900 Series HP 3000 Computer Systems

# Volume Management

HP Part No. 32650-90045
Printed in U.S.A. 1994

Third Edition
E0394

## Printing History

The following table lists the printings of this document, together with the respective release dates for each edition. The software version indicates the version of the software product at the time this document was issued. Many product releases do not require changes to the document. Therefore, do not expect a one-to-one correspondence between product releases and document editions.

| Edition | Date | Software Version |
|---|---|---|
| First Edition | November 1987 | A.01.00 |
| Update 1 | July 1988 | A.10.00 |
| Second Edition | October 1989 | A.30.00 |
| Third Edition | March 1994 | C.50.00 |

# Preface

MPE/iX, Multiprogramming Executive with Integrated POSIX, is the latest in a series of forward-compatible operating systems for the HP 3000 line of computers.

In HP documentation and in talking with HP 3000 users, you will encounter references to MPE XL, the direct predecessor of MPE/iX. MPE/iX is a superset of MPE XL. All programs written for MPE XL will run without change under MPE/iX. You can continue to use MPE XL system documentation, although it may not refer to features added to the operating system to support POSIX (for example, hierarchical directories).

Finally, you may encounter references to MPE V, which is the operating system for HP 3000s, not based on PA-RISC architecture. MPE V software can be run on the PA-RISC (Series 900) HP 3000s in what is known as *compatibility mode*.

## Conventions

`UPPERCASE` In a syntax statement, commands and keywords are shown in uppercase characters. The characters must be entered in the order shown; however, you can enter the characters in either uppercase or lowercase. For example:

```
COMMAND
```

can be entered as any of the following:

```
command        Command        COMMAND
```

It cannot, however, be entered as:

```
comm           com_mand       comamnd
```

*italics* In a syntax statement or an example, a word in italics represents a parameter or argument that you must replace with the actual value. In the following example, you must replace *filename* with the name of the file:

`COMMAND` *filename*

***bold italics*** In a syntax statement, a word in bold italics represents a parameter that you must replace with the actual value. In the following example, you must replace ***filename*** with the name of the file:

`COMMAND(`***filename***`)`

punctuation In a syntax statement, punctuation characters (other than brackets, braces, vertical bars, and ellipses) must be entered exactly as shown. In the following example, the parentheses and colon must be entered:

(*filename*):(*filename*)

underlining Within an example that contains interactive dialog, user input and user responses to prompts are indicated by underlining. In the following example, yes is the user's response to the prompt:

```
Do you want to continue? >>  yes
```

`{ }` In a syntax statement, braces enclose required elements. When several elements are stacked within braces, you must select one. In the following example, you must select either `ON` or `OFF`:

`COMMAND` $\left\{ \begin{matrix} \texttt{ON} \\ \texttt{OFF} \end{matrix} \right\}$

`[ ]` In a syntax statement, brackets enclose optional elements. In the following example, `OPTION` can be omitted:

`COMMAND` *filename* `[OPTION]`

When several elements are stacked within brackets, you can select one or none of the elements. In the following example, you can select `OPTION` or *parameter* or neither. The elements cannot be repeated.

`COMMAND` *filename* $\left[ \begin{matrix} \texttt{OPTION} \\ \textit{parameter} \end{matrix} \right]$

`[ ... ]` In a syntax statement, horizontal ellipses enclosed in brackets indicate that you can repeatedly select the element(s) that appear within the immediately preceding pair of brackets or braces. In the example below, you can select *parameter* zero or more times. Each instance of *parameter* must be preceded by a comma:

`[,`*parameter*`][...]`

In the example below, you only use the comma as a delimiter if *parameter* is repeated; no comma is used before the first occurrence of *parameter*:

`[`*parameter*`][,...]`

`| ... |` In a syntax statement, horizontal ellipses enclosed in vertical bars indicate that you can select more than one element within the immediately preceding pair of brackets or braces. However, each particular element can only be selected once. In the following example, you must select **`A, AB, BA`**, or **`B`**. The elements cannot be repeated.

$$\left\{ \begin{matrix} \texttt{A} \\ \texttt{B} \end{matrix} \right\} \texttt{| ... |}$$

`...` In an example, horizontal or vertical ellipses indicate where portions of an example have been omitted.

`Δ` In a syntax statement, the space symbol `Δ` shows a required blank. In the following example, *parameter* and *parameter* must be separated with a blank:

`(`*parameter*`)Δ(`*parameter*`)`

`[ ]` The symbol `[ ]` indicates a key on the keyboard. For example, `[RETURN]` represents the carriage return key or `[Shift]` represents the shift key.

`[CTRL]`*character* `[CTRL]`*character* indicates a control character. For example, `[CTRL]Y` means that you press the control key and the Y key simultaneously.

# Contents

# Figures



# Tables

# 1

# Introducing Volume Management

Volume management is designed to provide high data availability by efficiently managing disk storage space using volumes, volume sets, and volume classes.

Volume management is available on any 900 Series HP 3000 computer. Volume management proves particularly useful if you have four or more disk drives or a multisystem environment.

## Why Use Volume Management?

On many systems all the disk drives are configured as one set, the system volume set. This set contains the system control programs, working space for the system and storage for permanent files. Typically, the files are spread across multiple disk drives. Since system files must be online, and these files are spread across the drives, what happens if any drive goes down? The whole system goes down.

Volume management provides a solution to these problems by configuring the drives so that system files are stored together on a separate disk or set of disks. This prevents "nonsystem" disks, containing user data, from affecting the system. If a nonsystem drive goes down, the system continues running.

With volume management user data can be grouped on separate disks or sets of disks. This allows systems with removable disks to store specific data offline and easily restore it by mounting a disk pack.

### Volume Management Benefits

Volume management is particularly useful in managing disk storage space for the following reasons:

- High data availability.
- Reduced downtime.
- Efficient use of resources.
- Improved security.

### High Data Availability

Volume management attempts to keep as much data as possible available to the system. If all the volumes in the volume set are not mounted, the user is still able to access data from the available volumes in the volumes set.

Data can be partitioned into volume classes, so specific volumes can be grouped together to limit access to these volumes. Data can be grouped and stored on specific disks, so that only those disks need to be mounted.

### Reduced Downtime

If a drive goes offline for any reason, only the users who are currently accessing that drive or volume set of which this drive is a member are affected. Other users can continue normally unless they attempt to access the volume. If the drive comes back online, and there is no damage to the drive, all users can continue after logon. A removable disk may be transported to another disk drive on the current system or another system that has the necessary account structure and processing can continue.

In terms of the system volume set, volume management has the ability to add a new drive in critical situations. If in the middle of a large application there is not enough spoolfile space, or not enough permanent disk space, it is not necessary to bring the system down to add a new drive. Assuming that the I/O path had been previously configured, the new drive can be added to the system online, while the application is running.

Also, if a new system needs to be installed or a system needs to be rebooted, only the system volume set needs to be available.

### Efficient Use of Resources

If the system is configured as one large system volume set, then all files are on that set. If nonsystem volume sets are created, files can be partitioned and assigned to different sets. This allows a volume set to be taken offline when one group of users is finished and another set to be put online for another group of users. Even system volumes can be moved from one drive to another when using the same LDEV.

When using removable disk packs, volume sets containing applications needed during the day can be mounted each morning. Large batch jobs that can run overnight could reside on other volume sets, which can be mounted on the same drives in the evening.

### Improved Security

When using removable disks, sensitive data may be taken offline and placed in a secure location. Special capabilities also protect data on nonsystem volumes. Users without use volumes (UV) capabilities cannot access any files on a nonsystem volume, even if the files are released.

Volume management also keeps account security information on the system disk, so there is no possibility of nonsystem or transported disks interfering with the system security structure.

## What Is Volume Management?

Volume management is a facility of MPE used to manage disk space by creating volume sets and volume classes.

The basic units of volume management are listed below:

| | |
|---|---|
| Volume | A volume is a disk pack. |
| Volume sets | A volume set is a set of disks. |
| Volume classes | A volume class is a subset of the volumes in a volume set. |

### Volumes

A volume is a disk pack. A disk pack can be removable or nonremovable. For example, in Figure 1-1, six disks are shown connected to a system. Every disk connected to a system is a volume.

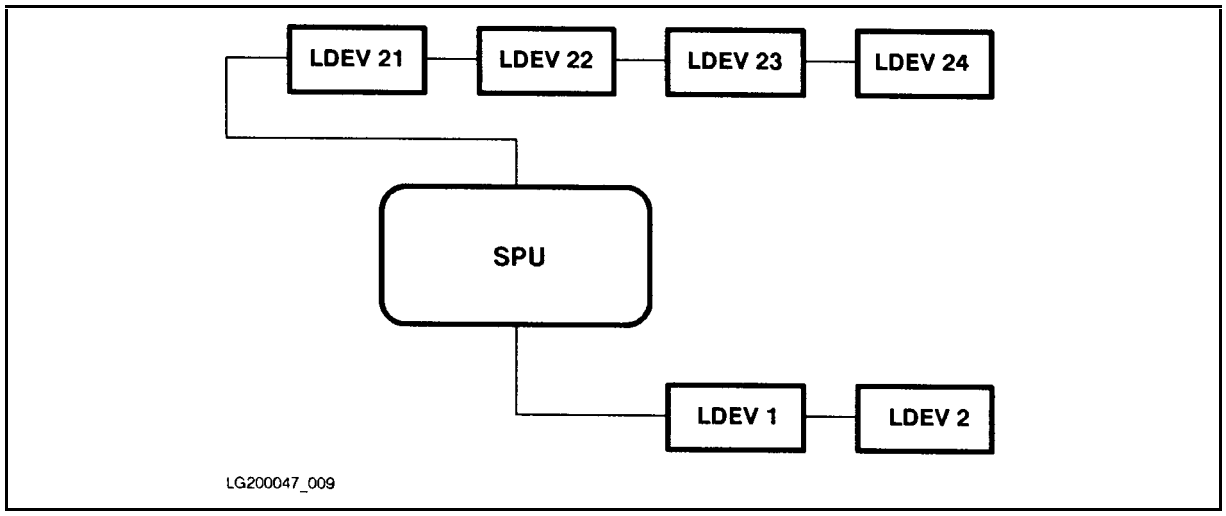



**Figure 1-1. Volumes**

## Volume Sets

A volume set is a group of volumes containing one master volume and optional member volumes. There are two types of volume sets available on the system, one system volume set that is initialized when the system is installed and nonsystem volume sets that you create. You can create nonsystem volume sets and add volumes to volume sets while the system is running.

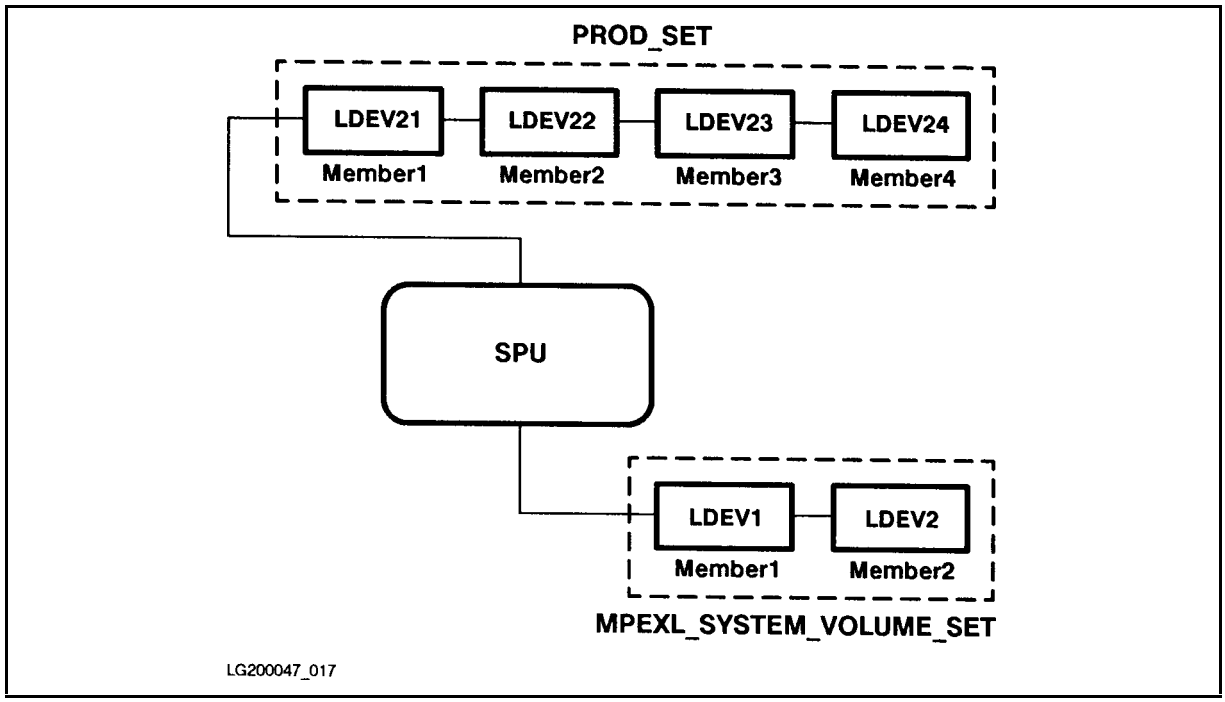



**Figure 1-2. Volume Sets**

## Volume Classes

A volume class is a subset of the volumes within one volume set. Volumes can belong to one volume class or several. Volume classes can be added to the system while the system is running.

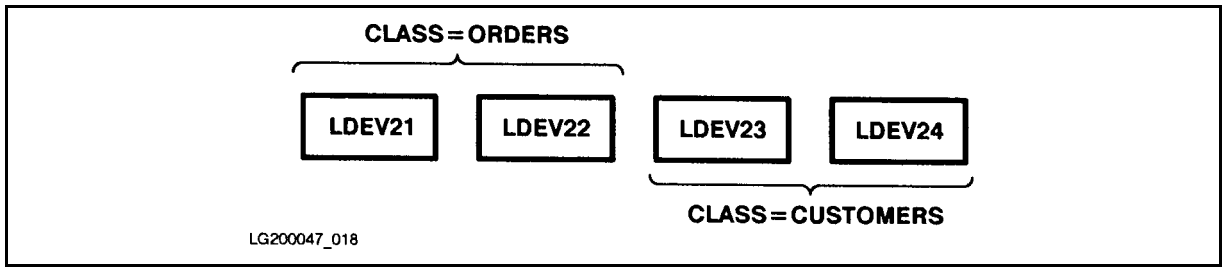



**Figure 1-3. Volume Classes**

# 2

# Understanding Volume Management

This chapter describes basic volume management tasks, terms, and how to plan for volume management.

## Volume Management Task Overview

Figure 2-1 shows the tasks involved in volume management.

