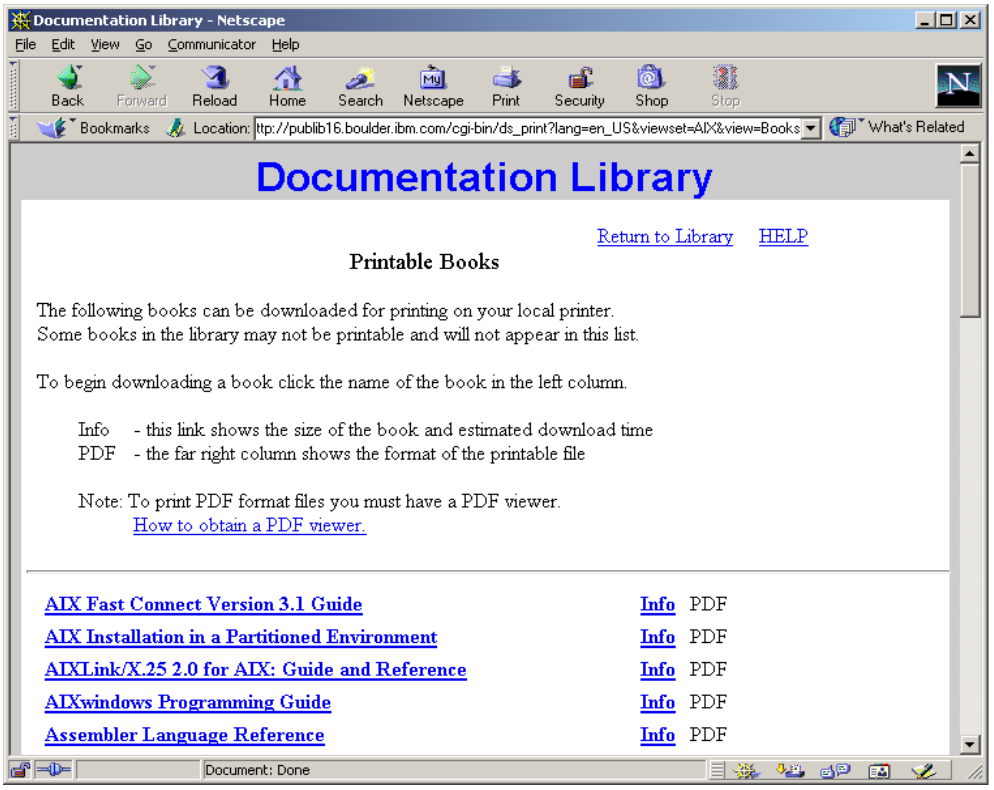


Figure B-10. Print AIX 5L V5.2 documentation AU1412.0

Notes:

Printing documentation

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 5L V5.2 documentation

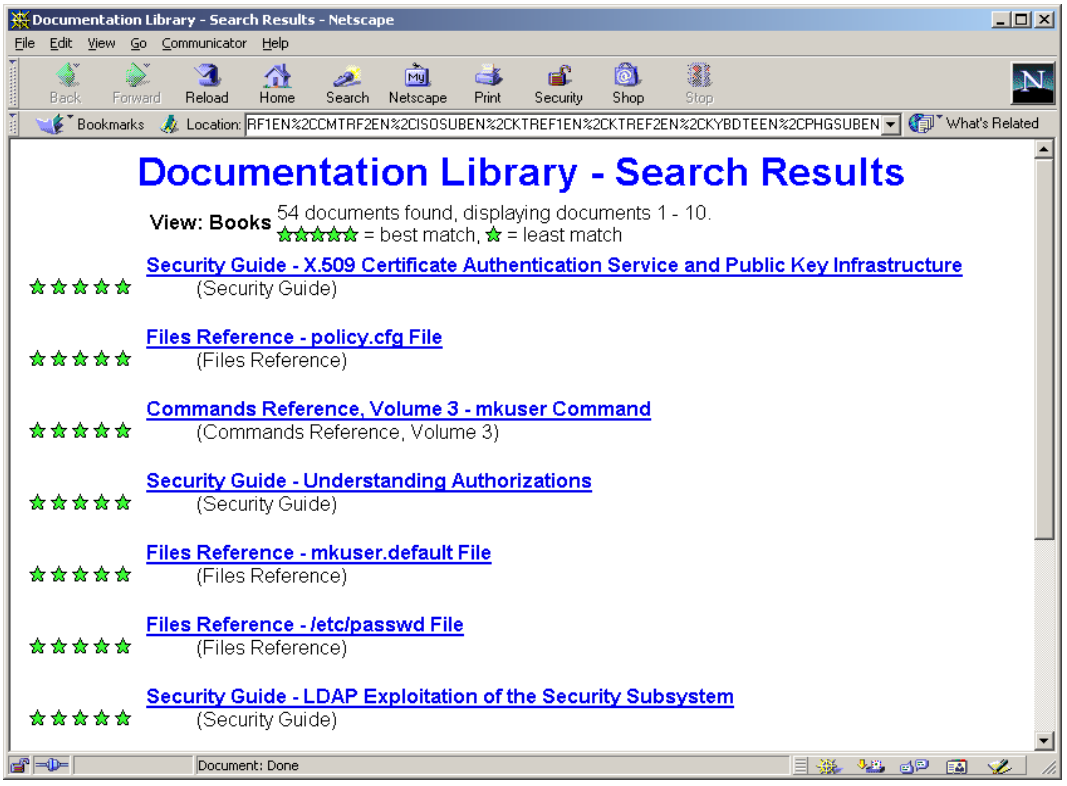


Figure B-11. Search AIX 5L V5.2 documentation AU1412.0

Notes:

Searching documentation

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

Above are 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 takes you to that document.

IBM pSeries Information Center

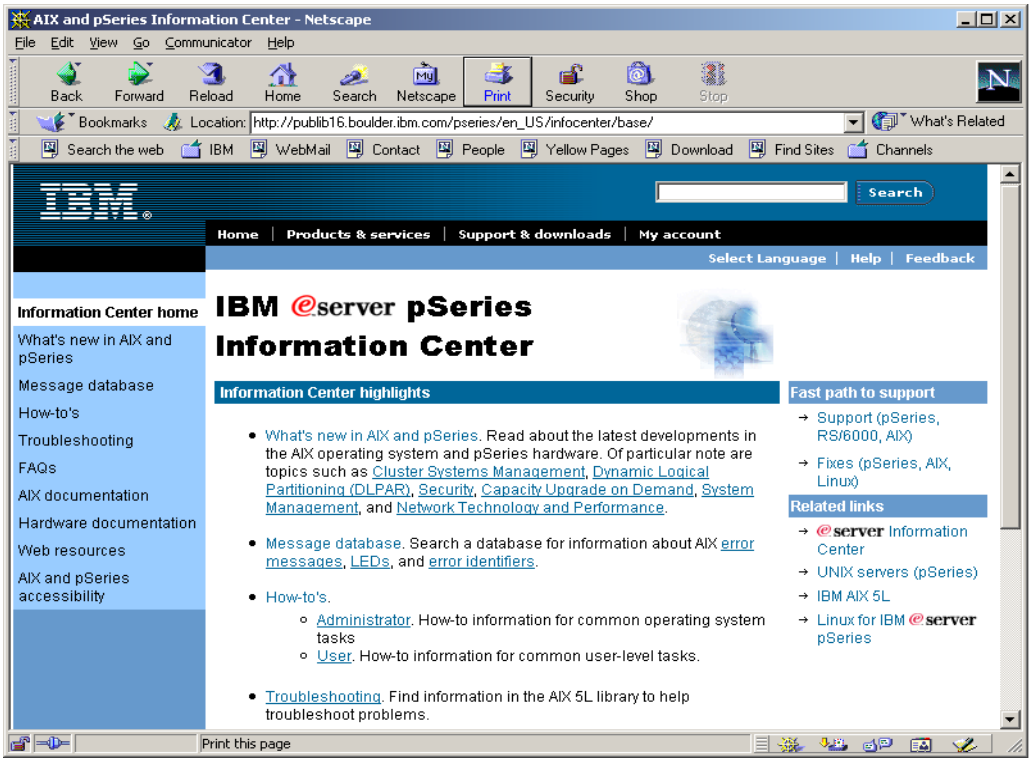


Figure B-12. IBM pSeries Information Center AU1412.0

Notes:

Accessing the Information Center

The IBM @server pSeries Information Center is a Web Site that serves as a focal point for all information pertaining to pSeries and AIX. It provides access to the AIX V4.3, AIX 5L V5.1 and V5.2 documentation, as well as access to a message database to search on error numbers, identifiers and LEDs. FAQs, How-To's, and many more features are provided.

- You can access the Information Center by using the URL:

  http://publib16.boulder.ibm.com/pseries/en_US/infocenter/base

- Run the command **`infocenter`** from the command line. This command starts the default browser with the URL previously mentioned.
- Start the Information Center with the **Information Center icon** located on the Help panel of the CDE desktop.

