AIX 5L System Administration II: Problem Determination

(Course Code AU16)

Student Exercises

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	Micro-Partitioning
MVS	OS/2	POWER
POWER4	POWER5	POWER Gt1
POWER Gt3	PS/2	pSeries
Redbooks	RS/6000	SP

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

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

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

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

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

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

December 2004 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™	Micro-Partitioning™
MVS™	OS/2®	POWER™
POWER4™	POWER5™	POWER Gt1™
POWER Gt3™	PS/2®	pSeries®
Redbooks™	RS/6000®	SP™

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

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

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

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

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

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

Exercise Description

Each exercise in this course is divided into sections as described below. Select the section that best fits your method of performing exercises. You may use a combination of these sections as appropriate.

Exercise Instructions

This section tells you what to accomplish. There are no definitive details on how to perform the tasks. You are given the opportunity to work through the exercise given what you learned in the unit presentation, utilizing the Student Notebook, your past experience, and maybe a little intuition.

Exercise Instructions With Hints

This section is an exact duplicate of the **Exercise Instructions** section except that in addition, hints (which are not solutions!) are provided to help step you through the exercise. A combination of using the **Exercise Instructions** section along with the **Exercise Instructions With Hints** section can make for a rewarding combination.

Exercise Instructions With Solutions

This section is also an exact duplicate of the **Exercise Instructions** and contains solutions and additional tips for the students. If very inexperienced students take part in this course, they should work with this section.

Students can use this part to compare their work with the solutions.

When showing the SMIT method to accomplish a task, each line in bold represents a submenu or selector screen. You will need to press the Enter key after selecting each item as listed. When you reach the dialog screen, the field descriptions will be in regular text and the items you need to fill in will be in bold. Only the items that need to be changed will be shown, not the entire screen. Once you have reached the dialog screen portion of SMIT, press Enter ONLY after all indicated entries have been made.

The SMIT steps will be shown for the ASCII version of SMIT. Under most circumstances these steps match the steps taken if using the graphics version of SMIT. The exceptions relate to the use of the function keys. When instructed to press the **F3** key back to a particular menu, when in graphics SMIT, you will instead click the Cancel box at the bottom of the screen. When instructed to press the **F9** key to shell out, in graphics mode, simply open another window.

Note: The new **wsm** interfaces are currently not shown.

Optional Exercise Parts

Some labs provide additional practice on a particular topic. Specific details and/or hints are provided to help step you through the **Optional Exercises**, if needed. Not all exercises include **Optional Exercises**. According to the group, the instructor can decide to do them or not. If there is time, the optional part should be executed by the students.

Exercise 1. Problem Determination Introduction

What This Exercise Is About

This exercise will acquaint you with the system that you will be using throughout the rest of this course. You will recall some basic administration commands.

What You Should Be Able to Do

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

- List volume groups, physical and logical volumes on your system
- Identify real memory and paging space on your system
- Identify the hardware platform and processor type of your system

Introduction

In this unit you will review some basic administration commands you should know.

The instructions in this exercise require **root** authority.

Exercise Instructions

Review Basic Administration Commands

__ 1. Collect the following details of your system and make a note of them:

Volume Groups:

Physical Volumes:

Logical Volumes in rootvg:

Paging Space:

Real Memory:

Hardware Platform:

Processor Type:

__ 2. Identify the logical volumes that reside on your **hdisk0**.

Write down the command you used:

_____

From the fact that the number of LPs is equal to the number of PPs, what can you conclude?

_____

END OF LAB

Exercise Instructions With Hints

Review Basic Administration Commands

__ 1. Collect the following details of your system and make a note of them:

Volume Groups:

Physical Volumes:

Logical Volumes in rootvg:

Paging Space:

Real Memory:

Hardware Platform:

Processor Type:

Hint: Use **lsps, bootinfo, lspv, lsvg, prtconf, getconf** to collect the information.

__ 2. Identify the logical volumes that reside on your **hdisk0**.

Write down the command you used:

_____

Hint: Use **lspv** or **lsvg** to determine the logical volumes on hdisk0.

From the fact that the number of LPs is equal to the number of PPs, what can you conclude?

_____

END OF LAB

Exercise Instructions With Solutions

Review Basic Administration Commands

__ 1. Collect the following details of your system and make a note of them:

Volume Groups:	Use **lsvg**
Physical Volumes:	Use **lspv**
Logical Volumes in rootvg:	Use **lsvg -l rootvg**
Paging Space:	Use **lsps -a**
Real Memory:	Use **bootinfo -r** or **prtconf -m**
Hardware Platform:	Use **bootinfo -p** or **getconf MACHINE_ARCHITECTURE**
Processor Type:	Use **prtconf | grep Processor**

__ 2. Identify the logical volumes that reside on your **hdisk0**.

Write down the command you used:

lspv -l hdisk0 or **lsvg -l rootvg**

From the fact that the number of LPs is equal to the number of PPs, what can you conclude?

No mirroring

END OF LAB

Exercise 2. The Object Data Manager (ODM)

What This Exercise Is About

This exercise will review some of the most important ODM files and how they are used in device configuration. Students will use the ODM command line interface.

What You Should Be Able to Do

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

- Define the meaning of some of the most important ODM files
- Work with the ODM command line interface
- Describe how ODM classes are used from device configuration commands

Introduction

This exercise has three parts:

1. Review of device configuration ODM classes (PdDv, PdAt, CuDv, CuAt, CuDep, CuDvDr).
2. Role of ODM during device configuration.
3. Optional Part: Creating self-defined ODM classes.
4. All instructions in this exercise require **root** authority.

Exercise Instructions

Review of device configuration ODM classes

__ 1. Execute the **lsdev** command and identify all devices that are supported on your system. Tell the **lsdev** command to provide column headers in the output.

What is the command you used?

_____

Which ODM class is used by the **lsdev** command to generate this output?

_____

__ 2. Execute the **lsdev** command and identify all SCSI-devices that are currently attached to your system. Tell the **lsdev** command to provide column headers in the output.

What is the command you used?

_____

Which ODM class is used by the **lsdev** command to generate this output?

_____

__ 3. From the output, complete the following list for disk **hdisk0**:

Name: _____

Status: _____

Location: _____

Description: _____

__ 4. Use the ODM command line interface and list the ODM object that describes this disk device.

What is the command you used?

_____

From the output, complete the following list for disk **hdisk0**:

Status: _____

Chgstatus: _____

Parent: _____

Location: _____

__ 5. From this output please answer the following questions.

What is the meaning of the descriptor **chgstatus**?

_____

_____

The **lsdev** command provides a description field, which is not part of ODM class CuDv. Where does the description come from?

_____

_____

_____

_____

__ 6. Execute the **lsattr** command, and identify the **physical volume identifier** for your **hdisk0**.

What is the command you used?

_____

Write down the physical volume ID for your **hdisk0**:

pvid: _____

__ 7. Use the ODM command line interface and list the ODM object that stores the **physical volume identifier**.

What is the command you used?

_____

__ 8. The /**dev** directory contains the special files to access the devices. Write down the major and minor number of the special file for **hdisk0**.

Major number: _____

Minor number: _____

Which ODM class is used to create this special file entry?

_____

__ 9. List all your logical volumes that are part of the **rootvg**.

What is the command you used?

_____

Query the ODM class **CuDep** and identify all logical volumes that belong to **rootvg**.

What is the command you used?

_____

Role of ODM during device configuration

__ 10. During the following steps we will simulate the configuration of an SCSI disk without using **cfgmgr**.

Important: This is just a simulation. You do not attach a real disk in this exercise.

The ODM contains predefined objects to support many types of different disks.

Use the **lsdev** command to list all predefined devices of class **disk**.

Write down the command you used.

_____

Identify the disk type **osdisk**, which means **other SCSI disk**. We will use this type of disk in the following steps.

Use **odmget** to identify the object in PdDv, that describes the disk type **osdisk**.

Write down the command you used.

_____

From the output complete the following:

type: _____

class: _____

subclass: _____

prefix: _____

Device Driver: _____

Configuration Method: _____

__ 11. A disk needs to be attached to a SCSI adapter. This adapter will get the **parent device** for the disk we are configuring.

Using the **lsdev** command, list all customized devices of class **adapter** and identify the logical device name for the SCSI adapter where the disk will be attached to.

Write down the command you used.

_____

Add the name of the adapter:

SCSI parent adapter: _____

__ 12. Before configuring the device, a free SCSI address must be identified. List all ODM objects in CuDv where the identified SCSI adapter is stored as the parent device.

Write down the command you used.

_____

From the output, write down the SCSI addresses which are in use:

SCSI addressed in use: _____

Choose a free SCSI address and write it down in the table. You need to specify this address later.

Free SCSI address: _____

__ 13. Get the disk into the **defined** state using the **mkdev -d** command. You need to pass the following information to **mkdev**:

- Device class
- Device subclass
- Device type
- Parent device
- SCSI address

Write down the command you used to define the disk.

_____

What device name has been assigned to the disk?

Device name: _____

__ 14. Using this assigned name, list the object that stores your disk in the customized database.

Write down the command you used:

_____

__ 15. Try to configure the disk using **mkdev -l**. What happens?

_____

_____

_____

__ 16. Finally, remove the disk from the system using **rmdev**.

Write down the command you used:

_____

Examine your CuDv object class. Did you find the removed disk in this object class?

_____

Optional Part: Creating self-defined ODM classes

__ 17. Before creating an ODM class you need to specify the descriptors that are contained in the class.

Using an editor, create a file **parts.cre** with the following class structure:

```
class parts {
long        part_number;
char        part_description[128];
char        warehouse[4];
long        contained_in;
}
```

__ 18. Create the ODM class using this class structure and check the structure of this class. Write down the command you used:

_____

Identify in your working directory, which files have been created during this step.

What do you think is the purpose of these files?

_____

_____

_____

Where does the ODM class **parts** reside?

_____

__ 19. Create some objects in ODM class **parts**, using the following data:

Part Number	Description	Warehouse	Contained In
10001	Wheel	a12	50001
10003	Frame	a19	50001
10005	Saddle	a01	50001
10006	Front wheel brake	a03	50001
10007	Rear wheel brake	a03	50001
50001	City Bike Easy Rider	x99	

Take the stanza file **parts.add** out of /**home**/**workshop**.

__ 20. List all objects that are contained in part 50001 (the City Bike Easy Rider). Write down the command you used:

_____

__ 21. Change the **warehouse** location for part **Wheel** to **b10**.

__ 22. Finally, remove ODM class **parts** from the system. Write down the command you used.

_____

END OF LAB

Exercise Instructions With Hints

Review of device configuration ODM classes

__ 1. Execute the **lsdev** command and identify all devices that are supported on your system. Tell the **lsdev** command to provide column headers in the output.

What is the command you used?

_____

Which ODM class is used by the **lsdev** command to generate this output?

_____

Hint: Use **man lsdev** and identify the example section.

__ 2. Execute the **lsdev** command and identify all SCSI-devices that are currently attached to your system. Tell the **lsdev** command to provide column headers in the output.

What is the command you used?

_____

Which ODM class is used by the **lsdev** command to generate this output?

_____

Hint: Use **man lsdev**. There is an option **-s** to specify a **subclass**.

__ 3. From the output, complete the following list for disk **hdisk0**:

Name: _____

Status: _____

Location: _____

Description: _____

__ 4. Use the ODM command line interface and list the ODM object that describes this disk device.

What is the command you used?

_____

Hint: Use **odmget -q** and specify **name=hdisk0** as search criteria.

From the output, complete the following list for disk **hdisk0**:

Status: _____

Chgstatus: _____

Parent: _____

Location: _____

__ 5. From this output please answer the following questions.

What is the meaning of the descriptor **chgstatus**?

_____

_____

Hint: Use the index in your student manual. Search for word **chgstatus**.

The **lsdev** command provides a description field, which is not part of ODM class CuDv. Where does the description come from?

_____

_____

_____

_____

Hint: Use the **odmshow** command and try to describe the last descriptor.

__ 6. Execute the **lsattr** command and identify the **physical volume identifier** for your **hdisk0**.

What is the command you used?

Write down the physical volume ID of the disk:

status: _____

Hint: Use **man lsattr** and check the example section.

__ 7. Use the ODM command line interface and list the ODM object that stores the **physical volume identifier**.

What is the command you used?

_____

Hint: Use **odmget** to query the object. The attribute **pvid** is a nondefault value.

__ 8. The /**dev** directory contains the special files to access the devices. Write down the major and minor number of the special file for **hdisk0**.

Major/Minor Number: _____

Which ODM class is used to create this special file entry?

_____

Hint: Check your /**dev** directory for a file **hdisk0**. The ODM file used to create this special file belongs to the customized database files of ODM.

__ 9. List all your logical volumes that are part of the **rootvg**.

What is the command you used?

_____

Query the ODM class **CuDep** and identify all logical volumes that belong to **rootvg**.

What is the command you used?

_____

Hint: Use **odmget** and list all logical volumes that have a *dependency* to **rootvg**.

Role of ODM during device configuration

__ 10. During the following steps we will simulate the configuration of an SCSI disk without using **cfgmgr**.

This is just a simulation. You do not attach a real disk in this exercise.

The ODM contains predefined objects to support many types of different disks.

Use the **lsdev** command to list all predefined devices of class **disk**.

Write down the command you used.

_____

Hint: Use **lsdev -P** and specify the classname **disk**.

Identify the disk type **osdisk**, which means **other SCSI disk**. We will use this type of disk in the following steps.

Use **odmget** to identify the object in PdDv, that describes the disk type **osdisk**.

Write down the command you used.

_____

Hint: Use **odmget** and specify as type **osdisk**.

From the output complete the following:

type: _____

class: _____

subclass: _____

prefix: _____

Device Driver: _____

Configuration Method: _____

__ 11. A disk needs to be attached to a SCSI adapter. This adapter will get the **parent device** for the disk we are configuring.

Using the **lsdev** command, list all customized devices of class **adapter** and identify the logical device name for the SCSI adapter where the disk will be attached to.

Write down the command you used.

_____

Hint: Use **lsdev -C** and specify class **adapter**.

Add the name of the adapter:

SCSI parent adapter: _____

__ 12. Before configuring the device, a free SCSI address must be identified. List all ODM objects in CuDv where the identified SCSI adapter is stored as the parent device.

Write down the command you used.

_____

Hint: Use **odmget** and specify the **parent** attribute as search criteria.

From the output, write down the SCSI addresses which are in use:

SCSI addressed in use: _____

Choose a free SCSI address and write it down in the table. You need to specify this address later.

Free SCSI address: _____

__ 13. Get the disk into the **defined** state using the **mkdev -d** command. You need to pass the following information to **mkdev**:

- Device class
- Device subclass
- Device type
- Parent device

- SCSI address

Write down the command you used to define the disk.

_____

Hint: mkdev has the following options: -d means define, -c for class, -s for subclass, -t for type, -p for parent device, -w for SCSI address.

What device name has been assigned to the disk?

Device name: _____

__ 14. Using this assigned name, list the object that stores your disk in the customized database.

Write down the command you used:

_____

Hint: Use **odmget** and identify the object in the customized devices class.

__ 15. Try to configure the disk using **mkdev -l**. What happens?

_____

_____

_____

Hint: There is a special program that will be called to configure the disk. Where do you find this program? Will this program be successful?

__ 16. Finally, remove the disk from the system using **rmdev**.

Write down the command you used:

_____

Hint: Use **man rmdev** and check the example section.

Examine your CuDv object class. Did you find the removed disk in this object class?

_____

Hint: Use **odmget** and use the **name** descriptor to search the corresponding device.

Optional Part: Creating self-defined ODM classes

__ 17. Before creating an ODM class you need to specify the descriptors that are contained in the class.

Using an editor, create a file **parts.cre** with the following class structure:

```
class parts {
long        part_number;
char        part_description[128];
char        warehouse[4];
long        contained_in;
}
```

__ 18. Create the ODM class using this class structure and check the structure of this class. Write down the command you used:

_____

Hint: Use the **odmcreate** and the **odmshow** command.

Identify in your working directory, which files have been created during this step.

What do you think is the purpose of these files?

_____

_____

_____

Hint: Can you access the ODM database files only by the command line interface?

Where does the ODM class **parts** reside?

_____

Hint: Which environment variable is used by the ODM command line interface?

__ 19. Create some objects in ODM class **parts**, using the following data:

Part Number	Description	Warehouse	Contained In
10001	Wheel	a12	50001
10003	Frame	a19	50001
10005	Saddle	a01	50001
10006	Front wheel brake	a03	50001
10007	Rear wheel brake	a03	50001
50001	City Bike Easy Rider	x99	

Take the stanza file **parts.add** out of /**home**/**workshop**.

__ 20. List all objects that are contained in part 50001 (the City Bike Easy Rider). Write down the command you used:

_____

Hint: Use **odmget**.

__ 21. Change the **warehouse** location for part **Wheel** to **b10**.

Hint: Use **odmget, vi, odmdelete, and odmadd**
-OR-
Use **odmget, vi and odmchange**

__ 22. Finally, remove ODM class **parts** from the system. Write down the command you used.

_____

Hint: Use the **odmdrop** command to remove the class from the system.

END OF LAB

Exercise Instructions With Solutions

Review of device configuration ODM classes

__ 1. Execute the **lsdev** command and identify all devices that are supported on your system. Tell the **lsdev** command to provide column headers in the output.

What is the command you used?

lsdev -P -H | pg

Which ODM class is used by the **lsdev** command to generate this output?

PdDv

__ 2. Execute the **lsdev** command and identify all SCSI-devices that are currently attached to your system. Tell the **lsdev** command to provide column headers in the output.

What is the command you used?

lsdev -C -s scsi -H

Which ODM class is used by the **lsdev** command to generate this output?

CuDv

__ 3. From the output, complete the following list for disk **hdisk0**:

Name: _____

Status: _____

Location: _____

Description: _____

__ 4. Use the ODM command line interface and list the ODM object that describes this disk device.

What is the command you used?

odmget -qname=hdisk0 CuDv

From the output complete the following list for disk **hdisk0**:

Status: _____

Chgstatus: _____

Parent: _____

Location: _____

__ 5. From this output please answer the following questions.

What is the meaning of the descriptor **chgstatus**?

The status of the disk device has not changed since the last reboot.

The **lsdev** command provides a description field, which is not part of ODM class CuDv. Where does the description come from?

The description of a device is shown via the descriptors setno, msgno and catalog from the object class PdDv. The attribute PdDvLn=disk/scsi/scsd in CuDv is a reference to the object in class PdDv.

__ 6. Execute the **lsattr** command and identify the **physical volume identifier** for your **hdisk0**.

What is the command you used?

lsattr -El hdisk0

Write down the physical volume ID of the disk:

pvid: _____

__ 7. Use the ODM command line interface, and list the ODM object that stores the **physical volume identifier**:

What is the command you used?

odmget -q"name=hdisk0 and attribute=pvid" CuAt

or

odmget CuAt | grep -p hdisk0 | grep -p pvid

__ 8. The /**dev** directory contains the special files to access the devices. Write down the major and minor number of the special file for **hdisk0**.