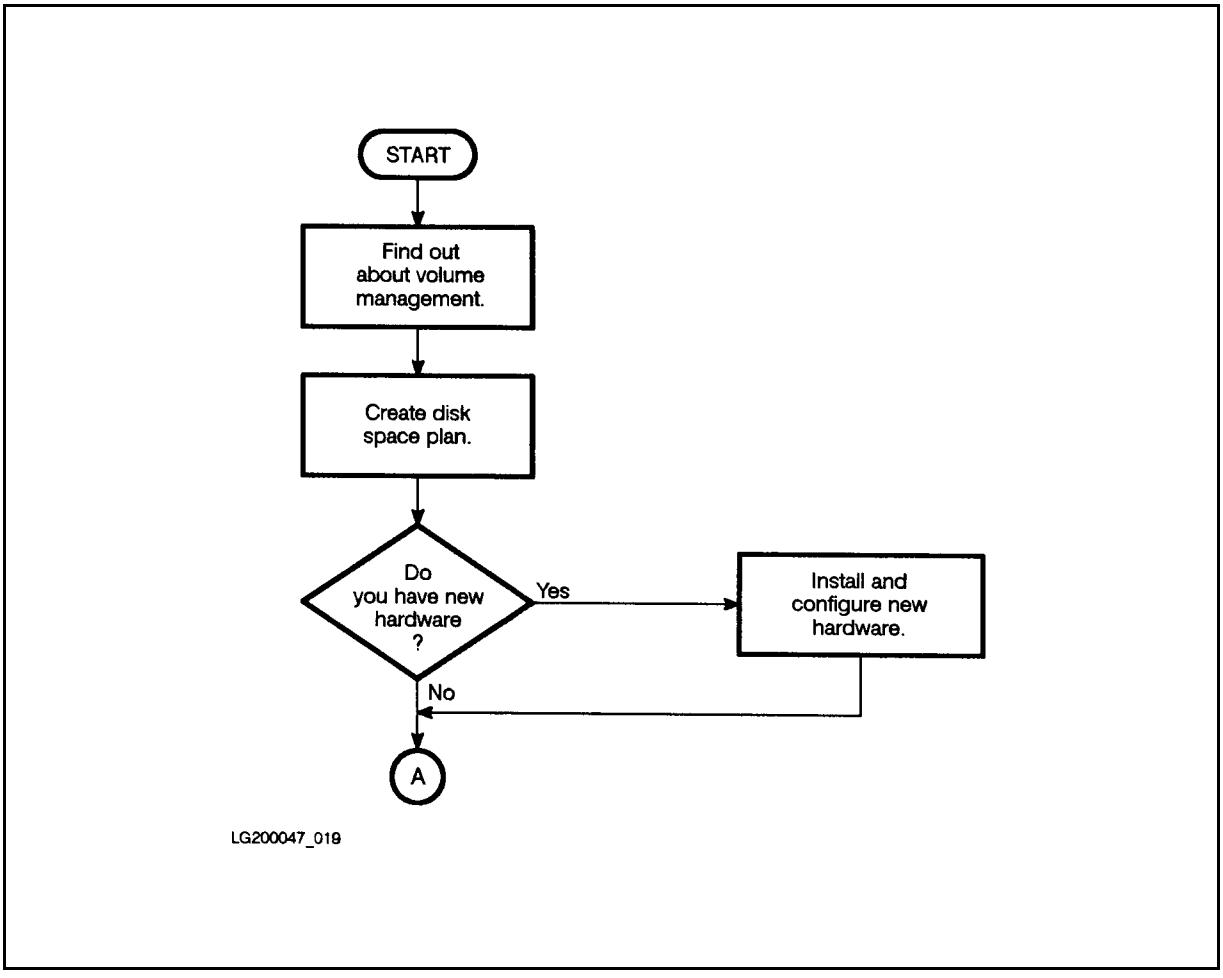



**Figure 2-1. Volume Management Task Overview**

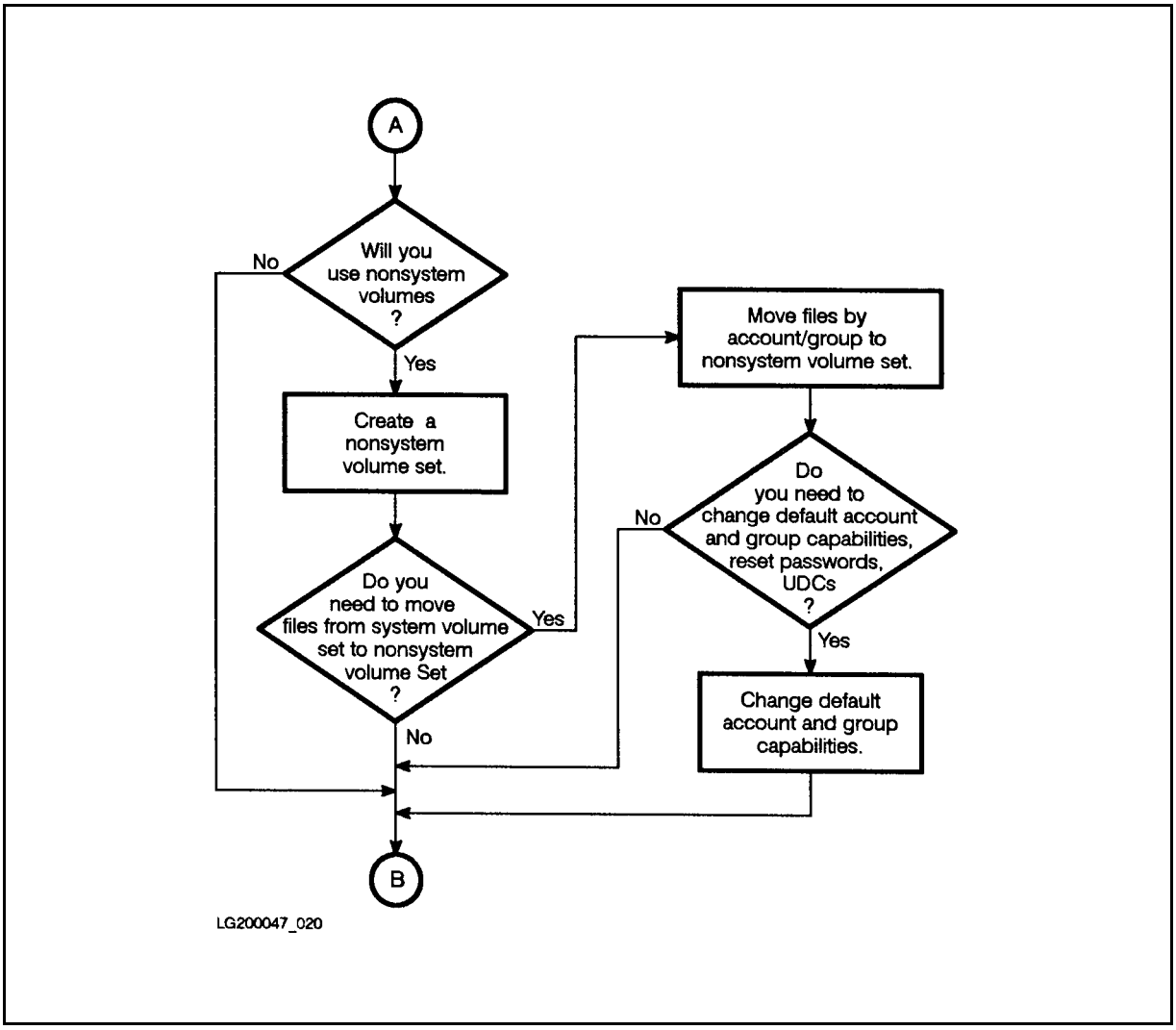



**Figure 2-2. Volume Management Overview (continued)**

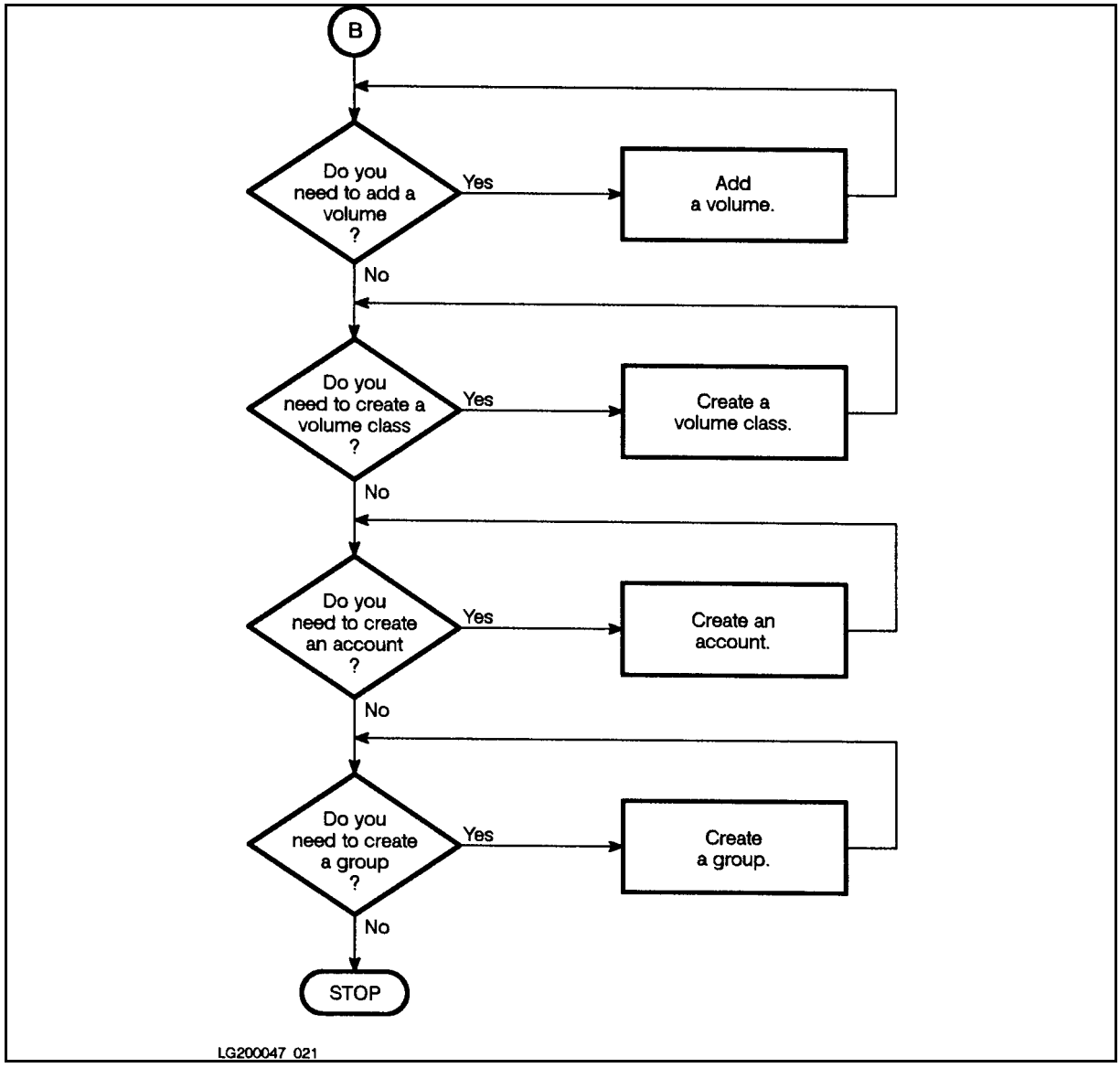



**Figure 2-3. Volume Management Overview (continued)**

The following tasks are involved in setting up volume management.

- Finding out about volume management, so you can create a disk space plan (described in this chapter).
- Installing additional hardware (if needed).
- Configuring additional disks (if needed).
- Using the VOLUTIL commands to set up volume sets and classes, according to your disk space plan (described in chapter 3).
- Using system commands to set up the account structure with users, accounts, and groups (described in chapter 3).

## Volume Management Terms

This section describes volumes, volume sets, and volume classes in more detail.

### Volumes

A volume is a disk pack. A disk pack can be removable or nonremovable. The system recognizes a volume in a particular volume state. The volume state determines whether the volume is available or not. Volume states are described in chapter 3 of this manual.

### Volume Sets

There are two types of volume sets, the system volume set and optional nonsystem volume sets. The system volume set is created when the operating system is installed and is always mounted when the system is operating. You can add extra volumes to the system volume set while the system is running. The system volume set contains two types of storage space, permanent and transient. Permanent space is used for files (permanent and temporary), label tables, and free space maps. Transient space is used for temporary data structures (stacks, heaps, and operating system data structures).

You create nonsystem volume sets when you want to restrict data for data accessibility, decrease the impact to system downtime (if a non-system volume is bad), or if you have to re-install the system. You can create and add extra volumes to nonsystem volume sets while the system is running. Nonsystem volume sets can be mounted on the system as needed and they contain only permanent space. If nonsystem volume sets are not defined on the system, all volumes will be part of the system volume set.

Table 2-1 lists the differences between the system volume set and nonsystem volume sets.

**Table 2-1. Differences between System and Nonsystem Volume Sets**

| System Volume Set | Nonsystem Volume Sets |
|---|---|
| Maximum one system volume set per system. | Up to 32 nonsystem volume sets per system. |
| Named `MPEXL_SYSTEM_VOLUME_SET`. | Can be any name up to 32 characters long. |
| Always mounted when system is running. | Can be removed (dismounted) while system running, if not needed. |
| Contains permanent and transient storage. (Disk space used for temporary structures.) | Allows permanent storage only. (Disk space used for files, label tables, and free space maps.) |
| User data and system data reside on same system disks. | User data can be partitioned onto separate disks. |
| Master contains system image and configuration(s). Requires master to be present to boot the system. | Requires master to be present to use the set. |
| Additional members can be added using the VOLUTIL utility while the system is operating. | Additional members can be added using the VOLUTIL utility while the system is operating. |

Nonsystem volume sets are also called *user* or *mountable volume sets* since they are not required for the system to run. You can remove (dismount) some or all of the volumes in a nonsystem volume set.

## Types of Volumes

A volume set consists of two types of volumes, a master volume and optional member volumes. A volume set is created when the master volume is created. The master volume is the controlling volume of each set and must be present in order to access the set. The master contains the Volume Set Information Table (VSIT), the free space map, the file label table, and the root node of the accounting directory for the set. The VSIT is a table that contains information about all of the volumes and classes in the set. A member volume contains a volume label that identifies its volume set, a free space map, and a file label table.

### Master Volumes

Table 2-2 lists the information contained on the master volume of a volume set.

**Table 2-2. Master Volume Information**

| Information | Description |
|---|---|
| volume label | Name of the master volume. |
| file label table | File labels for the files residing on the volume set. |
| free space map | Map of allocated and available disk sectors on the volume set. |
| directory root | Contains the directory of the volume set. |
| volume set's information table (VSIT) | Defines the volume set's configuration including volume set's name, names of the volumes and classes in the set, and the volumes in each class. |
| Files | User data. |

The volume master of a nonsystem volume set is the only volume needed to define a volume set. When a master volume is mounted, the volume set is considered mounted. The master volume must be mounted before any file access can be made to other members of the volume set.

The master volume is created when you create a nonsystem volume set. It contains information about the set, including a root directory which enables the system to recognize the volume set.

The system master volume is initialized as LDEV 1. After the system is rebooted, additional volumes may be added online to the system volume set using the VOLUTIL utility described in chapter 3 of this manual. The master volume for the system volume set contains all the system files and system configuration information. It must be mounted for the system to boot or run.

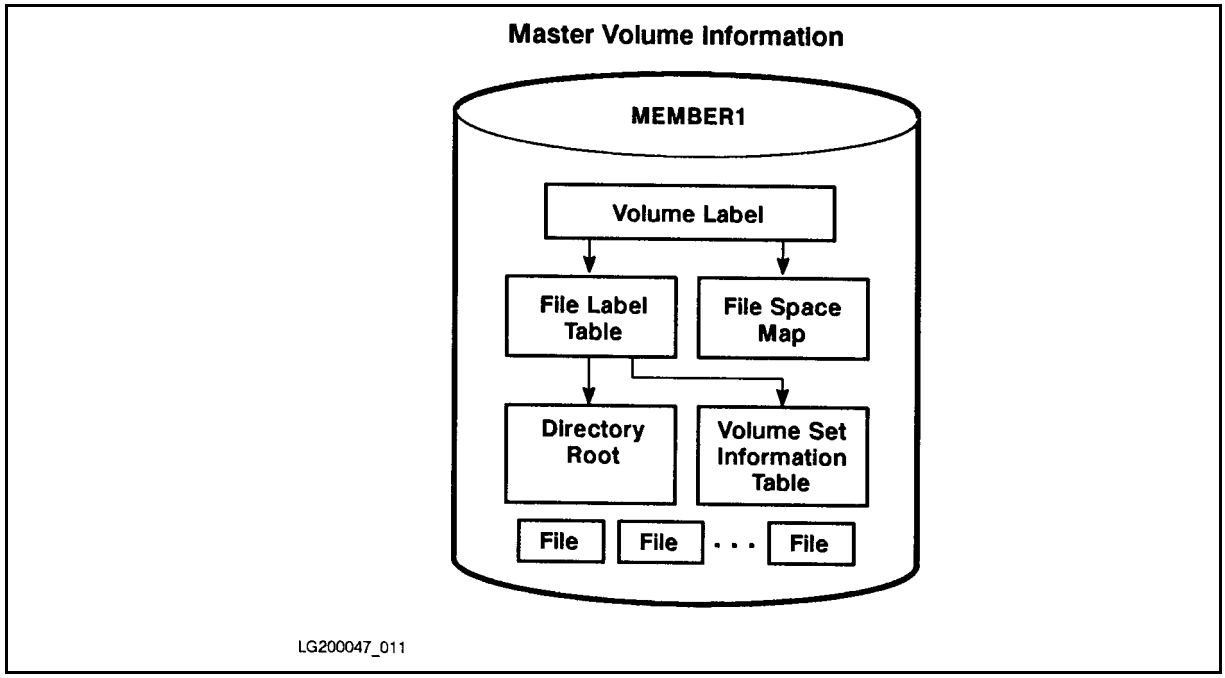



**Figure 2-4. A Master Volume**

**Note**

The master volume of a nonsystem volume set must be mounted for the system to recognize the volume set.

The master volume of the system volume set must be mounted to boot the system.

### Member Volumes

Member volumes can be added online after the volume set is created. They contain the information described in Table 2-3.

**Table 2-3. Member Volume Information**

| Information | Description |
|---|---|
| Volume Label | Name of the member volume. |
| File Label Table | Contains the file labels for the files residing on the volume set. |
| Free Space Map | Map of allocated and available disk sectors on the volume set. |
| Files | User data. |

Member volumes can belong to only one volume set since the root directory for their file information is contained on the master volume of the set.

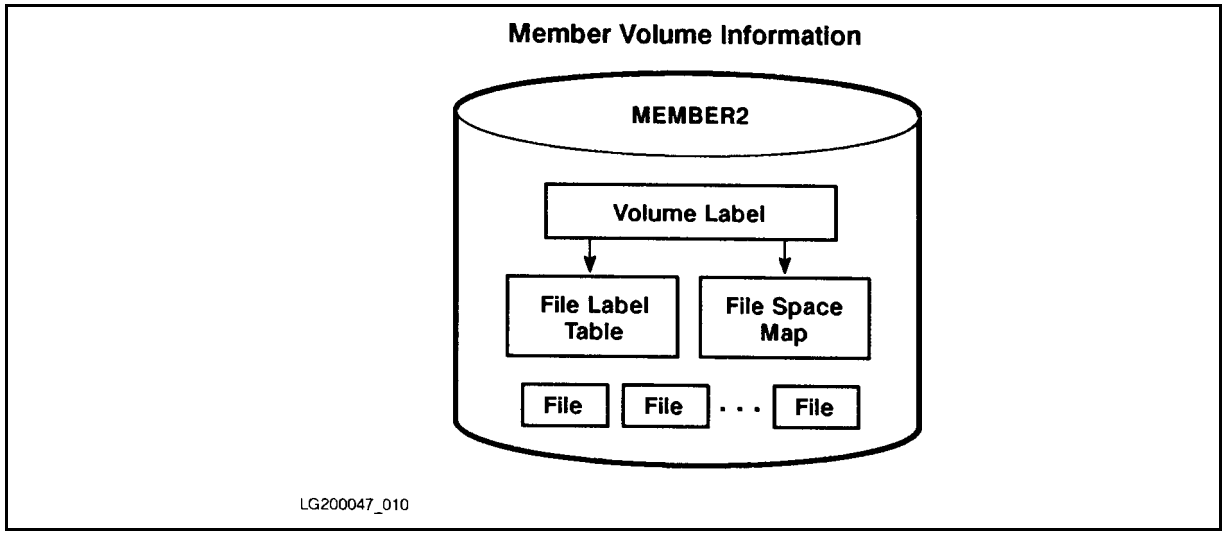



**Figure 2-5. A Member Volume**

### Volume Classes

A volume class is a subset of the volumes within one volume set. Volume classes are useful in restricting data to particular volumes in a volume set. Only the master volume and those particular volumes are needed to run applications. When data is restricted to a volume class, this limits the chance of losing data if a particular volume fails. Then problems on volumes outside that class, except the master volume, do not affect that data.

If resources are limited, more volumes can be defined in a set than there are physical drives to mount them. In this case, the volumes can be grouped into classes according to the types of information they contain. The volumes within a specific class can then be mounted as the data is needed.

Additionally, volume classes can be used to keep application programs on one drive and data on another, or to allow data to be mounted at different times.

---

**Note** When you create a new volume set, unless you specify another class, volume management assigns all volumes in the set to the default volume class, `DISC`.

---

## Planning for Volume Management

Volume management manages disk space to keep as much data as possible available to the system. This is accomplished by partitioning or restricting data according to your disk space plan.

The following guidelines will help you develop your own disk space plan.

- How will disk space be used? (need nonsystem volumes?)
- What files should exist on the system volume? on nonsystem volumes?
- What types of disk hardware are available (removable, capacity)?
- How should data be restricted?
- Who needs access to these volumes? (the account structure?)
- Are volume classes needed?
- What future expansion is planned?

### Restricting Data

At the heart of your disk space plan is data restriction. Who can access a file? Which files should reside on a volume set? How should files be restricted?

Restricting files to a volume set, a volume class, or a volume gives you greater control over data access and increases data availability. There are three levels of restricting data.

- Volume set (least restrictive).
- Volume class (more restrictive).
- Volume (most restrictive).

**Note** There may be a performance lag for being more restrictive, especially for large, active files such as databases. If many users are accessing a large file and the file is restricted to one volume, the disk drive may become a bottleneck for file access.

The default restriction is volume class DISC, on the home volume set of the group where the file resides. This means that the file extents are placed on any volume that is part of the volume class DISC within the volume set.

### Volume Set Restriction

The default volume set restriction is the least restrictive. Since a file cannot span volume sets, if the master volume fails, access to the entire volume set is denied. If a volume in the volume set fails, access to the file is denied.

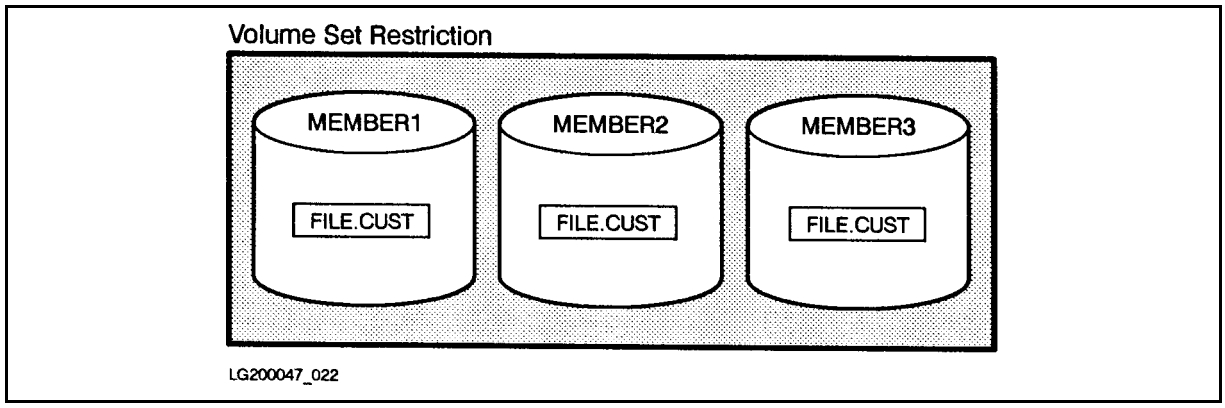



**Figure 2-6. Volume Set File Restriction**

### Volume Class Restriction

With the exception of the volume class `DISC`, any volume class restriction must be specified at file creation time. A file is placed only on the volumes within that volume class. If a volume in the class fails, then access to the file is denied.

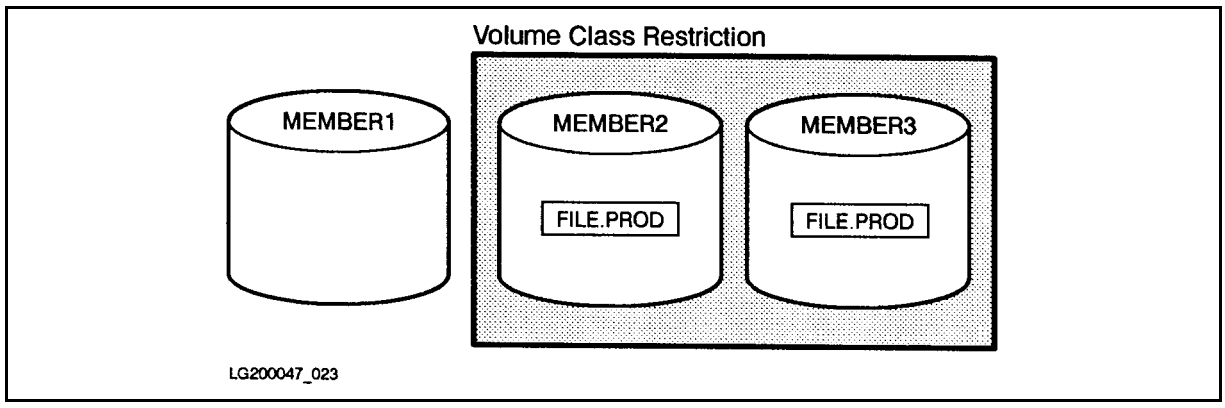



**Figure 2-7. Volume Class File Restriction**

### Volume Restriction

Volume restriction is the most restrictive/granular level of protection. Volume restriction must be specified at file creation time. The file extents are placed on only one volume. If the volume fails, access to the file is denied.

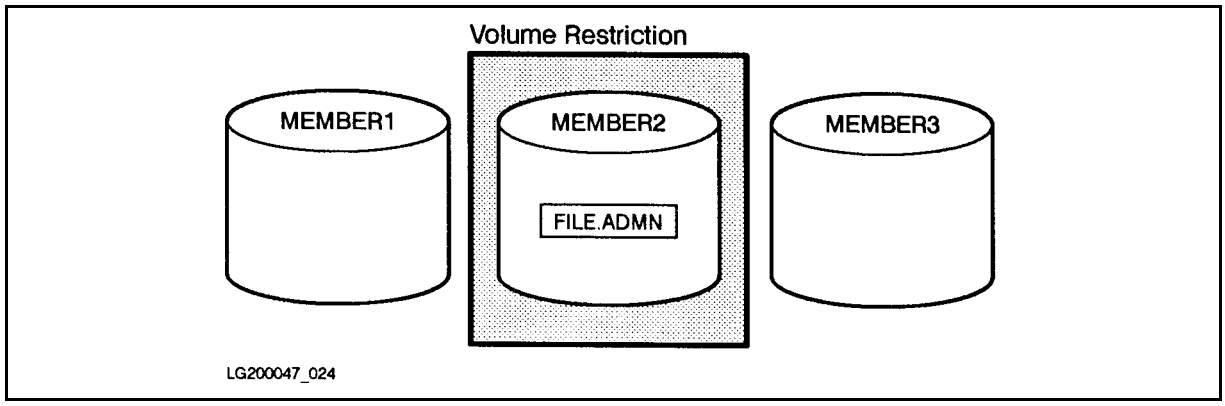



**Figure 2-8. Volume File Restriction**

## Disk Storage Space

There are two types of disk storage space, permanent and transient. Several commands in VOLUTIL (`ALTERVOL, NEWSET, NEWVOL, SHOWUSAGE`) allow you to set up, view, or change the allocations for each type of space.

Permanent space is disk space used for permanent structures such as files (permanent and temporary), the file label table , the free space map, and the directory on both system and nonsystem volume sets. Transient space is used for temporary structures such as stacks, heaps, and operating system data structures and is only allocated on the system volume set.

Each volume in the system volume set is given a percentage for permanent and transient space when it is defined with the VOLUTIL `NEWVOL` command. This value relates to the maximum amount of disk space of that type that can be allocated on the volume. For example, if you specify a permanent space value of 75% on an HP7937 disk drive (which has a total size of 2,232,192 sectors) only 1,674,144 sectors or 75% of the drive, would be available for permanent structures. This example could also be used for transient disk space.