Activity: Configuring Web-based documentation

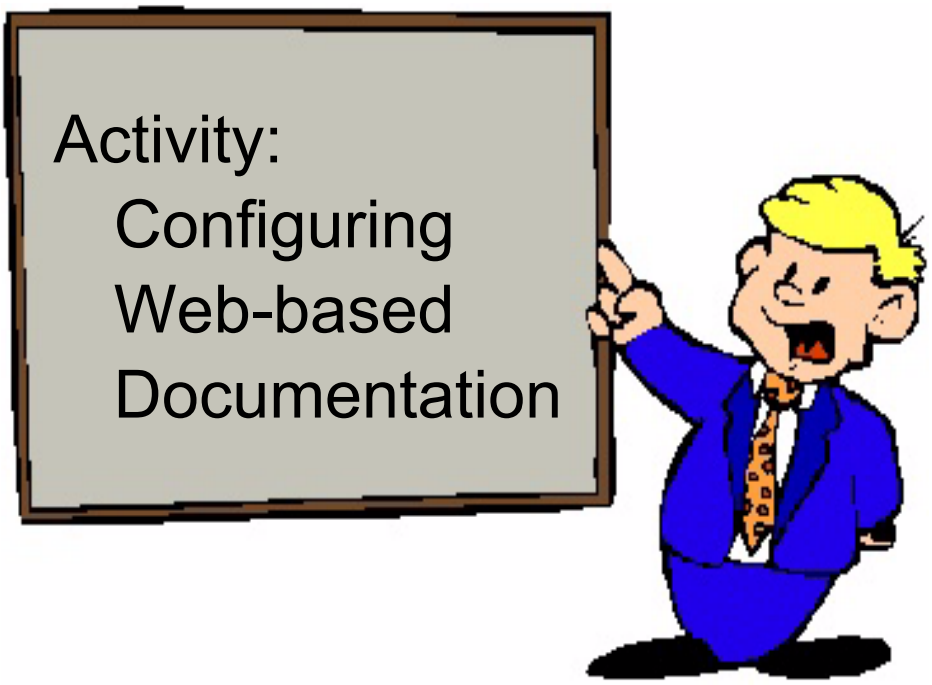


Figure B-13. Activity: Configuring Web-based documentation AU1412.0

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 allows you to access the AIX 5L V5.2 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 are 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 5L V5.2 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 permits, 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 allows you to access the AIX 5L V5.2 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 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 expands to display additional fields. If you are using the IBM HTTP Server Web server, the fields are already be filled in with the correct values. (In your own environment, if you are using some other Web server software, you 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 are 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 5L V5.2 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 5L V5.2 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

B.1. Web-based System Manager

Web-based System Manager

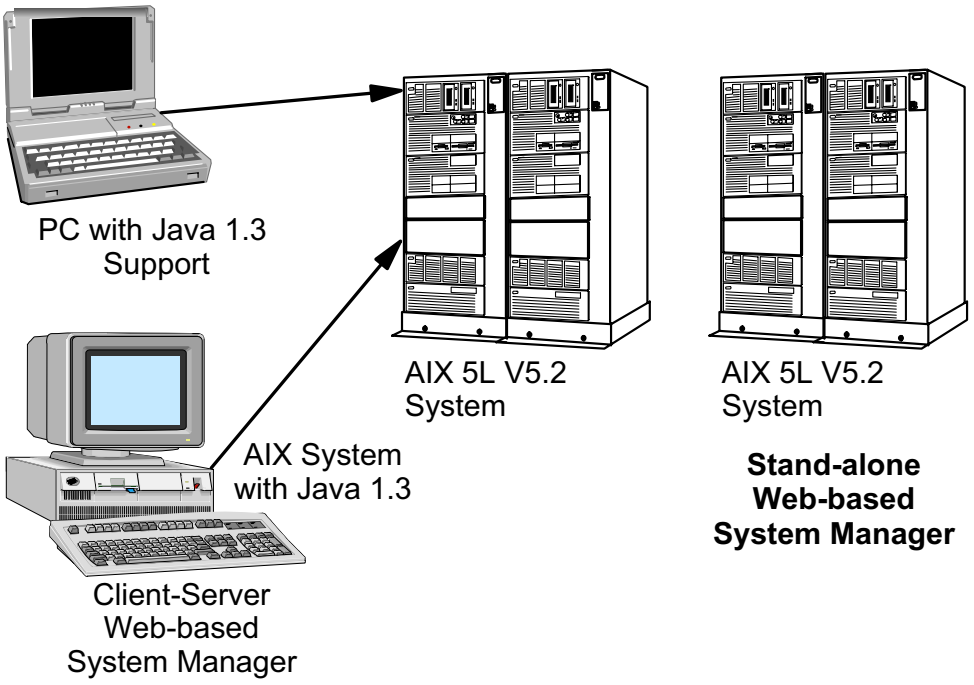


Figure B-14. Web-based System Manager AU1412.0

Notes:

Introduction

AIX V4.3 introduced the Web-based System Manager, which is the next step in the evolution of AIX system administration tools. There have been many enhancements to the Web-based System Manager since AIX 5L V5.1 it is called the default system administration tool for AIX.

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.

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 NT/2000/XP.

Web-based System Manager objectives

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 with a browser that supports Java 1.3 or use downloaded client code from an AIX 5L V5.2 code
- Enable AIX systems to be administered remotely
- Provide a system administration environment that provides a similar look and feel to the Windows NT/2000/XP, LINUX 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 needs to be used in the ASCII environment.

Client requirements

To download Web-based System Manager Client code from an AIX host use the address *http://<hostname>/remote_client.html*.

Supported clients are Microsoft Windows NT/2000/XP and RedHat Linux 7.2 and 7.3.

To download Windows Web-based System Manager Client code from an AIX host and start Install Shield use the address: *http://<hostname>/wsmship/pc_client/setup.html*.

The Windows Web-based System Manager Client installation needs around 64 MB disk space.

Accessing the Web-based System Manager

Standalone
 # wsm

Client-Server
With browser, URL:
http://<*hostname*>/wsm.html

As Standalone Java application
Double-click on the Web-based SystemManager Remote Client icon

From AIX client:
 # wsm -host <*hostname*>

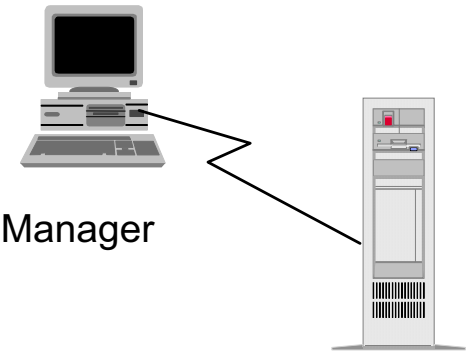


Figure B-15. Accessing the Web-based System Manager AU1412.0

Notes:

Stand-alone mode

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.

Client-server mode

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 use the appropriate URL to access the tool. The default URL is *http://<hostname>/wsm.html*. Be aware that AIX 5L V5.1 is using Java 1.3.0 and AIX

5L V5.2 is using Java 1.3.1 and that your browser plug-in-version must be compatible to the Java version on the AIX server.

- If the Web-based System Manager client is running as a stand-alone Java application, double click on the Web-based System Manager remote client icon.
- From an AIX 5L 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)