Major number: _____

Minor Number: _____

 # ls -l /dev/hdisk0

Which ODM class is used to create this special file entry?

CuDvDr

odmget -q value3=hdisk0 CuDvDr

__ 9. List all your logical volumes that are part of the **rootvg**.

What is the command you used?

lsvg -l rootvg

Query the ODM class **CuDep** and identify all logical volumes that belong to **rootvg**.

What is the command you used?

odmget -qname=rootvg CuDep

or

odmget -q parent=rootvg CuDv

Role of ODM during device configuration

__ 10. During the following steps we will simulate the configuration of an SCSI disk without using **cfgmgr**.

Important: This is just a simulation. You do not attach a real disk in this exercise.

The ODM contains predefined objects to support many types of different disks.

Use the **lsdev** command to list all predefined devices of class **disk**.

Write down the command you used.

lsdev -P -c disk

Identify the disk type **osdisk**, which means **other SCSI disk**. We will use this type of disk in the following steps.

lsdev -P -c disk -t osdisk

Use **odmget** to identify the object in PdDv, that describes the disk type **osdisk**.

Write down the command you used.

odmget -qtype=osdisk PdDv

From the output complete the following:

type: _____

class: _____

subclass: _____

prefix: _____

Device Driver: _____

Configuration Method: _____

__ 11. A disk needs to be attached to a SCSI adapter. This adapter will get the **parent device** for the disk we are configuring.

Using the **lsdev** command, list all customized devices of class **adapter** and identify the logical device name for the SCSI adapter where the disk will be attached to.

Write down the command you used.

lsdev -C -c adapter

Add the name of the adapter:

SCSI parent adapter: _____

__ 12. Before configuring the device, a free SCSI address must be identified. List all ODM objects in CuDv where the identified SCSI adapter is stored as the parent device.

Write down the command you used.

odmget -qparent=scsi0 CuDv

(or)

lsdev -Cs scsi

From the output, write down the SCSI addresses which are in use:

SCSI addressed in use: _____

Choose a free SCSI address and write it down in the table. You need to specify this address later.

Free SCSI address: _____

__ 13. Get the disk into the **defined** state using the **mkdev -d** command. You need to pass the following information to **mkdev**:

- Device class
- Device subclass
- Device type
- Parent device
- SCSI address

Write down the command you used to define the disk.

mkdev -d -c disk -s scsi -t osdisk -p scsi0 -w x,0

-w x,0; x is the free SCSI address identified in step 12

What device name has been assigned to the disk?

Device name: _____

lsdev -Cs scsi

__ 14. Using this assigned name, list the object that stores your disk in the customized database.

Write down the command you used:

odmget -qname=hdisk2 CuDv

__ 15. Try to configure the disk using **mkdev -l**. What happens?

 # mkdev -l hdisk2

The mkdev command will call the configuration method of the disk. The name of the configuration method is stored in PdDv. The configuration method detects that no disk has been attached to the SCSI adapter and will fail. The disk will still be in the defined state.

__ 16. Finally, remove the disk from the system using **rmdev**.

Write down the command you used:

rmdev -l hdisk2 -d

Examine your CuDv object class. Did you find the removed disk in this object class?

odmget -qname=hdisk2 CuDv
No. The object has been removed from the ODM.

Optional Part: Creating self-defined ODM classes

__ 17. Before creating an ODM class you need to specify the descriptors that are contained in the class.

 # mkdir /tmp/odm

 # cd /tmp/odm

Using an editor, create a file **parts.cre** with the following class structure:

```
class parts {
long        part_number;
char        part_description[128];
char        warehouse[4];
long        contained_in;
}
```

__ 18. Create the ODM class using this class structure and check the structure of this class. Write down the command you used:

odmcreate parts.cre

odmshow parts

Identify in your working directory, which files have been created during this step.

What do you think is the purpose of these files?

As administrator you use the ODM command line interface to access the ODM database files. For accessing the ODM from applications or system programs an ODM application programming interface (API) exists. These programs need to include the files that have been created by odmcreate.

Where does the ODM class **parts** reside?

/**etc**/**objrepos** (if you didn't change the ODMDIR variable)

__ 19. Create some objects in ODM class **parts**, using the following data:

Part Number	Description	Warehouse	Contained In
10001	Wheel	a12	50001
10003	Frame	a19	50001
10005	Saddle	a01	50001
10006	Front wheel brake	a03	50001
10007	Rear wheel brake	a03	50001
50001	City Bike Easy Rider	x99	

Take the stanza file **parts.add** out of /**home**/**workshop**.

 # cat parts.add
parts:
part_number = “10001”
part_description = “Wheel”
warehouse = “a12”
contained_in = “50001”

parts:
part_number = “10003”
part_description = “Frame”
warehouse = “a19”
contained_in = “50001”

... Add all the information from the chart.

 # odmadd parts.add

__ 20. List all objects that are contained in part 50001 (the City Bike Easy Rider). Write down the command you used:

 # odmget -qcontained_in=50001 parts

__ 21. Change the **warehouse** location for part **Wheel** to **b10**.

Extract the object and place it in a file.
 # odmget -qpart_description=Wheel parts>part_change

Edit the file to change the **warehouse** location.

 # vi part_change
....
warehouse="b10"
....

Option 1:

Remove the old record.
 # odmdelete -qpart_description=Wheel -oparts

Add modified record.
 # odmadd part_change

-OR-

Option 2:

 # odmchange -qpart_description=Wheel -oparts part_change

Verify the change.

odmget -qpart_description=Wheel parts

__ 22. Finally, remove ODM class parts from the system. Write down the command you used.

 # odmdrop -o parts

END OF LAB

Exercise 3. System Initialization Part 1

What This Exercise Is About

This exercise will review the hardware boot process of an RS/6000.

What You Should Be Able to Do

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

- Boot a machine in maintenance mode
- Repair a corrupted boot logical volume
- Alter boot lists on different RS/6000 hardware models

Introduction

This exercise has three parts:

1. Identify the boot lists of your system
2. Identify LVM information of your system
3. Repair a corrupted boot logical volume

All instructions in this exercise require **root** authority.

Required Materials

- The program /**home**/**workshop**/**ex3pro1**
- AIX 5.2 CD or **mksysb** tape

Exercise Instructions

Working with Boot Lists and Identifying information on your system

__ 1. Write down the boot sequence of your RS/6000 model for a **normal** boot:

1) boot device:

What is the command you used to determine the boot list:

_____

__ 2. Does your model support a boot list for **maintenance** mode?

_____

If your model supports a service boot list for maintenance mode, write down the boot sequence for this boot mode:

1) boot device:
2) boot device:
3) boot device:
4) boot device:

__ 3. Identify which disks are contained within the **rootvg**:

_____

Write down the command you used.

_____

Identify which disk is the bootable disk, that means the disk that contains the boot logical volume **hd5**:

_____

Write down the command you used.

_____

Identify the **logical volume type** of **hd5**:

_____

Write down the command you used.

_____

Identify LVM information from your system

__ 4. Alter the boot sequence of your system using the **bootlist** command. Set the normal boot list so it contains only the bootable hard disk.

__ 5. The Logical Volume Manager uses names **and** IDs when storing information. Complete the following table that maps names to IDs:

rootvg ID:	
first disk pvid:	
second disk pvid:	

Which command did you use to determine the **rootvg** ID?

_____

Which command did you use to determine the physical volume IDs?

_____

Using **odmget**, identify the attribute **pvid** of one of your disks from ODM class **CuAt**. Write down the command you used.

_____

What difference do you see when looking at the shown IDs?

_____

_____

Repair a corrupted boot logical volume

__ 6. Execute the program /**home**/**workshop**/**ex3pro1**. When the prompt is returned, shut down and **reboot** the system.

__ 7. What happens on your system during the reboot?

_____

_____

_____

_____

_____

__ 8. To fix this boot problem, the boot logical volume needs to be repaired. Boot your system in maintenance mode as described in the students notebook, using your AIX CD.

Write down the steps you execute to get to a maintenance shell, where you can repair the corrupted boot logical volume.

Access the **rootvg** with all mounted file systems.

_____

_____

_____

__ 9. In the maintenance shell, repair the boot logical volume. Write down the command you used.

_____

__ 10. If the command executes successfully, reboot your system in normal mode.

END OF LAB

Exercise Instructions With Hints

Working with Boot Lists and Identifying information on your system

__ 1. Write down the boot sequence of your RS/6000 model for a **normal** boot:

1) boot device:

What is the command you used to determine the boot list:

_____

Hint: Use **bootlist** command.

__ 2. Does your model support a boot list for **maintenance** mode?

_____

Hint: Use **bootlist** to check.

If your model supports a service boot list for maintenance mode, write down the boot sequence for this boot mode:

1) boot device:
2) boot device:
3) boot device:
4) boot device:

Hint: Use **bootlist** with mode **service**.

__ 3. Identify which disks are contained within the **rootvg**:

_____

Write down the command you used.

_____

Hint: Use **lsvg -p**.

Identify which disk is the bootable disk, that means the disk that contains the boot logical volume **hd5**:

_____

Write down the command you used.

_____

Hint: Use **lspv** command plus an option that shows all logical volumes.

Identify the **logical volume type** of **hd5**:

_____

Write down the command you used.

_____

Hint: Use **lsvg** command plus an option that shows all logical volumes.

Identify LVM information from your system

__ 4. Alter the boot sequence of your system using the **bootlist** command. Set the normal boot list so it contains only the bootable hard disk.

Hint: bootlist -m xxx.

__ 5. The Logical Volume Manager uses names **and** IDs when storing information. Complete the following table that maps names to IDs:

rootvg ID:	
first disk pvid:	
second disk pvid:	

Which command did you use to determine the **rootvg** ID?

_____

Which command did you use to determine the physical volume IDs?

_____

Hint: Use **lsvg** and **lspv**.

Using **odmget**, identify the attribute **pvid** of one of your disks from ODM class **CuAt**. Write down the command you used.

_____

Hint: odmget -q"name=xxxxx and xxxxxxxxx=pvid" xxxx

What difference do you see when looking at the shown IDs?

_____

_____

Hint: Check how many bytes are used for the physical volume ID.

Repair a corrupted boot logical volume

__ 6. Execute the program /**home**/**workshop**/**ex3pro1**. When the prompt is returned, shut down and **reboot** the system.

__ 7. What happens on your system during the reboot?

_____

_____

_____

_____

_____

Hint: Check the LED/LCD display. Do you get any errors during the POST?

__ 8. To fix this boot problem, the boot logical volume needs to be repaired. Boot your system in maintenance mode as described in the students notebook, using your AIX CD.

Write down the steps you execute to get to a maintenance shell, where you can repair the corrupted boot logical volume.

Important: Access the **rootvg** with all mounted file systems.

_____

_____

_____

Hint: Insert your AIX CD into your system. Check your student notebook on how to access a **rootvg** in maintenance mode.

__ 9. In the maintenance shell, repair the boot logical volume. Write down the command you used.

_____

Hint: Use **bosboot** command.

__ 10. If the command executes successfully, reboot your system in the normal mode.

END OF LAB

Exercise Instructions With Solutions

Working with Boot Lists and Identifying information on your system

__ 1. Write down the boot sequence of your RS/6000 model for a **normal** boot:

1) boot device:

What is the command you used, to determine the boot list:

For example: bootlist -m normal -o

__ 2. Does your model support a customized service boot list?

 # bootlist -m service -o

For 43P Model 150: YES

If your model supports a service boot list for maintenance mode, write down the boot sequence for this boot mode:

1) boot device:
2) boot device:
3) boot device:
4) boot device:

__ 3. Identify which disks are contained within the **rootvg**:

_____

Write down the command you used.

 # lsvg -p rootvg

Identify which disk is the bootable disk, that means the disk that contains the boot logical volume **hd5**:

_____

Write down the command you used.

lspv -l hdisk0 (for example) or **lslv -m hd5**

Identify the **logical volume type** of **hd5**.

Write down the command you used.

 # lsvg -l rootvg

TYPE: boot

Identify LVM information from your system

__ 4. Alter the boot sequence of your system using the **bootlist** command. Set the "normal" boot list so it contains only the bootable hard disk.

 # bootlist -m normal hdisk0

Or if you want to try out the system management services programs:
1. Reboot the system.
2. When icons or text icons appear on the screen (you'll hear an acoustic signal at the same time), press **F1** or **1**. From there select boot devices.

__ 5. The Logical Volume Manager uses names **and** IDs when storing information. Complete the following table that maps names to IDs:

rootvg ID:	
first disk pvid:	
second disk pvid:	

Which command did you use to determine the **rootvg** ID?

 # lsvg rootvg

Which command did you use to determine the physical volume IDs?

 # lspv

Using **odmget**, IDentify the attribute **pvid** of one of your disks from ODM class **CuAt**. Write down the command you used.

odmget -q "name=hdisk0 and attribute=pvid" CuAt

What difference do you see with the ID value?

The ODM stores physical volume IDs in a 32-number field, and adds 16 zeros to the ID of the disk. lspv just shows 16 bytes.

Repair a corrupted boot logical volume

__ 6. Execute the program /**home**/**workshop**/**ex3pro1**. When the prompt is returned, shut down and **reboot** the system.

 # /home/workshop/ex3pro1
 # shutdown -Fr

__ 7. What happens on your system during the reboot?

Your systems shows 20EE000B on the console display, a series of 4 digit "E" codes on the operator panel LED/LCD and continues with reboot attempts. The boot record at the begin of your boot disk pointing to hd5 has been removed.

__ 8. To fix this boot problem, the boot logical volume needs to be repaired. Boot your system in maintenance mode as described in the students notebook, using your AIX CD.

Write down the steps you execute to get to a maintenance shell, where you can repair the corrupted boot logical volume.

Access the **rootvg** with all mounted file systems.

Insert the AIX CD into the system and press F5 or 5 when you hear the tones and see the icons.

Select your terminal.
Select your language.
From the Installation and Maintenance Menu select:
3 Start maintenance Mode for System Recovery
1 Access a Root Volume Group
0 Continue
1 Volume Group (containing rootvg)
1 Access this Volume Group and start a shell

__ 9. In the maintenance shell, repair the boot logical volume. Write down the command you used.

 # ipl_varyon -i

 # bosboot -ad /dev/hdisk0 (for example)

 # ipl_varyon -i

 # sync

 # sync

__ 10. If the command executes successfully, reboot your system in normal mode.

Remove the AIX CD from the system.
Do a proper shutdown:
 # shutdown -Fr

END OF LAB

Exercise 4. System Initialization Part 2

What This Exercise Is About

This exercise will review the software boot process of an RS/6000.

What You Should Be Able to Do

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

- Boot a machine in maintenance mode
- Repair a corrupted log logical volume
- Analyze and fix an unknown boot problem

Introduction

This exercise has two parts:

1. Repair a corrupted log logical volume
2. Analyze and fix a boot failure

All instructions in this exercise require **root** authority.

Required Material

- Program /**home**/**workshop**/**ex4pro1**
- AIX 5.2 CD or **mksysb** tape

Exercise Instructions

Before starting the lab, read the following paragraph carefully.

If you have any questions, ask the instructor **before** starting the exercise steps.

Files or directories which are created or updated are stored with their i-nodes and the superblock of the file system in memory first. All write requests are always handled in memory first to improve system performance. Every minute or 16 KB of changes the **syncd** daemon writes the changes from memory to disk.

All changes in the AIX-JFS file systems (superblock, i-nodes, list of free data blocks, and so forth) are recorded in a log logical volume. The **rootvg** uses as default the log logical volume /**dev**/**hd8**. When the changes are written to the disk, the JFS transactions are removed from the log logical volume. This guarantees the integrity of a file system. Until the file system changes are written to disk, the changes are recorded and held in the log logical volume.

In the first part of the lab, we corrupt the jfslog to stress a boot failure.

Repair a Corrupted Log Logical Volume

__ 1. Execute the program /**home**/**workshop**/**ex4pro1**. This program takes about 30 seconds to run. It will shut down your machine.

__ 2. Power on your system.

__ 3. Reboot your system. What happens during the reboot? Examine your student notebook to find an explanation for the boot failure.

_____

_____

__ 4. Boot your machine in maintenance mode. From the maintenance menu access the rootvg **before** mounting the file systems. You need to do this, because the mount's of the file systems will fail due to the corrupted log logical volume.

__ 5. Initialize a new log logical volume. Be sure to do a file system check for all file systems that use /**dev**/**hd8**. If you like use set -o emacs or set -o vi. Write down the commands you used:

_____

_____

_____

_____

_____

_____

__ 6. Type **exit** to leave the maintenance shell. What happens?

_____

_____

_____

__ 7. Shut down your system and **reboot your system in normal mode**.

Analyze and fix a boot failure

__ 8. What happens during the reboot of the system? Write down the last LED code that is shown.

_____

_____

__ 9. Reboot the system in maintenance mode. Examine your system and find the corrupted file that leads to the boot failure.

Be sure to set the **TERM** variable to **lft**, if you are working on a graphical display. Otherwise **vi** or **smit** will not work correctly in the maintenance shell.

__ 10. Repair the corrupted file. You'll find an example in your student notebook. If you are not able to fix the boot failure, contact your instructor.

__ 11. Shut down your system and **reboot your system in normal mode**. Your machine should boot now without any boot failure.

END OF LAB

Exercise Instructions With Hints

Before starting the lab, read the following paragraph carefully.

If you have any questions, ask the instructor **before** starting the exercise steps.

Files or directories which are created or updated are stored with their i-nodes and the superblock of the file system in memory first. All write requests are always handled in memory first to improve system performance. Every minute or 16 KB of changes the **syncd** daemon writes the changes from memory to disk.

All changes in the AIX-JFS file systems (superblock, i-nodes, list of free data blocks, and so forth) are recorded in a log logical volume. The **rootvg** uses as default the log logical volume /**dev**/**hd8**. When the changes are written to the disk, the JFS transactions are removed from the log logical volume. This guarantees the integrity of a file system. Until the file system changes are written to disk, the changes are recorded and held in the log logical volume.

In the first part of the lab, we corrupt the jfslog to stress a boot failure.

Repair a Corrupted Log Logical Volume

__ 1. Execute the program /**home**/**workshop**/**ex4pro1**. This program takes about 30 seconds to run. It will shut down your machine.

__ 2. Power on your system.

__ 3. Reboot your system. What happens during the reboot? Examine your student notebook to find an explanation for the boot failure.

_____

_____

Hint: Search the LED in the index section of your student notebook.

__ 4. Boot your machine in maintenance mode. From the maintenance menu access the rootvg **before** mounting the file systems. You need to do this, because the mount's of the file systems will fail due to the corrupted log logical volume.