It is possible that the total percentage for permanent and transient space will be greater than 100%. For example, the volume can be set up for 100% permanent and 100% transient. In this case, the entire drive is available for either type of storage.

The system generally uses LDEV 1 for transient space. Therefore, it is not necessary to reserve a large amount of transient space on every system volume. Since nonsystem volume sets do not have transient space, they should be set to 100/100 (permanent/transient).

The VOLUTIL `SHOWUSAGE` command displays disk space usage of a volume. The main purpose of this command is to help the user locate the 60,000 sectors of contiguous disk space needed in order to perform successful system updates.

The DISCFREE utility can be used to view disk space allocations for each volume. Refer to the *MPE/iX Utilities Manual* (32650-90081) for more information.

## Volume Management Recommendations

In summary, the following volume management recommendations should guide you in designing your disk space plan.

- Volume management proves particularly useful if you have four or more disk drives or a multisystem environment.
- You should have most of the account structure on nonsystem volume sets for higher data availability. There is little difference in the time to access nonsystem volumes versus system volumes.
- Keep volume sets small. Having multiple volume sets with only a few members is similar to having multiple volume classes on one volume set, except that volume sets can be moved and backed up separately.
- Keep volume classes small. If a disk fails, it is less likely to prevent access to data.
- Use nonsystem volumes to reduce the risk that a disk failure will affect access to data.

# 3

# Using Volume Management

This chapter describes how to use volume management to perform the following tasks:

- Display volume status.
- Create nonsystem volume sets.
- Add volumes to a volume set.
- Create volume classes.
- Create accounts and groups.
- Mount a volume set.

## Volume Management Commands

Volume management tasks are performed by using three types of commands: VOLUTIL commands, system commands, and DISCUTIL commands.

| | |
|---|---|
| VOLUTIL commands | are used to create volume sets, create volume classes, and display volume status (described in this chapter). The VOLUTIL commands are described in chapter 5. |
| System commands | are used to create accounts, create groups, and mount volume sets (described in this chapter). The system commands that refer to volume management are described in chapter 5. |
| DISCUTIL commands | are used to recover data from damaged disks (described in chapter 5). |

## Using VOLUTIL

The volume utility, VOLUTIL, provides commands for manipulating system and nonsystem volume sets. VOLUTIL resides in PUB.SYS. These commands are used to manage and maintain individual volumes, volume sets, and volume classes and to display volume content, availability, and status.

---

**Note** `VOLUTIL` must not be executed simultaneously from two or more sessions, since many commands assume a single thread of execution on the entire system.

---

### Starting VOLUTIL

VOLUTIL can be invoked from the system prompt in two ways:

```
RUN VOLUTIL.PUB.SYS
```

or

```
VOLUTIL
```

You can use any system command, such as `DSTAT`, from within VOLUTIL by first entering a colon (:), then the command.

```
volutil: :DSTAT
```

You can execute VOLUTIL commands from the system prompt by using the `INFO` parameter of the `RUN VOLUTIL` command. To use the `SHOWSET` command, enter:

```
RUN VOLUTIL.PUB.SYS;INFO="SHOWSET PUB_SET"
```

### Exiting VOLUTIL

You can exit VOLUTIL in two ways depending on how you invoked VOLUTIL.

If you invoke `VOLUTIL` with an ;INFO= string, as above, `VOLUTIL` automatically terminates after executing the specified command.

If you invoke `VOLUTIL` without an ;INFO= parameter, you can terminate VOLUTIL by using the VOLUTIL `EXIT` command. Control then returns to the system (the system prompt is displayed).

## Displaying Volume Status

Volumes are recognized by the system in the states listed in Table 3-1. The "Accessible?" column in Table 3-1 shows whether a volume can be used in the corresponding state.

**Table 3-1. Volume States**

| State | Description | Accessible? |
|---|---|---|
| `MASTER` | A volume in this state is the master volume of a volume set. In order for the system to recognize the volume set, the master volume must be mounted. | Yes |
| `MEMBER` | A volume in this state belongs to a volume set whose master is mounted. If the master were not mounted, the volume would be in the `LONER` state. | Yes |
| `LONER` | A volume is in the `LONER` state when its master is not mounted or when the volume set is taken offline by the `VSCLOSE` command. | No |
| `SCRATCH` | A volume in the `SCRATCH` state can be initialized. It may contain data, but by scratching the volume, the user has indicated that the data is no longer needed. | No |
| `UNKNOWN` | A volume in the `UNKNOWN` state does not have a label that the system can recognize. The volume may be from another system, it may be a new disk pack, or it may be a volume that has been formatted. An `UNKNOWN` volume is available for initialization. | No |

There are two commands that display the state of a volume: the system command, `DSTAT`, and the VOLUTIL command, `SHOWSET`.

Use the system command, `DSTAT`, to display the status of nonsystem disks on the system. `DSTAT ALL` displays the status of all of the disks on the system, including the system disks.

**Note** It is a good idea to frequently display disk information to verify disk status when using VOLUTIL commands.

### Example: Displaying Volume Status

This example shows how to display the status of all of the disks recognized by the system by using the DSTAT command.

At the system prompt, use the DSTAT ALL command.

```
: DSTAT ALL

LDEV-TYPE  STATUS  VOLUME  (VOLUME SET - GEN)
- - - - -  - - -   - - - - - - - - - - - - - -
1-079350   MASTER  MEMBER1 (MPEXL_SYSTEM_VOLUME_SET-0)
2-079350   MEMBER  MEMBER2 (MPEXL_SYSTEM_VOLUME_SET-0)
10-07937   UNKNOWN
11-07937   LONER
12-07937   SCRATCH
```

The above example shows that there are two system volumes (LDEVs 1 and 2), and three volumes (LDEVS 10,11,and 12) available for initializing: one SCRATCH, one LONER, and one UNKNOWN volume.

This example shows how to use the SHOWSET command to display disk status.

1. Invoke VOLUTIL.
2. At the VOLUTIL prompt, use the SHOWSET command.

```
: VOLUTIL

volutil:SHOWSET PROD_SET:VOLUMES

Volume Name         Vol Status      Ldev
- - - - - - - -    - - - - - - - -  - - -
MEMBER1             UNKNOWN         10
MEMBER2             LONER           11
MEMBER3             SCRATCH         12
```

The DSTAT and SHOWSET commands are described in detail in chapter 5.

## Creating a Nonsystem Volume Set

A nonsystem volume set is created by initializing the master volume of the set by using the VOLUTIL NEWSET command. You cannot create a system volume set. Only one system volume set is supported on the system and that volume set, MPEXL_SYSTEM_VOLUME_SET, is created automatically at system initialization. Once a volume set is created by creating the master volume, the system recognizes the volume set and the volume is ready for use.

**Note** The new volume must be mounted in the SCRATCH or UNKNOWN state.

### Example: Creating a Nonsystem Volume Set

The following example shows how to create a nonsystem volume set.

1. Invoke VOLUTIL from the system prompt.
2. At the VOLUTIL prompt, use the DSTAT command to determine what volumes can be initialized.

```
volutil: :DSTAT

LDEV-TYPE  STATUS    VOLUME  (VOLUME SET - GEN)
- - - - -  - - - -  - - - - - - - - - - - - - -
10-07937   UNKNOWN
11-07937   LONER
12-07937   SCRATCH
```

3. Create the nonsystem volume set, PROD_SET, using the VOLUTIL NEWSET command. Refer to chapter 5 for more information on this command.

   If you do not specify a volume class, the default class DISC will be assigned to the volume set.

   Volume names can be up to 16 alphanumeric characters in length. The first character of the name must be alphabetic.

```
:VOLUTIL

volutil: NEWSET PROD_SET MEMBER1 10
```

4. The system responds with a question asking you to verify whether the information you input was correct. When you respond Y followed by Enter, the system displays process information.

5. After you create a nonsystem volume set, use the DSTAT command to verify that the volume set was created.

```
volutil: :DSTAT

  LDEV-TYPE   STATUS    VOLUME  (VOLUME SET - GEN)
  - - - - -   - - - -  - - - - - - - - - - - - - -
  10-07937    MASTER     MEMBER1 (PROD_SET-0)
  11-07937    LONER
  12-07937    SCRATCH
```

**Caution** Make sure you see the message verifying that the volume was initialized correctly. Any error that occurs during initialization means that the volume must be reinitialized.

The name of the volume set is PROD_SET. Do not include the "-0" when asked to input the volume set name. The "-0" is the generation number (GEN) displayed under the heading (VOLUME SET - GEN) in the DSTAT screen.

## Adding Volumes to a Volume Set

You can add volumes to a volume set while the system is running. To add a volume to a system or nonsystem volume set, use the NEWVOL command. Once a volume is added to a volume set and initialized, the system recognizes the volume and the volume is ready for use.

**Note** If the new volume is mounted, it must be in the SCRATCH or UNKNOWN state to be initialized and added to a volume set. It is possible to add a volume to a volume set even if that set is not mounted. Then when the volume set is mounted, it must be initialized using the INITVOL command before it can be used.

### Example: Adding Volumes to a Volume Set

This example shows how to add a volume to a nonsystem volume set.

1. At the VOLUTIL prompt, use the DSTAT command to determine what volumes can be initialized.

```
volutil: :DSTAT

  LDEV-TYPE   STATUS    VOLUME  (VOLUME SET - GEN)
  - - - - -   - - - -  - - - - - - - - - - - - - -
  10-07937   MASTER     MEMBER1 (PROD_SET-0)
  11-07937   LONER
  12-07937   SCRATCH
```

2. LDEV 11 is a LONER volume and you determine it. It no longer contains any needed data. Since a volume needs to be in the SCRATCH or UNKNOWN state in order to be added to a volume set, change LDEV 11 to a SCRATCH volume by using the SCRATCHVOL command. Refer to chapter 5 for more information on this command.

```
volutil: SCRATCHVOL 11
```

3. To add the volume (LDEV 11) to PROD_SET, use the NEWVOL command. If you do not want to initialize the new volume, do not use the *ldev* parameter in the command. It is possible to add a volume and not initialize it (if you do not have a drive available). You can then initialize it when a drive is available. Refer to the INITVOL command described in chapter 5 of this manual.

   If you do not specify a volume class, the default class DISC will be assigned to the volume.

   The volume name can be up to 32 alphanumeric characters in length. The first character must be alphabetic.

```
volutil: NEWVOL PROD_SET:MEMBER2 11
```

4. The system responds with a question asking you to verify whether the information you input was correct. When you respond Y followed by Enter, the system displays process information.
5. After you have added a volume to a volume set, use the DSTAT command to verify that the volume was added to the volume set.

---

**Note** A volume must be mounted in the LONER status before it can be SCRATCHED using the SCRATCHVOL command.

---

```
volutil: :DSTAT

  LDEV-TYPE   STATUS    VOLUME  (VOLUME SET - GEN)
  - - - - -  - - - -   - - - - - - - - - - - - - -
  10-07937   MASTER    MEMBER1 (PROD_SET-0)
  11-07937   MEMBER    MEMBER2 (PROD_SET-0)
  12-07937   SCRATCH
```

**Caution** Make sure you see the message verifying that the volume was initialized correctly. Any error that occurs during initialization means that the volume must be reinitialized.

Once a volume has been added as part of a volume set, its name and volume ID are loaded on the master volume. This information cannot be deleted and remains there as long as the volume set exists.

**Note** A volume cannot be deleted from a volume set.

## Creating Volume Classes

Volume classes are used to restrict data. To create a volume class on a system or nonsystem volume set, use the `NEWCLASS` command.

**Note** The default class `DISC` is assigned to nonsystem volumes unless another volume class is specified when using the `NEWSET` or `NEWVOL` commands. Also, the master volume does not need to be a member of every volume class in the volume set.

### Example: Creating a Volume Class

This example shows how to create a new volume class.

1. To create a new class, `ENGR`, on the `PROD_SET` nonsystem volume set, use the following command at the VOLUTIL prompt:

   The volume class name can be up to 32 alphanumeric characters in length. The first character must be alphabetic.

```
volutil: NEWCLASS PROD_SET:ENGR MEMBER1
```

2. Verify that the class has been created by using the `SHOWCLASS` command.

```
volutil: SHOWCLASS PROD_SET:ENGR

Volume class index: 2
Number of volumes in class: 1
```

Once a class is specified, the system recognizes the class and the class is ready for use.

**Note** A class cannot be deleted from the volume set. Likewise, a volume cannot be deleted from a class.

## Adding Volumes to a Class

To add a volume to an existing class on a system or nonsystem volume set, use the EXPANDCLASS command. Refer to chapter 5 for more information on this command.

### Example: Adding a Volume to a Class

This example shows how to add two volumes to an existing class.

1. To add two member volumes to the existing volume class, ENGR, use the following command at the VOLUTIL prompt.

```
volutil: EXPANDCLASS PROD_SET:ENGR (MEMBER2)
```

2. Verify that the volumes have been added to the class by using the SHOWCLASS command:

```
volutil: SHOWCLASS PROD_SET:ENGR DSTATUS

Volume name:        State:    ldev:     Type:     Path:
- - - - - -        - - -     - - -     - - -     - - -
MEMBER1            MASTER    11        079330    8.0.4
MEMBER2            MEMBER    12        079330    8.0.5
```

   This screen shows that the volumes have been added.

Once volume members have been added to a volume class, the system recognizes them, and they are available for use.

## Moving files

Depending on your disk space plan, you may need to move files from the system volume set to a nonsystem volume set.

System volume files can be moved to a nonsystem volume set by using the `RESTORE` command to move the entire group or account. For more information on this command, refer to the *Storing Files and Backing Up the System Reference Manual* (32650-90140) manual.

### Example: Moving Files

This example shows how to move all of the files in an account from the system volume set to a nonsystem volume set.

**Note** It is important to follow this example in order to create the account structure and directory entries necessary on both the system and nonsystem volume set.

1. Logon to the system.
2. Use the `REPORT` command to display information about the account, MANUF, on the system volume. You should also use the `LISTUSER` command to display users and their capabilities for the `@.MANUF` users, and the `LISTGROUP` command to display group information. Refer to the *MPE/iX Commands Reference Manual Volumes 1 and 2* (32650-90003 and 32650-90364) for more information.

```
:REPORT @.MANUF

ACCOUNT FILESPACE=SECTORS  CPU-SECONDS  CONNECT-MINUTES
 /GROUP      COUNT  LIMIT   COUNT  LIMIT   COUNT  LIMIT

MANUF        288     **       0     **       0     **
 /PROD       272     **       0     **       0     **
 /PUB         16     **       0     **       0     **
```

```
:LISTUSER @.MANUF

*************
USER: USER1.MANUF

HOME GROUP: PUB                    PASSWORD: **
MAX PRI   : 150                    LOC ATTR: $00000000
LOGON CNT : 0
CAP: AM,AL,GL,ND,SF,BA,IA

...
```

3. Use the STORE command to copy the system files to tape by account. Refer to the *Storing Files and Backing Up the System Reference Manual* (32650-90140) manual for more information on storing files.

```
:FILE T;DEV=TAPE
:STORE @.@.MANUF;*T;SHOW
```

4. Use the REPORT command to make sure the account does not exist on the nonsystem volume set. Here ONVS refers to "on the specified volume set." ONVS can also be used to refer to the system volume set.

```
:REPORT @.MANUF;ONVS=PROD_SET

NON-EXISTENT ACCOUNT (CIERR 909)
:
```

**Note** Remember to purge the account or group on the system volume first before using the RESTORE command with the CREATE option. The account structure will not be created correctly if the account or group exists on either the system or nonsystem volume set when the files are restored.

5. Use the PURGEACCT command to delete the account from the system volume set. If you were moving files by group, you would need to use the PURGEGROUP command to purge the group from the system volume set. Refer to the *MPE/iX Commands Reference Manual Volumes 1 and 2* (32650-90003 and 32650-90364) for more information.

```
:PURGEACCT MANUF
```

6. Use the RESTORE command with the CREATE option to restore the files from tape to the nonsystem volume set, PROD_SET.

```
:RESTORE *T;@.@.MANUF;CREATE;SHOW;VOLSET=PROD_SET
```

This command creates the necessary group and account entries on both the system and nonsystem volume sets.

**Note** This command will only create user who are creators of the files restored. If there are any @.MANUF users that only read or modify files but do not create them, these users will not be created. Hence the use of the LISTUSER command in Step 2 of this procedure.

```
RESTORE CREATED ACCOUNT "MANUF  "
          ON VOLUMESET "PROD-SET"
RESTORE CREATED ACCOUNT "MANUF  "  MIRROR
         FOR VOLUMESET "PROD_SET"
RESTORE CREATED GROUP "PROD  "
          ON VOLUMESET "PROD-SET"
RESTORE CREATED GROUP "PROD  "  MIRROR
         FOR VOLUMESET "PROD_SET"
WILL RESTORE  2 FILES; NUMBER OF FILES ON TAPE  2
FILENAME GROUP   ACCOUNT  VOLUME    RESTRICTIONS     SECTORS CODE   REEL
SAMPLE1 .PROD    .MANUF   PROD_SET                   :200    10      1
SAMPLE2 .PROD    .MANUF   PROD_SET                   :72     10      1
SAMPLE  .PUB     .MANUF   COULD NOT BE GIVEN SPECIFIED VOLUME RESTRICTIONS
SAMPLE  .PUB     .MPEXL_SYSTEM_VOLUME_SET            :16     10      1

FILES RESTORED:
```

**Caution** Make sure you see the message verifying that the files were successfully restored. Any error that occurs during file restoration means that the files need to be restored again.

"MIRROR" on the screen tells you that the account, MANUF was created on both the system volume set and the nonsystem volume set.

7. Use the REPORT command to make sure the account now exists on the nonsystem volume set. Notice that the PUB group still resides on the system volume set.

**Note** The PUB group normally resides on the system volume set. When files are restored from tape, the PUB group's files will automatically be restored to the system volume set. If you want to change the home group of the PUB group, you will need to use the `ALTGROUP` command. Refer to the *MPE/iX Commands Reference Manual Volumes 1 and 2* (32650-90003 and 32650-90364) for more information on this command.

```
:REPORT @.MANUF;ONVS=PROD_SET

ACCOUNT FILESPACE=SECTORS  CPU-SECONDS  CONNECT-MINUTES
 /GROUP      COUNT  LIMIT   COUNT  LIMIT   COUNT  LIMIT

MANUF          272    **       0     **       0     **
 /PROD         272    **       0     **       0     **
 /PUB            0    **       0     **       0     **

:REPORT @.MANUF

ACCOUNT FILESPACE=SECTORS  CPU-SECONDS  CONNECT-MINUTES
 /GROUP      COUNT  LIMIT   COUNT  LIMIT   COUNT  LIMIT

MANUF          16     **       0     **       0     **
 /PROD          0     **       0     **       0     **
 /PUB          16     **       0     **       0     **
```

**Note** Because the accounts, groups and users will be created with default capabilities, you may need to use the `ALTACCT`, `ALTGROUP` and `ALTUSER` commands to alter the account's, group's and user's capabilities. Remember to reset the passwords and UDCs because they are lost when you use `RESTORE` with the CREATE option.

Another way to avoid this would be to use the `NEWACCT;ONVS=`, `ALTGROUP;HOMEVS=`, and `NEWGROUP;ONVS=` (after having first stored the files), then restore the files. This keeps all capabilities, users, UDCs, and passwords intact.

Be sure to use `ALTUSER` on all users to add UV capability, or they will not be able to logon or access the files in this account.

## Creating Accounts

In order to create or use files on a nonsystem volume set, the account structure must exist on the system volume set and the nonsystem volume set. If you moved files from the system volume set according to the instructions in this chapter, you already have the necessary account structure present on both volume sets.

If you are going to create files in a new account on a new nonsystem volume set, you must first create the accounts and groups using system commands on both the system and nonsystem volume sets. If the files are to be built in an existing group and account the entries already exist on the system volume set. They only need to be modified and have appropriate entries built on the target UV set.

The system is designed to gain access to accounts through the system volume set, even though the accounts exist on the nonsystem volume set. Therefore, entries must be made in two directories (system and nonsystem directory), and two `NEWACCT` commands are necessary. The directory structures created on the system volume set and the nonsystem volume set are parallel. Account structure parameters must be used with the system commands to keep the directory structures consistent. Accounts may exist on different volume sets and groups in an account can be split across different volume sets. For example, `PUB.MANUF` on system volume set and `PROD.MANUF` on `PROD_SET`.

**Note** Files residing on nonsystem volume sets must have directory entries on the system and nonsystem volume set. Files residing on the system volume set do not need directory entries on nonsystem volume sets.

Refer to the *MPE/iX Commands Reference Manual Volumes 1 and 2* (32650-90003 and 32650-90364) for a detailed description of this command.

### Example: Creating an Account

This example shows how to create an account on a nonsystem volume set.