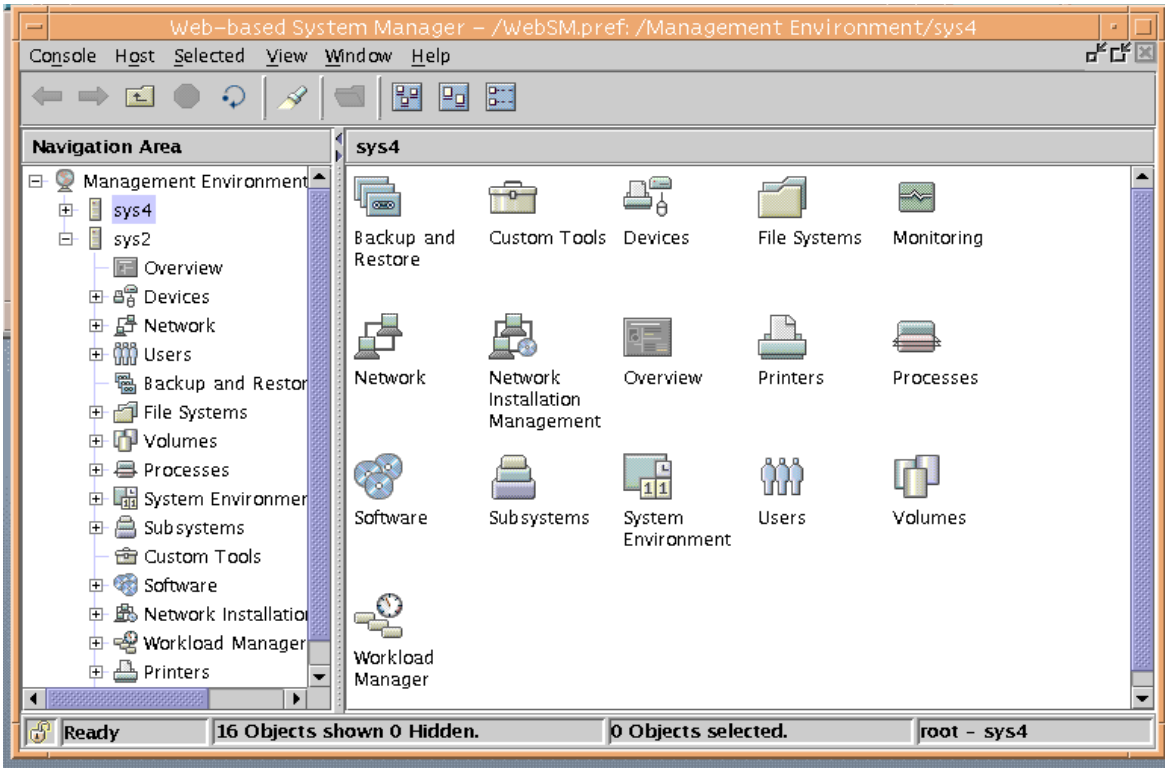


Figure B-16. Using the Web-based System Manager (1 of 3) AU1412.0

Notes:

Console window

This visual 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.

Session log

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)

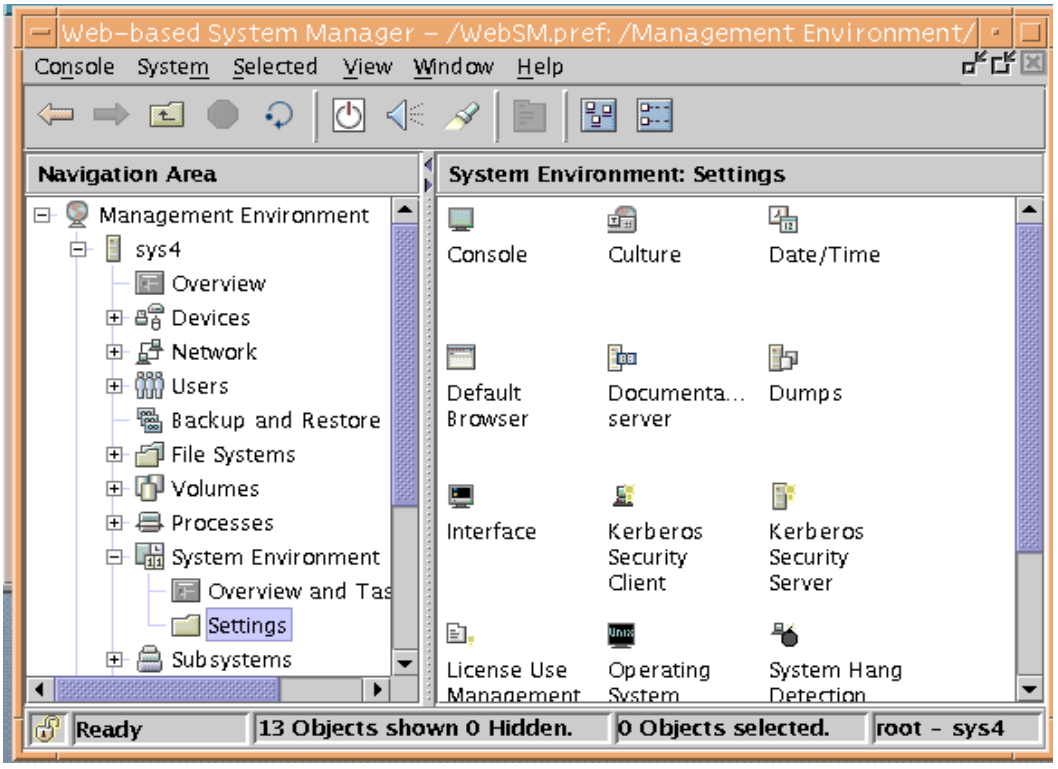


Figure B-17. Using the Web-based System Manager (2 of 3) AU1412.0

Notes:

Toolbar functions

The visual shows a Web-based System Manager window with the **System Environment: Settings** plug-in running in the **Contents** area. Above the **Navigation** Area and **Contents** Area, there is a toolbar with symbols for managing the window. 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.

Date/Time icon

If you select the **Date/Time** icon, this allows you to set the date and time on the system.

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

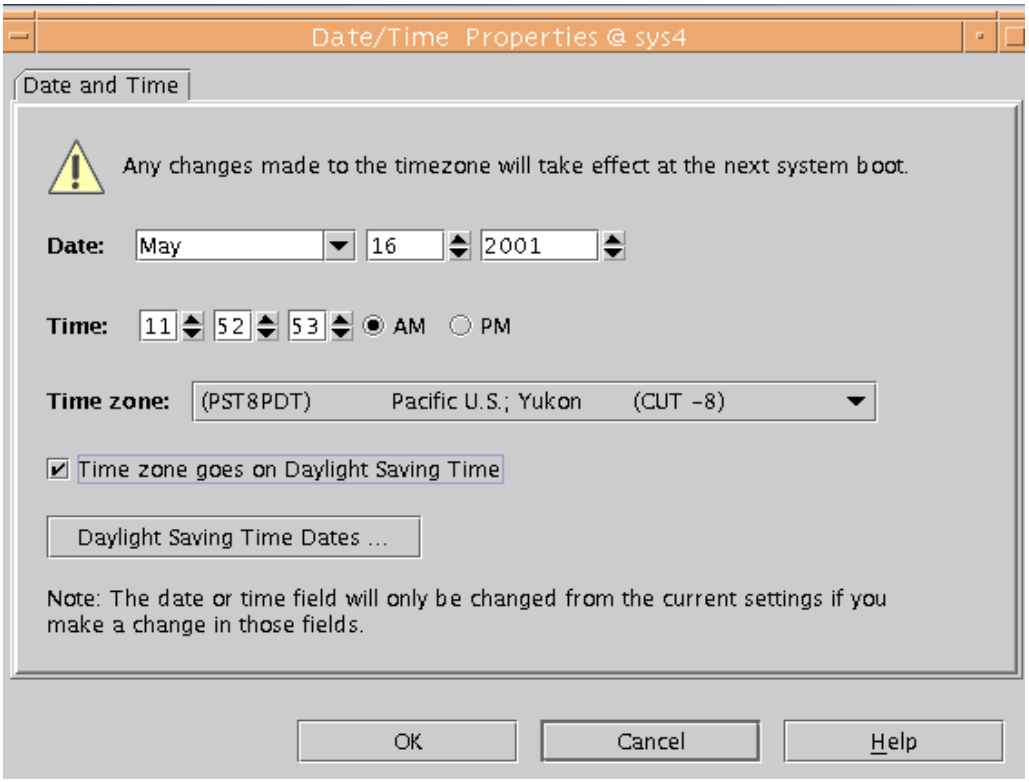


Figure B-18. Using the Web-based System Manager (3 of 3) AU1412.0

Notes:

Overview

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-1.ibm.com/servers/aix/wsm/ .*

Configuring the client/server Web-based System Manager

- Install the Web server
- Test the Web server
- Install the Web-based System Manager (usually done by default with the base)
- Define the Web server document directory

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

- Enable the Web-based System Manager server

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


Figure B-19. Configuring client/server Web-based System Manager AU1412.0

Notes:

Setting up the Web server

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.

The Web-based System Manager is installed by default in AIX 5L V5.1 and V5.2. The following filesets are installed from the AIX 5L 5.2 Base Installation media:

sysmgt.help.en_US.websm
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.websm.Webaccess

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

Next, enable the Web-based System Manager server
`/usr/websm/bin/wsmserver -enable`

This can also be done through smit using the fastpath
`smit web_based_system_manager` .

Which automatically runs
`/usr/websm/bin/wsmserver -enable`

To access the Web-based System Manager from the client machine, use the URL:
http://<hostname>/wsm.html

Checkpoint (1 of 2)

1. True or false? 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.
2. True or false? The AIX 5L V5.2 documentation is viewed using a Web browser.
3. True of false? The Web-based System Manager is available for client access automatically after the BOS is installed.

Figure B-20. Checkpoint (1 of 2) AU1412.0

Notes:

Checkpoint (2 of 2)

4. Which of the statements are true regarding the Web-based System Manager?
   a. System Manager includes TaskGuides that direct the user through An AIX 5L V5.2 system can be managed from a remote PC with appropriate JAVA and Web-browser code installed.
   b. In standalone mode use the **wsm** command to access the Web-based System Manager.
   c. It is possible to manage an AIX 5L V5.2 system from a remote AIX 5L V5.2 system using an ASCII terminal.
   d. The Web-based complex tasks.