__ 5. Initialize a new log logical volume. Be sure to do a file system check for all file systems that use /**dev**/**hd8**. If you like use set -o emacs or set -o vi. Write down the commands you used:

_____

_____

_____

_____

_____

_____

Hint: Search for **logform** in the index section of your student notebook.

__ 6. Type **exit** to leave the maintenance shell. What happens?

_____

_____

_____

__ 7. Shut down your system and **reboot your system in normal mode**.

Analyze and fix a boot failure

__ 8. What happens during the reboot of the system? Write down the last LED code that is shown.

_____

_____

Hint: Use the index section of your student notebook.

__ 9. Reboot the system in maintenance mode. Examine your system and find the corrupted file that leads to the boot failure.

Be sure to set the **TERM** variable to **lft**, if you are working on a graphical display. Otherwise **vi** or **smit** will not work correctly in the maintenance shell.

__ 10. Repair the corrupted file. You'll find an example in your student notebook. If you are not able to fix the boot failure, contact your instructor.

Hint: Look for the file in your index.

__ 11. Shut down your system and **reboot your system in normal mode**. Your machine should boot now without any boot failure.

END OF LAB

Exercise Instructions With Solutions

Before starting the lab, read the following paragraph carefully.

If you have any questions, ask the instructor **before** starting the exercise steps.

Files or directories which are created or updated are stored with their i-nodes and the superblock of the file system in memory first. All write requests are always handled in memory first to improve system performance. Every minute or 16 KB of changes the **syncd** daemon writes the changes from memory to disk.

All changes in the AIX-JFS file systems (superblock, i-nodes, list of free data blocks, and so forth) are recorded in a log logical volume. The **rootvg** uses as default the log logical volume /**dev**/**hd8**. When the changes are written to the disk, the JFS transactions are removed from the log logical volume. This guarantees the integrity of a file system. Until the file system changes are written to disk, the changes are recorded and held in the log logical volume.

In the first part of the lab, we corrupt the jfslog to stress a boot failure.

Repair a Corrupted Log Logical Volume

__ 1. Execute the program /**home**/**workshop**/**ex4pro1**. This program takes about 30 seconds to run. It will shut down your machine.

 # /home/workshop/ex4pro1

__ 2. Power on your system.

__ 3. What happens during the reboot? Examine your student notebook to find an explanation for the boot failure.

Machine stops with LED 557 (newer 2-lined LED/LCD will show 0553 & ROOT MNT FAILED). The mount of /dev/hd4 (root file system) failed.
Expected reason is a corrupted log logical volume.

__ 4. Boot your machine in maintenance mode. From the maintenance menu access the **rootvg** before mounting the file systems. You need to do this, because the mount's of the file systems will fail due to the corrupted log logical volume.

Boot your system either from the AIX CD or an mksysb tape.
(Remember to press F5 or 5 during bootup)
From the Installation and Maintenance Menu, select the Maintenance option (option 3).
From the Maintenance submenu, select option 1 to access a root volume group.
Select the volume group that is causing the problem.

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

(option 2).
Notice the error messages while rootvg is varied on. These provide more clues to the problem.

__ 5. Initialize a new log logical volume. Be sure to do a file system check for all file systems that use /dev/hd8. If you like use set -o emacs or set -o vi.

 # logform -V jfs /dev/hd8
logform: Destroy /dev/hd8 (y)? y

 # fsck -y -V jfs /dev/hd1
 # fsck -y -V jfs /dev/hd2
 # fsck -y -V jfs /dev/hd3
 # fsck -y -V jfs /dev/hd4
 # fsck -y -V jfs /dev/hd9var
 # fsck -y -V jfs /dev/hd10opt

__ 6. Type **exit** to leave the maintenance shell. What happens?

Normally when leaving this shell, the rootvg file systems are mounted for maintenance work. If these mounts work this is a good indication that this boot error has been fixed. In AIX V5.2 sometimes the system hangs due to the fact that the following mount of /dev/hd4 needs the -V jfs option. You have to Power-off/on the system instead of doing a shutdown -Fr as in Step 7.

__ 7. Shut down your system and reboot your system in normal mode.

 # shutdown -Fr

Then immediately remove the media from the system.

Analyze and fix a boot failure

__ 8. What happens during the reboot of the system? Write down the last LED code that is shown.

The system stops with LED 553 (newer 2-lined LED?LCD will have 0553 & PHASE 1 COMPLETE).
This is an indication for a corrupted /etc/inittab.

__ 9. Reboot the system in maintenance mode. Examine your system and find the corrupted file that leads to the boot failure.

Be sure to set the **TERM** variable to **lft**, if you are working on a graphical display. Otherwise **vi** or **smit** will not work correctly in the maintenance shell.

Boot your system either from the AIX CD or mksysb tape using F5.
From the Installation and Maintenance Menu, select the Maintenance Option (option 3).
From the Maintenance submenu, select option 1 to

access a root volume group.
Select the volume group that is causing the problem.
Access the rootvg and mount the file systems (option 1).

On a LFT terminal, issue:
 # export TERM=lft

The corrupted file is /etc/inittab.

__ 10. Repair the corrupted file. You'll find an example in your student notebook. If you are not able to fix the boot failure, contact your instructor.

Notice that the file has a semi-colon instead of a colon
as the first delimiter. Correct this by manually editing /etc/inittab.

 # vi /etc/inittab
:%s/;/:/g
:wq!

Shut down your system and reboot your system in normal mode. Your machine should boot now without any boot failure.

 # sync

 # sync

 # shutdown -Fr

END OF LAB

Exercise 5. Fixing LVM-Related ODM Problems

What This Exercise Is About

This exercise describes how to analyze and fix LVM-related ODM problems.

What You Should Be Able to Do

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

- Analyze an LVM-related ODM problem
- Fix an LVM-related ODM problem associated with the rootvg

Introduction

This exercise has two parts:

1. Analyze and fix an LVM ODM failure manually.
2. Analyze and fix an LVM ODM failure by using **rvgrecover**.

All instructions in this exercise require **root** authority.

Required Materials

- **/home/workshop/ex5_corrupt_pvid**
- **/home/workshop/ex5_corrupt_odm**
- **/home/workshop/rvgrecover**

Exercise Instructions

Analyze and Fix an LVM-related ODM Problem

__ 1. Execute **lspv** without any options to list all physical volumes in your system.

Complete the following table.

Disk name	PVID	Volume group

__ 2. Execute **lsvg -p** to list all physical volumes that are part of your **rootvg**.

Complete the following table:

PV_NAME	PV STATE	TOTAL PPs

__ 3. Execute **odmget -q** to see the disk information stored in ODM. Write down the command you used:

PV_Name	PVID

Also, write down the structure of the stanza (that is, information labels) output by the above command. You will need this information in a later lab step.

__ 4. Execute the program /**home/workshop/ex5_corrupt_pvid**.

__ 5. Repeat command **lspv** to list your physical volumes. Complete the table and compare with the table from step 1.

Disk name	PVID	Volume group

__ 6. Repeat command **lsvg -p** to list your physical volumes from **rootvg**.

What is the output from the command?

_____

_____

_____

_____

_____

_____

__ 7. You learned that LVM stores volume groups, physical volumes and logical volumes in the ODM. From the output of **lspv** and **lsvg -p** where is the data missing? Where is the problem?

- Volume group objects
- Physical volume objects
- Logical volume objects

Write down what you suspect:

_____

__ 8. Depending on your suspicion, identify the ODM entries which are shown in your **student notes** in **Unit 5**.

Find out which objects in which ODM class are missing by reviewing the material from your **student notes**.

_____

__ 9. Before you fix the problem, please consult one VGDA and compare the missing information with the data in the VGDA. Be sure that the information you wrote down in the tables above is correct, otherwise you will not be able to fix the problem.

What command allows you to query a VGDA?

_____

__ 10. Fix the ODM problem by adding the missing objects into the ODM. **Please work very carefully in this step!**

Use your **student notes** to find out the layout of the corresponding ODM class. Write down the steps you executed to fix the problem.

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

__ 11. Repeat the commands **lspv** and **lsvg -p** to check whether your fix works.

If you still have problems the stanza file you created contains a typo. Find the typo, delete the objects you just created and add the fixed file. Did you remember to include the 16 trailing zeros on your pvid value?

Analyze and Fix an LVM-related ODM Problem Using rvgrecover

__ 12. Execute the program /**home/workshop/ex5_corrupt_odm**.

__ 13. Verify the following information:

__ a. Check whether your volume groups are ok. Use **lsvg**.

__ b. Check whether your physical volumes are ok. Use **lspv**.

__ c. Check whether your logical volumes are ok. List all logical volumes that are part of your **rootvg**. Use **lsvg -l rootvg**.

What happens?

_____

_____

_____

__ 14. Display information for logical volume **hd2**. Use **lslv hd2**.

What happens?

_____

_____

_____

__ 15. Analyze the ODM problem by reviewing your **student notebook**. Compare the **ODM entries for logical volumes** from **unit 5** with the ODM objects from your system.

What causes the ODM problems?

_____

_____

__ 16. Fix the ODM problem by executing /**home**/**workshop**/**rvgrecover**. Ignore the error messages. This will take about 1 minute.

Check that your ODM problems have been fixed. Repeat **lsvg -l rootvg** and **lslv hd2**. They should work now without problems.

__ 17. Look into /**home**/**workshop**/**rvgrecover**. What two main steps fix your ODM problem?

_____

_____

END OF LAB

Exercise Instructions With Hints

Analyze and Fix an LVM-related ODM Problem

__ 1. Execute **lspv** without any options to list all physical volumes in your system.

Complete the following table.

Disk name	PVID	Volume group

__ 2. Execute **lsvg -p** to list all physical volumes that are part of your **rootvg**.

Complete the following table:

PV_NAME	PV STATE	TOTAL PPs

__ 3. Execute **odmget -q** to see the disk information stored in ODM. Write down the command you used:

PV_Name	PVID

__ 4. Execute the program **/home/workshop/ex5_corrupt_pvid**.

__ 5. Repeat command **lspv** to list your physical volumes. Complete the table and compare with the table from step 1.

Disk name	PVID	Volume group

Also, write down the structure of the stanza (that is, information labels) output by the above command. You will need this information in a later lab step.

__ 6. Repeat command **lsvg -p** to list your physical volumes from **rootvg**.

What is the output from the command?

_____

_____

_____

_____

_____

_____

Hint: Write down the IDs that are shown in the error message.

__ 7. You learned that LVM stores volume groups, physical volumes and logical volumes in the ODM. From the output of **lspv** and **lsvg -p** where is the data missing? Where is the problem?

- Volume group objects
- Physical volume objects
- Logical volume objects

Write down what you suspect:

_____

Hint: Query the VGDA and check the IDs. From this you can conclude what kind of data is wrong.

__ 8. Depending on your suspicion, identify the ODM entries which are shown in your **student notes** in **Unit 5**.

Find out which objects in which ODM class are missing by reviewing the material from your **student notes**.

_____

Hint: Use **odmget** and query CuDv, CuAt and CuDvDr as described in the student notes.

__ 9. Before you fix the problem, please consult one VGDA and compare the missing information with the data in the VGDA. Be sure that the information you wrote down in the tables above is correct, otherwise you will not be able to fix the problem.

What command allows you to query a VGDA?

_____

Hint: Use **lqueryvg** (Check your index in the student notes).

__ 10. Fix the ODM problem by adding the missing objects into the ODM. **Please work very carefully in this step!**

Use your **student notes** to find out the layout of the corresponding ODM class. Write down the steps you executed to fix the problem.

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

Hint: Create a stanza file. The layout of the ODM class is shown in the student notes. Use **odmadd** to add the objects to the ODM.

__ 11. Repeat the commands **lspv** and **lsvg -p** to check whether your fix works.

If you still have problems the stanza file you created contains a typo. Find the typo, delete the objects you just created and add the fixed file. Did you remember to include the 16 trailing zeros on your pvid value?

Hint: If the fix does not work, use **odmdelete** to delete the objects you added in the last step. Then check your stanza file.

Analyze and Fix an LVM-related ODM Problem Using rvgrecover

__ 12. Execute the program /**home**/**workshop**/**ex5_corrupt_odm**.

__ 13. Verify the following information:

__ a. Check whether your volume groups are OK. Use **lsvg**.

__ b. Check whether your physical volumes are OK. Use **lspv**.

__ c. Check whether your logical volumes are OK. List all logical volumes that are part of your **rootvg**. Use **lsvg -l rootvg**.

What happens?

_____

_____

_____

Hint: Check the TYPE information.

__ 14. Display information for logical volume **hd2**. Use **lslv hd2**.

What happens?

_____

_____

_____

Hint: Where does the matching of names to IDs take place?

__ 15. Analyze the ODM problem by reviewing your **student notebook**. Compare the **ODM entries for logical volumes** from **Unit 5** with the ODM objects from your system.

What causes the ODM problems?

_____

_____

Hint: Use **odmget** and query CuDv, CuAt and CuDvDr as described in the student notes.

__ 16. Fix the ODM problem by executing /**home**/**workshop**/**rvgrecover**. Ignore the error messages. This will take about 1 minute.

Check that your ODM problems have been fixed. Repeat **lsvg -l rootvg** and **lslv hd2**. They should work now without problems.

__ 17. Look into /**home**/**workshop**/**rvgrecover**. What two main steps fix your ODM problem?

_____

Hint: Review the description of **rvgrecover** in your student notebook.

END OF LAB

Exercise Instructions With Solutions

Analyze and Fix an LVM-related ODM Problem

__ 1. Execute **lspv** without any options to list all physical volumes in your system.

Complete the following table.

Disk name	PVID	Volume group

 # lspv

__ 2. **Execute lsvg -p** to list all physical volumes that are part of your **rootvg**.

Complete the following table:

PV_NAME	PV STATE	TOTAL PPs

 # lsvg -p rootvg

__ 3. **Execute odmget -q** to see the disk information stored in ODM. Write down the command you used:

odmget -q "name like hdisk? and attribute=pvid" CuAt

PV_Name	PVID

Also, write down the structure of the stanza (that is, information labels) output by the above command. You will need this information in a later lab step.

name =
attribute =
value =
type = “R”
generic = “D”
rep = “s”
nls_index = 2

__ 4. Execute the program /**home/workshop/ex5_corrupt_pvid**.

 # /home/workshop/ex5_corrupt_pvid

__ 5. **Repeat command lspv** to list your physical volumes. Complete the table and compare with the table from step 1.

Disk name	PVID	Volume group

 # lspv

__ 6. Repeat command **lsvg -p** to list your physical volumes from **rootvg**.

What is the output from the command?

 # lsvg -p rootvg

Depending on the number of disks in rootvg, you get error messages like the following:

0516-304 lsvg: Unable to find device id 00008371b5969c35 in the Device Configuration Database

The same table is shown as in step 2, but instead of disk names the PVIDs are shown.

__ 7. You learned that LVM stores volume groups, physical volumes and logical volumes in the ODM. From the output of **lspv** and **lsvg -p**. Where is the data missing? Where is the problem?

- Volume group objects
- Physical volume objects
- Logical volume objects

Write down what you suspect:

Physical volume objects

__ 8. Depending on your suspicion, identify the ODM entries which are shown in your student notes in unit 5.

Find out which objects in which ODM class are missing by reviewing the material from your **student notes**.

 # odmget -q "name like hdisk?" CuDv
 # odmget -q "name=hdisk0" CuAt
 # odmget -q "name=hdisk1" CuAt

The PVIDs for all disks in object class CuAt are missing.

__ 9. Before you fix the problem, please consult one VGDA and compare the missing information with the data in the VGDA. Be sure that the information you wrote down in the tables above is correct, otherwise you will not be able to fix the problem.

What command allows you to query a VGDA?

 # lqueryvg -p hdisk0 -At (for example, hdisk0)

Check the last two lines that show the PVIDs of the missing disks.

__ 10. Fix the ODM problem by adding the missing objects into the ODM. Please work very carefully in this step!

Use your **student notes** to find out the layout of the corresponding ODM class. Write down the steps you executed to fix the problem.

 # vi fix.add

Example objects (Use your information):

```
CuAt:
  name = "hdisk0"
  attribute = "pvid"
  value = "00008371b5969c3500000000000000000"
  type = "R"
  generic = "D"
  rep = "s"
  nls_index = 2

CuAt:
  name = "hdisk1"
  attribute = "pvid"
  value = "002106699b1dd4440000000000000000"
  type = "R"
  generic = "D"
  rep = "s"
  nls_index = 2
```

 # odmadd fix.add

__ 11. Repeat the commands **lspv** and **lsvg -p** to check whether your fix works.

If you still have problems the stanza file you created contains a typo. Find the typo, delete the objects you just created and add the fixed file. Did you remember to include the 16 trailing zeros on your pvid valve?

 # lspv
 # lsvg -p rootvg

If your fix does not work:

 # odmdelete -o CuAt -q"attribute=pvid"

Fix the typo, and add the fixed objects:
 # odmadd fix.add

Analyze and Fix an LVM-related ODM Problem Using rvgrecover

__ 12. Execute the program /**home/workshop/ex5_corrupt_odm**.

 # /home/workshop/ex5_corrupt_odm

__ 13. Verify the following information:

__ a. Check whether your volume groups are ok. Use **lsvg**.

 # lsvg

__ b. Check whether your physical volumes are ok. Use **lspv**.

 # lspv

__ c. Check whether your logical volumes are ok. List all logical volumes that are part of your **rootvg**. Use **lsvg -l rootvg**.

 # lsvg -l rootvg

What happens?

The TYPE information for some logical volume is not shown (???).

The logical volume type is stored in CuAt so we suspect that the logical volume objects in the ODM are not correct.

__ 14. Display information for logical volume **hd2**. Use **lslv hd2**.

 # lslv hd2

What happens?

The following error message occurs:
0516-306 lslv: Unable to find hd2 in the Device Configuration Database.

__ 15. Analyze the ODM problem by reviewing your student notebook. Compare the ODM entries for logical volumes from **unit 5** with the ODM objects from your system.

What causes the ODM problems?

 # odmget -q "name=hd2" CuDv
 # odmget -q "name=hd4" CuDv

==> The logical volumes are missing in CuDv.

 # odmget -q "name=hd2" CuAt
 # odmget -q "name=hd4" CuAt

==> The LVIDs are missing in CuAt.

__ 16. Fix the ODM problem by executing /**home**/**workshop**/**rvgrecover**. Ignore the messages. This will take about 1 minute.

 # /home/workshop/rvgrecover

Check that your ODM problems have been fixed. Repeat **lsvg -l rootvg** and **lslv hd2**. They should work now without problems.

 # lsvg -l rootvg

 # lslv hd2

__ 17. Look into /**home**/**workshop**/**rvgrecover**. What two main steps fix your ODM problem?

__ **a. Deleting all ODM objects from rootvg**

__ **b. Importing new ODM objects by reading the information from VGDA and LVCB on the boot disk.**

END OF LAB

Exercise 6. Mirroring rootvg

What This Exercise Is About

This exercise describes how to mirror the rootvg.