1. Use the `REPORT` command to find out whether the account, `ORDERS`, exists on the nonsystem volume set, `PROD_SET`.

```
:REPORT @.ORDERS; ONVS=PROD_SET
NON-EXISTENT ACCOUNT (CIERR 909)
:
```

2. The `ORDERS` account does not exist on `PROD_SET` (you should also have checked previously to make sure the `ORDERS` account does not exist on the system volume set). At the system prompt, create the account, `ORDERS`, on the system volume set with the user capabilities AM, SF, ND, CS, PH, IA, BA, UV and CV. Refer to the *MPE/iX Commands Reference Manual Volumes 1 and 2* (32650-90003 and 32650-90364) for a detailed description of user capabilities.

```
:NEWACCT ORDERS,MGR;CAP=AM,SF,ND,CS,PH,IA,BA,CV,UV
```

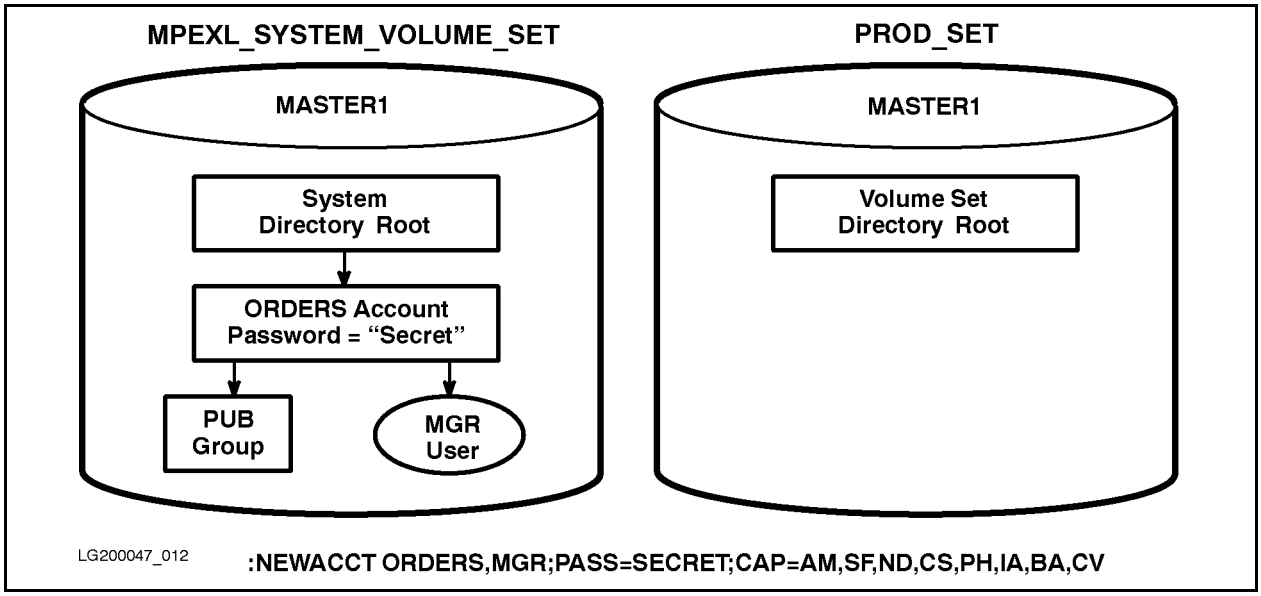



**Figure 3-1. Creating the Account on the System Volume**

3. Next create the same account on the nonsystem volume set PROD_SET, using `NEWACCT` with ONVS. ONVS tells the system to create the directory entry on the PROD_SET volume set.

```
:NEWACCT ORDERS,MGR;ONVS=PROD_SET
```

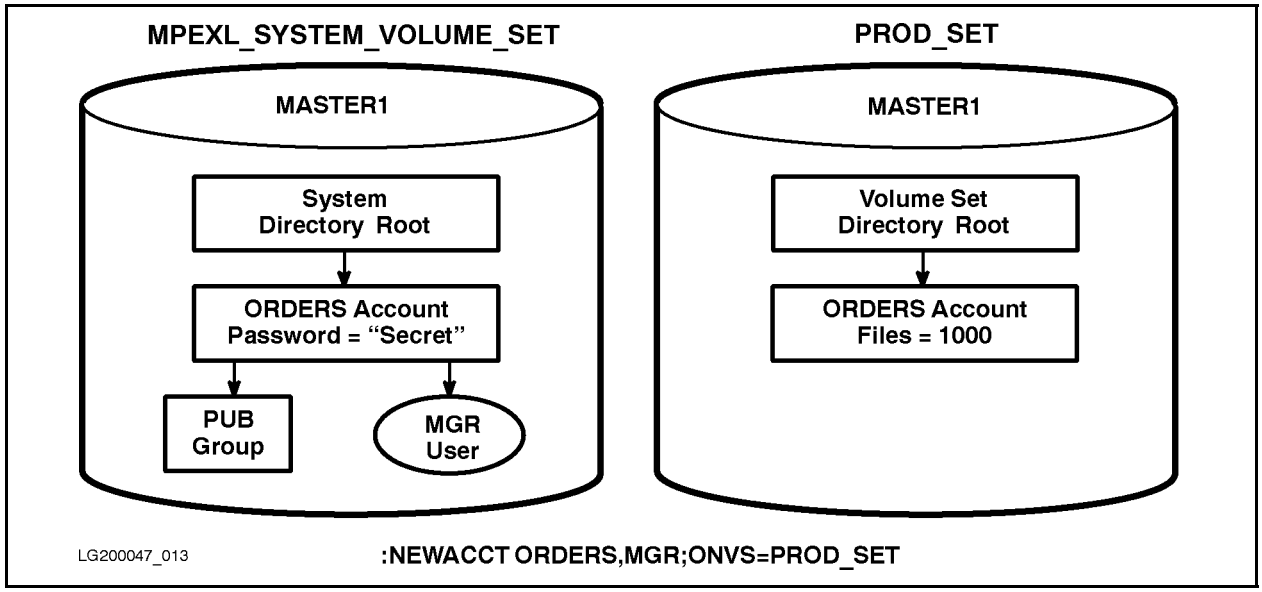



**Figure 3-2. Creating the Account on the Nonsystem Volume**

## Creating Groups

A group is specified on a nonsystem volume set to limit access to files. A group is contained within one volume set or one single volume.

The system is designed to gain access to groups through the system volume set, even though the groups exist on the nonsystem volume set. Therefore, entries must be made in two directories (system and nonsystem directory), and two `NEWGROUP` commands are necessary.

### Example: Creating a Group

This example shows how to create a group on a nonsystem volume set.

1. Log on to the new account.
2. At the system prompt, create the group, `DATABASE`, on the system volume set with the HOMEVS parameter. HOMEVS is only used with groups. HOMEVS tells the file system to place the files of a group on a particular volume set.

```
:HELLO MGR.ORDERS

:NEWGROUP DATABASE;HOMEVS=PROD_SET
```

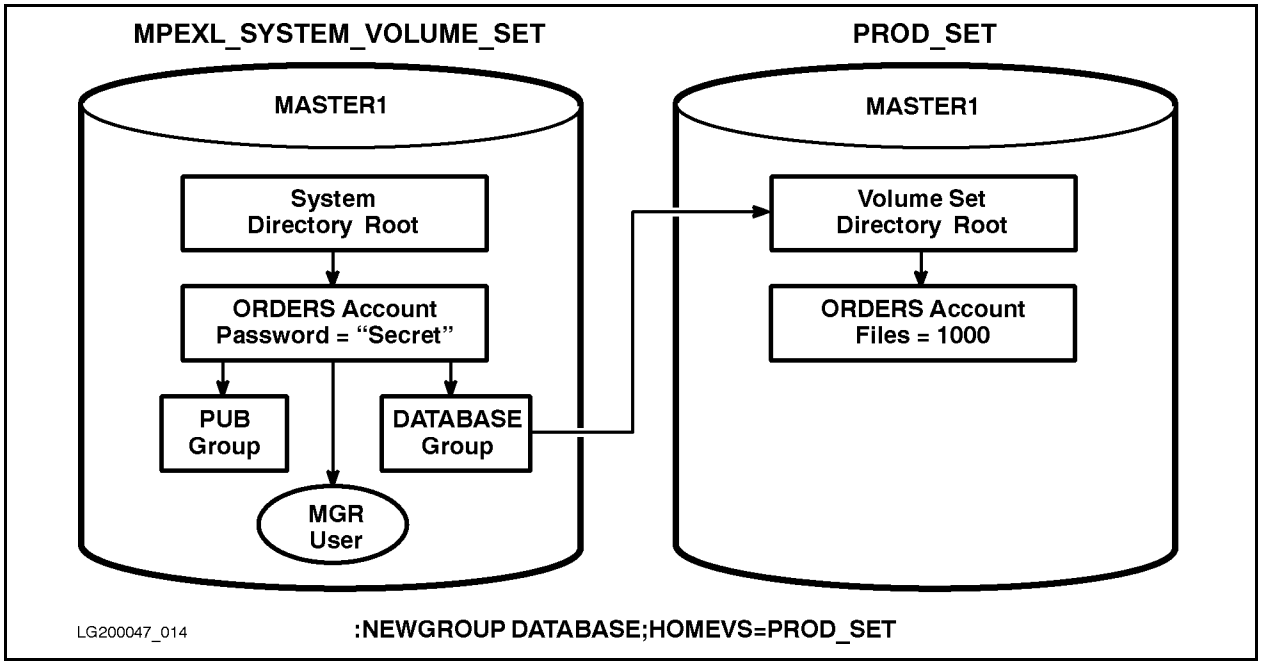



**Figure 3-3. Creating the Group on the System Volume**

3. Next create the group on the nonsystem volume set with the `ONVS` parameter. `ONVS` refers to the nonsystem volume. Refer to the *MPE/iX Commands Reference Manual Volumes 1 and 2* (32650-90003 and 32650-90364) for a detailed description of this command.

```
:NEWGROUP DATABASE;ONVS=PROD_SET
```

Account information is stored on the system volume. This means that even if the PROD_SET volume set is removed from the system, you can still log on to the `CUST` account, which is on a different volume set. This also means that if the PROD_SET volume set is moved to another system, the account structure (groups and accounts) moves with it and must also exist on the new system volume set's directory in order to access these files. System account information (like passwords and capabilities) resides on the system volume set, as does user information.

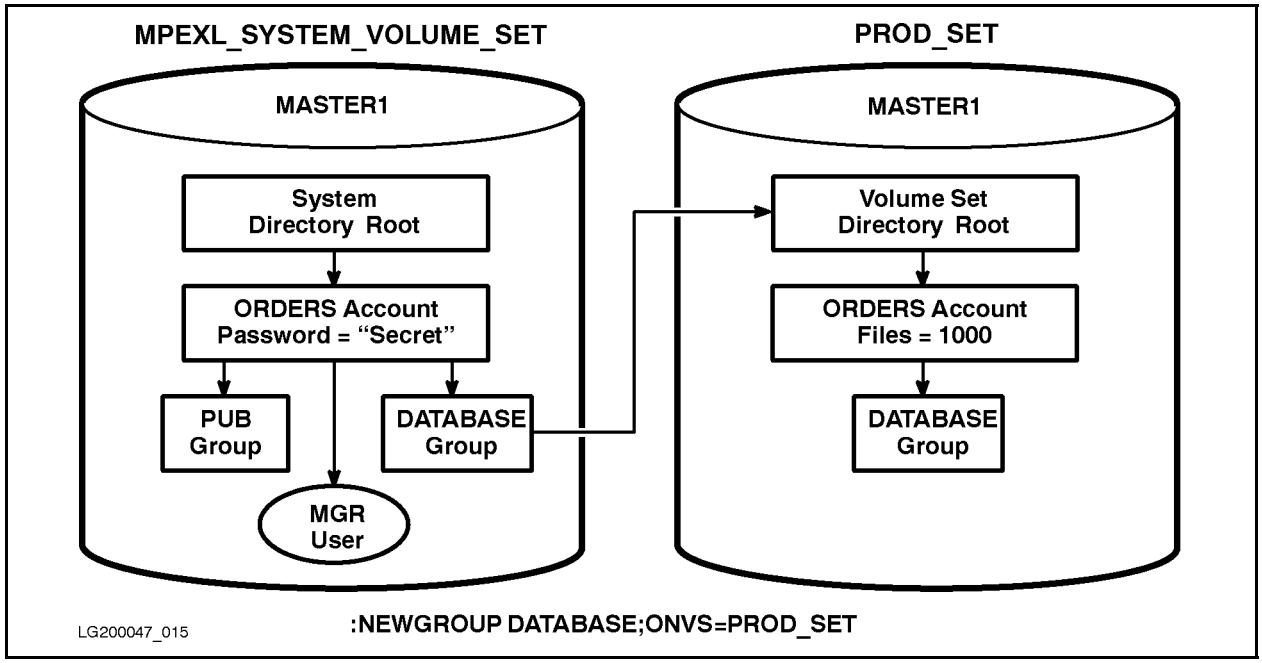



**Figure 3-4. Creating a Group on the Nonsystem Volume**

4. Use the REPORT command to verify account and group information.

```
:REPORT @.ORDERS

ACCOUNT FILESPACE=SECTORS  CPU-SECONDS  CONNECT-MINUTES
 /GROUP      COUNT  LIMIT   COUNT  LIMIT   COUNT  LIMIT


ORDERS       421     **       0     **       0     **
 /DATABASE   392     **       0     **       0     **
 /PUB         29     **       0     **       0     **

:REPORT @.ORDERS;ONVS=PROD_SET

ACCOUNT FILESPACE=SECTORS  CPU-SECONDS  CONNECT-MINUTES
 /GROUP      COUNT  LIMIT   COUNT  LIMIT   COUNT  LIMIT

ORDERS       392     **       0     **       0     **
 /DATABASE   392     **       0     **       0     **
 /PUB          0     **       0     **       0     **
```

## Creating Files

Once the account structure exists, you can use or build files. Building files is described in detail in *Controlling System Activity* (32650-90155).

This section describes the following file building information specific to volume management.

- File opening.
- File extents.
- File restriction.

### File Opening

When opening a file, the system searches for the file in the system directory root, traversing down to the group/file node, which points to the directory root on the nonsystem volume. The system then traverses this tree to find the file. However, this process only happens when a file is first accessed. All further reads and writes go directly to the file.

### File Extents

A file's extents may be spread across the volumes within a volume set. If you are using volume classes, you have to specify the volume class restriction at the time you create the file.

A master volume is needed when the file is opened because the volume set directory is on the master volume and the directory is needed to find the file's extents.

### File Restriction

Files can be restricted to a particular volume class or volume by using the `HPFOPEN` intrinsic. Refer to the *MPE/iX Intrinsics Reference Manual* (32650-90028) for more information.

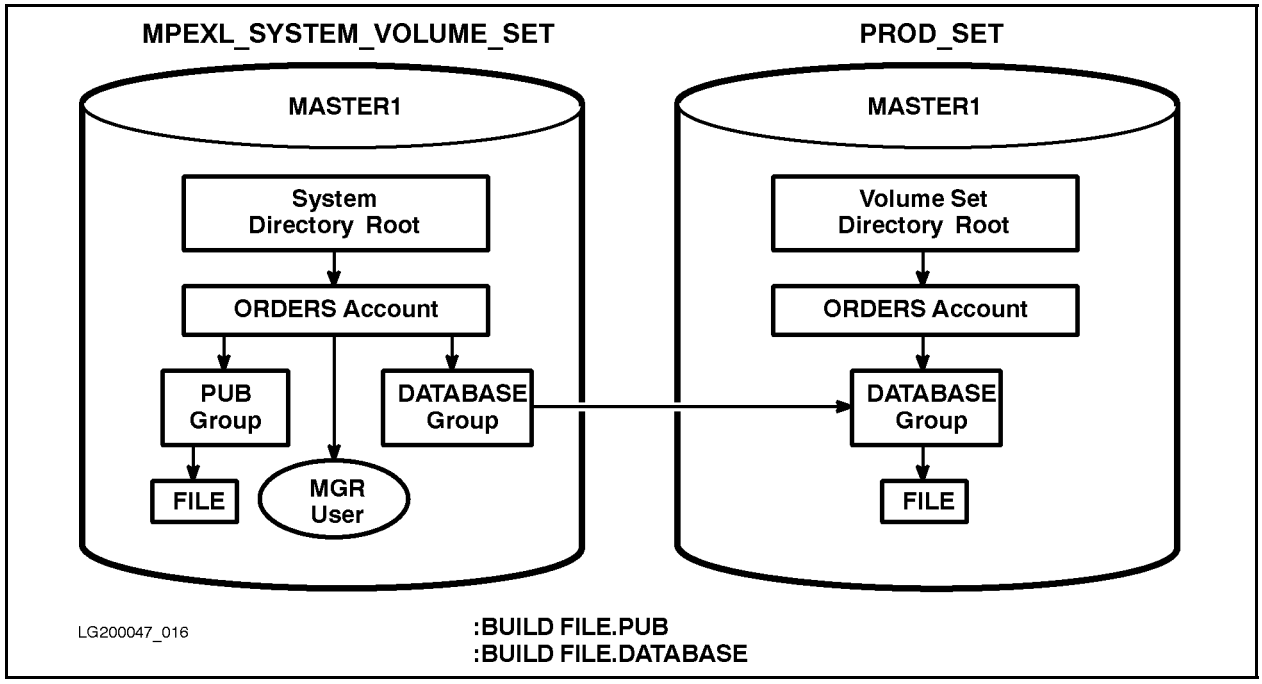



**Figure 3-5. Building Files**

## Mounting a Volume Set

Volume management makes volume sets available as soon as they are physically recognized by the system. The system automatically recognizes a volume set after the volumes have been added to the volume set, upon the power on of the disk drive, or at the boot of the system.

In order to access data from a volume set, only the volumes where the data is located and the master of the volume set need to be mounted. The system volume set master (LDEV 1) must be mounted on the disk drive at all times. Nonsystem volumes sets are not necessary to system operation and can be removed or replaced online without affecting system operation. This means that a nonsystem drive can fail without bringing down the system.

No mount command is needed to access the volume set. However, there are system mount commands that are used to notify the system that you want the volume set to remain physically mounted for a period of time. The system mount commands are described in chapter 5.

As long as the I/O path to the device has been configured, any volume set may be mounted on any system. A maximum of 32 volume sets may be mounted on the system at one time.

## System Volume Management Commands

There are two types of system volume management commands: the commands that support the system account structure and the commands that are used for volume mounting. The system account structure commands are listed here. The system volume mounting commands are described in chapter 5.

### System Account Structure Commands

The following system commands can be used with volume management to set up and update accounts and groups.

Refer to the *MPE/iX Commands Reference Manual Volumes 1 and 2* (32650-90003 and 32650-90364) for a detailed description of these commands and *Controlling System Activity* (32650-90155) for more information on account structure.

**Table 3-2. System Account Structure Commands**

| Command | Task |
|---|---|
| ALTACCT | Changes the attributes of an existing account. Requires system manager (SM) capability. Uses the ONVS keyword. |
| ALTGROUP | Changes one or more attributes of a group. Uses the ONVS and HOMEVS keywords. |
| NEWACCT | Creates a new account and an associated account manager and PUB group. Uses the ONVS keyword. |
| NEWGROUP | Creates a new group within an account. Uses the ONVS and HOMEVS keywords. |
| PURGEACCT | Removes an account and its groups and users from the directory of the specified volume set. Uses the ONVS keyword. |
| PURGEGROUP | Removes a group and its files from the system or from the directory of the specified volume set. Uses the ONVS keyword. |
| REPORT | Displays accounting information for the logon account and group. Uses the ONVS keyword. |
| RESTORE | Copies disk files from magnetic tape to disk. Users the VOLSET keyword. |
| STORE | Copies disk files onto a magnetic tape so that the files can be recovered with the RESTORE command. Uses the ONVS keyword. |

The ONVS keyword directs the command to take place "on the specified volume set." The HOMEVS keyword directs the command to take place "on the home volume set of the group" where files in a particular group are to be built.

# 4

# Troubleshooting

This chapter describes how to recover data from disks using the DISCUTIL utility and how to recover from a system and nonsystem disk failure.

For more information on recovering data, refer to the `SAVE` command in chapter 5.

## Using DISCUTIL

DISCUTIL is a standalone utility used to configure devices and to save files to tape from disks that have failed, systems that have failed or directories that have become corrupted. DISCUTIL is invoked at the initial system load (ISL) prompt and operates on all volumes that are configured and mounted. The ISL prompt is available after the system is restarted. Refer to the *System Startup, Configuration, and Shutdown Reference Manual* (32650-90042) for more information about how to restart a system.

The following example shows how to restart the system to use DISCUTIL.

1. Press CTRL B to reset the system.
2. Enter either `TC` to transfer control and save system memory or `RS` to reset the system and lose system memory at the `CM>` prompt.

```
CM>TC

Processor Dependent Code (PDC) version 2.8

Console path = 2/4.1.0.0.0.0.0
Primary boot path = 2/4.0.0.0.0.0.0
Alternate boot path = 6/4.3.0.0.0.0.0

Boot from primary path (Y or N)?> Y

Interact with IPL (Y or N)?> Y
```

```
MMSAVE Version 9.90
DUMPAREA found, save main memory to disc
ISL loaded

ISL Revision 2822 May, 1987

   Cannot find an auto-execute file. AUTOBOOT ABORTED.

ISL> DISCUTIL

discutil>
```

## Data Recovery

Data can be recovered from disks by using DISCUTIL in conjunction with the `RECOVER` command of VOLUTIL. This section describes two data recovery tasks:

- Saving files to tape using DISCUTIL.
- Recovering files to disk using VOLUTIL.

### Saving Files to Tape

Saving files to tape allows you to save data that you cannot access because the disk has failed, the system has failed, or a directory has become corrupted. The DISCUTIL `SAVE` command is used to save files.

#### Example: Saving Files to Tape

This example shows how to save disk files to tape using DISCUTIL.

1. Enter the `SAVE` command at the `discutil` prompt.
2. Enter the file set you want saved at the prompt.

```
discutil>SAVE

ENTER FILE SET TO BE SAVED: MYFILE.JOHN.SMITH
```

Use the *filename.groupname.acctname* syntax where:

*filename* is the file designator.

The file(s) must reside on a disk defined during startup of DISCUTIL or with the DISCUTIL `CONFIGURE` command. Refer to chapter 5 for more information on this command.

*groupname* is an optional group designator.

*acctname* is an optional account designator.

If you press RETURN without entering a file set, DISCUTIL ends the SAVE command.

The parameters *filename*, *groupname*, and *acctname* may be replaced by @ to signify "all members of the set". For example, @.@.MPEXL, @.UTIL.MPEXL, or @.@.@. @.@.@, @.@, and @ are equivalent.

The prompt is repeated after the file set is saved. To terminate SAVE, press RETURN.

3. Enter RETURN at the prompt below to indicate that "no" hierarchical directories are saved.

```
TRAVERSE DIRECTORIES UNDER FILESET (Y/N)?
```

4. Enter the LDEV at the prompt.

```
ENTER THE LDEV: 17
```

To save files from a particular logical device (disk), enter the LDEV now. SAVE searches only the specified disk for the file(s) you want to save. You may also press RETURN at this prompt to indicate you want SAVE to search each LDEV for file(s). The LDEV entered must be one that is known to DISCUTIL. Use the PDEV command to determine if an LDEV is valid. Refer to chapter 5 for more information on this command.