Figure B-21. Checkpoint (2 of 2) AU1412.0

Notes:

Unit summary

- 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 documentation searches are performed, they are done on the server computer and the results are then sent back to the user on the client computer.
- The Web-based System Manager supports system administration tasks in a standalone or client-server environment.

Figure B-22. Unit summary AU1412.0

Notes:

Appendix C. 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 **-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 and * 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. • **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) • **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/tty*nn***)

cal Calendar (**cal *month year***)

cut Cut out specific fields from each line of a file

diff Differences between two files

find Find files anywhere on disks. Specify location by path (will search all subdirectories under specified directory).

- **-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 command interactively with results of **find** command)
- **-o** (logical or)
- **-print** (display results. Usually included.)

find syntax: **find *path expression action***

For example:

- **find / -name "*.txt" -print**
- **find / -name "*.txt" -exec li -l {} \;**

(Executes **li -l** where names found are substituted for **{}**) **;** indicates end of command to be executed and **** removes usual interpretation as command continuation character)

grep Search for pattern, for example, **grep pattern files**. **pattern** can include regular expressions.

- **-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)

Expression metacharacters:

- [ ] 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 Sort and merge files
-r (reverse order); **-u** (keep only unique lines)

Editors

ed Line editor

vi Screen editor

INed LPP editor

emacs Screen editor +

Shells, redirection, and pipelining

< (read) Redirect standard input, for example, ***command < file*** reads input for ***command*** from ***file***.

> (write) Redirect standard output, for example, ***command > file*** writes output for ***command*** to ***file*** overwriting contents of ***file***.

>> (append) Redirect standard output, for example, ***command >> file*** appends output for ***command*** to the end of ***file***.

2> Redirect standard error (to append standard error to a file, use ***command 2>> file***) combined redirection examples:

- ***command < infile > outfile 2> errfile***
- ***command >> appendfile 2>> errfile < infile***

; Command terminator used to string commands on single line

| Pipe information from one command to the next command. For example, **ls | cpio -o > /dev/fd0** passes the results of the **ls** command to the **cpio** command.

\ Continuation character to continue command on a new line. Will be prompted with **>** for command continuation.

tee Reads standard input and sends standard output to both standard output and a file. For example,
ls | tee ls.save | sort results in ls output going to **ls.save** and piped to **sort** command.

Metacharacters

`*`	Any number of characters (0 or more)
`?`	Any single character
`[abc]`	`[ ]` any character from the list
`[a-c]`	`[ ]` match any character from the list range
`!`	Not any of the following characters (for example, leftbox !abc right box)
`;`	Command terminator used to string commands on a single line
`&`	Command preceding and to be run in background mode
`#`	Comment character
`\`	Removes special meaning (no interpretation) of the following character Removes special meaning (no interpretation) of character in quotes
`"`	Interprets only `$`, backquote, and `\` characters between the quotes.
`'`	Used to set variable to results of a command for example, **`now='date'`** sets the value of now to current results of the **`date`** command.
`$`	Preceding variable name indicates the value of the variable.

Physical and logical storage

`chfs`	Changes file system attributes such as mount point, permissions, and size
`compress`	Reduces the size of the specified file using the adaptive LZ algorithm
`crfs`	Creates a file system within a previously created logical volume
`extendlv`	Extends the size of a logical volume
`extendvg`	Extends a volume group by adding a physical volume
`fsck`	Checks for file system consistency, and allows interactive repair of file systems
`fuser`	Lists the process numbers of local processes that use the files specified
`lsattr`	Lists the attributes of the devices known to the system

lscfg	Gives detailed information about the AIX system hardware configuration
lsdev	Lists the devices known to the system
lsfs	Displays characteristics of the specified file system such as mount points, permissions, and file system size
lslv	Shows you information about a logical volume
lspv	Shows you information about a physical volume in a volume group
lsvg	Shows you information about the volume groups in your system
lvmstat	Controls LVM statistic gathering
migratepv	Used to move physical partitions from one physical volume to another
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
unmount	Exactly the same function as the umount command
varyoffvg	Deactivates a volume group so that it cannot be accessed
varyonvg	Activates a volume group so that it can be accessed

Variables

=	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/rmt0 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 file system created with **backup -level -u;**

for example, **restore -xv -f/dev/rmt0**

cpio Copies to and from an I/O device. Destroys all data previously on tape or diskette. For input, must be able to place files in the same relative (or absolute) path name as when copied out (can determine path names with **-it** option). For input, if file exists, compares last modification date and keeps most recent (can override with **-u** option).

- **-o** (output)
- **-i** (input),
- **-t** (table of contents)
- **-v** (verbose),
- **-d** (create needed directory for relative path names)
- **-u** (unconditional to override last modification date)

  for example, **cpio -o > /dev/fd0** or
  cpio -iv file1 < /dev/fd0

tapechk Performs simple consistency checking for streaming tape drives

tcopy Copies information from one tape device to another

tctl Sends commands to a streaming tape device

tar Alternative utility to 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 directory
uux	Execute on remote system (UNIX to UNIX execute)

System administration

df	Display file system 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 file system 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 `nl` 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 required)
Integer operators:
`-eq (=)` `-lt (<)` `-le (=<)`
`-ne (<>)` `-gt (>)` `-ge (=>)`
String operators:
`=` `!= (not eq.)` `-z (zero length)`
File status (for example, `-opt file1`)

- `-f` (ordinary file)
- `-r` (readable by this process)
- `-w` (writable by this process)
- `-x` (executable by this process)
- `-s` (non-zero length)

while loop

while *test expression*
do
command
done

Miscellaneous

sh
Execute shell script in the **sh** shell
-x (execute step by step - used for debugging shell scripts)

vi Editor

Entering vi

vi *file*
Edits the file named file

vi *file file2*
Edit files consecutively (via :n)

.exrc
File that contains the **vi** profile

wm=*nn*
Sets wrap margin to ***nn***. Can enter a file other than at first line by adding + (last line), +n (line n), or +/pattern (first occurrence of pattern).

vi -r
Lists saved files

vi -r file
Recover file named file from crash

:n
Next file in stack

:set all
Show all options

:set nu
Display line numbers (off when **set nonu**)

:set list
Display control characters in file

`: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
***n*G**	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 character in current line
o	Add line following current line
O	Add line before current line
<esc>	Return to command mode

Deleting text

<Ctrl>w	Undo entry of current word
@	Kill the insert on this line
x	Delete current character
dw	Delete to end of current word (observe punctuation)
dW	Delete to end of current word (ignore punctuation)
dd	Delete current line
d	Erase to end of line (same as d$)
d)	Delete current sentence
d}	Delete current paragraph
dG	Delete current line thru end-of buffer
d^	Delete to the beginning of line
u	Undo last change command
U	Restore current line to original state before modification

Replacing text

r*a*	Replace current character with ***a***
R	Replace all characters overtyped until <esc> is entered
s	Delete current character and append test until **<esc>**.
s/*s1*/*s2*	Replace ***s1*** with ***s2*** (in the same line only)
S	Delete all characters in the line and append text
cc	Replace all characters in the line (same as **S**)

ncx	Delete ***n*** text objects of type ***x***, w, b = words,) = sentences, } = paragraphs, $ = end-of-line, ^ = beginning of line) and enter append mode
C	Replace all characters from cursor to end-of-line.

Moving text

p	Paste last text deleted after cursor (**xp** will transpose 2 characters)
P	Paste last text deleted before cursor
nYx	Yank ***n*** text objects of type ***x*** (w, b = words,) = sentences, } = paragraphs, $ = end-of-line, and no "x" indicates lines. Can then paste them with ***p*** command. Yank does not delete the original.
"ayy"	Can use named registers for moving, copying, cut/paste with "ayy" for register a (use registers a-z). Can then paste them with **ap** command.