What You Should Be Able to Do

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

- Mirror the rootvg
- Describe physical volume states
- Unmirror the rootvg

Introduction

This exercise is to mirror and unmirror rootvg

All instructions in this exercise require **root** authority.

Required Materials

- **/home/workshop/ex6_diskfailure**

Exercise Instructions

Exercise: Mirror and Unmirror the Complete rootvg

__ 1. Write down on which disks rootvg resides. You might have a mixed installation, where the **rootvg** logical volumes are spread over two disks. This step is important because you need to specify later the target disk for the new mirror.

Which command displays the logical volumes that are contained on a disk?

_____

_____

__ 2. Now mirror each logical volume as described in your **student notebook**. If you have a mixed **rootvg** installation, you must be careful when specifying the target disk name.

Do not synchronize the logical volumes in this step.

Write down the commands you executed in this step:

_____

_____

_____

_____

_____

_____

_____

_____

_____

__ 3. Display information about your **rootvg** using **lsvg rootvg**.

Complete the following information:

- **Stale physical volumes:**
- **Stale physical partitions:**

__ 4. **Now synchronize your rootvg**. Depending on your system this step takes about 15 minutes to complete. After starting this command take a break.

Write down the command you executed:

_____

__ 5. Check by using **lsvg rootvg** that all partitions have been updated.

__ 6. Update your boot logical volumes and your boot list and reboot your system.

Write down the commands you executed:

_____

_____

__ 7. The procedure /**home**/**workshop**/**ex6_diskfailure** simulates a disk failure. This procedure requires that your rootvg is mirrored completely.

Execute the program /**home**/**workshop**/**ex6_diskfailure**. Create some files in the /**tmp** file system after running the program.

_____

_____

__ 8. Analyze your AIX error log. Working with the AIX error log is introduced later in this course, so use smit to display the information in the error log:

- **smitty errpt**
- Select **filename (defaults to stdout)**
- Select **1 no**: No CONCURRENT error reporting
- Select **detailed** error report.

Browse through your error report and identify the error logs that have been created by the LVM. Here is some room for you to make notices about the error logs:

_____

_____

_____

_____

_____

_____

_____

_____

_____

__ 9. Use **lspv** to display information about your physical volumes. Analyze the column **PV STATE** to check the states of your disks.

Which disk is causing problems? Which **PV STATE** has been allocated to the failing disk?

_____

_____

_____

__ 10. Review the page in your **student notebook** that describes **Physical Volume States (unit 5)**. Find the physical volume state of the failing disk on the picture.

__ 11. Execute a **varyonvg rootvg** and check whether this fixes the disk problem.

_____

What happens? Check the **PV STATE** of the failing disk:

_____

_____

_____

__ 12. By reviewing your **student notebook**, which command brings the disk back into the **ACTIVE** state?

_____

Execute this command and check the **PV STATE** of the failing disk.

_____

_____

__ 13. Check whether your **rootvg** still contains **stale** partitions.

_____

_____

__ 14. Again review your material. What command is the best to fix the stale partitions?

_____

Execute the command and check if the stale partitions are fixed.

_____

_____

_____

__ 15. Unmirror the **rootvg** of your system.

Important: Unmirror your **rootvg** in a way so that one disk is **completely empty**. We need an empty disk in our next exercise.

Decide which of your disks you want to unmirror.

Write down the command you executed to unmirror your **rootvg**:

_____

What recommendation do you get when executing this command?

_____

_____

Follow this recommendation and remove the old boot sector.

_____

__ 16. Check that all logical volumes have been removed from the disk.

_____

__ 17. Finally update your boot logical volume and your boot list.

Write down the commands you executed:

_____

_____

END OF LAB

Exercise Instructions With Hints

Exercise: Mirror and Unmirror the Complete rootvg

__ 1. Write down on which disks your rootvg resides. You might have a mixed installation, where the **rootvg** logical volumes are spread over two disks. This step is important because you need to specify later the target disk for the new mirror.

Which command displays the logical volumes that are contained on a disk?

_____

_____

Hint: Use **lspv**.

__ 2. Now mirror each logical volume as described in your **student notebook**. If you have a mixed **rootvg** installation, you must be careful when specifying the target disk name.

Do not synchronize the logical volumes in this step.

Write down the commands you executed in this step:

_____

_____

_____

_____

_____

_____

_____

_____

_____

Hint: Use **mirrorvg** or **mklvcopy**. Do not forget to extend your rootvg by a new disk and disable the quorum.

__ 3. Display information about your **rootvg** using **lsvg rootvg**.

Complete the following information:

- **Stale physical volumes:**
- **Stale physical partitions:**

__ 4. Now synchronize your **rootvg**. Depending on your system this step takes about 10 minutes to complete. After starting this command take a break.

Write down the command you executed:

_____

Hint: Use **syncvg**.

__ 5. Check by using **lsvg rootvg** that all partitions have been updated.

__ 6. Update your boot logical volumes and your boot list and reboot your system.

Write down the commands you executed:

_____

_____

Hint: Use **bosboot** and **bootlist**.

__ 7. The procedure /**home**/**workshop**/**ex6_diskfailure** simulates a disk failure. This procedure requires that your rootvg is mirrored completely.

Execute the program /**home**/**workshop**/**ex6_diskfailure**. Create some files in the /**tmp** file system after running the program.

_____

_____

__ 8. Analyze your AIX error log. Working with the AIX error log is introduced later in this course, so use smit to display the information in the error log:

- **smitty errpt**
- Select **filename (defaults to stdout)**
- Select **1 no**: No CONCURRENT error reporting
- Select **detailed** error report.

Browse through your error report and identify the error logs that have been created by the LVM. Here is some room for you to make notices about the error logs:

_____

_____

_____

_____

_____

_____

_____

_____

_____

__ 9. Use **lspv** to display information about your physical volumes. Analyze the column **PV STATE** to check the states of your disks.

Which disk is causing problems? Which **PV STATE** has been allocated to the failing disk?

_____

_____

_____

__ 10. Review the page in your **student notebook** that describes **Physical Volume States (unit 5)**. Find the physical volume state of the failing disk on the picture.

__ 11. Execute a **varyonvg rootvg** and check whether this fixes the disk problem.

_____

What happens? Check the **PV STATE** of the failing disk:

_____

_____

_____

__ 12. By reviewing your **student notebook**, which command brings the disk back into the **ACTIVE** state?

_____

Hint: Use **chpv**.

Execute this command and check the **PV STATE** of the failing disk.

_____

_____

Hint: Use **lspv** to check the PV state.

__ 13. Check whether your **rootvg** still contains **stale** partitions.

_____

_____

Use **lsvg**.

__ 14. Again review your material. What command is the best to fix the stale partitions?

_____

Hint: Use **varyonvg**.

Execute the command and check if the stale partitions are fixed.

_____

_____

_____

__ 15. Unmirror the **rootvg** of your system.

Important: Unmirror your **rootvg** in a way so that one disk is **completely empty**. We need an empty disk in our next exercise.

Decide which of your disks you want to unmirror.

Write down the command you executed to unmirror your **rootvg**:

_____

Hint: Use **unmirrorvg**.

What recommendation do you get when executing this command?

_____

_____

Follow this recommendation and remove the old boot sector.

_____

__ 16. Check that all logical volumes have been removed from the disk.

_____

Hint: Use **lspv**.

__ 17. Finally update your boot logical volume and your boot list.

Write down the commands you executed:

_____

_____

Hint: Use **bosboot** and **bootlist**.

END OF LAB

Exercise Instructions With Solutions

Exercise: Mirror and Unmirror the Complete rootvg

__ 1. Write down on which disks your rootvg resides. You might have a mixed installation, where the **rootvg** logical volumes are spread over two disks. This step is important because you need to specify later the target disk for the new mirror.

Which command displays the logical volumes that are contained on a disk?

 # lspv

__ 2. Now mirror each logical volume as described in your student notebook. If you have a mixed **rootvg** installation, you must be careful when specifying the target disk name.

Do not synchronize the logical volumes in this step.

Write down the commands you executed in this step:

Solution A:

```
# extendvg rootvg hdisk1
# chvg -Qn rootvg          (Disable quorum)
# mirrorvg -s rootvg
```

or

Solution B:

```
# extendvg rootvg hdisk1 (for example, target disk hdisk1)
# chvg -Qn rootvg          (Disable quorum)
# mklvcopy hd1 2 hdisk1
# mklvcopy hd2 2 hdisk1
# mklvcopy hd3 2 hdisk1
# mklvcopy hd4 2 hdisk1
# mklvcopy hd5 2 hdisk1
# mklvcopy hd6 2 hdisk1
# mklvcopy hd8 2 hdisk1
# mklvcopy hd9var 2 hdisk1
# mklvcopy hd10var 2 hdisk1

Do the same for other rootvg logical volumes
```

__ 3. Display information about your **rootvg** using **lsvg rootvg**.

 # lsvg rootvg

Complete the following information:

- **Stale physical volumes:**

- **Stale physical partitions:**

__ 4. Now synchronize your **rootvg**. Depending on your system this step takes about 10 minutes to complete. After starting this command take a break.

Write down the command you executed:

 # syncvg -v rootvg

__ 5. Check by using **lsvg rootvg** that all partitions have been updated.

 # lsvg rootvg

__ 6. Update your boot logical volumes and your boot list.

Write down the commands you executed:

 # bosboot -a
(without -d, bosboot will default to the disk used on last boot)
 # bootlist -m normal hdisk1 hdisk0 (for example, hdisk1 and hdisk0)
 # ipl_varyon -i
 # lslv -m hd5
 # shutdown -Fr
 # bootinfo -b

__ 7. The procedure /**home**/**workshop**/**ex6_diskfailure** simulates a disk failure. This procedure requires that your rootvg is mirrored completely.

Execute the program /**home**/**workshop**/**ex6_diskfailure**. Create some files in the /**tmp** file system after running the program.

 # errclear 0
 # /home/workshop/ex6_diskfailure
 # touch /tmp/data1
 # touch /tmp/data2

__ 8. Analyze your AIX error log. Working with the AIX error log is introduced later in this course, so use smit to display the information in the error log:

- **smitty errpt**
- Select **filename (defaults to stdout)**
- Select **1 no**: No CONCURRENT error reporting
- Select **detailed** error report.

Browse through your error report and identify the error logs that have been created by the LVM. Here is some room for you to make notices about the error logs:

You should see many error log entries like the following:

1. LABEL: LVM_SA_STALEPP
Description
PHYSICAL PARTITION MARKED STALE

2. LABEL: LVM_SA_PVMISS

Description
PHYSICAL VOLUME DECLARED MISSING

__ 9. Use **lspv** to display information about your physical volumes. Analyze the column **PV STATE** to check the states of your disks.

Which disk is causing problems? Which **PV STATE** has been allocated to the failing disk?

 # lspv hdisk0
 # lspv hdisk1

One of the disks is in the REMOVED state, and causes stale partitions.

__ 10. Review the page in your **student notebook** that describes **Physical Volume States (unit 5)**. Find the physical volume state of the failing disk on the picture.

__ 11. Execute a **varyonvg rootvg** and check whether this fixes the disk problem.

 # varyonvg rootvg

What happens? Check the PV STATE of the failing disk:

Many errors occur.

The disk still has the REMOVED state.

__ 12. By reviewing your **student notebook**, which command brings the disk back into the **ACTIVE** state?

 # chpv -v a hdisk1 (for example, hdisk1)

Execute this command and check the PV STATE of the failing disk.

 # lspv hdisk1 (for example, hdisk1)

The disk is back in the active state.

__ 13. Check whether your **rootvg** still contains **stale** partitions.

 # lsvg rootvg

rootvg still contains stale partitions.

__ 14. Again review your material. What command is the best to fix the stale partitions?

 # varyonvg rootvg

Execute the command and check if the stale partitions are fixed.

varyonvg starts a syncvg in background. Note that the synchronization takes some time. varyonvg is better than syncvg, because it recovers a disk reservation. syncvg is not able to do this.

__ 15. Unmirror the **rootvg** of your system.

Important: Unmirror your **rootvg** in a way so that one disk is **completely empty**. We need an empty disk in our next exercise.

Decide which of your disks you want to unmirror.

Write down the command you executed to unmirror your **rootvg**:

 # unmirrorvg rootvg hdisk1 (for example, hdisk1)

What recommendation do you get when executing this command?

rmlvcopy recommends to do a chpv -c <diskname> to remove an old boot sector.

Follow this recommendation and remove the old boot sector.

 # chpv -c hdisk1 (for example, hdisk1)

__ 16. Check that all logical volumes have been removed from the disk.

 # lspv -l hdisk1 (for example, hdisk1)

hdisk1 is empty now.

 # reducevg rootvg hdisk1 (for example, hdisk1)

__ 17. Finally update your boot logical volume and your boot list.

Write down the commands you executed:

 # bosboot -ad /dev/hdisk0 (for example, hdisk0)
 # bootlist -m normal hdisk0
 # shutdown -Fr

END OF LAB

Exercise 7. Exporting and Importing Volume Groups

What This Exercise Is About

This exercise describes the steps to export and import volume groups.

What You Should Be Able to Do

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

- Export a volume group
- Import a volume group

Introduction

This exercise has two parts:

- Export and Import a Volume Group
- Analyze Import Messages (optional)

This exercise requires one disk to be completely empty. This disk will be used to create a new volume group. This volume group will be exported and imported.

All instructions in this exercise require **root** authority.

Exercise Instructions

In this exercise you will discover how exporting and importing of volume groups works.

As you learned in the units of this course, export and import is not only a way to move data from one system to another, sometimes it's the only way to correct ODM failures with non-rootvg volume groups.

Export and Import a Volume Group

__ 1. Create a **new** volume group **datavg** on the disk that is **empty**. Check that this disk does not belong to another volume group. Set the physical partition size to 16 MB.

Write down the command you (or smit) executed, to create the new volume group:

_____

__ 2. Check if the new volume group has been varied on automatically. Write down the command you used:

_____

__ 3. Use the fastpath **smit mklv** to create a logical volume in **datavg**. Please use **lv_raw** as the logical volume. Use 1 partition as **size**.

__ 4. Use the fastpath **smit jfs2** to create **two** enhanced journaled file systems in **datavg**. Use **/home/jupiter** and **/home/mars** as names. Both file systems should have a size of 16 MB (32768 512-byte blocks).

Write down the corresponding **logical volume names** that have been created for the file systems:

Logical volume for /home/jupiter:

Logical volume for /home/mars:

Verify the LVs are in datavg with **lsvg**.

__ 5. **Mount** the file systems and **create** some files in both file systems.

_____

_____

__ 6. **Export** the volume group **datavg** from your system. Write down all steps you executed to export the volume group:

_____

_____

_____

_____

_____

_____

__ 7. Analyze your system to see if it contains any reference to the exported volume group. For example, check whether the file systems you have created exist. Check /**etc**/**filesystems**.

_____

_____

__ 8. **Import** the volume group into your system. Specify volume group name **datavg**, otherwise the system will generate a new volume group name.

Write down the command you (or smit) executed:

_____

__ 9. Check whether the imported volume group **datavg** is varied on.

_____

Check to see if the file system information is back.

__ 10. Mount the file systems /**home**/**jupiter** and /**home**/**mars**. Check that no files have been lost.

_____

_____

Optional Part: Analyze Import Messages

In the first part of this exercise, the export and import worked without problems as the logical volumes and file systems did not exist during the import of the volume group.

In this second part we will change this a little bit.

__ 11. **Export** the volume group **datavg** again. Repeat the steps from the last export.

__ 12. Use the fastpath **smit mklv** to create a logical volume **lv_raw** but this time in your **rootvg**. Size should be 1 partition.

__ 13. Use the fastpath **smit jfs2** to create **two** enhanced journaled file systems in **rootvg**. Use the same names as above /**home**/**jupiter** and /**home**/**mars**. Size should be 16 MB.

Write down the corresponding **logical volume names** that has been created for the file system:

lsfs

Logical volume for /home/jupiter:

Logical volume for /home/mars:

__ 14. **Mount** the file systems /**home**/**jupiter** and /**home**/**mars** and add a few files to each.

__ 15. At this stage the following problems will come up when we import our **datavg**:

- The logical volume **lv_raw** exists already in **rootvg**.
- The file system /**home**/**jupiter** exists already in **rootvg**.
- The file system /**home**/**mars** exists already in **rootvg**.

Let's see how **importvg** will react to this situation.

Import the volume group **datavg** into the system.

_____

__ 16. Write down the new logical volume names that are created during the import.

_____

_____

_____

__ 17. Another problem that you should see at this stage is that the file systems /**home**/**jupiter** and /**home**/**mars** already exist in **rootvg**.

To fix this problem **umount** the file systems /**home**/**jupiter** and /**home**/**mars** from **rootvg** first.

_____

_____

__ 18. **Mount** the file systems from **datavg** over the corresponding **mount points**. Use the new logical volume names that have been created. You have to specify the log device that is part of **datavg**.

Write down the commands you executed:

_____

_____

Correct the problem and mount the file systems.

__ 19. Check the files you have created in /**home**/**jupiter** and /**home**/**mars**. They should exist in these directories.

__ 20. At the end of this exercise we want all four file systems be mounted at the same time.

Start with unmounting /**home**/**jupiter** and /**home**/**mars**.

_____

_____

__ 21. Create two new directories /**datavg**/**jupiter** and /**datavg**/**mars**. These will be the new mount points for our file systems from **datavg**.

_____

_____

__ 22. Create two new stanzas in /**etc**/**filesystems** that describe the file systems from **datavg**. You must use the **new logical volume names** that have been created during the import of **datavg**.

Here is some room for you to write down the stanzas you've created:

__ 23. **Mount** the file systems /**datavg**/**jupiter** and /**datavg**/**mars**.

_____

_____

__ 24. Verify you can access all the files.

__ 25. **Unmount** the file systems /**datavg**/**jupiter** and /**datavg**/**mars.**

__ 26. **Varyoff** the volume group datavg.

__ 27. **Export** the volume group datavg

__ 28. **Remove** the file systems /**home**/**jupiter** and /**home**/**mars** from the volume group rootvg.

END OF LAB

Exercise Instructions With Hints

In this exercise you will discover how exporting and importing of volume groups works.

As you learned in the units of this course, export and import is not only a way to move data from one system to another, sometimes it's the only way to correct ODM failures with non-rootvg volume groups.

Export and Import a Volume Group

__ 1. Create a **new** volume group **datavg** on the disk that is **empty**. Check that this disk does not belong to another volume group. Set the physical partition size to 16 MB.

Write down the command you (or smit) executed, to create the new volume group:

_____

Hint: Use **mkvg**. You may have to remove hdisk1 from rootvg first.

__ 2. Check if the new volume group has been varied on automatically. Write down the command you used:

_____

Hint: Use **lsvg**.

__ 3. Use the fastpath **smit mklv** to create a logical volume in **datavg**. Please use **lv_raw** as the logical volume. Use 1 partition as **size**.