5. The SAVE command prompts you for a volume set name if you did not specify an LDEV number.

The volume set's name is the name of the volume set you want to save. If a volume set name is not given, DISCUTIL saves the file sets from all disks. If you enter an LDEV, DISCUTIL will not prompt for a volume set.

```
ENTER THE VOLUME SET NAME: PROD_SET
```

Enter the volume set's name if you want SAVE to search for file(s). The volume set's name must be known to DISCUTIL. Use the PDEV command to determine valid volume set names. You may also press RETURN to indicate you want SAVE to search all disks for the file(s).

6. Enter the modification date at the prompt.

```
ENTER THE MODIFICATION DATE (MM/DD/YYYY): 04/01/1989
```

Only files modified since this date are saved to tape. DD is a two-digit number for the day of the month, MM is a two-digit number for the month, and YYYY is a four-digit number for the year. Omit the date specification and press RETURN at the prompt to save all files requested.

```
discutil> SAVE

WARNING BLOCK

ENTER FILE SET TO BE SAVED: MYFILE.JOHN.SMITH

TRAVERSE DIRECTORIES UNDER FILESET (Y/N)?

ENTER THE LDEV: 17

ENTER THE MODIFICATION DATE (MM/DD/YYYY): 04/01/1989

ENTER THE TAPE LDEV 7

MYFILE.JOHN.SMITH - LDEV 17 - ADDR $0002CA0 - FOUND
MYFILE.JOHN.SMITH - LDEV 17 - ADDR $0002CA0 - SAVED

WARNING BLOCK

ENTER FILE SET TO BE SAVED:
```

**Modified Files Saved**

DISCUTIL now searches each LDEV specified for the designated file(s). Each time a file is found a message is printed. When the file is successfully written to tape another message is printed. Errors are also reported. If the end of a tape is reached, DISCUTIL prompts to mount a new tape. Once a tape is mounted, the save continues until all specified files are written.

7. Enter the TAPE LDEV: ldev
8. After all specified files are saved, the ENTER FILE SET TO BE SAVED: prompt is displayed and you may specify additional files.

### To Make Multiple Tapesets

To make multiple tapesets, the current SAVE command must finish with the current tapeset. To do this, type (RETURN) at the ENTER FILE SET TO BE SAVED: prompt. This returns you to the DISCUTIL> prompt, and the following message displays.

```
MANUALLY REWINDING THE TAPE DRIVE AND STARTING A NEW TAPESET
BEFORE THE CURRENT SAVE COMMAND IS COMPLETE WILL CAUSE FILES TO BE LOST!!!
```

### Example: Making Multiple Filesets

This example shows how to make multiple filesets.

1. Invoke SAVE.
2. The system displays the following:

```
                        WARNING!!!
MANUALLY REWINDING THE TAPE DRIVE AND STARTING A NEW TAPESET
BEFORE THE CURRENT SAVE COMMAND IS COMPLETE WILL CAUSE FILES TO BE LOST!!!

PRESS (RETURN) AT "ENTER FILE SET TO BE SAVED:" TO COMPLETE THE SAVE
PROPERLY.  SEE HELP ON SAVE.

ENTER FILE SET TO BE SAVED:  @.@.@

TRAVERSE DIRECTORIES UNDER FILE SET (Y/N)?  (RETURN)

ENTER THE LDEV:  (RETURN)

ENTER C (RETURN) TO CONTINUE

ENTER THE VOLUME SET NAME:  (RETURN)

ENTER THE MODIFICATION DATE:  (RETURN)

ENTER THE TAPE LDEV: 7

ENTER FILE SET TO BE SAVED:  (RETURN)
:
```

### Tape Errors During a DISCUTIL SAVE

If a tape ends in the middle of a file, the rest of the file will be saved on the next tape. If a tape write error occurs, DISCUTIL resaves all files on the bad tape to a new tape. If part of a file is on the previous tape, then the entire file is resaved on the new tape. DISCUTIL marks

a tape anytime it cannot save an entire file. The VOLUTIL RECOVER command aborts the recovery of that file when it reads the mark on a file. It then continues with the next file.

### Exiting DISCUTIL

DISCUTIL is terminated by using the EXIT command. Control then returns to the system and the system prompt is displayed.

## Recovering Files to Disk

VOLUTIL's RECOVER command recreates files from the output tape produced by DISCUTIL. You will need SM, system manager, capability to use the VOLUTIL RECOVER command.

### Example: Recovering Files to Disk

This example shows how to recover disk files from tape after the operating system has been successfully restarted.

1. Load the tape that contains the files you want to recover.
2. Invoke VOLUTIL.
3. Use the RECOVER command with the NOKEEP option to start the copy. The NOKEEP option replaces files on the disk with files of the same name on the tape. As each file is copied to the disk, messages are displayed.
4. The system will prompt you to mount the next tape if the files continue onto additional tapes.

```
volutil: RECOVER;NOKEEP

(fname)   (group)   (acct)       IS RESTORED
TEXT      .PROD     .MANUF       IS RESTORED
INTRO     .PROD     .MANUF       IS RESTORED
VOLM2     .PROD     .MANUF       IS RESTORED

END OF VOLUME, RECOVER DONE
```

## Troubleshooting

This section describes how to recover from the following failures:

- System disk failure.
- Nonsystem disk failure.

### System Disk Failure

If a member of the system volume set fails because of a hardware problem, it causes a system abort. You will not be able to reboot until the disk is fixed. An `INSTALL` is usually required.

### Nonsystem Disk Failure

If a member of a nonsystem volume set fails because of a hardware problem, it does not necessarily cause a system abort. Even if it does, you can reboot after removing the defective disk. Users who do not need the failed volume can continue to work.

# 5

# Volume Management Commands

Volume management uses three types of commands: VOLUTIL commands, system commands, and DISCUTIL commands. All of the these commands are described in this chapter.

## VOLUTIL Commands

VOLUTIL commands are organized into groups depending on the ending of the command. Commands that end with `SET`, `CLASS`, or `VOL` operate on sets, classes, and volumes, respectively.

All of the VOLUTIL commands are described in this chapter. Enter a VOLUTIL command after the VOLUTIL prompt is displayed. The VOLUTIL prompt, `volutil:` , displays when the VOLUTIL utility is running.

```
volutil:  SHOWSET PROD_SET VOLUMES
```

VOLUTIL commands can be input in uppercase or lowercase like any system command.

**Table 5-1. VOLUTIL Commands**

| Command | Type | Task |
|---|---|---|
| `ALTERVOL` | Volume | Changes permanent and transient disk storage space. |
| `COPYVOL` | Volume | Copies data from one disk volume to another. |
| `DSECTORSVOL` | Volume | Detects defective sectors. |
| `FORMATVOL` | Volume | Formats a volume. |
| `INITVOL` | Volume | Initializes a volume that was previously defined by `NEWVOL`. |
| `NEWVOL` | Volume | Adds a new volume to a volume set. |
| `SCRATCHVOL` | Volume | Places a volume in the `SCRATCH` state. |
| `SHOWVOL` | Volume | Displays volume information. |

**Table 5-1. VOLUTIL Commands (continued)**

| Command | Type | Task |
|---|---|---|
| UNSCRATCHVOL | Volume | Unscratches a volume. |
| VERIFYVOL | Volume | Verifies that the data on a volume can be read. |
| COPYSET | Set | Copies data from one volume set to another. |
| NEWSET | Set | Creates a new volume set. |
| SETDEFAULTSET | Set | Changes the default volume set. |
| SHOWDEFAULTSET | Set | Shows the default volume set. |
| SHOWSET | Set | Displays volume set information. |
| EXPANDCLASS | Class | Adds a volume to an existing volume class. |
| NEWCLASS | Class | Creates a new volume class. |
| SHOWCLASS | Class | Displays volume class information. |
| DO | Misc. | Reexecutes commands from the command history stack. |
| EXIT | Misc. | Exits VOLUTIL. |
| HELP | Misc. | Displays information about VOLUTIL commands online. |
| LISTREDO | Misc. | Lists commands in the command history stack. |
| LOG | Misc. | Sends user input and output to a log file. |
| RECOVER | Misc. | Copies files, from tape, that have been previously saved by the DISCUTIL utility. |
| REDO | Misc. | Reexecutes a command from the command history stack. |
| SHOWUSAGE | Misc. | Displays contiguous workspace that can be reallocated to make free space. |
| USE | Misc. | Processes VOLUTIL commands in an ASCII file. |

# ALTERVOL

The ALTERVOL command changes the permanent and transient disk space allocation for a particular volume. Permanent disk space is reserved for files, the label table, and free space map. The default value for permanent storage is 100%. Transient storage on system volumes is used for stacks and other temporary operating structures. For LDEV 1 the system master volume, the default value for transient space is 75% and the default value for permanent storage space is 75%.

The volume to be altered must be in the MASTER or MEMBER state.

**Note** When updating a new version of MPE XL, it is recommended that all volumes except LDEV 1 have permanent space set to 100% and transient space set to 100%.

## Task

Sets permanent and transient disk storage space.

## Capability

CV, create mountable volume set

## Syntax

```
ALTERVOL [VNAME=]sname:vname
         [PERM=]percentperm [TRANS=]percenttrans
```

## Parameters

*sname* The volume set that contains the volume to be modified. The master volume of this set must be mounted in the MASTER state. **Required.**

*vname* The volume to be modified. The volume must already be initialized. **Required.**

*percentperm* A number between 0 and 100 specifying the new maximum percentage of disk space that can be allocated as permanent space on system and nonsystem volumes. **Required.**

*percenttrans* A number between 0 and 100 specifying the new maximum percentage of disk space that can be allocated as transient space on system volumes only. **Required.**

## Example

This example shows how to change disk space allocation.

1. Set the maximum allocation of 100% for permanent and transient storage space.

```
volutil: ALTERVOL SAMPLE_SET:SAMPLE_VOL 100 100

Verify: Set maximum permanent to 100[Y/N]? Y

Verify: Set maximum transient to 100[Y/N]? Y
```

2. Use the DISCFREE utility to display disk space allocations. Refer to the *MPE/iX Utilities Manual* (32650-90081) for more information on this utility.

# COPYSET

The COPYSET command copies one or more members of a volume set to another volume set. The master must be copied first, because it contains information that must be placed on the member volumes. The source volumes must be mounted in the LONER state, and the destination volumes must be mounted in the SCRATCH or UNKNOWN state.

Allows the copying of an entire volume set. Additionally, it enforces the integrity of the target and source volume sets to prevent intermixing of original and copied volumes.

## Task

Copies the contents of one volume set to another volume set.

## Capability

CV, create mountable volume set

## Syntax

```
volutil: COPYSET [SNAME=]sname [FROM=]from_ldev
                 [TO]=to_ldev [GEN]=gen_number
```

## Parameters

| | |
|---|---|
| *sname* | The volume set that will be copied. The master volume of the set must be copied first. **Required**. |
| *from_ldev* | A number from 1 to 100 specifying the logical device number of the destination volume that must be configured into the device class *DISC* and have a volume mounted in the LONER state. Use the DSTAT command to determine the state of the volume. **Required**. |
| *to_ldev* | Must identify a volume mounted in the SCRATCH or UNKNOWN state. **Required**. |
| *gen_number* | A number from 1 to 32767 specifying the new generation number of the copied volume set. **Optional**. If it is omitted, the generation number will be one greater than the original volume set. |

## Example

This example shows how to copy one member of a volume set to another.

1. Place the master of the volume set you want to copy in the LONER state by using the VSCLOSE system command. If the master volume is not online when you mount the member volumes, the member volumes will mount in the LONER state.
2. Use the DSTAT command to verify that the source volumes are mounted in the LONER, and the destination volumes are mounted in the SCRATCH or UNKNOWN state.

3. Copy the master to the other volume set.

```
volutil: :VSCLOSE BSET

volutil: :DSTAT

LDEV-TYPE  STATUS  VOLUME (VOLUME SET - GEN)
---------  ------  -------------------------
21-079330  LONER   BVOL1    (BSET-0)
22-079330  LONER   BVOL2    (BSET-0)
23-079350  LONER   BVOL6    (BSET-0)
41-079350  SCRATCH
42-079350  SCRATCH
43-079350  SCRATCH

volutil: COPYSET SNAME=BSET FROM=21 TO=41
```

After the volume has been copied, both volumes (the source and the target) would be LONER volumes.

```
volutil: :DSTAT

LDEV-TYPE  STATUS  VOLUME (VOLUME SET - GEN)
---------  ------  -------------------------
21-079330  LONER   BVOL1    (BSET-0)
22-079330  LONER   BVOL2    (BSET-0)
23-079350  LONER   BVOL6    (BSET-0)
41-079350  LONER   BVOL1    (BSET-0)
42-079350  SCRATCH
43-079350  SCRATCH
```

# COPYVOL

The COPYVOL command copies the contents of one volume to a second volume. This is a bit-wise copy of the source volume. The source must be mounted in the SCRATCH, UNKNOWN, or LONER state. The destination volume must be mounted in the SCRATCH or UNKNOWN state. Both volumes must have the same physical characteristics. Use the DSTAT command to determine the state of any volume.

## Task

Copies the contents of one volume to another volume.

## Capability

CV, create mountable volume set

## Syntax

COPYVOL [FROM=] *from_ldev* [TO=] *to_ldev*

## Parameters

*from_ldev* A number from 1 to 100 specifying the logical device number of the source volume. **Required.**

*to_ldev* A number from 1 to 100 specifying the logical device number of the destination volume. **Required.**

## Example

This example shows how to copy data from one volume to another volume.

1. Use the VSCLOSE command to place the source volume in the LONER state (this is only necessary if the volume is mounted). The destination volume must be mounted in the SCRATCH or UNKNOWN state. Both volumes must have the same physical characteristics.
2. Use the COPYVOL command to copy the volume.

```
volutil: :VSCLOSE BSET

volutil: COPYVOL FROM=22 TO=42

volutil: :DSTAT

LDEV-TYPE  STATUS  VOLUME (VOLUME SET - GEN)
---------  ------  -------------------------
21-079330  LONER    BVOL1    (BSET-0)
22-079330  LONER    BVOL2    (BSET-0)
23-079350  LONER    BVOL6    (BSET-0)
41-079330  LONER    BVOL1    (BSET-2)
42-079350  LONER    BVOL2    (BSET-2)
```

3. Use the DSTAT command to verify that the volume was copied.

**Caution** Mixing copied volumes with original volumes could have disastrous effects. For example, if you make a backup of a volume set, then continue to use the original volume set, file and accounting information on the original may no longer correspond to the copy. Any attempt to integrate a duplicate volume into the original volume set would be disastrous.

# DSECTORSVOL

The DSECTORSVOL command finds defective disk sectors and lists them in the bad sector table. Sectors in the bad sector table will be recovered at system startup or whenever a volume is mounted. Execute this command when you get read and write errors on the volume.

**Caution** Use this command when there is no system activity and when the only user is the system manager.

## Task

Finds defective sectors on a disk.

## Capability

SM, system manager

## Syntax

DSECTORSVOL [VOL=] { *ldev* | *[sname:]vname* }

## Parameters

*ldev* A number from 1 to 100 specifying the logical device number of the volume that will be processed. **Optional**.

*sname* The volume set containing the volume to be processed. If omitted, the current default volume set is used. **Optional.**

*vname* The volume to be processed. The volume must already be initialized. **Optional.**

## Example

This example shows how to recover defective disk sectors.

1. Use the DSTAT ALL command to verify that the volumes are online in the MASTER or MEMBER state.
2. Use the DSECTORSVOL command to check MEMBER1 for files that span defective sectors.

```
volutil: :DSTAT ALL

LDEV-TYPE  STATUS  VOLUME (VOLUME SET - GEN)
---------  ------  -------------------------
 1-079350  MASTER  MEMBER1  (MPEXL_SYSTEM_VOLUME_SET-0)
 2-079350  MEMBER  MEMBER2  (MPEXL_SYSTEM_VOLUME_SET-0)
21-079330  MASTER  BVOL1    (BSET-0)
22-079330  MEMBER  BVOL2    (BSET-0)
23-079350  MEMBER  BVOL6    (BSET-0)

volutil: DSECTORSVOL MPEXL_SYSTEM_VOLUME_SET:MEMBER1
```

# DO

The DO command reexecutes a command from the command history stack. Refer to the LISTREDO and REDO commands in this chapter.

## Task

Reexecutes a command from the command history stack.

## Capability

No special capability required.

## Syntax

```
DO [[CMD=]cmdid]
```

## Parameters

*cmdid* Identifies a particular command in the command history stack. It can be one of the following:

- A relative command number (executes the *n*th number in the stack, starting with the most recent).
- An absolute number (executes number *n* in the stack).
- A string (executes the most recent command starting with string).
- Omitted (executes last command).

**Optional.**

## Example

This example shows how to reexecute a command from the command history stack using the LISTREDO command. The example on the following pag

1. List the commands in the command history stack using the LISTREDO command.
2. Reexecute command 30.

```
volutil: LISTREDO

    24) LIST REDO
    25) LISTREDO
    26) REDO 24
    27) LIST REDO
    28) LISTREDO
    29) LIST REDO
    30) HELP
    31) HELP RECOVER
    32) HELP LISTREDO
    33) LISTREDO

volutil: DO 30
```

# EXIT

The `EXIT` command terminates VOLUTIL and returns to the system prompt.

## Task

Exits VOLUTIL.

## Capability

No special capability required.

## Syntax

```
EXIT
```

## Example

This example shows how to exit VOLUTIL.

```
volutil: EXIT

:
```

# EXPANDCLASS

The EXPANDCLASS command allows you to add additional volumes to an existing volume class.

## Task

Adds volume(s) to an existing volume class.

## Capability

CV, create mountable volume set

## Syntax

```
EXPANDCLASS [CNAME=]sname:cname
            [VOLUMES=](vname[[,vname]...])]
```

## Parameters

| | |
|---|---|
| *sname* | The volume set containing the volume class to be expanded. The master volume of this set must be mounted in the MASTER state. **Required**. |
| *cname* | The existing volume class to be expanded. **Required**. |
| *vname* | The volume to be assigned to the existing class. The volume must have been previously defined with the NEWVOL command. **Required**. |

## Example

This example shows how to add volumes to an existing volume class.

1. Use the SHOWCLASS command to get information about the volumes in the volume class, ENGR.
2. Add BVOL2 and BVOL3 to the class.

```
volutil:SHOWCLASS BSET:ENGR INFO=DSTATUS

Volume name:  State:     ldev:     Type:      Path:
- - - - - -   - - -     - - -     - - -      - - -
BVOL1         MASTER     21        079330     8.0.4

volutil: EXPANDCLASS BSET:ENGR VOLUMES=(BVOL2,BVOL3)

Verify: Expand volume class BSET:ENGR [Y/N]? Y
```

3. Verify the volumes that have been added. BVOL3 is not physically mounted, so it is shown as a member of the class that is not available.

```
volutil: SHOWCLASS BSET:ENGR DSTATUS

Volume name:   State:    ldev:     Type:     Path:
- - - - - -    - - -     - - -     - - -     - - -
BVOL1          MASTER    21        079330    8.0.4
BVOL2          MEMBER    22        079330    8.0.5
BVOL3        (VOLUME NOT AVAILABLE)
```

# FORMATVOL

The FORMATVOL command formats a disk pack (volume) to get it ready for initialization with the NEWSET, NEWVOL, or INITVOL commands . This command is needed only on rare occasions when an irrecoverable disk error has been detected. After it is formatted, the volume is in the UNKNOWN state.

**Caution** FORMATVOL destroys all data on the specified volume.

## Task

Formats a volume.

## Capability

CV, create mountable volume set

## Syntax

FORMATVOL [LDEV=] *ldev*

## Parameters

*ldev* A number from 1 to 100 specifying the logical device number of the volume to be formatted. **Required**.

## Example

This example shows how to format a volume.

1. Use the DSTAT command to check the status of the volume you want to format.

```
volutil: :DSTAT

LDEV-TYPE  STATUS   VOLUME (VOLUME SET - GEN)
---------  ------   -------------------------
21-079350  MASTER   BVOL1  (BSET-0)
22-079350  MEMBER   BVOL2  (BSET-0)
23-07950   MEMBER   BVOL5  (BSET-0)
```

2. Use the VSCLOSE command to place the volume set in the LONER state. The VSCLOSE command is described in this chapter. Then use the DSTAT command to display the states of all the volumes.

```
volutil: :VSCLOSE BSET
volutil: :DSTAT

LDEV-TYPE  STATUS   VOLUME (VOLUME SET - GEN)
---------  -------  --------------------------
21-079330  LONER   BVOL1   (BSET-0)
22-079350  LONER   BVOL2   (BSET-0)
23-079350  LONER   BVOL5   (BSET-0)
```

3. Now that you have verified that the volume set is in the LONER state, you can use the SCRATCHVOL command to place the volume in the SCRATCH state. Refer to the SCRATCHVOL command for more information.
4. Use the FORMATVOL command to format the volume on LDEV 23. After you format the volume, the volume will be in the UNKNOWN state. Again use the DSTAT command to verify the state of the volume.

```
volutil: SCRATCHVOL 23

*Verify: Scratch volume on ldev 23 [Y/N] ? Y
*Note: Loner volume scratched on ldev 23.

volutil: FORMATVOL 23

Verify: Format volume on ldev 23 [Y/N] ? Y
Note: Volume has been successfully formatted on ldev 23

volutil: :DSTAT

LDEV-TYPE  STATUS   VOLUME (VOLUME SET - GEN)
---------  ------   -------------------------
21-079330  LONER   BVOL1   (BSET-0)
22-079350  LONER   BVOL2   (BSET-0)
23-079350  UNKNOWN
```

## HELP

The HELP command provides online information about the VOLUTIL commands. When used by itself, the command displays a list of available commands and a brief syntax of each command. When HELP is used with a command name (for example, HELP SHOWSET), HELP displays detailed information about the particular command, along with a detailed syntax diagram for the command and an example of its use.