Miscellaneous

.	Repeat last command
J	Join current line with next line

Appendix D. 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 are 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, 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:

```
#!/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 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 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
```

The following shell script, **fragcopy**, is used in the Managing File Systems Exercise:

```
#!/usr/bin/ksh
# fragcopy
usage ()
{
clear
echo " "
echo " "
echo " "
echo " "
echo "Usage: fragcopy numfiles dir/sourcefilename dir/targetfilename"
echo " "
echo " "
echo " "
echo " "
exit
}
# Main...
integer i=0
integer cnt=$1
source=$2
target=$3
if [ $# != 3 ]
then
usage
fi
while true
do
cp $source $target.$i
if [ $? != 0 ]
then
echo " "
exit
fi
i=i+1
echo "  Files copied: \c"
echo "$i\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\c"
if [ $i = $cnt ]
then
echo " "
break
fi
done
```

Appendix E. AIX Control Book creation

AIX Control Book creation

List the licensed program products	`lslpp -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`
List system resources	`lsattr -EHl sys0`
List the VPD (Vital Product Data)	`lscfg -v`
Document the tty setup	`lscfg` or SMIT screen capture F8
Document the print queues	`qchk -A`
Document disk Physical Volumes (PVs)	`lspv`
Document Logical Volumes (LVs)	`lslv`
Document Volume Groups (long list)	`lsvg -l` *vgname*
Document Physical Volumes (long list)	`lspv -l` *pvname*
Document File Systems	`lsfs` *fsname* **/etc/filesystems**
Document disk allocation	`df`
Document mounted file systems	`mount`
Document paging space (70 - 30 rule)	`lsps -a`
Document paging space activation	`/etc/swapspaces`
Document users on the system	**/etc/passwd** `lsuser -a id home ALL`
Document users attributes	**/etc/security/user**
Document users limits	**/etc/security/limits**
Document users environments	**/etc/security/environ**
Document login settings (login herald)	**/etc/security/login.cfg**
Document valid group attributes	**/etc/group** `lsgroup ALL`
Document system wide profile	**/etc/profile**
Document system wide environment	**/etc/environment**
Document cron jobs	**/var/spool/cron/crontabs/***
Document skulker changes if used	`/usr/sbin/skulker`
Document system startup file	**/etc/inittab**
Document the hostnames	**/etc/hosts**
Document network printing	**/etc/hosts.lpd**
Document remote login host authority	**/etc/hosts.equiv**

Directories to monitor

/var/adm/sulog	Switch user log file (ASCII file). Use **`cat`**, **`pg`** or **`more`** to view it and **`rm`** to clean it out.
/etc/security/failedlogin	Failed logins from users. Use the **`who`** command to view the information. Use **`cat /dev/null > /etc/failedlogin`** to empty it.
/var/adm/wtmp	All login accounting activity. Use the **`who`** command to view it. Use **`cat /dev/null > /var/adm/wtmp`** to empty it.
/etc/utmp	Who has logged in to the system. Use the **`who`** command to view it. Use **`cat /dev/null > /etc/utmp`** to empty it.
/var/spool/lpd/qdir/*	Left over queue requests
/var/spool/qdaemon/*	Temporary copy of spooled files
/var/spool/*	Spooling directory
smit.log	SMIT log file of activity
smit.script	SMIT log of commands and scripts

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*

Unit objectives

After completing this unit, you 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

Figure F-1. Unit objectives AU1412.0

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 AU1412.0

Notes:

Self-configuring and non-self-configuring devices

During the bootup of an AIX system, 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, the SCSI adapter for CD-ROM disks, tape drives, and so forth, 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

Figure F-3. Adding a terminal AU1412.0

Notes:

How to add a TTY

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 AU1412.0

Notes:

Login attributes

Appropriate values for the **login** attribute are:

`disable`	The port is still defined, but it is only available as a dial-out port for an asynchronous connection to another machine
`enable`	The port is enabled for login, a **`getty`** process runs 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
  - 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

Figure F-5. Port attributes AU1412.0

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, (that is, 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. Others 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
```

```
                          TTY

  Move cursor to desired item and press Enter.

    List All Defined TTYs
    Add a TTY
    Move a TTY to Another Port
    Change / Show Characteristics of a TTY
    Remove a TTY
    Configure a Defined TTY
    Generate an Error Report
    Trace a TTY


  F1=Help      F2=Refresh      F3=Cancel     Esc+8=Image
  F9=Shell     F10=Exit        Enter=Do
```


Figure F-6. SMIT TTY menu AU1412.0

Notes:

Configuring TTY devices

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
  tty vcom  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   1P-03-11 16-Port RAN EIA-232 for 128-Port adapter
 sa3   Available   1P-03-12 16-Port RAN EIA-232 for 128-Port adapter
 sa4   Available   1P-03-13 16-Port RAN EIA-232 for 128 Port adapter
```


Figure F-7. Attachment AU1412.0

Notes:

Defining the TTY type and parent adapter

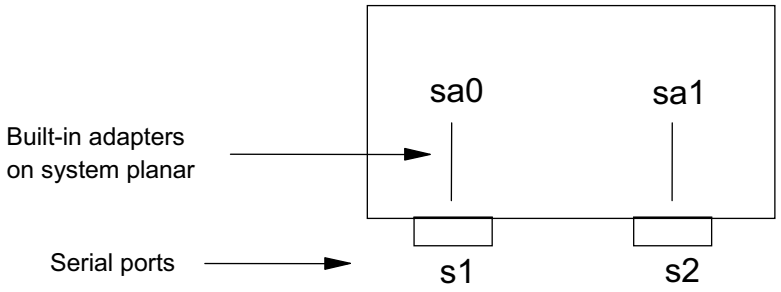
When you select **Add a TTY** from the TTY menu you are presented with two pop-ups to select the TTY type and adapter.

TTYs can either be connected to an RS232, RS422 or vcon adapter.

Once a type has been selected, you are presented with a list of installed adapters that support that method of attachment.

Device nomenclature

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



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

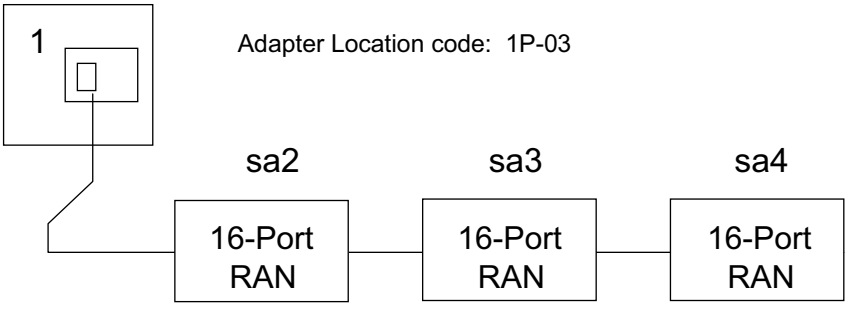


Figure F-8. Device nomenclature AU1412.0

Notes:

Pictorial view

This visual shows a picture and the associated nomenclature for the scenario we're discussing on adding a TTY.

Add a TTY

```
                                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                                sa2
* PORT number                                   []                  +
  Enable LOGIN                                   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...31]

F1=Help             F2=Refresh          F3=Cancel           F4=List
Esc+5=Reset         Esc+6=Command       Esc+7=Edit          Esc+8=Image
Esc+9=Shell         Esc+0=Exit          Enter=Do
```


Figure F-9. Add a TTY AU1412.0

Notes:

Port number

There is only one mandatory field on this screen and that is the **PORT number**. The **F4** key provides 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

Enable LOGIN

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

Asynchronous line characteristics

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.

TERMINAL type

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 is 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-10. **terminfo** AU1412.0

Notes:

Database of terminal characteristics

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 5L through the file /**etc**/**termcap**.)

TERM environment variable

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**.

Supported terminal types

There are a number of terminal types supported by default on AIX 5L, 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...35]

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


Figure F-11. Change the characteristics of a TTY AU1412.0

Notes:

Changing characteristics

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 (1 of 2)

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 Volume Level	7
Message Type	NON-DISPLAY	Cursor	Steady-block

General	Communication	Keyboard/Printer	Function
Operation Mode	ECHO	Interface	RS-232C
Line 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-12. IBM 3151 setup menus (1 of 2) AU1412.0

Notes:

Example

The example shows the settings for the **UK-English AIX Multiuser Enhancement Cartridge** to work with AIX 5L. The menus 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 (2 of 2)

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 dependent on the options cartridge

Figure F-13. IBM 3151 setup menus (2 of 2) AU1412.0

Notes:

Function menu

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 rmvtty
```

```
                        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-14. Deleting TTYs AU1412.0

Notes:

Preparing to remove a TTY

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 needs 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 needs to log out before you attempt to delete the TTY, otherwise problems may occur. When they log out a new **getty** process is 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-15. **penable/pdisable** . AU1412.0

Notes:

The penable command

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

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*n***
    - Then, **stty sane </dev/tty*n***
      or
    - **kill -9 *pid_of_login_shell***

Figure F-16. TTY problems AU1412.0

Notes:

Questions to ask

When approaching a terminal problem, there are several issues to investigate:

- Can the system communicate with the terminal? Try the command **echo hello > /dev/ttyn** and check if any output is sent to the TTY.
- Are cabling, power, brightness, contrast correct?
- Are there any processes running on the terminal? Verify this using the **fuser -u /dev/ttyn** command.

These questions normally produce a resolution to the problem.

Things to try

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 needs 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

Figure F-17. Documenting TTY setup AU1412.0

Notes:

Documentation

Since the setup of serial devices is relatively complex and involves an amount of manual labor, it is recommended that at least a map of where the devices exist, their port numbers, and the specific settings (**`stty`**, etc) be drawn up.

This will make life a lot easier for the system administrator who looks after the system.

Checkpoint

1. True or false? If a device, like a TTY, is left for **`cfgmgr`** to configure automatically, it picks 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 are not 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?