__ 4. Use the fastpath **smit jfs2** to create **two** enhanced journaled file systems in **datavg**. Use /**home**/**jupiter** and /**home**/**mars** as names. Both file systems should have a size of 16 MB (32768 512-byte blocks).

Write down the corresponding **logical volume names** that have been created for the file systems:

Logical volume for /home/jupiter:

Logical volume for /home/mars:

Verify the LVs are in datavg with **lsvg**.

__ 5. **Mount** the file systems and **create** some files in both file systems.

_____

_____

Hint: Use **mount**.

__ 6. **Export** the volume group **datavg** from your system. Write down all steps you executed to export the volume group:

_____

_____

_____

_____

_____

_____

Hint: Unmount all file systems. Vary off the volume group. Export the volume group.

__ 7. Analyze your system to see if it contains any reference to the exported volume group. For example, check whether the file systems you have created exist. Check /**etc**/**filesystems**.

_____

_____

Hint: Use **lsfs**.

__ 8. **Import** the volume group into your system. Specify volume group name **datavg**, otherwise the system will generate a new volume group name.

Write down the command you (or smit) executed:

_____

Hint: Use **importvg**.

__ 9. Check whether the imported volume group **datavg** is varied on.

_____

Hint: Use **lsvg**.

Check to see if the file system information is back.

__ 10. Mount the file systems /**home**/**jupiter** and /**home**/**mars**. Check that no files have been lost.

_____

_____

Hint: Use **mount**.

Optional Part: Analyze Import Messages

In the first part of this exercise, the export and import worked without problems as the logical volumes and file systems did not exist during the import of the volume group.

In this second part we will change this a little bit.

__ 11. **Export** the volume group **datavg** again. Repeat the steps from the last export.

Hint: Unmount all file systems. Vary off the volume group. Export the volume group.

__ 12. Use the fastpath **smit mklv** to create a logical volume **lv_raw** but this time in your **rootvg**. Size should be 1 partition.

__ 13. Use the fastpath **smit jfs2** to create **two** enhanced journaled file systems in **rootvg**. Use the same names as above /**home**/**jupiter** and /**home**/**mars**. Size should be 16 MB.

Write down the corresponding **logical volume names** that has been created for the file system:

lsfs

Logical volume for /home/jupiter:

Logical volume for /home/mars:

__ 14. **Mount** the file systems /**home**/**jupiter** and /**home**/**mars** and add a few files to each.

Hint: Use **mount** and **touch**.

__ 15. At this stage the following problems will come up when we import our **datavg**:

- The logical volume **lv_raw** exists already in **rootvg**.
- The file system /**home**/**jupiter** exists already in **rootvg**.
- The file system /**home**/**mars** exists already in **rootvg**.

Let's see how **importvg** will react to this situation.

Import the volume group **datavg** into the system.

_____

Hint: Use **importvg**.

__ 16. Write down the new logical volume names that are created during the import.

_____

_____

_____

__ 17. Another problem that you should see at this stage is that the file systems /**home**/**jupiter** and /**home**/**mars** already exist in **rootvg**.

To fix this problem **umount** the file systems /**home**/**jupiter** and /**home**/**mars** from **rootvg** first.

_____

_____

Hint: Use **umount**.

__ 18. **Mount** the file systems from **datavg** over the corresponding **mount points**. Use the new logical volume names that have been created. You have to specify the log device that is part of **datavg**.

Write down the commands you executed:

_____

_____

Hint: mount -o log=xxx /dev/xxx mount-point

Write down the commands you executed:

_____

_____

__ 19. Check the files you have created in /**home**/**jupiter** and /**home**/**mars**. They should exist in these directories.

__ 20. At the end of this exercise we want all four file systems be mounted at the same time.

Start with unmounting /**home**/**jupiter** and /**home**/**mars**.

_____

_____

Hint: Use **umount**.

__ 21. Create two new directories /**datavg**/**jupiter** and /**datavg**/**mars**. These will be the new mount points for our file systems from **datavg**.

_____

_____

Hint: Use **mkdir**.

__ 22. Create two new stanzas in /**etc**/**filesystems** that describe the file systems from **datavg**. You must use the **new logical volume names** that have been created during the import of **datavg**.

Here is some room for you to write down the stanzas you've created:

/datavg/jupiter:

dev = /dev/lvxx

vfs = xxx

log = /dev/loglvxx

mount = false

options = xx

account = xx

/datavg/mars:

dev = /dev/lvxx

__ 23. **Mount** the file systems /**datavg**/**jupiter** and /**datavg**/**mars**.

_____

_____

Hint: Use **mount**.

__ 24. Verify you can access all the files.

__ 25. **Unmount** the file systems /**datavg**/**jupiter** and /**datavg**/**mars.**

Hint: Use **umount**.

__ 26. **Varyoff** the volume group datavg.

Hint: Use **varyoffvg**.

__ 27. **Export** the volume group datavg

Hint: Use **exportvg**.

__ 28. **Remove** the file systems /**home**/**jupiter** and /**home**/**mars** from the volume group rootvg.

Hint: Use **rmfs**.

END OF LAB

Exercise Instructions With Solutions

In this exercise you will discover how exporting and importing of volume groups works.

As you learned in the units of this course, export and import is not only a way to move data from one system to another, sometimes it's the only way to correct ODM failures with non-rootvg volume groups.

Export and Import a Volume Group

__ 1. Create a **new** volume group **datavg** on the disk that is **empty**. Check that this disk does not belong to another volume group. Set the physical partition size to 16 MB.

Write down the command you (or smit) executed, to create the new volume group:

 # lspv

You may need to remove hdisk1 from rootvg. If so, use:

 # reducevg rootvg hdisk1

 # mkvg -s 16 -y datavg hdisk1

__ 2. Check if the new volume group has been varied on automatically. Write down the command you used:

 # lsvg -o (lists only the active volume groups)
datavg
rootvg

In AIX 4.3 and subsequent AIX releases the volume group is automatically varied on.

__ 3. Use the fastpath **smit mklv** to create a logical volume in **datavg**. Please use **lv_raw** as the logical volume. Use 1 partition as **size**.

 # smit mklv

__ 4. Use the fastpath **smit jfs2** to create **two** enhanced journaled file systems in **datavg**. Use /**home**/**jupiter** and /**home**/**mars** as names. Both file systems should have a size of 16 MB (65536 512-byte blocks).

 # smit jfs2

Write down the corresponding logical volume and lvlog information that has been created for datavg in the last two steps. Use the lsvg -l datavg command to gather the information. Include lv_raw, the logical volume names for /home/jupiter and /home/mars, and the log logical volume.

LV NAME	TYPE	MOUNT POINT
1.		
2.		
3.		
4.		

Verify the new LVs are in datavg with lsvg.

 # lsvg -l datavg

__ 5. **Mount** the file systems and **create** some files in both file systems.

 # mount /home/jupiter
 # mount /home/mars

 # cd /home/jupiter
 # touch j1 j2 j3

 # cd /home/mars
 # touch m1 m2 m3

__ 6. **Export** the volume group **datavg** from your system. Write down all steps you executed to export the volume group:

 # cd
 # umount /home/jupiter
 # umount /home/mars
 # varyoffvg datavg
 # exportvg datavg

__ 7. Analyze your system to see if it contains any reference to the exported volume group. For example, check whether the file systems you have created exist. Check **/etc/filesystems**.

 # lsfs
/home/jupiter and /home/mars do not exist on the system.

 # more /etc/filesystems
/etc/filesystems contains no reference to a file system that has been exported.

__ 8. **Import** the volume group into your system. Specify volume group name **datavg**, otherwise the system will generate a new volume group name.

Write down the command you (or smit) executed:

 # importvg -y datavg hdisk1

__ 9. Check whether the imported volume group **datavg** is varied on.

 # lsvg -o (In AIX 4.3.2 and subsequent AIX releases it should be varied on)

Check to see if the file system information is back.

 # lsfs
 # more /etc/filesystems
 # mount

References are there but file systems are not mounted.

__ 10. Mount the file systems /**home/jupiter** and /**home/mars**. Check that no files have been lost.

 # mount /home/jupiter
 # mount /home/mars

 # ls /home/jupiter
 # ls /home/mars

Optional Part: Analyze Import Messages

In the first part of this exercise, the export and import worked without problems as the logical volumes and file systems did not exist during the import of the volume group.

In this second part we will change this a little bit.

__ 11. **Export** the volume group **datavg** again. Repeat the steps from the last export.

 # umount /home/jupiter
 # umount /home/mars
 # varyoffvg datavg
 # exportvg datavg

__ 12. Use the fastpath **smit mklv** to create a logical volume **lv_raw** but this time in your **rootvg**. Size should be 1 partition.

 # smit mklv

__ 13. Use the fastpath **smit jfs2** to create **two** enhanced journaled file systems in **rootvg**. Use the same names as above /**home/jupiter** and /**home/mars**. Size should be 16 MB.

 # smit jfs2

Write down the corresponding **logical volume names** that has been created for the file system:

 # lsfs

Logical volume for /home/jupiter: /dev/lv00 (for example)
Logical volume for /home/mars: /dev/lv01 (for example)

__ 14. **Mount** the file systems /**home/jupiter** and /**home/mars** and add a few files to each.

 # mount /home/jupiter
 # mount /home/mars
 # cd /home/jupiter
 # touch j20 j21 j22
 # cd /home/mars
 # touch m20 m21 m22

__ 15. At this stage the following problems will come up when we import our **datavg**:

- The logical volume **lv_raw, lv00** and **lv01** exists already in **rootvg**.
- The file system /**home/jupiter** exists already in **rootvg**.
- The file system /**home/mars** exists already in **rootvg**.

Let's see how **importvg** will react to this situation.

Import the volume group **datavg** into the system.

 # importvg -y datavg hdisk1

__ 16. Write down the new logical volume names that are created for datavg during the import.

lv_raw has been changed to fslv00 (Example names)
lv00 has been changed to fslv01
lv01 has been changed to fslv02

loglv00 has been changed to loglv01

mount point /home/jupiter already exists in /etc/filesystems
mount point /home/mars already exists in /etc/filesystems

__ 17. Another problem that you should see at this stage is that the file systems /**home/jupiter** and /**home/mars** already exist in **rootvg**.

To fix this problem **umount** the file systems /**home/jupiter** and /**home/mars** from **rootvg** first.

 # umount /home/jupiter
 # umount /home/mars

__ 18. **Mount** the file systems from **datavg** over the corresponding **mount points**. Use the new logical volume names that have been created. You have to specify the log device that is part of **datavg**.

Write down the commands you executed:

 # mount -o log=/dev/loglv01 -V jfs2 /dev/fslv01 /home/jupiter
 # mount -o log=/dev/loglv01 -V jfs2 /dev/fslv02 /home/mars

__ 19. Check the files you have created in /**home/jupiter** and /**home/mars**. They should exist in these directories.

 # ls /home/jupiter
 # ls /home/mars

__ 20. At the end of this exercise we want all four file systems mounted at the same time.

Start with unmounting /**home/jupiter** and /**home/mars**.

 # umount /home/jupiter
 # umount /home/mars

__ 21. Create two new directories /**datavg/jupiter** and /**datavg/mars**. These will be the new mount points for our file systems from **datavg**.

 # mkdir -p /datavg/jupiter
 # mkdir -p /datavg/mars

__ 22. Create two new stanzas in /**etc/filesystems** that describe the file systems from **datavg**. You must use the new logical volume names that have been created during the import of **datavg**.

Here is some room for you to write down the stanzas you've created:

/datavg/jupiter:
dev = /dev/fslv01
vfs = jfs2
log = /dev/loglv01
mount = false
options = rw
account = false

/datavg/mars:
dev = /dev/fslv02
vfs = jfs2
log = /dev/loglv01
mount = false
options = rw
account = false

__ 23. **Mount** the file systems /**datavg/jupiter** and /**datavg/mars**.

 # mount /datavg/jupiter
 # mount /datavg/mars
 # mount /home/jupiter
 # mount /home/mars

__ 24. Verify you can access all the files.

 # ls /datavg/jupiter
 # ls /datavg/mars
 # ls /home/jupiter
 # ls /home/mars

__ 25. **Unmount** the file systems /**datavg/jupiter** and /**datavg/mars.**

 # umount /datavg/jupiter
 # umount /datavg/mars

__ 26. **Varyoff** the volume group datavg.

 # varyoffvg datavg

__ 27. **Export** the volume group datavg

 # exportvg datavg

__ 28. **Remove** the file systems /**home/jupiter** and /**home/mars** from the volume group rootvg.

 # unmount /home/jupiter
 # unmount /home/mars
 # rmfs /home/jupiter
 # rmfs /home/mars

END OF LAB

Exercise 8. Saving and Restoring a User Volume Group

What This Exercise Is About

This exercise provides an opportunity to back up a non-rootvg volume group and then restore the data to simulate a failure of a complete volume group.

What You Should Be Able to Do

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

- Use the **savevg** command
- Change volume group characteristics
- Use the **restvg** command

Introduction

All instructions in this exercise require **root** authority. There must be enough free space on the other disk to backup the user volume group. This disk will probably be the rootvg disk. If there are two students using the system, they must work on this exercise together. **NOTE:** It is imperative that **exercise 7** (Exporting and Importing Volume Groups) was completed. This exercise assumes a user volume group (datavg) was created.

Exercise Instructions

__ 1. Check that your user volume group **datavg** is defined and varied on.

_____

__ 2. Write down the **physical partition size** of **datavg**:

_____

_____

__ 3. Write down the number of partitions that are allocated for /**home**/**jupiter**:

_____

_____

__ 4. Execute the **mkvgdata** command to create a control file for the **savevg** command. Write down the command you executed:

_____

__ 5. Before saving the volume group **datavg**, change the control file that is used during the restore process. Edit the file and change the following volume group characteristic:

- Change the number of logical partitions that are allocated for /**home**/**jupiter** to **4**.

_____

_____

_____

_____

__ 6. Back up your user volume group **datavg** to a file image. Do not use **smit** to save the volume group. Write down the command you used:

_____

__ 7. Unmount all file systems from **datavg**. Write down the commands you used:

_____

_____

__ 8. Varyoff the volume group **datavg**. Write down the command you used:

_____

__ 9. Export the volume group from the system. Write down the command you used:

_____

__ 10. Execute the **restvg** command and restore the volume group from your backup image. Write down the command you used:

_____

__ 11. Write down the number of partitions that are allocated for /**home/jupiter**:

_____

_____

__ 12. Using **smit**, save the volume group **datavg** again. Specify the same backup file image as before. Write down the command that **smit** executes:

_____

_____

__ 13. Execute the same steps as before (umounts, varyoffvg, exportvg) to remove the complete volume group **datavg** from the system:

_____

_____

_____

_____

__ 14. Using **smit**, restore the volume group **datavg** from the file image. In **smit**, specify a bigger partition size, for example **32** MB. Write down the command that **smit** executes:

_____

_____

__ 15. After restoring the volume group, check the partition size of **datavg**:

_____

__ 16. Write down the number of partitions that are allocated for /**home/jupiter**:

_____

_____

Do you still have four partitions for /**home/jupiter**?

_____

_____

END OF LAB

Exercise Instructions With Hints

__ 1. Check that your user volume group **datavg** is defined and varied on.

_____

Hint: Use **lsvg**. And may have to use **importvg**.

__ 2. Write down the **physical partition size** of **datavg**:

_____

_____

Hint: Use **lsvg** again.

__ 3. Write down the number of partitions that are allocated for /**home**/**jupiter**:

_____

_____

Hint: Use **lsvg -l**.

__ 4. Execute the **mkvgdata** command to create a control file for the **savevg** command. Write down the command you executed:

_____

Hint: mkvgdata vgname

__ 5. Before saving the volume group **datavg**, change the control file that is used during the restore process. Edit the file and change the following volume group characteristic:

- Change the number of logical partitions that are allocated for /**home**/**jupiter** to **4**. Also change the size of the filesystem to 4 times its current value.

_____

_____

_____

_____

Hint: The file is located in /**tmp**/**vgdata**/**datavg**. You must change the LPs attribute in the lv_data stanza.

__ 6. Back up your user volume group **datavg** to a file image. Do not use **smit** to save the volume group. Write down the command you used:

_____

Hint: savevg -f filename vgname

__ 7. Unmount all file systems from **datavg**. Write down the commands you used:

_____

_____

Hint: Use **umount**.

__ 8. Varyoff the volume group **datavg**. Write down the command you used:

_____

Hint: Use **varyoffvg**.

__ 9. Export the volume group from the system. Write down the command you used:

_____

Hint: Use **exportvg**.

__ 10. Execute the **restvg** command and restore the volume group from your backup image. Write down the command you used:

_____

Hint: restvg -f filename diskname

__ 11. Write down the number of partitions that are allocated for /**home**/**jupiter**:

_____

_____

Hint: Use **lsvg -l**.

__ 12. Using **smit**, save the volume group **datavg** again. Specify the same backup file image as before. Write down the command that **smit** executes:

_____

_____

Hint: Use **smit savevg**.

__ 13. Execute the same steps as before (umounts, varyoffvg, exportvg) to remove the complete volume group **datavg** from the system:

_____

_____

_____

_____

__ 14. Using **smit**, restore the volume group **datavg** from the file image. In **smit**, specify a bigger partition size, for example **32** MB. Write down the command that **smit** executes:

_____

_____

Hint: Use **smit restvg**.

__ 15. After restoring the volume group, check the partition size of **datavg**:

_____

Hint: Use **lsvg**.

__ 16. Write down the number of partitions that are allocated for /**home/jupiter**:

_____

_____

Do you still have four partitions for /**home/jupiter**?

_____

_____

Hint: Remember: What happens if the partition size has been changed?

END OF LAB

Exercise Instructions With Solutions

__ 1. Check that your user volume group **datavg** is defined and varied on.

 # lsvg -o

If its not defined, use

 # importvg -y datavg hdisk1

__ 2. Write down the physical partition size of **datavg**:

 # lsvg datavg

(Should be 16 megabytes.)

__ 3. Write down the number of partitions that are allocated for /**home**/**jupiter**:

 # lsvg -l datavg

(Should be 1 partition)

__ 4. Execute the **mkvgdata** command to create a control file for the **savevg** command. Write down the command you executed:

 # mkvgdata datavg

__ 5. Before saving the volume group **datavg**, change the control file that is used during the restore process. Edit the file and change the following volume group characteristic:

- Change the number of logical partitions that are allocated for /**home**/**jupiter** to **4**.

  # vi /tmp/vgdata/datavg/datavg.data

  lv_data:
  ...
  LPs=4
  MOUNT_POINT=/home/jupiter
  ...

  fs_data:
  fs_name=/home/jupiter
  fs_size=131072 (4 times original value; 4 x 32768 = 131072)
  ...