### Task

Displays online command information.

### Capability

No special capability required.

### Syntax

HELP [ [COMMAND=] *cmdname*]

### Parameters

*cmdname* The VOLUTIL command name for which help is desired. If omitted, a list of all commands is displayed. **Optional**.

### Example

This example shows how to display command information online.

```
volutil:HELP

NEWSET      sname vname ldev [perm][trans][gen_number]
            [(cname[[,cname]...])]
SHOWSET     [sname] [ CLASSES | VOLUMES | SETINFO | DSTATUS |
                            STORAGE [;FREE] [;PERM] [;TRANS] |
                            LABELS [;MPEXL] | STRUCT ]
COPYSET         sname from_ldev to_ldev
SETDEFAULTSET   [sname]
SHOWDEFAULTSET
NEWCLASS        sname:cname [(vname [[,vname]...])]
EXPANDCLASS     sname:cname (vname [[,vname]...])
SHOWCLASS       [sname:]cname [ VOLUMES | SETS | CLASSINFO | DSTATUS |
                                STORAGE [;FREE] [;PERM] [;TRANS] |
                                LABELS [;MPEXL] ]


NEWVOL          sname:vname ldev [perm] [trans]
                     [(cname[[,cname]...])]
INITVOL         sname:vname ldev
ALTERVOL        sname:vname [perm] [trans]
SHOWVOL         [sname:]vname [ SETS | CLASSES | VOLINFO | DSTATUS |
                                STORAGE [;FREE] [;PERM] [;TRANS] |
                                LABEL [;MPEXL] ]
```

```
FORMATVOL       ldev
SCRATCHVOL      ldev
UNSCRATCHVOL    ldev
DSECTORSVOL     [sname:]vname | ldev
VERIFYVOL       [sname:]vname | ldev
COPYVOL         from_ldev to_ldev


HELP            [command_name]
USE             filename
LOG             filename
REDO            [cmd_id]
DO              [cmd_id]
LISTREDO
RECOVER         [;KEEP | ;NOKEEP]
EXIT
```

This example shows how to use the HELP command to find out about the SHOWSET command.

```
volutil: HELP SHOWSET

Description:

The SHOWSET command is used to display information about a
particular volume set. The master volume of the volume set
must be mounted in the MASTER state as displayed by the
DSTAT command.

Syntax:

                                        [CLASSES]
                                        [VOLUMES]
                                        [SETINFO]
volutil: SHOWSET [SNAME=]sname [INFO=]  [DSTATUS]
                                        [STORAGE] [;FREE][;PERM][;TRANS]
                                        [LABELS]  [;MPEXL]
                                        [STRUCT]

Example:


volutil: SHOWSET sample_set volumes
```

# INITVOL

The INITVOL command initializes a volume that was previously defined by the NEWVOL command.

If a volume was not mounted when the NEWVOL command was used without the *ldev* parameter, it was only defined. When a volume is defined only, the volume name, initial class assignments, and permanent and transient space allocations are recorded in the volume set information table (VSIT) of the volume set master.

You can then initialize a volume with the INITVOL command and create a volume label, label table, and free space map.

This command is used when it is necessary to create a volume set with more volumes than you have available disks on the system. When the other volumes become available, they need only be initialized.

## Task

Initializes a volume that was defined by NEWVOL.

## Capability

CV, create mountable volume set

## Syntax

```
INITVOL [VNAME=]sname:vname
        [LDEV=]ldev
```

## Parameters

*sname* The volume set containing the volume to be initialized. The master volume of this set must be mounted in the MASTER state. This command will not use the default volume set. **Required**.

*vname* The volume to be initialized. This volume must already be defined by a NEWVOL command. **Required**.

*ldev* A number from 1 to 100 specifying the logical device number of the volume that will be initialized. It must specify a drive with a volume in the SCRATCH or UNKNOWN state. INITVOL asks for verification before proceeding. **Required**.

## Example

This example shows how to initialize a volume that the NEWVOL command previously defined.

1. Use the INITVOL command to initialize BVOL6.
2. Use SHOWSET to display volume status.
3. Use DSTAT ALL to display that the volume is available.

```
volutil: INITVOL BSET:BVOL6 LDEV=23

Verify: Initialize new member volume BSET:BVOL6 on ldev 23 [Y/N] ?Y
Note: New member volume has been initialized on ldev 23.

volutil: SHOWSET BSET DSTATUS

Volume name:               State:   Ldev:      Type:     Path:
-----------                -----    ----       ----      ----

  BVOL1                   MASTER      21     079330     8.0.4
  BVOL2                   MEMBER      22     079350     8.0.5
  BVOL3           (VOLUME NOT AVAILABLE)
  BVOL4           (VOLUME NOT AVAILABLE)
  BVOL5           (VOLUME NOT AVAILABLE)
  BVOL6                   MEMBER      23     079350     8.0.6

volutil: :DSTAT ALL

 LDEV-TYPE    STATUS      VOLUME (VOLUME SET - GEN)
-----------  ---------   ---------------------------
 1-079350     MASTER      MEMBER1         (MPEXL_SYSTEM_VOLUME_SET-0)
 2-079350     MEMBER      MEMBER2         (MPEXL_SYSTEM_VOLUME_SET-0)
21-079330     MASTER      BVOL1           (BSET-0)
22-079350     MEMBER      BVOL2           (BSET-0)
23-079350     MEMBER      BVOL6           (BSET-0)
```

# LISTREDO

The LISTREDO command displays the command history stack. The output order of the stack is from the oldest to the most recent command with command reference numbers preceding each command. The command reference number is used with the DO and REDO commands.

## Task

Displays the command history stack.

## Capability

No special capability required.

## Syntax

```
LISTREDO
```

## Example

This example shows how to list the command history stack by using the LISTREDO command.

```
volutil: LISTREDO

    24) LIST REDO
    25) LISTREDO
    26) REDO 24
    27) LIST REDO
    28) LISTREDO
    29) LIST REDO
    30) HELP
    31) HELP RECOVER
    32) HELP LISTREDO
    33) LISTREDO

volutil:
```

# LOG

The LOG command allows the user to save in a file the dialog between the user and VOLUTIL that is displayed on the screen.

## Task

Saves screen input and output to a file.

## Capability

No special capability required.

## Syntax

LOG [FILENAME=]*filename*

## Parameters

*filename* Any valid file that has read and write access. If the file does not exist, it is created. **Required**.

## Example

This example shows how to enable logging.

1. Use the LOG command to start logging all input and output to the file, logfilea.
2. Use the SHOWSET command to display all volumes in the set.

```
volutil: LOG logfilea

volutil: SHOWSET BSET VOLUMES

Volume name:              Index:
-----------               -----
  BVOL1                     1
  BVOL2                     2
  BVOL3                     3
  BVOL4                     4
  BVOL5                     5
  BVOL6                     6
```

3. Use the `LOG $STDLIST` command to disable logging and close the log file.
4. You can now display the contents of `logfilea` by using the system `PRINT` command. If you log to the same file again, you will overwrite its contents.

```
volutil: LOG $STDLIST

volutil: :PRINT logfilea
```

# NEWCLASS

The NEWCLASS command adds a volume class to a volume set and optionally adds volumes to that class. You can create a class without assigning any volumes to it. However, if you wish to assign volumes, those volumes must be already defined. Refer to the NEWVOL command. You can assign additional volumes to the class with the EXPANDCLASS command.

## Task

Creates a new volume class.

## Capability

CV, create mountable volume set

## Syntax

NEWCLASS[CNAME=]*sname:cname*
[VOLUMES=][(*vname*[[,*vname*]...])]

## Parameters

| | |
|---|---|
| *sname* | The volume set to contain the new class. The master volume of this set must be mounted in the MASTER state. **Required**. |
| *cname* | The new class. The name must be unique within the volume set. **Required**. |
| *vname* | The names of the volumes to be assigned to the new class. The volumes must have been previously defined with the NEWVOL command. **Optional**. |

## Example

This example shows how to create a new volume class and assign a volume to the class.

1. Create the new volume class, ENGR.
2. Verify that the class has been created by using the SHOWCLASS command with different parameters.

```
volutil: NEWCLASS CNAME=BSET:ENGR VOLUMES=(BVOL1)

Verify: Create new volume class BSET ENGR [Y/N]? Y

volutil: SHOWCLASS BSET:ENGR

Volume class index: 2
Number of volumes in class: 1
```

```
volutil: SHOWCLASS CNAME=BSET:ENGR INFO=VOLUMES

Volume name:   Index:
- - - - - -    - - -
BVOL1          1

volutil: SHOWCLASS BSET:ENGR CLASSINFO

Volume class index: 2
Number of volumes in class: 1

volutil: SHOWCLASS BSET:ENGR DSTATUS

Volume name:   State:    ldev:     Type:    Path:
- - - - - -    - - -    - - -    - - -    - - -
BVOL1          MASTER     21      079330   8.0.4
```

# NEWSET

The NEWSET command creates a new volume set by initializing the master of the volume set. Nonsystem volumes have only permanent space and should be set to 100 for *percentperm*.

The volume master must be mounted in the SCRATCH or UNKNOWN state in order for the NEWSET command to initialize a new volume set.

## Task

Creates a new volume set.

## Capability

CV, create mountable volume set

## Syntax

```
NEWSET [SNAME=]sname
       [MASTER=]master
       [LDEV=]ldev
       [[PERM=]percentperm] [[TRANS=]percenttrans]
       [[GEN=]gen_number]
       [[CLASSES=](cname[[,,cname]...])]
```

## Parameters

*sname* — The name assigned to the new volume set. It is used to reference and identify the set, so it must be unique to the system. **Required**.

*master* — The name assigned to the master volume of the set. The name assigned to the master volume need not be the same as the name of the volume set. **Required**.

*ldev* — A number from 1 to 100 specifying the logical device on which the master volume being created is mounted. It must specify a drive with a volume in the SCRATCH or UNKNOWN state. **Required**.

*percentperm* — A number between 0 and 100 specifying the maximum percentage of disk space that can be allocated as permanent. **Optional**.

*percenttrans* — For system volumes only, a number between 0 and 100 specifying the maximum percentage of disk space that can be allocated as transient. **Optional**.

*gen_number* — A number from 0 to 32767 specifying the generation of the new volume. The default is 0. **Optional**.

*cname* — The names of the volume classes to be initially created in the volume set. The master volume (volume being initialized) is assigned to these classes. If this parameter is omitted, the volume class DISC is created, and the master volume is assigned to it. **Optional**.

## Example

This example shows how to create a new volume set from a volume in either the UNKNOWN or SCRATCH state.

1. Use the DSTAT command to check the states of the volumes in the volume set.
2. Use the NEWSET command to create a new volume set, BSET.
3. Use the DSTAT command to verify that the new set was created.

```
volutil: :DSTAT

LDEV-TYPE  STATUS   VOLUME (VOLUME SET - GEN)
- - - - -  - - - -  - - - - - - - - - - - - -
21-079330  UNKNOWN
22-079350  SCRATCH
23-079350  SCRATCH

volutil: NEWSET SNAME=BSET MASTER=BVOL1 LDEV=21 PERM=100

*Verify: Initialize new volume set BSET:BVOL1 on ldev 21 [Y/N]? Y
*Note: New master volume has been initialized on ldev 21.

volutil: :DSTAT

LDEV-TYPE   STATUS    VOLUME (VOLUME SET - GEN)
- - - - -   - - - -   - - - - - - - - - - - - -
21-079330   MASTER    BVOL1     (BSET - 0)
22-079350   SCRATCH
23-079350   SCRATCH
```

4. Use the SHOWSET commands to display volume set information.

```
volutil: SHOWSET SNAME=BSET

Volume - set name: BSET
Creation date: THU, OCT 15, 1987, 4:48 PM
Generation number: 1
Number of volumes in set: 1
Number of classes in set: 1
Volume-class name:                Index:
- - - - - - - - -                - - -
DISC                                1

volutil: SHOWSET SNAME=BSET INFO=VOLUMES
    Volume name:                  Index:
    - - - - - -                   - - -
    BVOL1                           1

volutil: SHOWSET SNAME=BSET INFO=SETINFO

Volume - set name: BSET
Creation date: THU, OCT 15, 1987, 4:48 PM
Generation number: 1
Number of volumes in set: 1
Number of classes in set: 1

volutil: SHOWSET SNAME=BSET INFO=DSTATUS

Volume name:          State:     ldev:      Type:      Path:
- - - - - -          - - -      - - -      - - -      - - -
BVOL1                MASTER      21        079330     8.0.4


volutil: SHOWSET SNAME=BSET INFO=STORAGE

Volume name:      Free:    Permanent:    Transient:   Device
- - - - - -       - - -    - - - - -     - - - - -    - - - -
BOLV1            1178032     401872           0         15

Total:           1178032     401872           0         15
```

```
volutil: SHOWSET SNAME=BSET INFO=LABELS

Volume name: BSET:BVOL1
Initialization date: THU, OCT 15, 1987, 4:48 PM   Volume type: 2
Member number: 1    Number in set: 1
Label Table Address: $00000500          MVT Address: $00000000
Free Space Map Address: $00000070       Cold Load ID: $00000000
Logical Volume ID: $05570001       Physical Volume ID: $05570001


volutil: SHOWSET SNAME=BSET INFO=STRUCT

Volumes in set: BSET
BVOL1
Classes in set: BSET
DISC
Volumes in class: BSET:DISC
BVOL1
```

# NEWVOL

The NEWVOL command adds a new member volume to an existing volume set. If you include the *ldev* parameter, then the command initializes the new volume. That is, the command can define a volume in the set without the volume being present online. Defining a volume means that an entry identifying the volume is put on the volume set information table (VSIT) on the master volume, but the volume will not be initialized or available to the system.

The NEWVOL and INITVOL commands work together. If the volumes are not online, you can define the volumes with the NEWVOL command by not including the *ldev* parameter. Later, when the volume is available and online, you can use the INITVOL command to initialize the volume.

## Task

Adds a new volume to a volume set.

## Capability

CV, create mountable volume set

## Syntax

```
NEWVOL [VNAME=]sname:vname [ [LDEV=]ldev
                             [PERM=]percentperm
                             [TRANS=]percenttrans
                             [CLASSES=](cname[[,,cname]...]) ]
```

## Parameters

*sname* — The name of the volume set where the new volume will reside. **Required**.

*vname* — The name assigned to the new volume. The name must be unique. **Required**.

*ldev* — A number from 1 to 100 specifying the logical device number of the volume being created. It must specify a drive with a volume in the SCRATCH or UNKNOWN state. **Optional**.

**Note**

If the *ldev* parameter is omitted, the volume is defined only; that is, an entry is placed in the volume set information table (VSIT), but the volume is not initialized. You must then use INITVOL to initialize the volume. (Refer to the INITVOL command in this chapter.)

*percentperm* — A number between 0 and 100 specifying the maximum percentage of disk space that can be allocated as permanent storage. **Optional**.

*percenttrans* — For system volumes only, a number between 0 and 100 specifying the maximum percentage of disk space that can be allocated as transient storage. **Optional**.

*cname* The class assigned to the new volume. The class must have been created previously with the NEWCLASS or NEWSET command. If this parameter is omitted, the volume will be assigned to the volume class DISC. **Optional**.

## Example

This example shows how to add a volume to an existing volume set.

1. Before attempting to create a new volume, use the DSTAT ALL command to make certain the master volume of the volume set is in the MASTER state.
2. Next, use the SHOWSET command with the STRUCT parameter to determine the structure of the volume set, BSET. The STRUCT parameter shows that there are five volumes in volume set BSET and two classes, XL_DATA and the default class, DISC. It also shows the volumes in the classes.

```
volutil: :DSTAT ALL

LDEV-TYPE  STATUS   VOLUME (VOLUME SET - GEN)
---------  -------  -------------------------
1-079350   MASTER   MEMBER1  (MPEXL_SYSTEM_VOLUME_SET-0
2-079350   MEMBER   MEMBER2  (MPEXL_SYSTEM_VOLUME_SET-0
21-079330  MASTER   BVOL1    (BSET-0)
22-079350  MEMBER   BVOL2    (BSET-0)
23-079350  UNKNOWN

volutil: SHOWSET BSET STRUCT

     Volumes in set: BSET

      BVOL1
      BVOL2
      BVOL3
      BVOL4
      BVOL5

      Classes in set: BSET

      XL_DATA
      DISC
```

```
        Volumes in class: BSET:XL_DATA

             BVOL1
             BVOL3

        Volumes in class: BSET:DISC

             BVOL1
             BVOL2
             BVOL3
             BVOL4
             BVOL5
```

3. Now add another volume to the set. For this example, the volume is not physically mounted, so it cannot be initialized. Since a class is not specified, the volumes are automatically added to the `DISC` class.
4. Use `SHOWSET STRUCT` to verify that the volume has been added.

```
volutil: NEWVOL BSET:BVOL6

 *Verify: Define new member volume BSET:BVOL6 [Y/N]? Y

 *Warning: Volume only defined and not physically
           initialized.

volutil: SHOWSET BSET STRUCT


     Volumes in set: BSET

     BVOL1
     BVOL2
     BVOL3
     BVOL4
     BVOL5
    *BVOL6
```

```
Classes in set: BSET

  XL_DATA
  DISC

Volumes in class: BSET:XL_DATA

  BVOL1
  BVOL3

Volumes in class: BSET:DISC

  BVOL1
  BVOL2
  BVOL3
  BVOL4
  BVOL5
 *BVOL6
```

**Note** The * indicates that the volume has been defined but not initialized. Refer to the INITVOL command in this chapter.

# RECOVER

The RECOVER command recreates files from the output produced by DISCUTIL. Refer to chapter 4 for more information on recovering data.

## Task

Copies files from tape that have been saved by the DISCUTIL utility.

## Capability

SM, system manager

## Syntax

```
RECOVER [;KEEP]
        [;NOKEEP]
```

## Parameters

| | |
|---|---|
| KEEP | Files on the disk are not replaced by files of the same name on the tape. KEEP is the default. **Optional.** |
| NOKEEP | Replaces files on the disk with files of the same name on the tape. **Optional.** |

## Example

This example shows how to copy disk files from tape.

1. Load the tape.
2. Use the RECOVER command to start the copy. As each file is copied to the disk, messages are displayed.
3. The system prompts you to mount the next tape if the files continue onto additional tapes.

```
volutil: RECOVER;NOKEEP

(fname)   (group)   (acct)       IS RESTORED
TEXT      .PROD     .MANUF       IS RESTORED
INTRO     .PROD     .MANUF       IS RESTORED
VOLM2     .PROD     .MANUF       IS RESTORED

END OF VOLUME, RECOVER DONE
```

# REDO

The REDO command allows you to edit a command from the command history stack and then reexecute the command.

## Task

Allows editing and reexecution of a command from the command history stack.

## Capability

No special capability required.

## Syntax

REDO [[CMD=]*cmdid*]

## Parameters

*cmdid* Identifies a particular command in the command history stack. It can be one of the following:

- A relative command number (executes the *n*th number in the stack, starting with the most recent).
- An absolute number (executes number *n* in the stack).
- A string (executes the most recent command starting with string).
- Omitted (executes last command).

**Optional.**

## Example

This example shows how to reexecute a command from the command history stack.

1. List the commands in the command history stack using the LISTREDO command.
2. Reexecute command 30.

```
volutil: LISTREDO
    24) LIST REDO
    25) LISTREDO
    26) REDO 24
    27) LIST REDO
    28) LISTREDO
    29) LIST REDO
    30) HELP
    31) HELP RECOVER
    32) HELP LISTREDO
    33) LISTREDO

volutil: REDO 30

volutil: HELP
```

# SCRATCHVOL

The SCRATCHVOL command places a volume in the SCRATCH state. Although the data remains on the volume after the command scratches (erases) the volume, the user can consider the data deleted, unless the UNSCRATCHVOL command is issued to undo this command. The command is typically used to reinitialize a volume. The volume must be in the LONER state in order to be scratched.

## Task

Deletes data on a volume.

## Capability

CV, create mountable volume set

## Syntax

SCRATCHVOL [LDEV=]*ldev*

## Parameters

*ldev* A number from 1 to 100 specifying the logical device number of the volume to be scratched. The volume must be in the LONER state. **Required**.

## Example

This example shows how to put a volume in the SCRATCH state.

1. Use the DSTAT command to check the status of the volume to be scratched.
2. Close the volume set to put the volume in the LONER state.

```
volutil: :DSTAT

LDEV-TYPE  STATUS  VOLUME (VOLUME SET - GEN)
---------  ------- --------------------------
21-079330  MASTER  BVOL1   (BSET-0)
22-079350  MEMBER  BVOL2   (BSET-0)
23-079350  MEMBER  BVOL5   (BSET-0)

volutil: :VSCLOSE BSET
```

3. Use the SCRATCHVOL command to place the volume in the SCRATCH state.
4. Use the DSTAT command to verify that the volume is in the SCRATCH state.

```
volutil: SCRATCHVOL 23
*Verify:  Scratch volume on ldev 23 [Y/N]? Y
*Note:  Loner volume scratched on ldev 23.

volutil:  :DSTAT

LDEV-TYPE  STATUS  VOLUME (VOLUME SET - GEN)
---------  ------- --------------------------
21-079330  LONER   BVOL1   (BSET-0)
22-079350  LONER   BVOL2   (BSET-0)
23-079350  SCRATCH
```

# SETDEFAULTSET

The SETDEFAULTSET command sets the default volume set. The default volume set is normally the system volume set. When a command has the volume set name as a parameter, the user is free to select any available volume set. If the user does not select a volume set, the command selects the default volume set.

## Task

Changes the default volume set.

## Capability

No special capability required.

## Syntax

SETDEFAULTSET [[SNAME=]*sname*]

## Parameters

*sname* The new default volume set. The master volume of this set must be mounted in the MASTER state. If omitted, the system volume set becomes the default volume set. **Optional**.

## Example

This example shows how to set the default volume set.