Figure F-18. Checkpoint AU1412.0

Notes:

Exercise Appendix B: Serial devices

- Verify a terminal
- Remove a terminal
- Add a terminal
- Check terminal communication settings
- Change terminal characteristics

Figure F-19. Exercise Appendix B: Serial devices AU1412.0

Notes:

Introduction

This exercise can be found in your Student Exercise Guide.

Unit summary

- Serial devices, such as TTYs and modems must be configured manually, either through SMIT or by a 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.

Figure F-20. Unit summary AU1412.0

Notes:

Appendix G. 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 AIX 5L.

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 (`lpsched`)
  - 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	*AIX 5L V5.3 Guide to Printers and Printing, Chapter 6. System V Printer Configuration*
SG24-6018	*Printing for Fun and Profit under AIX 5L* (*http://www.redbooks.ibm.com*)
Online	*AIX Commands Reference*

Unit objectives

After completing this unit, you 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 G-1. Unit objectives AU1412.0

Notes:

AIX 5L printing: What's new?

- System V print subsystem
- Changes to traditional AIX print subsystem
- Administration
- System management tools

Figure G-2. AIX 5L printing: What's new? AU1412.0

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 5L, 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 preprinted forms.

Traditional AIX print subsystem changes

Both the traditional AIX print subsystem and the new System V print subsystem are available. In order to support two print subsystems at the same time, some minor changes to AIX print subsystem file locations have been made.

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

The System V print subsystem includes system management using the Web-based System Manager, SMIT or the command line.

AIX 5L 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 G-3. AIX 5L printing environments AU1412.0

Notes:

Introduction

The visual gives an overview of the different approaches that can be taken to printing under AIX 5L. In the next two visuals, System V printing will be compared to the traditional AIX print subsystem. The remainder of this unit focuses 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, **`lpsched`**, 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 backend 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)

The 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/internet/wwsites.nsf/vwwebpublished/ipmaix_ww

System V print subsystem: Advantages

- Compatibility
- Availability of interface programs
- Security
- Support for forms
- Standard PostScript filters
- Long term strategy

Figure G-4. System V print subsystem: Advantages AU1412.0

Notes:

Compatibility

System administrators with experience in other UNIX variants that use System V printing 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 rewritten. This has led to a limited number of printers supported under AIX. With the support of System V printing in AIX 5L, it is 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 backend program must be customized to restrict user access.

Support for forms

If you are printing to preprinted 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 are added to the System V print subsystem in later releases, while the AIX print subsystem is 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
- Customizable spooling subsystem

Figure G-5. AIX print subsystem: Advantages AU1412.0

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 the Web-based System Manager or SMIT, as well as the command line. 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.

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

- 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 G-6. Software packaging AU1412.0

Notes:

Print subsystems installed

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 visual shows the filesets which comprise the System V print subsystem.

AIX print subsystem filesets

Default AIX 5L installation includes the base AIX print subsystem but no printers. When you add a printer, you are 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 the Web-based System Manager
- Switching:
  - **`switch.prt -s AIX`**
  - **`switch.prt -s SystemV`**
  - SMIT or the Web-based System Manager
- What happens during the switch

Figure G-7. Switching between print subsystems AU1412.0

Notes:

Introduction

Either the AIX print subsystem or the System V subsystem can be active, but not both at once.

Status

Use SMIT, the Web-based System Manager, or the **`switch.prt -d`** command to display the active print subsystem.

Switching

Use SMIT, the Web-based System Manager, or the **`switch.prt -s`** ***`subsystem-type`*** 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: `All print jobs must be terminated`.
2.	Stops the **`qdaemon`**, **`writesrv`**, and **`lpd`** daemons.
3.	Modifies **/etc/inittab** so that the AIX daemons will not be started on the next boot and 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 adds System V links for the common commands.
8.	Launches the System V print daemon (**`/usr/lib/lp/lpsched`**).

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: `All print jobs must be terminated`.
2.	Stops **`lpsched`** using the **`lpshut`** command.
3.	Modifies **/etc/inittab** so that **`lpsched`** will not be started on the next boot and 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.
8.	Launches the AIX print daemons.

Disabled queues or printers

If there are disabled queues or printers with waiting jobs, they remain disabled if the print subsystem is switched. If the original print subsystem is reactivated, they remain disabled. If the queue or printer is then enabled, the jobs are 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 are queued and 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**
- Main pages cover both versions

Figure G-8. Print commands overview AU1412.0

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: `AIX Print Subsystem`.

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: `System V Print Subsystem`.

System V printing overview

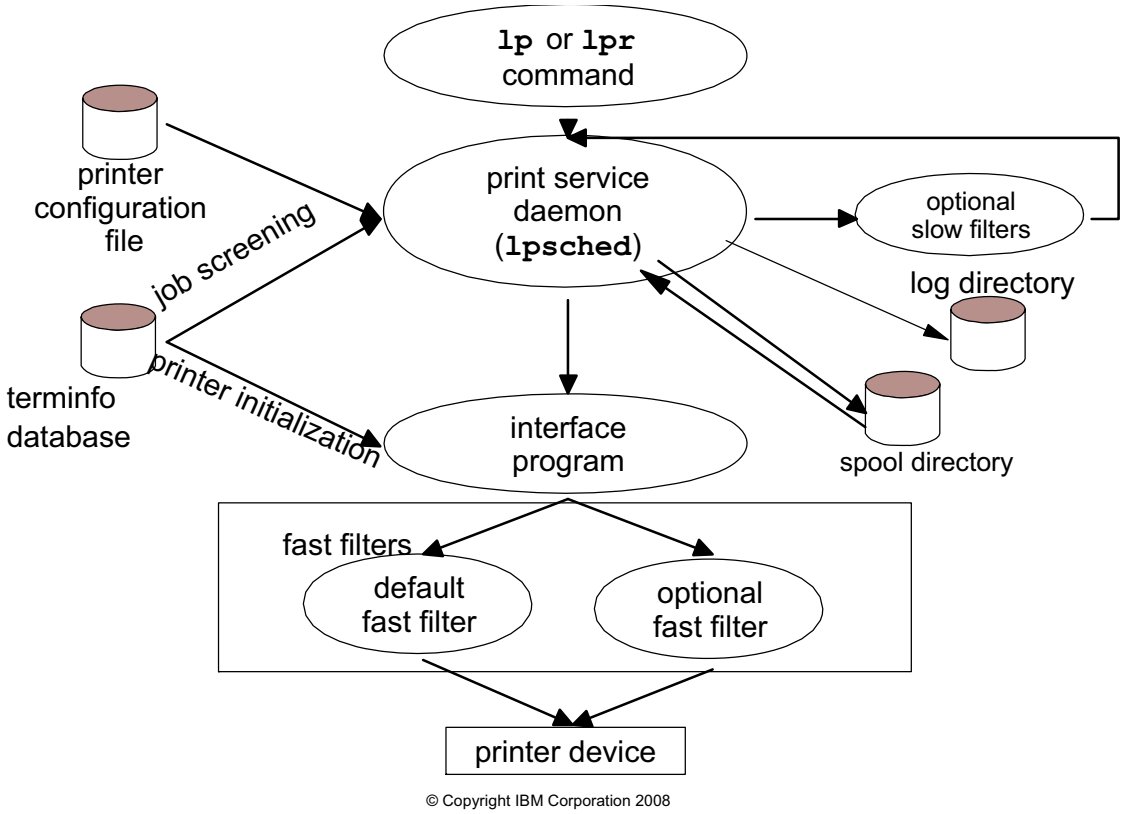


Figure G-9. System V printing overview AU1412.0

Notes:

Introduction

This visual provides an overview of the System V printing process. In the following pages, we 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 is 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 (`lpsched`)

When **`lpsched`** 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. **`lpsched`** 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, a printer configuration file is created (or modified). This file describes the printer, including:

- Content types this printer accepts
- 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. **`lpsched`** 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 (`lpsched`)

If the job can be printed, **`lpsched`** 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, **`lpsched`** sends 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. **`lpsched`** 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

`lpsched` 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)

`lpsched` may decide to use a combination of several filters. Slow filters are run directly by **`lpsched`**. Fast filters are run by the interface program, as directed by **`lpsched`**. Several filters may be piped together to achieve the desired file format.

Printing (`lpsched` and the interface program)

When a job moves to the top of a queue, **`lpsched`** 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

`lpsched` is responsible for monitoring job status and updating files in the log directory.

System V terminology

Term	Description	Examples
Printer device	The device driver this printer queue uses	**/dev/lp01**
System V printer	The printer queue	**myprinter**
Printer type	The **terminfo** entry used for this printer	**PS**
Content type	The types of files this printer can handle	**postscript**
Interface type	The interface script to use with this printer	**/usr/lib/lp/model/PS**
Class	A class is a group of printers	**bldg5**

Sample command to create a printer:

```
# lpadmin -p myprinter -v /dev/lp01 -c bldg5 -T PS \
-I postscript -m PS
```


Figure G-10. System V terminology AU1412.0

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 visual describes some System V terms. The **lpadmin** command at the bottom of the visual 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 Printer Command Language (PCL). 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 accepts 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 is rejected.