__ 6. Back up your user volume group **datavg** to a file image. Do not use **smit** to save the volume group. Write down the command you used:

 # savevg -f /tmp/datavg.img datavg

__ 7. Unmount all file systems from **datavg**. Write down the commands you used:

 # umount /home/mars
 # umount /home/jupiter

__ 8. Varyoff the volume group **datavg**. Write down the command you used:

 # varyoffvg datavg

__ 9. Export the volume group from the system. Write down the command you used:

 # exportvg datavg

__ 10. Execute the **restvg** command and restore the volume group from your backup image. Write down the command you used:

 # restvg -f /tmp/datavg.img hdisk1

__ 11. Write down the number of partitions that are allocated for /**home/jupiter**:

 # lsvg -l datavg

(Should be 4 partitions now)

__ 12. Using **smit**, save the volume group **datavg** again. Specify the same backup file image as before. Write down the command that **smit** executes:

 # smit savevg
Backup DEVICE or FILE **/tmp/datavg.img**
VOLUME GROUP to back up **datavg**

__ 13. Execute the same steps as before (umounts, varyoffvg, exportvg) to remove the complete volume group **datavg** from the system:

 # umount /home/mars
 # umount /home/jupiter
 # varyoffvg datavg
 # exportvg datavg

__ 14. Using **smit**, restore the volume group **datavg** from the file image. In **smit**, specify a bigger partition size, for example **64** MB. Write down the command that **smit** executes:

 # smit restvg
Restore DEVICE or FILE /tmp/datavg.img
Physical partition SIZE 64

__ 15. After restoring the volume group, check the partition size of **datavg**:

 # lsvg datavg

__ 16. Write down the number of partitions that are allocated for /**home/jupiter**:

 # lsvg -l datavg

==> Should be 1 partition now, because the PP size was increased.

Do you still have four partitions for /datavg/jupiter?

No. When the partition size is not the same as specified in vgname.data, each logical volume will be altered with respect to the new partition size.

END OF LAB

Exercise 9. Working with syslogd and errnotify

What This Exercise Is About

This exercise allows the user to work with **syslogd** daemon and the ODM Error Notification class **errnotify**.

What You Should Be Able to Do

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

- Identify errors and warnings sent by the **syslogd** daemon
- Create and maintain the /**etc**/**syslog.conf** file
- Automate error logging with **errnotify**
- Redirect **syslogd** messages to the Error Log

Introduction

All instructions in this exercise require **root** authority.

Exercise Instructions

Working with syslogd

__ 1. Edit the /**etc**/**syslog.conf** file and configure the **syslogd** daemon to log all daemon messages to a file with the name /**tmp**/**syslog.debug**.

Write down the line that you added to /**etc**/**syslog.conf**:

_____

__ 2. Execute the **touch** command and create the file /**tmp**/**syslog.debug**.

_____

__ 3. Refresh the **syslogd** daemon so it will pick up the changes. Write down the command that you used:

_____

__ 4. Stop the **inetd** daemon and restart it in debug mode. Use the corresponding **System Resource Controller** command to start the **inetd** daemon with these options **inetd -a "-d"**. Write down the commands that you used:

_____

_____

__ 5. **Telnet** back to your own system, log in, and then log back out of the telnet session. This step is performed to log several debug messages. Use your login name when telnet to your system.

_____

__ 6. Stop the **inetd** daemon and restart it without debug mode. Use the corresponding **System Resource Controller** command to start the inetd daemon. Write down the commands that you used:

_____

_____

__ 7. Analyze the content of the file /**tmp**/**syslog.debug**. Many debug messages from the **inetd** daemon processes are shown.

_____

__ 8. Change your /**etc**/**syslog.conf**. All messages should be directed to the AIX error log. Write down what you've changed:

_____

__ 9. Refresh the **syslogd** subsystem. Write down the command that you used:

_____

__ 10. Generate a **syslogd** message, for example use an invalid password during a login. Check that the message is posted to the error log.

_____

_____

Error Notification with errnotify

__ 11. Create an **errnotify** object that mails a message to root, whenever an **operator message** is posted to the errlog. Write down the stanza, that you added:

_____

_____

_____

_____

_____

_____

_____

__ 12. Execute the **errlogger** command and create an entry in the errlog. Write down the command that you used:

_____

__ 13. After a short time, check the mail for the root user.

_____

END OF LAB

Exercise Instructions With Hints

Working with syslogd

__ 1. Edit the /**etc**/**syslog.conf** file and configure the **syslogd** daemon to log all daemon messages to a file with the name /**tmp**/**syslog.debug**.

Write down the line that you added to /**etc**/**syslog.conf**:

_____

Hint: The format is **facility.level action**.

__ 2. Execute the **touch** command and create the file /**tmp**/**syslog.debug**.

_____

__ 3. Refresh the **syslogd** daemon so it will pick up the changes. Write down the command that you used:

_____

Hint: refresh -s

__ 4. Stop the **inetd** daemon and restart it in debug mode. Use the corresponding **System Resource Controller** command to start the **inetd** daemon with these options **inetd -a "-d"**. Write down the commands that you used:

_____

_____

Hint: Use **stopsrc** and **startsrc**.

__ 5. **Telnet** back to your own system, log in, and then log back out of the telnet session. This step is performed to log several debug messages. Use your login name when telnet to your system.

_____

__ 6. Stop the **inetd** daemon and restart it without debug mode. Use the corresponding **System Resource Controller** command to start the inetd daemon. Write down the commands that you used:

_____

_____

__ 7. Analyze the content of the file /**tmp**/**syslog.debug**. Many debug messages from the **inetd** daemon processes are shown.

_____

__ 8. Change your /**etc**/**syslog.conf**. All messages should be directed to the AIX error log. Write down what you've changed:

_____

Hint: All messages are: ***.debug**.

__ 9. Refresh the **syslogd** subsystem. Write down the command that you used:

_____

__ 10. Generate a **syslogd** message, for example use an invalid password during a login. Check that the message is posted to the error log.

_____

_____

Error Notification with errnotify

__ 11. Create an **errnotify** object that mails a message to root, whenever an **operator message** is posted to the errlog. Write down the stanza, that you added:

_____

_____

_____

_____

_____

_____

_____

Hint: Refer to the **syslogd** configuration examples in your student notes.

__ 12. Execute the **errlogger** command and create an entry in the errlog. Write down the command that you used:

_____

__ 13. After a short time, check the mail for the root user.

_____

END OF LAB

Exercise Instructions With Solutions

Working with syslogd

__ 1. Edit the /**etc**/**syslog.conf** file and configure the **syslogd** daemon to log all daemon messages to a file with the name /**tmp**/**syslog.debug**.

Write down the line that you added to /**etc**/**syslog.conf**:

daemon.debug /tmp/syslog.debug

__ 2. Execute the **touch** command and create the file /**tmp**/**syslog.debug**.

 # touch /tmp/syslog.debug

__ 3. Refresh the **syslogd** daemon so it will pick up the changes. Write down the command that you used:

 # refresh -s syslogd

__ 4. Stop the **inetd** daemon and restart it in debug mode. Use the corresponding **System Resource Controller** command to start the **inetd** daemon with these options **inetd -a "-d"**. Write down the commands that you used:

 # stopsrc -s inetd
 # startsrc -s inetd -a "-d"

__ 5. **Telnet** back to your own system, log in, and then log back out of the telnet session. This step is performed to log several debug messages. Use your login name when telnet to your system.

 # telnet host (Use your own hostname)
 # exit

__ 6. Stop the **inetd** daemon and restart it without debug mode. Use the corresponding **System Resource Controller** command to start the inetd daemon. Write down the commands that you used:

 # stopsrc -s inetd
 # startsrc -s inetd

__ 7. Analyze the content of the file /**tmp**/**syslog.debug**. Many debug messages from the **inetd** daemon processes are shown.

 # pg /tmp/syslog.debug

__ 8. Change your /**etc**/**syslog.conf**. All messages should be directed to the AIX error log. Write down what you've changed:

***.debug errlog**

__ 9. Refresh the **syslogd** subsystem. Write down the command that you used:

 # refresh -s syslogd

__ 10. Generate a **syslogd** message, for example use an invalid password during a login. Check that the message is posted to the error log.

 # login (Use an invalid password)
After three bad attempts, return to your command prompt and check the error log.
 # errpt | more

Error Notification with errnotify

This exercise demonstrates how to automate working with the error log.

__ 11. Create an errnotify object that mails a message to root, whenever an **operator message** is posted to the errlog. Write down the stanza that you added:

```
# vi notify.add
errnotify:
en_name="sample"
en_persistenceflg=0
en_class="O"
en_method="errpt -a -l $1 | mail -s ERRLOG root"

# odmadd notify.add
```

__ 12. Execute the **errlogger** command and create an entry in the errlog. Write down the command that you used:

 # errlogger Test-entry in the log

__ 13. After a short time, check the mail for the root user.

```
# mail
? t
```

END OF LAB

Exercise 10. System Dump

What This Exercise Is About

This exercise allows the student to become familiar with the AIX dump facility. Students will execute the **kdb** command, but only on a very high level.

What You Should Be Able to Do

After completing this exercise, students should be able to:

- Initiate a dump
- Identify LED codes associated with the dump facility
- Use the **snap** command

Introduction

In this exercise you will create a dump and use the **kdb** command to look at that dump.

The instructions in this exercise require **root** authority.

Exercise Instructions

Note: All users must perform this lab together, if there is more than one user on your system.

Working with the AIX Dump Facility

__ 1. Execute the command to display the **estimated size of a dump**. Also, determine what your **primary and secondary dump devices** are and where the **copy directory** is.

Write down the commands you executed:

_____

_____

__ 2. Verify that the dump **copy directory** is large enough to store a dump.

If there is not enough space, you must increase the size of the corresponding file system.

Write down the commands you (or smit) executed:

_____

_____

__ 3. Using the smit fastpath **smit dump**, start a dump to the primary dump device.

Write down the LED codes that occur:

_____

__ 4. After the dump completes, reboot the machine in normal mode. Write down the filename and the filesize of your dump:

Filename:

Filesize:

Find out the real memory size of your system and compare the dump size with the real memory size. Analyze the rule of thumb, that a dump is about 25% of real memory.

Real memory size:

__ 5. Uncompress the dump file (/var/adm/ras/vmcore.0.Z) and then execute the **kdb** command on the uncompressed dump that was created. Write down the command you used to start the **kdb** command:

_____

__ 6. Use the **kdb** subcommands to show the system name and time of the dump and the process which was scheduled to the CPU, when the dump occurred. Leave the **kdb** command afterwards.

__ 7. Increase your /**tmp** file system so that at least **16 MB** free space are available. We need this space in the next step.

Write down the command you (or smit) executed:

_____

__ 8. Run the command **snap -a** and review the output.

This will produce a list of all the directories to where the **snap** command writes its output. The files listed are directories. In these directories you will find files that end in **.snap** which are ASCII files.

Review the content of a few.

Note that this command will take approximately 10 minutes to run.

END OF LAB

Exercise Instructions With Hints

Note: All users must perform this lab together if there is more than one user on your system.

Working with the AIX Dump Facility

__ 1. Execute the command to display the **estimated size of a dump**. Also, determine what your **primary and secondary dump devices** are and where the **copy directory** is.

Write down the commands you executed:

_____

_____

Hint: Use the **sysdumpdev** command.

__ 2. Verify that the dump **copy directory** is large enough to store a dump.

If there is not enough space, you must increase the size of the corresponding file system.

Write down the commands you (or smit) executed:

_____

_____

Hint: Use **df** and **smit chfs**.

__ 3. Using the smit fastpath **smit dump**, start a dump to the primary dump device.

Write down the LED codes that occur:

_____

__ 4. After the dump completes, reboot the machine in normal mode. Write down the filename and the filesize of your dump:

Filename:

Filesize:

Hint: The dump is located in the **copy directory**.

Find out the real memory size of your system and compare the dump size with the real memory size. Analyze the rule of thumb, that a dump is about 25% of real memory.

Real memory size:

Hint: Use the **bootinfo** command.

__ 5. Uncompress the dump file (/var/adm/ras/vmcore.0.Z) and then execute the **kdb** command on the uncompressed dump that was created. Write down the command you used to start the **kdb** command:

_____

Hint: Locate **kdb** in the index section of your student notes.

__ 6. Use the **kdb** subcommands to show the system name and time of the dump and the process which was scheduled to the CPU, when the dump occurred. Leave the **kdb** command afterwards.

Hint: Use **stat** and **status**.

__ 7. Increase your /**tmp** file system so that at least **16 MB** free space are available. We need this space in the next step.

Write down the command you (or smit) executed:

_____

Hint: Use **smit chfs**.

__ 8. Run the command **snap -a** and review the output.

This will produce a list of all the directories to where the **snap** command writes its output. The files listed are directories. In these directories you will find files that end in **.snap** which are ASCII files.

Review the content of a few.

Note that this command will take approximately 10 minutes to run.

END OF LAB

Exercise Instructions With Solutions

Note: All users must perform this lab together if there is more than one user on your system.

Working with the AIX Dump Facility

__ 1. Execute the command to display the **estimated size of a dump**. Also, determine what your **primary and secondary dump devices** are and where the **copy directory** is.

Write down the commands you executed:

 # sysdumpdev -e
 # sysdumpdev -l

__ 2. Verify that the dump **copy directory** is large enough to store a dump.

If there is not enough space, you must increase the size of the corresponding file system.

Write down the command you (or smit) executed:

 # df -k
 # chfs -a size=+x /var
where x represents the number of 512 byte blocks that /var
must be increased by to hold the dump.
64000 is a good estimate.

You could also use the command /usr/lib/ras/dumpcheck to check if the size of the copy directory is large enough. Execute the errpt command to see if an error message was generated. If there is no message, then the size is sufficient.

__ 3. Using the smit fastpath **smit dump**, start a dump to the primary dump device.

 # smit dump

- Select: Start a Dump to the Primary Dump Device

Write down the LED codes that occur:

0c2. After approx. 1 minute 0c0 is shown.

__ 4. After the dump completes, reboot the machine in normal mode. Write down the filename and the filesize of your dump:

 # sysdumpdev -L

Filename: /var/adm/ras/vmcore.0 (if it's the first dump)

Filesize: 9507840 (for example)

Find out the real memory size of your system and compare the dump size with the real memory size. Analyze the rule of thumb, that a dump is about 25% of real memory.

Real memory size:

 # bootinfo -r
131072 (for example 128 MB)

Your dump should be a bit smaller than explained by the rule of thumb.

__ 5. Uncompress the dump file (/var/adm/ras/vmcore.0.Z) and then execute the **kdb** command on the uncompressed dump that was created. Write down the command you used to start the **kdb** command:

 # kdb /var/adm/ras/vmcore.0 (if it's the first dump)

__ 6. Use the **kdb** subcommands to show the system name and time of the dump and the process which was scheduled to the CPU, when the dump occurred. Leave the **kdb** command afterwards.

```
> stat        (Example entries)
  ...
  sysname: AIX
  nodename: localhost
  release: 1
  version: 5
  machine: 000400B24C00
  time of crash: Tue Jun 5 20:41:56 2001
  age of system: 1 hr., 15 min., 57 sec.
  xmalloc debug: disabled

 > status      (Example entries)

  CPU    TID    TSLOT     PID    PSLOT   PROCNAME
    0    5CCB       92    409E       64    sysdumpstart
 > q
```

__ 7. Increase your /**tmp** file system so that at least **16 MB** free space are available. We need this space in the next step.

Write down the command you (or smit) executed:

 # chfs -a size=+32000 /tmp (512 byte blocks!)

__ 8. Run the command **snap -a** and review the output.

 # snap -a

This will produce a list of all the directories to where the **snap** command writes its output. The files listed are directories. In these directories you will find files that end in **.snap** which are ASCII files.

Review the content of a few.

Note that this command will take approximately 10 minutes to run.

 # cd /tmp/ibmsupt

END OF LAB

Exercise 11. Basic Performance Commands

What This Exercise Is About

The purpose of this exercise is to provide basic performance commands.

What You Should Be Able to Do

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

- Use **ps** to identify CPU and memory-intensive programs
- Execute a basic performance analysis
- Implement a korn shell job queue
- Work with **nice** and **renice** to change the priorities of processes

Introduction

All instructions in this exercise should be executed with **root** authority.

Exercise Instructions

Working with ps, nice, and renice

__ 1. Implement an alias **top** that shows a sorted output from **ps aux** according to the CPU usage. Write down the alias definition.

_____

__ 2. Execute **top** and identify the process that consumes the most CPU.

_____

_____

__ 3. Start the program /**home**/**workshop**/**ex11_prog1** in background. Execute **ps -elf** and identify the assigned **priority** and **nice value**.

Priority:

Nice value:

__ 4. Stop **ex11_prog1** and restart it in background with a very low priority. Write down the command that you used.

_____

Again write down the **nice value** and **priority** that have been assigned to the process:

Priority:

Nice Value:

__ 5. Without restarting **ex11_prog1**, increase the priority of the process. Write down the command you used.

_____

Check that the priority has been increased.

_____

__ 6. Stop the program **ex11_prog1**.

_____

Basic Performance Analysis

__ 7. Start the program /**home**/**workshop**/**ex11_cpu** in background. Execute the **sar** command to analyze CPU usage on your system. Set it up to collect the data at **two second intervals for five times**. Write down the command that you used to monitor CPU usage.

_____

From the output, what can you conclude?

_____

_____

_____

__ 8. Use **ps -elf** and check the priority that has been assigned to the process. Is the priority high or low?

_____

_____

__ 9. Stop the program **ex11_cpu**.

_____

__ 10. Start the program /**home**/**workshop**/**ex11_disk** in background. Execute the **iostat** command to analyze your disk I/O. Look at **iostat** disk information for **two second intervals five times**. Write down the command that you used to monitor disk I/O.

_____

From the output, what can you conclude?

_____

_____

__ 11. Stop the program **ex11_disk**.

_____

__ 12. Start the memory intensive process /**home**/**workshop**/**ex11_memory** in the background. Execute the **vmstat** command to analyze your memory utilization. Run **vmstat** at five second intervals. Write down the command that you used to measure memory.

Working with a Korn Shell Job Queue

__ 13. Create a **korn shell job queue** as shown in your student notes. Write down the definitions for the queue and the queue device:

__ 14. Bring down the queue. Write the command you used.

_____

__ 15. Put the job /**home**/**workshop**/**ex11_job** into the **ksh** queue. Write down the command you used.

_____

__ 16. Verify that the job is queued. Write down the command you used.

_____

__ 17. Bring up the queue. Write down the command you used.

_____

What happens?

_____

_____

END OF LAB

Exercise Instructions With Hints

Working with ps, nice, and renice

__ 1. Implement an alias **top** that shows a sorted output from **ps aux** according to the CPU usage. Write down the alias definition.

_____

Hint: alias top="ps aux |…" (Check your student notes) If you want add the definition to your korn shell environment file.