1. Use the SHOWDEFAULTSET to find out which volume set is the default set.
2. To change the default volume set, use the SETDEFAULTSET command.

```
volutil: SHOWDEFAULTSET

Current default volume set: MPEXL_SYSTEM_VOLUME_SET

volutil: SETDEFAULTSET BSET

Note: Current default volume set changed to BSET.

volutil: SHOWDEFAULTSET

Current default volume set: BSET
```

# SHOWCLASS

The SHOWCLASS command displays information about a volume class. Included in this information is a list of volumes that are members of the volume class and the amount of storage (permanent and transient) available in that volume class. The volume set containing the class must be mounted and online in order to obtain information about the class.

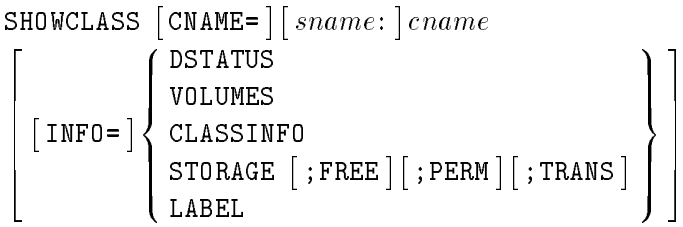
## Task

Displays volume class information.

## Capability

No special capability required.

## Syntax




## Parameters

*sname* — The volume set containing the volume class information to be displayed. The master volume of this set must be mounted in the MASTER state. If omitted, the current default volume set is assumed. **Optional**.

*cname* — The volume class whose information is requested. **Required**.

VOLUMES — Displays the volume name and volume index for each volume assigned to the class. Volume names preceded by an asterisk (*) have been defined only and not initialized. Refer to the NEWVOL and INITVOL commands in this chapter. **Optional**.

CLASSINFO — Displays the volume class index and current number of volumes assigned to the class. This is the default if the INFO keyword parameter is omitted. **Optional**.

DSTATUS — Like the DSTAT command. Displays the volume name, *ldev*, device type, physical path for each volume used by the class, and the mount state. **Optional.**

STORAGE — Displays information about disk storage for the specified volume class. The FREE, PERM, and TRANS options, respectively, show the available free space and the permanent and transient space allocations. All measurements are in sectors. The default options are FREE, PERM, and TRANS. **Optional**.

LABEL — Displays the contents of the volume label for each volume of the class. **Optional**.

## Example

The example on the following page shows how to display class information using the SHOWCLASS command with different parameters.

```
volutil: SHOWCLASS CNAME=BSET:XL_DATA

Volume class index: 1
Number of volumes in class: 2

volutil: SHOWCLASS BSET:XL_DATA INFO=VOLUMES

Volume name:             Index:
-----------              -----
  BVOL1                     1
 *BVOL3                     3

volutil: SHOWCLASS BSET:XL_DATA INFO=CLASSINFO

Volume class index: 1
Number of volumes in class: 2

volutil: SHOWCLASS BSET:XL_DATA INFO=DSTATUS

Volume name:               State:   Ldev:      Type:     Path:
-----------                -----    ----       ----      ----
  BVOL1                   MASTER      21     079330     8.0.4
  BVOL3           (VOLUME NOT AVAILABLE)

volutil: SHOWCLASS BSET:XL_DATA INFO=STORAGE

Volume name:                Free:    Permanent:    Transient:  Device size
-----------                 ----     ---------     ---------   ---------
  BVOL1                  1178176        401728             0       15799

  Total:                 1178176        401728             0       15799

volutil: SHOWCLASS CNAME=BSET:XL_DATA INFO=LABELS

Volume name: BSET:BVOL1
Initialization date: THU, OCT 15, 1987,  4:48 PM      Volume type: 2
Member number: 1     Number in set: 3

Label Table Address: $00000500         MVT Address: $00000000 Free Space
Map Address: $00000070      Cold Load ID: $00000000 Logical Volume ID:
$05570001 4810C1F7  Physical Volume ID: $05570001

Volume name:  BSET:BVOL3 *Warning: Volume not available.
```

# SHOWDEFAULTSET

The SHOWSET command displays the current default volume set.

## Task

Displays the default volume set.

## Capability

No special capability required.

## Syntax

```
SHOWDEFAULTSET
```

## Parameters

No special capability required.

## Example

This example shows how to display the current default set using the SHOWDEFAULTSET command.

```
volutil: SHOWDEFAULTSET

Current default volume set: BSET
```

# SHOWSET

The SHOWSET command displays information about a particular volume set. The command shows all the classes and volumes that are members of the volume set. Also, it displays device, label, and other information about the volume set. Volume names preceded by an asterisk (*) have been defined only and not initialized. Refer to the NEWVOL and INITVOL commands in this chapter.

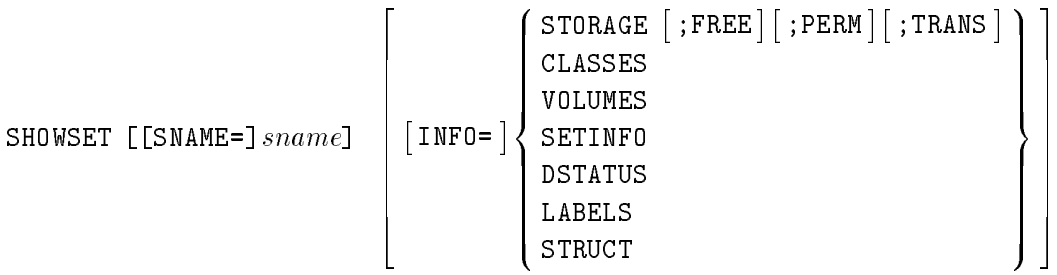
## Task

Displays volume set information.

## Capability

No special capability required.

## Syntax




## Parameters

*sname* — The volume set whose information is requested. The master volume of this set must be mounted in the MASTER state. Default is the current default volume set. **Optional**.

CLASSES — Displays the class name and index for each class in the set. **Optional**.

VOLUMES — Displays the volume name and index for each volume in the set. Volume names preceded by an asterisk (*) have been defined only and not initialized. Refer to the NEWVOL and INITVOL commands in this chapter. **Optional**.

SETINFO — Displays the creation date, generation number, and number of volumes and classes in the set. This is the default if the INFO parameter is omitted. **Optional**.

DSTATUS — Like the DSTAT command. Displays the volume name, *ldev*, device type, physical path for each volume in the volume set that is currently online, and the mount state. **Optional**.

STORAGE — Displays disk storage information for volumes currently available in the specified volume set. The FREE, PERM, and TRANS options respectively show the available free space and the permanent and transient space allocations. All measurements are in sectors. The default options are FREE, PERM, and TRANS. **Optional**.

LABELS — Displays the contents of the volume label for those volumes of the set that are currently mounted and online. **Optional**.

STRUCT — Displays a summary of the current structure or class/volume hierarchy for the specified set. **Optional**.

## Example

This example shows how to display set information using the SHOWSET command with different parameters.

```
volutil: SHOWSET SNAME=BSET INFO=CLASSES

Volume-class name:                        Index:
------------------                        -----

  DISC                                       1

volutil: SHOWSET SNAME=BSET INFO=DSTATUS

Volume name:               State:   Ldev:      Type:     Path:
------------               -----    ----       ----      ----

  BVOL1                   MASTER      21     079330     8.0.4

volutil: SHOWSET SNAME=BSET INFO=STORAGE

Volume name:       Free:  Permanent:    Transient:  Device size:
------------       ----   ---------     ---------   ------------

  BVOL1         1178032      401872             0       1579904

  Total:        1178032      401872             0       1579904

volutil: SHOWSET SNAME=BSET INFO=LABELS

Volume name: BSET:BVOL1
Initialization date: TUE, OCT 20, 1987,  3:02 PM      Volume type:    2
Member number: 1     Number in set: 1
Label Table Address: $00000500         MVT Address: $00000000
Free Space Map Address: $00000070      Cold Load ID: $00000000
Logical Volume ID: $05570001 494F0A24  Physical Volume ID: $0557000   1 494
```

```
volutil: SHOWSET SNAME=BSET INFO=STRUCT

Volumes in set: BSET

  BVOL1

Classes in set: BSET

  DISC

Volumes in class: BSET:DISC

  BVOL1

volutil: SHOWSET BSET INFO=VOLUMES

Volume name:              Index:
------------              -----

  BVOL1                      1
```

# SHOWUSAGE

The SHOWUSAGE command displays disk space usage of a volume.

## Task

Displays usage of a volume.

## Capability

No special capability is required to execute this command.

## Syntax

```
SHOWUSAGE [VOL=][[sname:]{vname | ldev}]  [[CONTIG=]contiguous_sectors]
[[FILE=]{ALL | RESTRICT | NONRESTRICT}]  [[FORMAT=]{DETAIL | SUMMARY}]  [;PERM][;FREE][;TRANS]
[;MAX][;PATH]
```

## Parameters

| | |
|---|---|
| *sname* | The name of the volume set containing the volume for which information is desired. This is an optional parameter. If it is omitted, the current default volume set is assumed. **Optional.** |
| *vname* | The name of the volume for which information is requested. The volume must be mounted and online. Volume must be mounted in the "MASTER" or "MEMBER" state, as displayed by the MPE/iX DSTAT command. Required if *ldev* is omitted. **Optional.** |
| *ldev* | An integer value used to specify the logical device on which the volume to be examined is mounted. Required if *vname* is omitted. **Optional.** |
| *contiguous_sectors* | An integer value used to specify the number of contiguous sectors requested. It requests that the contiguous blocks whose total sectors are approximately equal to the requested sectors to be displayed. **Optional.** |
| ALL, RESTRICT, or NONRESTRICT | Used as a filter of files. The keyword FILE applies to permanent space only. RESTRICT requests that only files that are restricted to this volume be displayed. NONRESTRICT requests that only files that are *not* restricted to this volume be displayed. ALL shows both restricted and nonrestricted files on the volume. The default is ALL. **Optional.** |

**Note** A file may be required to reside on a particular volume. This restriction is imposed upon file creation. Refer to the FOPEN and HPFOPEN intrinsics for more information.

| | |
|---|---|
| DETAIL or SUMMARY | Specifies the output format. DETAIL displays the sector address, size, and what occupies the location. SUMMARY only displays file names which are |

| | |
|---|---|
| | associated with the permanent space. Free and transient space are not shown in this format. The default is `DETAIL`. **Optional.** |
| `PERM`, `FREE`, or `TRANS` | The `PERM`, `FREE`, and `TRANS` option displays information about the volume's current permanent, free, and transient space. `PERM`, `FREE`, and `TRANS` space usage are displayed by default. |
| `MAX` | The `MAX` option is used to control the algorithm that searches for the contiguous space. When used, a contiguous block, whose total sectors has met the requested size, will not be displayed until this block reaches its maximum size. |
| `PATH` | The `PATH` option displays all file names in Hierarchical File System (HFS) syntax. |

## Examples

The following are examples on usage of the SHOWUSAGE command.

1. To display what files are on a LDEV 1 enter the following:

```
volutil: SHOWUSAGE VOL=1 FORMAT=SUMMARY ;PERM

PERM space on LDEV 1:

FILE1.GROUP5.ACCT5
FILE2.GROUP1.ACCT1
FILE1.GROUP1.ACCT1
/dir/subdir1/subdir2/file1
FILE1.GROUP2.ACCT1
FILE2.GROUP2.ACCT1
/ACCT3/GROUP2/d1/f1
FILE3.GROUP2.ACCT4
```

2. To display which files consume the most space on LDEV1, enter the following:

```
volutil:SHOWUSAGE VOL=1 ;PERM

PERM space on LDEV 1:

SECTOR ADDRESS       SIZE       SPACE USED BY
                  (in sectors)
--------------   ------------   -------------

  $00079DD0             672      FILE1.GROUP5.ACCT5
  $000DDD00              16      FILE2.GROUP1.ACCT1
  $000DDD10              16      FILE2.GROUP1.ACCT1
  $000E2E40             208      FILE1.GROUP1.ACCT1
  $000E3F60            1200      FILE2.GROUP1.ACCT1
  $000E4160             464      FILE1.GROUP5.ACCT5
```

   To find out the actual size of a file, add all of the sectors for that file, for example, for FILE2.GROUP1.ACCT1, add:

```
  16
  16
1200
----
1232
```

3. To display all files in the HFS (Hierarchical File System) syntax enter the following:

```
volutil: SHOWUSAGE VOL=1 FORMAT=SUMMARY ;PERM ;PATH

PERM space on LDEV 1:

/ACCT5/GROUPS/FILE1
/ACCT1GROUP1/FILE2
/ACCT1/GROUP1/FILE1
/dir1/subdir1/subdir2/file1
/ACCT1/GROUP2/FILE1
/ACCT1/GROUP2/FILE2
/ACCT3/GROUP2/d1/f1
/ACCT4/GROUP2/FILE3
```

4. To quickly identify the minimum number of files that can be removed from LDEV 1 to free up 60000 sectors of contiguous disk space (for example to perform an `UPDATE`), enter the following:

```
volutil: showusage 1 60000 nonrestrict summary ;perm ;free

*Note: Only PERM space is displayed in SUMMARY format.

CONTIG PERM, FREE space on LDEV 1:

FILE1.GROUP2.ACCT1
FILE1.GROUP1.ACCT1
FILE2.GROUP2.ACCT1
FILE3.GROUP3.ACCT4

Total sectors: 62176

FILE1.GROUP3.ACCT1
/ACCT1/GROUP2/dir1/f2
FILE1.GROUP1.ACCT1
FILE5.GROUP5.ACCT5

Total sectors:  80416
```

5. To display the total sectors provided by the previous example containing the number of sectors used by free space (note that free space is not displayed in the previous example), enter the following:

```
volutil:showusage 1 60000 nonrestrict ;perm ;free

CONTIG PERM, FREE space on LDEV 1;

SECTOR ADDRESS       SIZE        SPACE USED BY
                  (in sectors)
--------------   ------------    -------------

 $00111820             2112      FILE1.GROUP2.ACCT1
 $00112060             3680      FILE1.GROUP1.ACCT1
 $00112EC0            16816      FILE2.GROUP2.ACCT1
 $00117070              864      <free space>
 $001173D0             1200      FILE2.GROUP2.ACCT1
 $00117880             1824      FILE2.GROUP2.ACCT1
 $00117FA0             2048      FILE2.GROUP2.ACCT1
 $001187A0              480      <free space>
 $00118980             4284      FILE1.GROUP1.ACCT1
 $00119AA0             4688      FILE2.GROUP2.ACCT1
 $0011ACF0             5648      FILE3.GROUP2.ACCT4
 $$0011D7E0           13088      FILE3.GROUP2.ACCT4

Total sectors         62176
```

6. To display the maximum amount of contiguous block space enter the following:

```
volutil: SHOWUSAGE BSET:MEMBER2 10000 FORMAT=SUMMARY ;PERM ;MAX

CONTIG PERM space on BSET:MEMBER2:

FILEX.GROUP3.ACCTX
FILE1.GROUP1.ACCTX
/ACCTA/GROUPA/tmp/junk
FILE3.GROUP1.ACCTY
FILEX.GROUPM.ACCTB

Total sectors: 20264

FILEX.GROUPN.ACCTM
```

The following example, where the ;MAX option is not entered, is being presented as a comparison to the previous screen:

```
volutil: SHOWUSAGE BSET:MEMBER2 10000 FORMAT=SUMMARY ;PERM

CONTIG PERM space on BSET:MEMBER2:

FILEX.GROUP3.ACCTX
FILE1.GROUP1.ACCTX
/ACCTA/GROUPA/temp/junk
```

**Examples**

```
FILE3.GROUP1.ACCTY

Total sectors: 12032

FILEX.GROUPN.ACCTM
```

# SHOWVOL

The SHOWVOL command displays information about a volume in a volume set. It displays the classes of which the volume is a member; the available permanent, transient, and free space on the volume; volume label; and other information. The volume set containing the volume must be mounted and online in order for the SHOWVOL command to obtain information about the volume.

## Task

Displays volume information.

## Capability

No special capability required.

## Syntax

```
SHOWVOL [VNAME=][sname:]vname
[        { DSTATUS                                  }  ]
[        { CLASSES                                  }  ]
[ [INFO=]{ VOLINFO                                  }  ]
[        { STORAGE [;FREE][;PERM][;TRANS]           }  ]
[        { LABEL                                    }  ]
```

## Parameters

*sname* — The volume set containing the volume whose information is requested. The master volume of this set must be mounted in the MASTER state.

If omitted, the current default volume set is assumed. **Optional**.

*vname* — The volume whose information is displayed. The volume must be mounted and in the MASTER or MEMBER state. **Required**.

CLASSES — Displays all classes and the volumes that are members of those classes. **Optional**.

VOLINFO — Displays the name, index, and permanent and transient space allotments for the specified volume. This is the default if the INFO parameter is omitted. **Optional**.

DSTATUS — Like the DSTAT command. Displays the volume name, *ldev*, device type, physical path, and mount state. This may require the volume to be mounted. **Optional**.

STORAGE — Displays disk storage information for the specified volume. The FREE, PERM, and TRANS options show the available free space and the permanent and transient space allocation. All measurements are in sectors. The default options are FREE, PERM, and TRANS. **Optional**.

LABEL — Displays the contents of the volume label. This may require the volume to be mounted. **Optional**.

## Example

This example shows how to display volume information using the SHOWVOL command with different parameters.

```
volutil: SHOWVOL MPEXL_SYSTEM_VOLUME_SET:MEMBER1

Volume set index: 1
Maximum permanent space: 100% (1579905 sectors)
Maximum transient space: 100% (1579905 sectors)

volutil: SHOWVOL MEMBER1 INFO=CLASSES

Volume-class name:                      Index:
------------------                      -----
  DISC                                     1

volutil: SHOWVOL BSET:BVOL2 INFO=DSTATUS

Volume name:              State:   Ldev:      Type:     Path:
-----------               -----    ----       ----      ----
  BVOL2                  MEMBER      22     079350     8.0.5

volutil: SHOWVOL BSET:BVOL1 INFO=STORAGE

Volume name:     Free:    Permanent:    Transient:  Device size:
-----------      ----     ---------     ---------   -----------
  BVOL1       1178176        401728             0       1579904

  Total:      1178176        401728             0       1579904

volutil: SHOWVOL VNAME=BSET:BVOL1 INFO=LABEL

Volume name: BSET:BVOL1
Initialization date: THU, OCT 15, 1987, 4:48 PM   Volume type: 2
Member number: 2    Number in set: 3
Label Table Address: $00000500          MVT Address: $00000000
Free Space Map Address: $00000070       Cold Load ID: $00000000
Logical Volume ID: $05570001 4810C1F7
Physical Volume ID: $05570001
```

# UNSCRATCHVOL

The UNSCRATCHVOL command unscratches a volume that was previously scratched and is in the SCRATCH state. The only time you would use this command is when you have unintentionally scratched (erased) a volume that has necessary information on it. This command allows the user to undo the effects of a SCRATCHVOL command.

## Task

Removes a volume from the SCRATCH state.

## Capability

CV, create mountable volume set.

## Syntax

UNSCRATCHVOL [LDEV=]*ldev*

## Parameters

*ldev* A number from 1 to 100 specifying the logical device number of the volume to be unscratched. Only volumes in the SCRATCH state can be unscratched. **Required**.

## Example

This example shows how to use the UNSCRATCHVOL command to undo the SCRATCHVOL command and place a volume in the LONER state.

```
volutil: UNSCRATCHVOL LDEV=23
*Verify: Unscratch volume on ldev 23 [Y/N] ? Y
*Note: Scratched volume has been unscratched on ldev 23

volutil: :DSTAT ALL

LDEV-TYPE   STATUS  VOLUME (VOLUME SET - GEN)
---------   ------- --------------------------
 1-079350   MASTER  MEMBER1 (MPEXL_SYSTEM_VOLUME_SET-0)
 2-079350   MEMBER  MEMBER2 (MPEXL_SYSTEM_VOLUME_SET-0)
21-079330   LONER   BVOL1   (BSET-0)
22-079350   LONER   BVOL2   (BSET-0)
23-079350   LONER   BVOL3   (BSET-0)
```

# USE

The USE command processes VOLUTIL commands from a file of ASCII characters. The USE commands may be nested. File equations and backreferences are allowed.

## Task

Processes VOLUTIL commands in an ASCII command file.

## Capability

No special capability required.

## Syntax

USE [FILENAME=]*filename*

## Parameters

*filename* Any valid file for which you have READ access. **Required**.

## Example

This example shows how to process VOLUTIL commands from a file.