Interface or model type

The *interface* specifies which interface program is used by the printer. You specify the interface using one of the following **lpadmin** flags:

Flag	Description
-i ***interface_path***	Copy the script specified by ***interface_path*** (full path name) and use it as the interface script for this printer.
-e ***printer_name***	Copy the interface script already defined for ***printer_name*** and use it for this printer.
-m ***model***	Copy the file ***model*** in /**usr**/**lib**/**lp**/**model** and use it as the interface script for this printer.

In the example, **lpadmin** 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 prints 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 **bldg5**.

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 lpsched 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 G-11. Let's review 1 AU1412.0

Notes:

Adding a System V printer

- Create a printer device:
  - The **mkdev** command
  - SMIT
  - Web-based System Manager
- Create a System V printer:
  - The **lpadmin** command
  - SMIT
  - Web-based System Manager

Figure G-12. Adding a System V printer AU1412.0

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:

```
# lsplp 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 G-13. Printer device overview AU1412.0

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 using **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 chose 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.

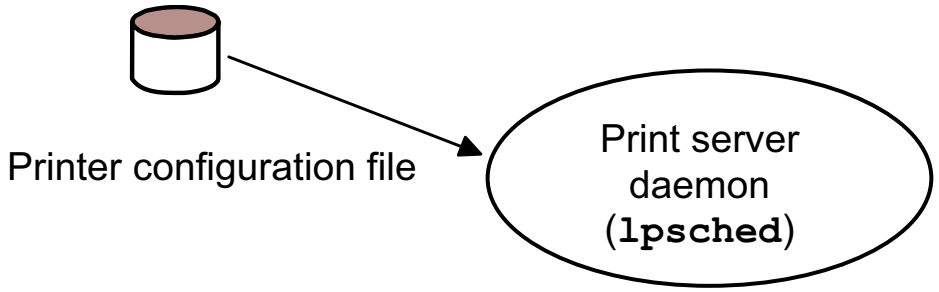
Printer device attributes

You can list printer device attributes using **`lsattr -El`** ***`printer_device_name`*** or using **`splp`** ***`printer_device_name`*** as shown in the visual. 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
```



- 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 G-14. Creating a local System V printer AU1412.0

Notes:

Introduction

System V printers are added, or modified, using **lpadmin**, SMIT or the Web-based System Manager. This visual shows a printer created using **lpadmin**.

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`

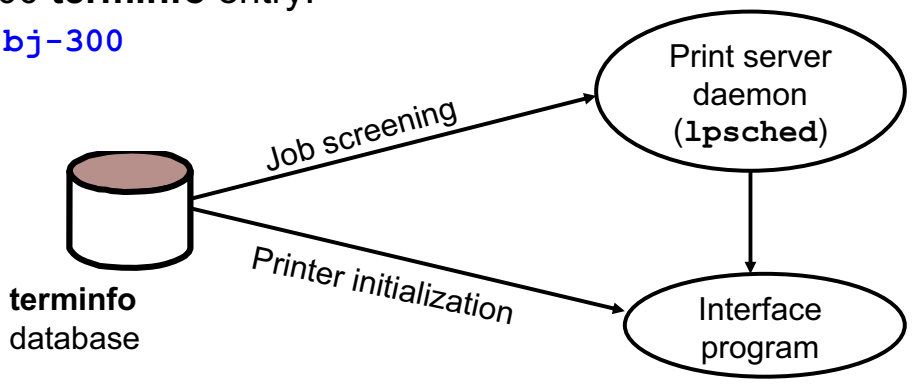


Figure G-15. Printer types AU1412.0

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 `lpsched` 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 5L 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 returns, form feeds, line feeds, set underline mode, as so forth

No printer type

If you do not specify a printer type, it defaults to **unknown**. Depending on how you are using the printer, this may not be a problem. It does mean that:

- Your printer is not 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 are 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:

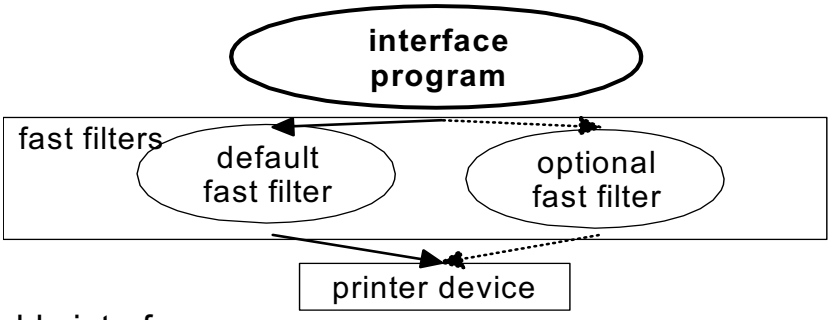
- If you are using a PCL printer, you can usually use the **hplaserjet** printer type. Set the content type to **pcl**.
- If you are using a PostScript printer, set the content type to **postscript** and choose one of the following printer types:
  - For serial connected printers: **PS**, **PSR**, or **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 chose 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 `lpsched`

Figure G-16. Interface programs AU1412.0

Notes:

Introduction

The interface is responsible for performing the functions as listed in the visual:

- 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 a filter
- Handle any printer errors from the filter and return exit status (success or failure) back to `lpsched`

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** selects 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 is 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, refer to: *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 G-17. Spool and log files AU1412.0

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 fails. 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 are 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
.SEQF	This file is used to keep track of the next job number.
***X*-0**	These files are the actual request files, where ***X*** is the job number. Notice that the printer name is not part of the request file name, but rather is stored within the file.
X-N	If files are being copied to the spool directory, there may be additional files (***X*-1**, ***X*-2** and so forth) that contain the actual data to be printed.

Example

In the example in the visual, job 10 was created without copying while copying was requested for job 11. Excerpts from the request files are shown in the visual. 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 size 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 grows until it fills the file system unless you manage it.

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

- Start printing:
  - **enable *printer***
- Stop printing:
  - **disable [-c] [-r *reason*] [-W] *printer***
- Start queuing:
  - **accept *printer***
- Stop queuing:
  - **reject [-r *reason*] *printer***
- Creating and enabling a new printer:
  - **mkdev**
  - **lpadmin**
  - **accept**
  - **enable**

Figure G-18. Managing printers AU1412.0

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 are 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` ***`reason`***	Assign a reason for disabling the printers. The reason is reported by **`lpstat -p`**. *`reason`* 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` ***`reason`***	Assign a reason for rejecting requests. The reason is reported by **`lpstat -a.`** The flag ***`reason`*** 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 G-19. Using the print service AU1412.0

Notes:

Introduction

This visual summarizes the commands (accessible by any user) to utilize the System V print service. Refer to 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 visual. ***dest*** can be either a printer or a class of printers. If a class is specified, the print service chooses 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 are accepted, if the printer can handle them. If the job has started printing, it is 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
cancel ***request-id-list***	Cancel the jobs specified. You can get the ***request-ids*** using **lpstat**.
cancel ***printer-list***	The user's currently printing job for the requested printer will be cancelled.
cancel -u ***user-list*** **[*printer-list*]**	Cancel all jobs for specified users. If ***printer-list*** 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
-o [*list*] [-l]	Reports the status of print requests. ***list*** can be printers, classes, or ***request-ids***. if **-l** is used, additional status for each job is reported.
-p [*list*] [-D] [-l]	Reports printing status: • Enabled/disabled • What is currently printing • Device status (available/defined) With **-D**, a brief description of each printer is included. With **-l**, a full description is included. ***list*** is a list of printers.
-a [*list*]	Reports queue status (accepting/rejecting). ***list*** can be printers or classes.
-u [*list*]	Reports status for users in ***list***.
-t [*list*]	Reports total status (similar to combined output of **-o**, **-p**, and **-a**).