__ 2. Execute **top** and identify the process that consumes the most CPU.

_____

_____

__ 3. Start the program /**home**/**workshop**/**ex11_prog1** in background. Execute **ps -elf** and identify the assigned **priority** and **nice value**.

Priority:

Nice value:

Hint: Check the PRI and NI columns

__ 4. Stop **ex11_prog1** and restart it in background with a very low priority. Write down the command that you used.

_____

Hint: Use **nice -n**.

Again write down the **nice value** and **priority** that have been assigned to the process:

Priority:

Nice Value:

__ 5. Without restarting **ex11_prog1**, increase the priority of the process. Write down the command you used.

_____

Hint: Use **renice -n**.

Check that the priority has been increased.

_____

__ 6. Stop the program **ex11_prog1**.

_____

Basic Performance Analysis

__ 7. Start the program /**home**/**workshop**/**ex11_cpu** in background. Execute the **sar** command to analyze CPU usage on your system. Set it up to collect the data at **two second intervals for five times**. Write down the command that you used to monitor CPU usage.

_____

Hint: Use **sar -u**.

From the output, what can you conclude?

_____

_____

_____

__ 8. Use **ps -elf** and check the priority that has been assigned to the process. Is the priority high or low?

_____

_____

__ 9. Stop the program **ex11_cpu**.

_____

__ 10. Start the program /**home**/**workshop**/**ex11_disk** in background. Execute the **iostat** command to analyze your disk I/O. Look at **iostat** disk information for **two second intervals five times**. Write down the command that you used to monitor disk I/O.

_____

Hint: Use **iostat <interval> <number>**.

From the output, what can you conclude?

_____

_____

__ 11. Stop the program **ex11_disk**.

_____

__ 12. Start the memory intensive process /**home**/**workshop**/**ex11_memory** in the background. Execute the **vmstat** command to analyze your memory utilization. Run **vmstat** at five second intervals. Write down the command that you used to measure memory.

Hint: Use **vmstat**

Working with a Korn Shell Job Queue

__ 13. Create a **korn shell job queue** as shown in your student notes. Write down the definitions for the queue and the queue device:

__ 14. Bring down the queue. Write the command you used.

_____

Hint: Use **qadm -D** or **disable**.

__ 15. Put the job /**home**/**workshop**/**ex11_job** into the **ksh** queue. Write down the command you used.

_____

Hint: Use **qprt -P** or **enable**.

__ 16. Verify that the job is queued. Write down the command you used.

_____

Hint: Use **lpstat**.

__ 17. Bring up the queue. Write down the command you used.

_____

What happens?

_____

_____

Use **qadm -U**.

END OF LAB

Exercise Instructions With Solutions

Working with ps, nice, and renice

__ 1. Implement an alias **top** that shows a sorted output from **ps aux** according to the CPU usage. Write down the alias definition.

 # alias top="ps aux | tail +2 | sort -k 1.15,1.19nr"

Check how ENV is defined. If ENV=$HOME/.kshrc, add the alias to the file $HOME/.kshrc.

__ 2. Execute **top** and identify the process that consumes the most CPU.

 # top

Probably the wait process (PID 516) consumes the most CPU.

__ 3. Start the program /**home/workshop/ex11_prog1** in background. Execute **ps -elf** and identify the assigned **priority** and **nice value**.

 # /home/workshop/ex11_prog1 &
 # ps -elf

Priority:

Nice value:

__ 4. Stop **ex11_prog1** and restart it in background with a very low priority. Write down the command that you used.

 # kill %1

 # nice -n 15 /home/workshop/ex11_prog1 &

 # ps -elf

Again write down the **nice value** and **priority** that have been assigned to the process:

Priority:

Nice Value:

__ 5. Without restarting **ex11_prog1**, increase the priority of the process. Write down the command you used.

 # renice -n -10 3688 (use corresponding PID)

Check that the priority has been increased.

 # ps -elf

__ 6. Stop the program **ex11_prog1**.

 # kill %1

Basic Performance Analysis

__ 7. Start the program /**home/workshop/ex11_cpu** in background. Execute the **sar** command to analyze CPU usage on your system. Set it up to collect the data at **two second intervals for five times**. Write down the command that you used to monitor CPU usage.

 # /home/workshop/ex11_cpu &

 # sar -u 2 5

From the output, what can you conclude?

After starting ex11_cpu, the CPU is active the whole time.
The program is very CPU-intensive, and causes the system to be CPU bound.

__ 8. Use **ps -elf** and check the priority that has been assigned to the process. Is the priority high or low?

 # ps -elf

The priority is very low. As the process consumes a lot of CPU time, the system protects itself by assigning a low priority to the process.

__ 9. Stop the program **ex11_cpu**.

 # kill %1

__ 10. Start the program /**home/workshop/ex11_disk** in background. Execute the **iostat** command to analyze your disk I/O. Look at **iostat** disk information for **two second intervals five times**. Write down the command that you used to monitor disk I/O.

 # /home/workshop/ex11_disk &

 # iostat 2 5

From the output, what can you conclude?

After starting ex11_disk, the disk activity is relatively high.
Depending on the system you work on, the CPU has to wait for outstanding

I/Os.
In this case your system is I/O bound.

__ 11. Stop the program **ex11_disk**.

 # kill %1

__ 12. Start the memory intensive process /**home**/**workshop**/**ex11_memory** in the background. Execute the **vmstat** command to analyze your memory utilization. Run **vmstat** at five second intervals. Write down the command that you used to measure memory.

 # /home/workshop/ex11_memory &

 # vmstat 5

From the output, what can you conclude?

After starting **ex11_memory** the system begins paging.
If paging takes place the system approaches it's limits, because the real memory is not sufficient.

If you don't see any increase in the **pi** and **po** values (they stay at zero), use the **rmss -c 32** command to simulate a smaller memory size. Once you are finished, be sure to set the memory back to its original size with the **rmss -r** command.

Working with a Korn Shell Job Queue

__ 13. Create a **korn shell job queue** as shown in your student notes. Write down the definitions for the queue and the queue device:

```
# vi /etc/qconfig

ksh:
device = kshdev
discipline = fcfs

kshdev:
backend = /usr/bin/ksh
```

__ 14. Bring down the queue. Write the command you used.

 # qadm -D ksh or # disable ksh

__ 15. Put the job /**home**/**workshop**/**ex11_job** into the **ksh** queue. Write down the command you used.

 # qprt -P ksh /home/workshop/ex11_job

__ 16. Verify that the job is queued. Write down the command you used.

 # lpstat

__ 17. Bring up the queue. Write down the command you used.

 # qadm -U ksh or #enable ksh

What happens?

The program /home/workshop/ex11_job will be executed.

END OF LAB

Exercise 12. PDT

What This Exercise Is About

The purpose of this exercise is to give students an opportunity to use the system monitor facility **PDT**.

What You Should Be Able to Do

After completing this exercise, students should be able to use the Performance Diagnostic Tool (PDT) for ongoing data capture and analysis of critical system resources.

Introduction

This exercise deals with the Performance Diagnostic Tool (PDT) for on-going data capture and analysis of system resources.

Exercise Instructions

Performance Diagnostic Tool

__ 1. AIX Version 5.1 comes with a program that includes procedures that can be used to monitor system activity and produce reports on a regular basis. The name of the program is the Performance Diagnostic Tool (PDT). Verify that PDT is loaded on your exercise system. Start PDT to enable default data collection and reporting.

What commands did you use?

_____

_____

__ 2. The **adm** user is needed to run this procedure. **su** to the **adm** user and change the crontab entry so PDT will collect data **within 10 minutes from now** and run the reports **5 minutes** later. Make sure you check the system **date** first.

What commands did you use?

_____

_____

_____

_____

Important: Move on to step 3. Step 3 needs to run while PDT is collecting information.

__ 3. Run the script **ex12_perf** located in /**home**/**workshop**. **You need to run this as root**. This script will create some items that should be reported when PDT runs in 10 minutes.

_____

__ 4. After the time frame is over in which the report should have been created (based on your entries in the **crontab** file), view the report.

_____

_____

__ 5. To change the severity level to severity level 2 and the user to whom the report is mailed, execute the **pdt_config** program. Once you are finished making the changes, exit the program.

Which **pdt_config** menu items did you select?

_____

_____

__ 6. Run PDT again, this time from the command line. Do it twice. Once to see a severity level 2 report. The other time to do a severity level 3 report.

What command did you use for severity level 2?

_____

What command did you use for severity level 3?

_____

END OF LAB

Exercise Instructions With Hints

Performance Diagnostic Tool

__ 1. AIX Version 5.1 comes with a program that includes procedures that can be used to monitor system activity and produce reports on a regular basis. The name of the program is the Performance Diagnostic Tool (PDT). Verify that PDT is loaded on your exercise system. Start PDT to enable default data collection and reporting.

What commands did you use?

_____

_____

Hint: Use **lslpp, pdt_config**, and then **select option 4 and then option 7**

__ 2. The **adm** user is needed to run this procedure. **su** to the **adm** user and change the crontab entry so PDT will collect data within **10 minutes from now** and run the reports **5 minutes** later. Make sure you check the system **date** first.

What command did you use?

_____

_____

_____

Hint: Use **crontab -e**, and then the time entries as **min hour day-of-month month day-of-week**

Important: Move on to step 3. Step 3 needs to run while PDT is collecting information.

__ 3. Run the script **ex12_perf** located in /**home**/**workshop**. **You need to run this as root**. This script will create some items that should be reported when PDT runs in 10 minutes.

_____

__ 4. After the time frame is over in which the report should have been created (based on your entries in the **crontab** file), view the report.

What command did you use to view the report?

_____

Hint: The report is located in /**var**/**perf**/**tmp**/**PDT_REPORT**

__ 5. To change the severity level to severity level 2 and the user to whom the report is mailed, execute the **pdt_config** program. Once you are finished making the changes, exit the program.

Which **pdt_config** menu items did you select?

_____

_____

Hint: Select menu items for **modify**/**enable** and then **exit**

__ 6. Run PDT again, this time from the command line. Do it twice. Once to see a severity level 2 report. The other time to do a severity level 3 report.

What command did you use for severity level 2?

_____

What command did you use for severity level 3?

_____

Hint: Use the command **pdt_report** followed by the severity level.

END OF LAB

Exercise Instructions With Solutions

Performance Diagnostic Tool

__ 1. AIX Version 5.1 comes with a program that includes procedures that can be used to monitor system activity and produce reports on a regular basis. The name of the program is the Performance Diagnostic Tool (PDT). Verify that PDT is loaded on your exercise system. Start PDT to enable default data collection and reporting.

 # lslpp -L bos.perf.diag_tool
 # /usr/sbin/perf/diag_tool/pdt_config

Select Item number 4, modify/enable PDT collection and then item number 7, exit pdt_config.

__ 2. The **adm** user is needed to run this procedure. **su** to the **adm** user and change the crontab entry so PDT will collect data within **10 minutes from now** and run the reports **five minutes** later. Make sure you check the system **date** first.

 # su adm
$ date
$ crontab -e (If you have problems here, check your EDITOR variable)

5 3 * * * /usr/sbin/perf/diag_tool/Driver_ daily
10 3 * * * /usr/sbin/perf/diag_tool/Driver_ daily2

(This is an example to be tailored for the current date and time on your system.)

Change the entries for /usr/sbin/perf/diag_tool/Driver_ daily and /usr/sbin/perf/diag_tool/Driver_ daily2 so that the collector step will run 10 minutes from now and reporters step will run 15 minutes from now

Move on to step 3. Step 3 needs to run while PDT is collecting information.

__ 3. Run the script **ex12_perf** located in **/home/workshop**. **You need to run this as root**. This script will create some items that should be reported when PDT runs in 10 minutes.

 # /home/workshop/ex12_perf

__ 4. After the time frame is over in which the report should have been created (based on your entries in the **crontab** file), view the report.

 # pg /var/perf/tmp/PDT_REPORT

__ 5. To change the severity level to severity level 2 and the user to whom the report is mailed, execute the **pdt_config** program. Once you are finished making the changes, exit the program.

 # /usr/sbin/perf/diag_tool/pdt_config

**Select item 2, modify/enable PDT reporting, from the menu.
When prompted for recipient, enter new user to send reports and
when prompted for severity level, enter 2.**

Select item 7, exit pdt_config.

__ 6. Run PDT again, this time from the command line. Do it twice. Once to see a severity level 2 report. The other time to do a severity level 3 report.

 # /usr/sbin/perf/diag_tool/pdt_report 2 | more

 # /usr/sbin/perf/diag_tool/pdt_report 3 | more

END OF LAB

Exercise 13. Authentication and Access Control Lists

What This Exercise Is About

This exercise will familiarize you with three security features: the **login.cfg** file, authentication methods, and access control lists.

What You Should Be Able to Do

After completing this exercise students should be able to:

- Customize the **login.cfg** file
- Add an additional primary **authentication method** for a user
- Implement **access control lists (ACLs)**

Introduction

This exercise consists of three parts:

1. Customizing the **login.cfg** file
2. Adding a **primary authentication method**
3. **Access Control Lists**

Required Materials

- Program /**home**/**workshop**/**ex14_login**

Exercise Instructions

Setting a New Login Herald

__ 1. Log in as **root** and edit /**etc**/**security**/**login.cfg.** Change the herald message to read:

```
*Restricted Access*
Authorized Users Only
Login:
```

__ 2. Log out. You must be at the command line login to see your changes. If you are using the CDE graphical login, click the **options** button and select **Command line login**.

Does it look correct? If not, try step one again.

__ 3. Log in as **root**.

__ 4. Review the fail login attempts made on your machine.

__ 5. Review the **su** activity on your machine.

__ 6. Review all the logins on your system.

__ 7. Review all **root** logins on your system.

Adding A Primary Authentication Method

In /**home**/**workshop** you find a procedure with the name **ex14_login**, which implements an additional primary authentication method. This method restricts a user to **one login session** on a system.

__ 8. With **root** authority, change to /**home**/**workshop** and analyze the procedure **ex14_login**. Which statement indicates a valid or invalid login?

_____

_____

Check that **ex14_login** is executable.

__ 9. Install the procedure **ex14_login** as additional authentication method on your system. Write down the stanza definition you've created in /**usr**/**lib**/**security**/**methods.cfg**:

_____

_____

__ 10. Install the additional authentication method for user **team01** in /**etc**/**security**/**user**. Write down the stanza definition for **team01**:

_____

_____

__ 11. Working in a graphical environment, open two windows and execute the **login** command in both of them. Login as **team01**. The second login should fail.

__ 12. Remove the additional authentication method from **team01**.

_____

Access Control Lists

__ 13. Log in as **team01** and switch to **root**. Create two new users named **michael** and **sarah**. Assign a password the same as the login names.

_____

_____

_____

_____

__ 14. Return to your **team01** user ID.

__ 15. Create a shell script in your home directory named **sample** with the following content:

tput clear
banner We love AIX
print End of Program

Set the base permissions for sample to **700** and verify that the script works.

_____

_____

__ 16. Log out as **team01** and log in as **michael**. Change directory to /**home**/**team01**.

__ 17. Try to display the **sample** script, using the **cat** command. Try to execute **sample**. You should not be able to do either. Log out as **michael**.

_____

_____

__ 18. Log in as **team01**. Set and export the **EDITOR** variable to /**usr**/**bin**/**vi** in your **.profile**.

Log out and log in again as **team01**.

_____

_____

__ 19. Use the **acledit** command to change the extended permissions of the **sample** script so that **michael** can **rwx** the script, and **sarah** can only **r-x** the script. Apply the modified ACL.

_____

_____

_____

_____

__ 20. Execute **ls -e** and check that **extended permissions** are set for **sample**. Log out as **team01** afterwards.

_____

_____

__ 21. Log in as **michael**. Change to the /**home**/**team01** directory and test the extended permissions by trying to add the **date** command to the end of the script. Execute the **sample** script afterwards.

_____

_____

__ 22. Log out as **michael** and log in as **sarah**. Change to /**home**/**team01** and try to change the **sample** script by removing the **date** command. Does it work?

_____

__ 23. Log out as **sarah** and log in as **team01**. Change the **base** and **extended** permissions to the **sample** script so that members of the **staff** group can **read** and **execute** the script, **except** for **michael**.

_____

_____

_____

_____

__ 24. Log out as **team01** and log in as **michael**. Issue the **groups** command to ensure you are part of the **staff** group. Change directory to /**home**/**team01**. Can you execute the **sample** script?

_____

_____

__ 25. Log out as **michael** and log in as **team01**. Create a new file named **sample2**. Type a couple of lines in the file. Use **aclget** to see that no extended permissions are set on **sample2**.

_____

__ 26. Using **aclget** and **aclput**, copy the access control information of the file **sample** to the new file **sample2**. Verify that the ACLs were copied over to **sample2**.

_____

_____

__ 27. Execute a **chmod 700** on **sample**. Execute **acledit** on **sample**. What is different?

_____

_____

END OF LAB

Exercise Instructions With Hints

Setting a New Login Herald

__ 1. Log in as **root** and edit /**etc**/**security**/**login.cfg.** Change the herald message to read:

```
*Restricted Access*
Authorized Users Only
Login:
```

Hint: Modify the /etc/security/login.cfg file. Do not use the <ENTER> key in your herald string. You must use /n for new lines and /r for return. When you are editing the line, if you reach the end of the line, let it wrap to the next line. Again, do not use the <ENTER> key.

__ 2. Log out. You must be at the command line login to see your changes. If you are using the CDE graphical login, click the **options** button and select **Command line login**.

Does it look correct? If not, try step one again.

__ 3. Log in as **root**.

__ 4. Review the fail login attempts made on your machine.

Hint: /etc/security/failedlogin file

__ 5. Review the **su** activity on your machine.

Hint: /var/adm/sulog file

__ 6. Review all the logins on your system.

Hint: /var/adm/wtmp file

__ 7. Review all **root** logins on your system.

Adding A Primary Authentication Method

In /**home**/**workshop** you find a procedure with the name **ex14_login**, which implements an additional primary authentication method. This method restricts a user to **one login session** on a system.

__ 8. With **root** authority, change to /**home**/**workshop** and analyze the procedure **ex14_login**. Which statement indicates a valid or invalid login?

_____

_____

The statement returns 0 for success, otherwise a non-zero value.

Hint: Check that **ex14_login** is executable.

__ 9. Install the procedure **ex14_login** as additional authentication method on your system. Write down the stanza definition you've created in **/usr/lib/security/methods.cfg**:

_____

_____

Hint: See **Customized Authentication** in your student materials.

__ 10. Install the additional authentication method for user **team01** in **/etc/security/user**. Write down the stanza definition for **team01**:

_____

_____

Hint: See **Customized Authentication** in your student materials.

__ 11. Working in a graphical environment, open two windows and execute the **login** command in both of them. Login as **team01**. The second login should fail.

__ 12. Remove the additional authentication method from **team01**.

_____

Access Control Lists

__ 13. Log in as **team01** and switch to **root**. Create two new users named **michael** and **sarah**. Assign a password the same as the login names.

_____

_____

_____

_____

Hint: Use the following commands: **mkuser**, **passwd**

__ 14. Return to your **team01** user ID.

__ 15. Create a shell script in your home directory named **sample** with the following content:

tput clear
banner We love AIX
print End of Program

Set the base permissions for sample to **700** and verify that the script works.

_____

_____

Hint: Use **chmod 700**.

__ 16. Log out as **team01** and log in as **michael**. Change directory to /**home**/**team01**.

__ 17. Try to display the **sample** script, using the **cat** command. Try to execute **sample**. You should not be able to do either. Logout as **michael**.

_____

_____

__ 18. Log in as **team01**. Set and export the **EDITOR** variable to /**usr**/**bin**/**vi** in your **.profile**.

Log out and log in again as **team01**.

_____

_____

__ 19. Use the **acledit** command to change the extended permissions of the **sample** script so that **michael** can **rwx** the script, and **sarah** can only **r-x** the script. Apply the modified ACL.

_____

_____

_____

_____

Hint: See **ACL Keywords: permit and specify** in your student materials.

__ 20. Execute **ls -e** and check that **extended permissions** are set for **sample**. Log out as **team01** afterwards.

_____

_____

Hint: Do you see the **+** - sign?

__ 21. Log in as **michael**. Change to the /**home**/**team01** directory and test the extended permissions by trying to add the **date** command to the end of the script. Execute the **sample** script afterwards.

_____

_____

__ 22. Log out as **michael** and log in as **sarah**. Change to /**home**/**team01** and try to change the **sample** script by removing the **date** command. Does it work?

_____

__ 23. Log out as **sarah** and log in as **team01**. Change the **base** and **extended** permissions to the **sample** script so that members of the **staff** group can **read** and **execute** the script, **except** for **michael**.

_____

_____

_____

_____

Hint: See **ACL Keywords: deny** in your student materials.

__ 24. Log out as **team01** and log in as **michael**. Issue the **groups** command to ensure you are part of the **staff** group. Change directory to /**home**/**team01**. Can you execute the **sample** script?

_____

_____

__ 25. Log out as **michael** and log in as **team01**. Create a new file named **sample2**. Type a couple of lines in the file. Use **aclget** to see that no extended permissions are set on **sample2**.

_____

Hint: See **ACL Commands** in your student materials.

__ 26. Using **aclget** and **aclput**, copy the access control information of the file **sample** to the new file **sample2**. Verify that the ACLs were copied over to **sample2**.

_____

_____

Hint: See **ACL Commands** in your student materials.

__ 27. Execute a **chmod 700** on **sample**. Execute **acledit** on **sample**. What is different?

_____

_____

END OF LAB

Exercise Instructions With Solutions

Setting a New Login Herald

__ 1. Log in as **root** and edit /**etc**/**security**/**login.cfg.** Change the herald message to read:

```
* Restricted Access *
Authorized Users Only
Login:
```

 # vi /etc/security/login.cfg

In the default stanza add the herald information.

default:

herald = "\n\n\n\n* Restricted Access *\n\rAuthorized Users Only\n\rLogin: "

Note: Do not use the <ENTER> key in your herald string. You must use \n for new lines and \r for return. When you are editing the line, if you reach the end of the line, let it wrap to the next line. Again, do not use the <ENTER> key.

__ 2. Log out. You must be at the command line login to see your changes. If you are using the CDE graphical login, click the **options** button and select **Command line login**.

Does it look correct? If not, try step one again.

__ 3. Log in as **root**.

__ 4. Review the fail login attempts made on your machine.

 # who /etc/security/failedlogin | more

__ 5. Review the **su** activity on your machine.

 # more /var/adm/sulog

__ 6. Review all the logins on your system.

 # last | more or **# who /var/adm/wtmp | more**

__ 7. Review all root logins on your system.

 # last root

Adding A Primary Authentication Method

In /**home**/**workshop** you'll find a procedure with the name **ex14_login**, which implements an additional primary authentication method. This method restricts a user to **one login session** on a system.

__ 8. With **root** authority, change to /**home**/**workshop** and view the file **ex14_login**. Which statement indicates a valid or invalid login?

exit 0 means valid login
exit 1 means invalid login

Check that ex14_login is executable.

__ 9. Install the procedure **ex14_login** as additional authentication method on your system. Write down the stanza definition you've created in **/usr/lib/security/methods.cfg**:

```
COUNT:
program = /home/workshop/ex14_login

AIX 5.3 - you must add the stanza to /etc/security/login.cfg.
```

__ 10. Install the additional authentication method for user **team01** in **/etc/security/user**. Write down the stanza definition for **team01**:

```
team01:
auth1 = SYSTEM,COUNT
```

__ 11. Working in a graphical environment, open two windows and execute the **login** command in both of them. Login as **team01**. The second login should fail.

__ 12. Remove the additional authentication method from **team01**.

 # vi /etc/security/user

Remove the attribute auth1 in the **team01** stanza.

Access Control Lists

__ 13. Log in as **team01** and switch to **root**. Create two new users named **michael** and **sarah**. Assign a password the same as the login names.

$ su
 # mkuser michael
 # mkuser sarah
 # passwd michael
 # passwd sarah

__ 14. Return to your **team01** user ID.

__ 15. Create a shell script in your home directory named **sample** with the following content:

tput clear
banner We love AIX
print End of Program

Set the base permissions for **sample** to **700** and verify that the script works.

$ vi sample
(insert above statements)

$ chmod 700 sample
$ sample

__ 16. Log out as **team01** and log in as **michael**. Change directory to /**home**/**team01**.

__ 17. Try to display the **sample** script, using the **cat** command. Try to execute **sample**. You should not be able to do either. Log out as **michael**.

$ cd /home/team01
$ cat sample
cat: cannot open sample
$ sample
ksh: sample: Execute permission denied
$ exit

__ 18. Log in as **team01**. Set and export the **EDITOR** variable to /**usr**/**bin**/**vi** in your **.profile**.

Execute your .profile to pick up the changes.

...
$ vi .profile

export EDITOR=/usr/bin/vi

$ . .profile

__ 19. Use the **acledit** command to change the extended permissions of the **sample** script so that **michael** can **rwx** the script, and **sarah** can only **r-x** the script. Apply the modified ACL.

$ acledit sample

Replace the word disabled with enabled
for extended permissions.

Add the following two lines to the file, under the word enabled

permit rwx u:michael
permit r-x u:sarah

__ 20. Execute **ls -e** and check that **extended permissions** are set for **sample**. Logout as **team01** afterwards.

$ ls -e sample
$ exit

__ 21. Log in as **michael**. Change to the /**home**/**team01** directory and test the extended permissions by trying to add the **date** command to the end of the script. Execute the **sample** script afterwards.

$ cd /home/team01
$ vi sample
...
date
$ sample

__ 22. Log out as **michael** and log in as **sarah**. Change to /**home**/**team01** and try to change the **sample** script by removing the **date** command. Does it work?

No. The user sarah has read and execute, but no write permission.

__ 23. Log out as **sarah** and log in as **team01**. Change the **base** and **extended** permissions to the **sample** script so that members of the **staff** group can **read** and **execute** the script, **except** for **michael**.

$ acledit sample
Change the Base Permissions for group (staff) from --- to r-x

Delete the two permit lines under extended permissions.
Add the following line:
deny rwx u:michael

__ 24. Log out as **team01** and login as **michael**. Issue the **groups** command to ensure you are part of the **staff** group. Change directory to /**home**/**team01**. Can you execute the **sample** script?

$ groups
staff
$ cd /home/team01
$ sample
ksh: sample: Execute permission denied.
==> The keyword deny denies the access to the file.

__ 25. Log out as **michael** and login as **team01**. Create a new file named **sample2**. Type a couple of lines in the file. Use **aclget** to see that no extended permissions are set on **sample2**.

$ vi sample2
(type in a couple of lines)
$ aclget sample2

__ 26. Using **aclget** and **aclput**, copy the access control information of the file **sample** to the new file **sample2**. Verify that the ACLs were copied over to **sample2**.

$ aclget sample | aclput sample2
$ aclget sample2

__ 27. Execute a **chmod 700** on **sample**. Execute **aclget** on **sample**. What is different?

$ chmod 700 sample
$ aclget sample
The extended permissions have been disabled.

END OF LAB

Appendix A. Auditing

What This Exercise Is About

This exercise is an introduction to the use of the AIX auditing subsystem to trace and record security-relevant information.

What You Should Be Able to Do

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

- Audit **objects** and **application events**
- Create **audit classes**
- Audit **users**
- Set up auditing in **bin** and **stream** mode

Introduction

All instructions in this exercise require **root** authority.

Required Materials

- **/home/workshop/ex13_job**

Exercise Instructions

Bin Mode Auditing

__ 1. Answer the following question first: Where do you specify file system objects that should be audited?

_____

__ 2. Set up auditing of the program /**usr**/**bin**/**passwd**. When you're finished, whenever a user calls **passwd** you should get an audit record. Add this object to the corresponding audit configuration file.

_____

_____

_____

Write down the event name you've created:

_____

__ 3. In which file do you have to specify the format definitions for your new event?

_____

__ 4. Add the format definition for your new event to the corresponding audit configuration file.

_____

_____

__ 5. In which file do you specify the **start mode** for the auditing subsystem?

_____

__ 6. Create a directory /**var**/**myaudit**. We want to use this directory to collect all audit-related files.

__ 7. Change the corresponding configuration file to startup the auditing subsystem in **bin mode**. Specify the following bin files:

bin1 = /var/myaudit/bin1
bin2 = /var/myaudit/bin2
trail = /var/myaudit/trail

_____

_____

__ 8. Start the auditing subsystem. Write down the command you used.

_____

__ 9. Log in as **team01**. Use the password **team01.** When prompted to change password, set it back to **team01**. If you use a graphical environment, execute the **login** command in a separate window.

__ 10. Execute the **passwd** command and change the password for **team01**.

__ 11. With **root** authority, stop the auditing subsystem. Write down the command you used.

_____

__ 12. Change to /**var**/**myaudit** and display the audit records that have been recorded. Write down the command you used.

_____

_____

Stream Mode Auditing

__ 13. In which file do you configure **audit classes** and **audit users**?

_____

__ 14. Change this configuration file in the following way:

- The auditing subsystem starts up in **stream mode**.
- Create an audit class **kill**, that contains an audit event whenever a process gets killed.
- Remove the **root** user in the **users** stanza.
- The user **team01** should be audited for the audit classes **kill** and **tcpip**.

_____

_____

_____

_____

__ 15. In which file do you configure the **auditstream** daemon?

_____

__ 16. Before starting the auditing subsystem in stream mode, change the configuration file for the **auditstream** daemon. All audit records shall be written to file /**var**/**myaudit**/**stream.out**. Be sure to terminate the command with a &-sign.

_____

_____

__ 17. Start your auditing system.

_____

__ 18. Use the **touch** command and create an empty file /**var**/**myaudit**/**stream.out**. Use the tail command in a separate window to display the audit records real-time.

_____

_____

__ 19. Log in as **team01** and trigger the events that you are auditing for this user:

- Execute the **ftp** command. Use your local host as destination host. You should see the corresponding audit records in file /**var**/**myaudit**/**stream.out**.
- Start the program /**home**/**workshop**/**ex13_job** in **background**. Kill the started program afterwards. You should see audit records for the **kill** audit class you've created.

_____

_____

__ 20. Stop the auditing subsystem.

_____

END OF LAB

Exercise Instructions With Hints

Bin Mode Auditing

__ 1. Answer the following question first: Where do you specify file system objects that should be audited?

_____

Hint: /etc/security/audit/o...

__ 2. Set up auditing of the program /**usr**/**bin**/**passwd**. When you're finished, whenever a user calls **passwd** you should get an audit record. Add this object to the corresponding audit configuration file.

_____

_____

_____

Hint: Add an **x-event** for the program /**usr**/**bin**/**passwd**.

Write down the event name you've created:

_____

__ 3. In which file do you have to specify the format definitions for your new event?

_____

Hint: /etc/security/audit/e...

__ 4. Add the format definition for your new event to the corresponding audit configuration file.

_____

_____

Hint: Specify the event name and add a printf definition.

__ 5. In which file do you specify the **start mode** for the auditing subsystem?

_____

Hint: /etc/security/audit/c...

__ 6. Create a directory /**var**/**myaudit**. We want to use this directory to collect all audit-related files.

__ 7. Change the corresponding configuration file to startup the auditing subsystem in **bin mode**. Specify the following bin files:

bin1 = /var/myaudit/bin1
bin2 = /var/myaudit/bin2
trail = /var/myaudit/trail

_____

_____

Hint: You must change the **start** and **bin** stanzas.

__ 8. Start the auditing subsystem. Write down the command you used.

_____

Hint: Execute **audit**...

__ 9. Log in as **team01**. Use password **team01**. When prompted to change password, set it back to **team01**. If you use a graphical environment, execute the **login** command in a separate window.

__ 10. Execute the **passwd** command and change the password for **team01**.

__ 11. With **root** authority, stop the auditing subsystem. Write down the command you used.

_____

Hint: Execute **audit**...

__ 12. Change to /**var**/**myaudit** and display the audit records that have been recorded. Write down the command you used.

_____

_____

Use the **auditpr** command and query the **trail**.

Stream Mode Auditing

__ 13. In which file do you configure **audit classes** and **audit users**?

_____

Hint: /etc/security/audit/c...

__ 14. Change this configuration file in the following way:

- The auditing subsystem starts up in **stream mode**.
- Create an audit class **kill**, that contains an audit event whenever a process gets killed.
- Remove the **root** user in the **users** stanza.
- The user **team01** should be audited for the audit classes **kill** and **tcpip**.

Hint: You must change the **start**, **classes** and **users** stanzas.

_____

_____

_____

_____

__ 15. In which file do you configure the **auditstream** daemon?

_____

Hint: /etc/security/audit/str...

__ 16. Before starting the auditing subsystem in stream mode, change the configuration file for the **auditstream** daemon. All audit records shall be written to file **/var/myaudit/stream.out**. Be sure to terminate the command with a &-sign.

_____

_____

Hint: Change the **auditpr** command.

__ 17. Start your auditing system.

_____

__ 18. Use the **touch** command and create an empty file **/var/myaudit/stream.out**. Use the tail command in a separate window to display the audit records real-time.

Hint: # tail -f /var/myaudit/stream.out

_____

_____

__ 19. Log in as **team01** and trigger the events that you are auditing for this user:

- Execute the **ftp** command. Use your local host as destination host. You should see the corresponding audit records in file /**var**/**myaudit**/**stream.out**.
- Start the program /**home**/**workshop**/**ex13_job** in **background**. Kill the started program afterwards. You should see audit records for the **kill** audit class you've created.

_____

_____

Hint: Execute **ftp localhost**. Use the **kill** command to kill the command that runs in background.

__ 20. Stop the auditing subsystem.

END OF LAB

Exercise Instructions With Solutions

Bin Mode Auditing

__ 1. Answer the following question first: Where do you specify file system objects that should be audited?

/etc/security/audit/objects

__ 2. Set up auditing of the program /**usr**/**bin**/**passwd**. When you're finished, whenever a user calls **passwd** you should get an audit record. Add this object to the corresponding audit configuration file.

 # vi /etc/security/audit/objects

/usr/bin/passwd:
x = "X_EVENT"

Write down the event name you've created:

X_EVENT

__ 3. In which file do you have to specify the format definitions for your new event?

/etc/security/audit/events

__ 4. Add the format definition for your new event to the corresponding audit configuration file.

 # vi /etc/security/audit/events

X_EVENT = printf "%s"

__ 5. In which file do you specify the **start mode** for the auditing subsystem?

/etc/security/audit/config

__ 6. Create a directory /**var**/**myaudit**. We want to use this directory to collect all audit-related files.

 # mkdir /var/myaudit

__ 7. Change the corresponding configuration file to startup the auditing subsystem in **bin mode**. Specify the following bin files:

bin1 = /var/myaudit/bin1
bin2 = /var/myaudit/bin2
trail = /var/myaudit/trail

 # vi /etc/security/audit/config

bin:
trail = /var/myaudit/trail
bin1 = /var/myaudit/bin1
bin2 = /var/myaudit/bin2

binsize = 10240
cmds = /etc/security/audit/bincmds

__ 8. Start the auditing subsystem. Write down the command you used.

 # audit start

__ 9. Log in as **team01**. Use password **team01**. When prompted to change password, set it back to **team01**. If you use a graphical environment, execute the **login** command in a separate window.

 # login

__ 10. Execute the **passwd** command and change the password for **team01**.

$ passwd

__ 11. With **root** authority, stop the auditing subsystem. Write down the command you used.

su
 # audit shutdown

__ 12. Change to /**var**/**myaudit** and display the audit records that have been recorded. Write down the command you used.

 # cd /var/myaudit
 # auditpr -v < trail

You should see the audit event **X_EVENT** that has been created, triggered by the execution of **passwd**.

Stream Mode Auditing

__ 13. In which file do you configure **audit classes** and **audit users**?

/etc/security/audit/config

__ 14. Change this configuration file in the following way:

- The auditing subsystem starts up in **stream mode**.
- Create an audit class **kill**, that contains an audit event whenever a process gets killed.
- Remove the **root** user in the **users** stanza.
- The user **team01** should be audited for the audit classes **kill** and **tcpip**.

 # vi /etc/security/audit/config

start:
binmode = off
streammode = on

classes:

kill = PROC_Kill

users:
team01 = kill,tcpip

__ 15. In which file do you configure the **auditstream** daemon?

/etc/security/audit/streamcmds

__ 16. Before starting the auditing subsystem in stream mode, change the configuration file for the **auditstream** daemon. All audit records shall be written to file **/var/myaudit/stream.out**. Be sure to terminate the command with a &-sign.

 # vi /etc/security/audit/streamcmds

/usr/sbin/auditstream | auditpr -v > /var/myaudit/stream.out &

__ 17. Start your auditing system.

 # audit start

__ 18. Use the **touch** command and create an empty file **/var/myaudit/stream.out**. Use the tail command in a separate window to display the audit records real-time. Start aixwindow for the next step (**# xinit**).

 # touch /var/myaudit/stream.out

 # tail -f /var/myaudit/stream.out

__ 19. In a separate window log in as **team01** and trigger the events that you are auditing for this user:

 # login

- **Enter login name as team01**. Provide teams01's password.
- Execute the **ftp** command. Use your local host as destination host. You should see the corresponding audit records in file **/var/myaudit/stream.out**.

  $ ftp localhost
  Name: team01
  Password: team01

  ftp>bye

- Start the program **/home/workshop/ex13_job** in **background**. Kill the started program afterwards. You should see audit records for the **kill** audit class you've created.

  $ /home/workshop/ex13_job &
  $ kill %1

__ 20. Stop the auditing subsystem.

```
# audit shutdown
```

END OF LAB