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 5L includes two interface programs _____ and _____.
5. If you want the print service to copy a file to the spool directory, you 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 G-20. Let's review 2 AU1412.0

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 G-21. Using filters AU1412.0

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 accepts 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 5L 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
```

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 G-22. Using forms AU1412.0

Notes:

Introduction

A form is a preprinted 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 visual. We summarize the procedures for using forms here.

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 is queued, but is not 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
-A ***alert_type***	Send alerts to user **lp** when ***form_name*** is requested. ***alert_type*** can be: **mail**: Send mail to user **lp**. **write**: Send message to the terminal where **lp** is logged in. **none**: Do not alert. ***shell-command***: Execute named command. **quiet**: Do not send any more messages for current form request.
-Q ***number***	Send alerts after ***number*** of form requests have accumulated in the queue. Default is one.
-W ***interval***	Repeat alert every ***interval*** 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: # **disable** ***printer_name***
2.	Physically load the form in the printer.
3.	Inform the print service that the form is ready: # **lpadmin -p** ***printer*** **-M -f** ***form_name*** **[-a] [-o filebreak]** where: **-M -f** ***form_name*** Informs print service that ***form_name*** is mounted. **-a** 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. **-o filebreak** Inserts a form feed at the end of each alignment pattern. If not specified, no form feed is added.
4.	Align the form, if required. (See **-a** option above.)
5.	Physically load the form in the printer.
6.	Enable the printer. Queued jobs for this form will now be printed: # **enable** ***printer_name***

Unmounting a form

To unmount a form, follow these steps:

Step	Action
1.	Disable the printer. # **disable** ***printer_name***
2.	Physically remove the form from the printer.
3.	Inform the print service that the form is removed: # **lpadmin -p** ***printer*** **-M -f none**
4.	Enable the printer. # **enable** ***printer_name***

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 G-23. Planning a local System V printer AU1412.0

Notes:

Introduction

This visual 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.

Printer type

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 are you 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?

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`**. Refer to 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?

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 rejects 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
lpadmin -o banner	Banners are required (default)
lpadmin -o nobanner	Users are allowed to request that the banner not be printed
lp -o nobanner	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. Refer to 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
- The `lpNet` daemon
- **/etc/lp/Systems and lpsystem**
- Printing to JetDirect-attached printers

Figure G-24. System V network printing AU1412.0

Notes:

Introduction

This visual 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 visuals.

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 `lpsched`.

On client machines, `lpsched` 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 `lpsched`. If the request can be printed, `lpsched` 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 `lpsystem`

Remote systems with which you want the print service to communicate (client or server) must be registered in the **/etc/lp/Systems** file. The `lpsystem` 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. Refer to *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 G-25. Configuring a network print server AU1412.0

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
system_name	This parameter specifies the name or IP address of the remote system. *system_name* can be * to allow access from any system.
-T *timeout*	This option specifies the length of time the print service will allow a network connection to be idle. If idle time exceeds *timeout*, the connection is dropped. It will be re-established if there are more requests. *timeout* can be: • n: never timeout. This is the default. • 0: timeout immediately. • *N*: timeout after *N* minutes.
-R *retry*	This option specifies the length of time to wait to re-establish a connection if the connection was abnormally dropped. *retry* can be: • n: do not retry until there is more work. • 0: try to reconnect immediately. • *N*: try to reconnect after *N* minutes. The default is 10.
-y *comment*	This option allows you to add a free form comment. The comment must be quoted if it contains spaces.
-l *system_name*	This option lists the parameters defined for *system_name*, including any comment.
-r *system_name*	This option removes *system_name* 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
-p ***printer***	Specifies the name of the printer on the server.
-u allow	This parameter specifies the users who are allowed access.
-u deny	This parameter specifies the users who are denied access.
user_list	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: • ***userID***: a user on the local system • ***system_name*!*userID***: a user on system_name • ***system_name*!all**: all users on system_name • **all!*userID***: a user on all systems • **all!all**: 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 G-26. Configuring a remote print client AU1412.0

Notes:

Introduction

Configuring your system to print using a remote LPD printer is done in three steps:

1. Register remote system (the printer or print server) on the client
2. Define the printer queue on the client
3. **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**, 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
-p *local_name*	Specifies the name of the print queue on the client.
-s *server_name*	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.
server_printer_name	Specifies the name of the printer on the server. The *local_name* and the *server_printer_name* 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 gives 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 makes 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
accept	Permits jobs to be queued for specified destination (printer or class).
reject	Prevents jobs from being queued for specified destination.
enable	Activates the named printers so they will print from the queue.
disable	Deactivates named printers.
cancel	Cancels print jobs.
lpadmin	Create or modify printer configuration.
lpfilter	Manages filters.
lpforms	Manages forms. (Use **lpadmin** to mount a form.)
lpmove	Move print jobs to another destination.
lpsched	Start the print service.
lpshut	Stop the print service.
lpsystem	Register remote systems with the print service.
lpusers	Manages default priority and priority limits for printer service users.
lpstat	Report print service status.

Figure G-27. System V administrative command summary AU1412.0

Notes:

Introduction

This visual provides a brief summary of all the System V administrative commands. A summary of the options to **lpadmin** is included on the next page. Refer to the relevant **man** page for complete information.

For a comparison of the commands for the System V print subsystem and the AIX print subsystem, refer to:

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
`-p` ***`printer`***	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)
`-v` ***`device`***	Used to configure a local printer. Associates a device with a printer.
`-s` ***`server`*** **`[!`***`server_printer_name`***`]`**	Used to configure a remote printer. ***`server`*** 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. ***`server_printer_name`*** specifies the name of the printer queue on the server.
`-x` ***`destination`***	Remove ***`destination`*** (which can be a printer or a class) from the print service.
`-i` ***`interface`*** **`-m`** ***`model`*** **`-e`** ***`printer_name`***	Used to specify the printer interface. Only one of these options can be specified. If none of these are specified, the `standard` interface is used. **`-i`** ***`interface`*** specifies a full file path. **`-m`** ***`model`*** specifies one of the supplied interface programs (a file in **/usr/lib/lp/model**). **`-e`** ***`printer_name`*** directs **`lpadmin`** to copy the interface used for ***`printer_name`*** to the printer being added or changed (specified with **`-p`** ***`printer`***).
`-T` ***`printer_type`***	Identifies an entry in the **terminfo** database, which is used by the interface program and some filters.
`-I` ***`content_type_list`***	Identifies one or more types of content that this printer can handle without filtering. If not specified, default is **simple**.

Option	Description
-c ***class***	Add ***printer*** to ***class***. Create ***class*** if it does not already exist.
-r ***class***	Remove ***printer*** from ***class***. If ***printer*** is the last member of ***class***, then remove ***class***.
-O {copy|nocopy}	**-O copy** specifies that files should always be copied to the spool directory. **-O nocopy** 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). This flag sets the value of the **copy-files** parameter in the **/etc/default/lp** file to **on** (**copy**) or **off** (**nocopy**).
-A ***alert_type*** **[-W** ***minutes*****]**	Specifies the type of alert used to notify the administrator of printer faults. The default is to send the alert message via mail. **-W** ***minutes*** specifies the interval between alerts. **0** or **once** is the default, which indicates sending only one alert for a fault.
-M -f ***form_name*** **[-o filebreak]**	Mount ***form_name*** on ***printer***. **-o filebreak** specifies that a form feed be inserted between each copy of the alignment pattern.
-f allow:*form_list* **-f deny:***form_list*	Allow or deny the forms in ***form_list*** to be printed on ***printer***. By default, all forms are denied.
-u allow:*user_list* **-u deny:***user_list*	Allow or deny the users in ***user_list*** to access ***printer***.

System V user command summary

Command	Description
`cancel`	Cancels print jobs.
`lp`	Submit a print job to a printer
`lpstat`	Report print service status.

Figure G-28. System V user command summary AU1412.0

Notes:

Introduction

This visual provides a brief summary of the System V user commands. Refer to the relevant man page for complete information.

`cancel`

The `cancel` command is used to cancel print jobs, as shown in the table. Regular users can only cancel their own jobs.

`lpstat`

The `lpstat` command displays information about the current status of the line printer.

`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
`-d` *`destination`*	Specifies the printer destination (printer or class) where the job is to be printed.
`files`	Specifies one or more files to be printed. Files are printed in the order specified. Use `-` to specify standard input.
`-c`	Copy files to spool directory before printing. Default is not to copy unless `lpadmin -O copy` has been used.
`-n` *`number`*	Print *`number`* copies. Default is one.
`-T` *`content_type`*	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.
`-f` *`form_name`*	Print the job on form *`form_name`*. 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 `lpform` command.)
`-o` *`options`*	`-o` 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.
`-m`	Send notification via mail when job has been printed. Default is no mail.

Option	Description
-y ***mode_list***	Print the job according to the modes in ***mode_list***. This option may only be used if there is a filter available to handle the requested modes; otherwise the print job is 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. **`lpsched`** 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 G-29. Checkpoint AU1412.0

Notes:

Exercise Appendix C: The System V print subsystem

- Switch print subsystems
- Add a System V printer
- Submit print jobs
- (Optional) Configure a remote print server and client

Figure G-30. Exercise Appendix C: The System V print subsystem AU1412.0

Notes:

Introduction

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

Unit summary

- AIX 5L supports both the AIX print subsystem and the System V subsystem.
- Either System V or AIX (not both) can be active at one time.
- 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.

Figure G-31. Unit summary AU1412.0

Notes:

Glossary

Note:

The entries in this glossary were developed a number of years ago and indicate the use of various terms at a particular point in UNIX history. Hence, some of the definitions may not be applicable to current UNIX implementations such as AIX 5L, and some other statements in the entries may not be current. However, this glossary still provides valuable information regarding the historical use of the terms listed here.

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, and so forth.

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.

MotifT 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 (that is, 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, and so forth, 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 FoundationT (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 a 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, and so forth).

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 de facto 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

zeroeth argument The command name; the argument before the first.

IBM®