1. Use an editor to create an ASCII file that contains VOLUTIL commands. Name the ASCII file SHOWIT.

```
:EDITOR
HP32201A.07.17 EDIT/3000 MON, DEC  7, 1987,  3:00 PM
d(C) HEWLETT-PACKARD CO. 1985
/  A
1       SHOWSET MPEXL_SYSTEM_VOLUME_SET CLASSES
2       SHOWSET MPEXL_SYSTEM_VOLUME_SET VOLUMES
3       SHOWSET MPEXL_SYSTEM_VOLUME_SET LABELS
4       //
/  K SHOWIT
/  E

END OF SUBSYSTEM
```

2. Now that you have created the ASCII file `showit`, run VOLUTIL.
3. Use the `PRINT` command to examine your file. Do not forget the colon in front of the `PRINT` command, because it is a system command.

```
:VOLUTIL

Volume Utility A.00.00, (C) Hewlett-Packard Co., 1987.

     volutil: :PRINT SHOWIT

     SHOWSET MPEXL_SYSTEM_VOLUME_SET CLASSES
     SHOWSET MPEXL_SYSTEM_VOLUME_SET VOLUMES
     SHOWSET MPEXL_SYSTEM_VOLUME_SET LABELS
```

4. Now enter the `USE` command to process the file. All the commands in the file will execute.

```
 volutil: USE SHOWIT
 volutil: SHOWSET MPEXL_SYSTEM_VOLUME_SET CLASSES

 Volume-class name:                        Index:
 ------------------                        -----

   DISC                                       1

 volutil: SHOWSET MPEXL_SYSTEM_VOLUME_SET VOLUMES

 Volume name:              Index:
 ------------              -----

   MEMBER1                    1
   MEMBER2                    2
  *MEMBER3                    3
  *MEMBER4                    4
```

```
volutil: SHOWSET MPEXL_SYSTEM_VOLUME_SET LABELS

Volume name: MPEXL_SYSTEM_VOLUME_SET:MEMBER1
Initialization date: TUE, OCT 13, 1987, 10:18 AM      Volume type:    0
Member number: 1     Number in set: 2
Label Table Address: $00000570         MVT Address: $00063F33
Free Space Map Address: $000000E0      Cold Load ID: $0000000D
Logical Volume ID: $05570001 478A4AF5  Physical Volume ID: $05570001 478A

Volume name: MPEXL_SYSTEM_VOLUME_SET:MEMBER2
Initialization date: TUE, OCT 13, 1987, 10:18 AM      Volume type:    0
Member number: 1     Number in set: 1
Label Table Address: $00000500         MVT Address: $00063F33
Free Space Map Address: $00000070      Cold Load ID: $00000001
Logical Volume ID: $05570002 478A4AF5  Physical Volume ID: $05570001 478A

Volume name: MPEXL_SYSTEM_VOLUME_SET:MEMBER3
Warning: Volume not available.

Volume name: MPEXL_SYSTEM_VOLUME_SET:MEMBER4
Warning: Volume not available.

volutil: :EOD
volutil: EXIT

:
```

# VERIFYVOL

The VERIFYVOL command verifies that the data on a disk pack can be read. All disk states can be verified.

**Caution** This command should be used only when there is no system activity, because it is a disruptive command and takes the volume offline for about 15 minutes while it verifies the media.

## Task

Verifies that the data on a volume can be read.

## Capability

CV, create mountable volume set

## Syntax

VERIFYVOL [VOL=] { *ldev* | [*sname*:]*vname* }

## Parameters

*ldev* A number from 1 to 100 specifying the logical device number of the volume being verified. **Required**.

*sname* The volume set containing the volume whose information is verified. If omitted, the current default volume set is assumed. **Optional**.

*vname* The volume to be verified. **Required**.

## Example

This example shows how to verify the media on a disk pack using the VERIFYVOL command.

1. Use the DSTAT ALL command to display all mounted volumes.
2. Use the VERIFYVOL command to verify that the information on LDEV 21 can be read.

```
volutil: :DSTAT ALL

LDEV-TYPE  STATUS  VOLUME (VOLUME SET - GEN)
---------  ------- --------------------------
 1-079350  MASTER  MEMBER1 (MPEXL_SYSTEM_VOLUME_SET-0)
 2-079350  MEMBER  MEMBER2 (MPEXL_SYSTEM_VOLUME_SET-0)
21-079330  MASTER  BVOL1   (BSET-0)
22-079350  MEMBER  BVOL2   (BSET-0)
23-079350  MEMBER  BVOL6   (BSET-0)

volutil: VERIFYVOL VOL=BSET:BVOL1

Verify: Verify media on ldev 21 [Y/N] ? Y
Note: Media on ldev 21 passed verification.
```

## System Volume Mounting Commands

The following system commands can be used with volume management to notify the system that you want a volume set to remain physically mounted for a period of time. Refer to the *MPE/iX Commands Reference Manual Volumes 1 and 2* (32650-90003 and 32650-90364) for more information on these commands.

**Table 5-2. System Volume Mounting Commands**

| Command | Task |
|---|---|
| VMOUNT | Enables/disables the MPE XL nonsystem volume facility. |
| VSCLOSE | Closes a specified volume set and takes it offline. |
| VSOPEN | Reopens a volume set closed with VSCLOSE. |
| VSRELEASE | Releases a volume set that was explicitly reserved by the user with VSRESERVE. |
| VSRELEASESYS | Cancels a previously issued VSRESERVESYS command for a specified volume set. |
| VSRESERVE | Reserves a particular volume set online. |
| VSRESERVESYS | Reserves a volume set online system-wide. |
| VSUSER | Lists all users of a currently reserved, nonsystem volume set. |

### MPE XL and MPE V/E Commands

A number of MPE V/E commands have been changed in MPE XL. The XL commands and their corresponding V/E commands are shown in Table 5-3. This section gives detailed descriptions and examples for these commands.

There are two user commands, VSRESERVE and VSRELEASE. The other commands require system manager capability and can only be executed from the console.

**Table 5-3. MPE XL and MPE V/E Commands**

| MPE XL | MPE V/E |
|---|---|
| VMOUNT | VMOUNT |
| VSOPEN | None. |
| VSCLOSE | None. |
| VSRELEASE | DISMOUNT |
| VSRELEASESYS | LDISMOUNT |
| VSRESERVE | MOUNT |
| VSRESERVESYS | LMOUNT |
| VSUSER | VSUSER |

# VMOUNT

The VMOUNT command is used by the operator to enable and disable the mountable (nonsystem) volume facility. At system startup the volume facility is enabled to automatically answer nonsystem volume requests, such as VSRESERVE and VSRESERVESYS. Without the AUTO parameter, the operator has to respond to the requests.

## Task

Enables nonsystem volumes.

## Capability

SM, system manager capability and can only be issued from the console.

## Syntax

```
VMOUNT { ON[,AUTO] } [;ALL]
       { OFF       }
```

## Parameters

| | |
|---|---|
| ON or ON,AUTO | Enables the use of nonsystem volumes. When ON is used without AUTO, the operator must reply to all reserve-related requests. When ON,AUTO is used, the system attempts to satisfy user reserve-related requests without operator intervention. |
| OFF | The operator can set VMOUNT to OFF, which will cause the system to reject all requests to use nonsystem volumes. Users who have already reserved volume sets will not be affected. The command will take effect when the last file is closed. |
| ALL | The ALL parameter displays all reserve-related messages on the system console. **Optional**. |

## Example

The example on the following page shows how to disable and enable the volume set facility.

1. Disable the volume sets by entering the VMOUNT command at the system prompt.

```
:VMOUNT OFF
```

2. Verify that the nonsystem volume sets cannot be accessed by trying to open a file on a nonsystem volume that you know is online. Verify that it cannot be opened.

3. Enable the volume set facility by entering the VMOUNT command.

```
:VMOUNT ON,AUTO
```

4. Verify that the volume sets can be accessed by opening a file on a volume set you know is online. Verify that it can be opened.

# VSCLOSE

The VSCLOSE command is used by the operator to put a volume set in the LONER state. If there are no files open in the volume set, the VSCLOSE command closes the set at that time. If there are files open, the volume set goes into a close-pending (CP) state. That means the set will be closed when the last file is closed. Closing takes the volume set offline and puts the volumes in the LONER state. The volumes can now be removed from the system or deleted and reused.

This command is used by the operator to notify users that a volume set is going to be removed. This command restricts access to the volume set. Command patiently waits until all files are closed on the volume set unless NOW specified then all users of volume set will be aborted and the volume set will be ready for removal.

Any job/session that 1)has not done an explicit RESERVE/MOUNT on the volume set and 2)currently has no files open on the volume set, will be denied access to the volume set.

## Task

Takes volumes offline.

## Capability

SM, system manager capability and can only be issued from the console.

## Syntax

```
VSCLOSE  volsetname  [; NOW]
```

## Parameters

| | |
|---|---|
| *volsetname* | The volume set that will be taken offline. **Required.** |
| NOW | Aborts all users of the volume set immediately. **Optional.** |

## Example

This example shows how to take a volume set offline and put it in the LONER state.

1. Try to take the volume set offline by using the VSCLOSE command when there is an open file or a VSRESERVE is in effect.
2. Verify that the “close pending” message displays.

3. Use DSTAT to verify that the volume set is still online. The CP* indicates a close pending state for the volume set.

```
:VSCLOSE BSET

VOLUME SET IN USE, CLOSE PENDING APPLIED TO VOLUME SET.
(CIERR 10637)

:DSTAT
LDEV-TYPE   STATUS      VOLUME (VOLUME SET - GEN)
---------   ---------   -------------------------
21-079350   CP*MASTER   BVOL1          (BSET-0)
22-079350   CP*MEMBER   BVOL2          (BSET-0)
23-079350   CP*MEMBER   BVOL3          (BSET-0)
```

4. Close the last file and verify that the volume set is automatically taken offline and put in the LONER state.

```
:DSTAT

LDEV-TYPE   STATUS   VOLUME (VOLUME SET - GEN)
---------   ------   -------------------------
21-079350   LONER    BVOL1          (BSET-0)
22-079350   LONER    BVOL2          (BSET-0)
23-079350   LONER    BVOL3          (BSET-0)
```

# VSOPEN

The VSOPEN command cancels a VSCLOSE command and places the volume set online.

After a VSOPEN command is issued, the volume set is ready for use. A volume set is open unless explicitly closed by the VSCLOSE command.

## Task

Places a previously closed volume set online.

## Capability

SM, system manager capability and can only be issued from the console.

## Syntax

VSOPEN *volsetname*

## Parameter

*volsetname* The volume set that will be placed online. **Required**.

## Example

This example shows how to bring a volume set online.

1. To bring a volume set online, enter the VSOPEN command.
2. Verify that VSOPEN was effective by entering the DSTAT command. The master/member status indicates that the set is online.

```
:VSOPEN BSET

:DSTAT

LDEV-TYPE   STATUS   VOLUME (VOLUME SET - GEN)
---------   ------   -------------------------
21-079350   MASTER   BVOL1          (BSET-0)
22-079350   MEMBER   BVOL2          (BSET-0)
23-079350   MEMBER   BVOL3          (BSET-0)
```

# VSRELEASE

The VSRELEASE command allows the user to cancel the VSRESERVE command and release a volume set so that it can be taken offline. Logging off the system also releases the volume set.

If the volume set is not specified, then the home volume set of the user's logon group and account is used.

## Task

Releases a volume set.

## Capability

No special capability required.

## Syntax

VSRELEASE [ *volsetname* ]

## Parameter

*volsetname* The volume set that will be placed online. If you omit the volume set name, the request will be issued for the home volume set of the user's logon group and account. **Optional.**

## Example

This example shows how to cancel a VSRESERVE command.

1. Enter the VSRELEASE command from the system prompt.
2. Verify that the volume set has been released.

```
:VSRELEASE BSET

:VSUSER

        NO VOLUME SETS CURRENTLY RESERVED (CIERR 10625)
```

# VSRELEASESYS

The VSRELEASESYS command allows the operator to cancel the VSRESERVESYS command and release a volume set to take it offline.

Negates the VSRESERVESYS command. Indicates that the system wide reservation of the volume set is no longer in effect. It has no effect on VSRESERVE commands issued by individual users on the system.

## Task

Releases a volume set.

## Capability

SM, system manager capability and can only be issued from the console.

## Syntax

VSRELEASESYS *volsetname*

## Parameter

*volsetname* The volume set that will be placed offline that was previously issued a VSRESERVESYS command. **Required.**

## Example

This example shows how to cancel a VSRESERVESYS command by using the VSRELEASESYS command.

```
:VSRELEASESYS BSET
```

# VSRESERVE

The VSRESERVE command allows a user to reserve a specified volume set. This means that the volume set cannot be taken offline by the operator. Whenever a user opens a file on a volume set, the system issues a VSRESERVE. When the user closes the file, the volume set is released and may be taken offline. A VSRESERVE is usually issued when a user may be opening and closing files over a period of time and wants to make sure the volume set remains available.

The VSRESERVE command is canceled when the user issues a VSRELEASE command, or when the user logs off the system.

Reserves the volume set between file opens for the user. That is, it notifies the system when the user does not explicitly have a file open on the volume set, further access will be forthcoming. This prevents the operator from taking the volume set offline. If no volume set is specified, then the request is for the home volume set of the user's logon group and account. Otherwise, the user must specify the full volume set name. The reservation of the volume automatically ends when the user logs off.

## Task

Keeps a volume set online.

## Capability

No special capability required.

## Syntax

VSRESERVE [ *volsetname* ] [;GEN= *genindex*]

## Parameters

*volsetname* The volume set that will be kept online. If this parameter is omitted, the home volume set of the user's logon group and account will be used. **Optional.**

*genindex* A number from 1 to 100 specifying the new generation number of the copied volume set. If it is omitted, the generation number will be one greater than the original volume set. **Optional**.

## Example

This example shows how to reserve a volume set.

1. Check to make sure that the volume set is mounted and online. Enter the DSTAT command from the system prompt. The master/member status indicates that the set is online.
2. Reserve the volume set BSET by entering the VSRESERVE command from the system prompt.

3. Verify that the reserve has taken effect by using the VSUSER command.

```
:DSTAT

LDEV-TYPE   STATUS   VOLUME (VOLUME SET - GEN)
---------   ------   ------------------------
21-079350   MASTER   BVOL1          (BSET-0)
22-079350   MEMBER   BVOL2          (BSET-0)
23-079350   MEMBER   BVOL3          (BSET-0)

:VSRESERVE BSET

:VSUSER

        VOLUME SET NAME   JOBNUM   JOBNAME
        ---------------   ------   -------
        BSET              #S6      (USER.ACCT)
```

# VSRESERVESYS

The VSRESERVESYS command is an operator command that reserves a volume set on a system-wide basis. Logging on and off does not cancel the reserve.

This command can be used when the operator wants to stream several jobs that use files on the volume set. This system-wide reserve remains in effect during the logging on and logging off of the jobs that use the volume set.

Reserves the volume set for the entire system. This command indicates to the system that the volume set is to remain online until an VSRELEASESYS command is issued. Unlike the VSRESERVE command, this command is unaffected by logging off.

## Task

Keeps a volume set online.

## Capability

SM, system manager capability, and can only be issued from the console.

## Syntax

VSRESERVESYS *volsetname*

## Parameter

*volsetname* The volume set that will be kept online. **Required.**

## Example

This command shows how to reserve a volume set on a system-wide basis.

1. Use DSTAT to verify that the volume set is mounted and online. The master/member status indicates that the set is online.
2. Reserve the volume set.

```
:DSTAT

LDEV-TYPE   STATUS   VOLUME (VOLUME SET - GEN)
---------   ------   ------------------------
21-079350   MASTER   BVOL1          (BSET-0)
22-079350   MEMBER   BVOL2          (BSET-0)
23-079350   MEMBER   BVOL3          (BSET-0)

:VSRESERVESYS BSET
```

3. Verify that the set is reserved by attempting to close it.
4. Verify that the volume set is still online. The master/member status indicates that the set is online.

```
:VSCLOSE BSET

VSRESERVESYS (OR LMOUNT) IS IN EFFECT ON VOLUME SET.
(CIERR 10634)

:DSTAT


LDEV-TYPE   STATUS   VOLUME (VOLUME SET - GEN)
---------   ------   ------------------------
21-079350   MASTER   BVOL1          (BSET-0)
22-079350   MEMBER   BVOL2          (BSET-0)
23-079350   MEMBER   BVOL3          (BSET-0)
```

# VSUSER

The VSUSER command lists all users who have reserved a nonsystem volume set. It also displays the volume set name, job number, and the job names of all users currently performing a reserve function.

## Task

Lists all users of a currently reserved nonsystem volume set.

## Capability

No special capability required.

## Syntax

VSUSER [*volsetname*]

## Parameter

*volsetname* The volume set that will be kept online. **Optional.**

## Example

This example shows how to list all of the currently reserved volume sets by using the VSUSER command.

```
:VSUSER

        VOLUME SET NAME   JOBNUM   JOBNAME
        ---------------   ------   -------
        BSET              #S6      (USER.ACCT)
```

## DISCUTIL Commands

DISCUTIL is a standalone utility used to configure devices and to save files to tape from disks that have failed, systems that have failed or directories that have become corrupted. The commands in Table 5-4 are available from DISCUTIL. All of the DISCUTIL commands are described in this section.

**Table 5-4. DISCUTIL Commands**

| Command | Task |
|---|---|
| CONFIGURE | Configures additional devices. |
| DISMOUNT | Dismounts a disk volume making it inaccessible to DISCUTIL. |
| DO | Reexecutes a command from the command history stack. |
| DSTAT | Displays information about each mounted volume. |
| EXIT | Terminates DISCUTIL. |
| HELP | Lists available commands. |
| LISTREDO | Displays the command history stack. |
| MOUNT | Mounts a disk volume making it accessible to DISCUTIL. |
| PDEV | Lists the disks that are currently configured and mounted. |
| REDO | Edits and reexecutes a command from the command history stack. |
| SAVE | Saves user specified files from disks to tape. |
| SHOWDEV | Displays information about each configured device. |
| TAPE | Displays and selects the current tape LDEV. |
| UNCONFG | Unconfigures a device. |

# CONFIGURE

The CONFIGURE command adds new devices to the current configuration. DISCUTIL can use the device after it has been configured. If the device is a disk, DISCUTIL tries to mount it. DISCUTIL automatically configures the system console, the tape drive, and all master and member disks that were mounted at the time of the failure.

## Task

Configures devices.

## Capability

No special capability required.

## Syntax

CONFIGURE [LDEV=] *ldev* [CLASS=] { TAPE | PRINTER | TERMINAL | DISC } [PATH=] *path*

## Parameters

| | |
|---|---|
| *ldev* | A number from 1 to 100 specifying the logical device number of the device that will be configured. **Required**. |
| TAPE | Configure the device as a tape. |
| PRINTER | Configure the device as a printer. |
| TERMINAL | Configure the device as a terminal. |
| DISC | Configure the device as a disk. |
| *path* | The numbers associated with the hardware cards connected to the device. **Required**. |

## Example

This example shows how to configure a disk drive using the CONFIGURE command.

```
discutil> CONFIGURE 17 DISC 8.0.1

*Note: LDEV 17 CONFIGURED AS DISC (PATH: 8.0.1)

*Note: VOLUME MOUNTED ON LDEV 17
```

# DISMOUNT

The DISMOUNT command logically dismounts a volume. The volume entry corresponding to the specified LDEV is removed from the mounted volume table in DISCUTIL. For removable disk drives, a volume can be dismounted, then the media or disk pack on the corresponding LDEV can be powered off and removed. A new disk pack may then be mounted in the disk drive, powered on, and mounted with the MOUNT command. DISCUTIL then recognizes the new disk pack.

## Task

Dismounts a disk.

## Capability

No special capability required.

## Syntax

```
DISMOUNT [LDEV=] ldev
```

## Parameter

*ldev* A number from 1 to 100 specifying the logical device number of the volume that will be dismounted. **Required**.

## Example

This example shows how to dismount a disk pack using the DISMOUNT command.

```
discutil> DISMOUNT 17

*Note: VOLUME DISMOUNTED ON LDEV 17
```

# DO

The DO command reexecutes a command from the command history stack. Refer to the LISTREDO and REDO commands in this chapter.

## Task

Reexecutes a command from the command history stack.

## Capability

No special capability required.

## Syntax

```
DO [[CMD=]cmdid]
```

## Parameters

*cmdid* Identifies a particular command in the command history stack. It can be one of the following:

- A relative command number (executes the *n*th number in the stack, starting with the most recent).
- An absolute number (executes number *n* in the stack).
- A string (executes the most recent command starting with string).
- Omitted (executes last command).

**Optional.**

## Example

This example shows how to reexecute a command from the command history stack using the LISTREDO command.

1. List the commands in the command history stack using the LISTREDO command.
2. e xcute command 30.

```
discutil> LISTREDO

    24) LIST REDO
    25) LISTREDO
    26) REDO 24
    27) LIST REDO
    28) LISTREDO
    29) LIST REDO
    30) HELP
    31) HELP RECOVER
    32) HELP LISTREDO
    33) LISTREDO

discutil> DO 30
```

# DSTAT

DSTAT displays disk information for all disks known to DISCUTIL. This information includes the LDEV where the volume is mounted, the type of volume, the volume set name, and volume name.

## Task

Displays volume information.

## Capability

No special capability required.

## Syntax

```
DSTAT
```

## Example

This example shows how to display disk information using the DSTAT command.

```
discutil>DSTAT


LDEV-TYPE   STATUS   VOLUME (VOLUME SET - GEN)
---------   ------   -------------------------
11-079350   MASTER   BVOL1          (BSET-0)
12-079350   MEMBER   BVOL2          (BSET-0)
13-079350   MEMBER   BVOL3          (BSET-0)
```

## EXIT

The EXIT command terminates DISCUTIL.

### Task

Terminates DISCUTIL.

### Capability

No special capability required.

### Syntax

```
EXIT
```

### Example

This example shows how to exit DISCUTIL using the EXIT command.

```
discutil> EXIT

END OF DISCUTIL/XL

AUTO-REBOOT TO FOLLOW...

AUTO-REBOOT IN PROGRESS...
```

# HELP

This command provides online information about all available DISCUTIL commands.

## Task

Lists DISCUTIL command information.

## Capability

No special capability required.

## Syntax

```
HELP [ [CMD=] cmdname]
```

## Parameter

*cmdname* The command you want to find out more about. **Optional.**

## Example

This example shows how to list the DISCUTIL commands using the HELP command.

```
discutil> HELP

THE COMMANDS AVAILABLE IN DISCUTIL ARE:

CONFIGURE
DISMOUNT
DO
DSTAT
EXIT
HELP
LISTREDO
MOUNT
PDEV
REDO
SAVE
SHOWDEV
TAPE
```

## LISTREDO

The LISTREDO command displays the command history stack. The output order of the stack is from the oldest to the most recent command with command reference numbers preceding each command. The command reference number is used with the DO and REDO commands.

### Task

Displays the command history stack.

### Capability

No special capability required.

### Syntax

```
LISTREDO
```

### Example

This example shows how to list the command history stack by using the LISTREDO command.

```
discutil> LISTREDO

    24) LIST REDO
    25) LISTREDO
    26) REDO 24
    27) LIST REDO
    28) LISTREDO
    29) LIST REDO
    30) HELP
    31) HELP RECOVER
    32) HELP LISTREDO
    33) LISTREDO

discutil>
```

# MOUNT

The MOUNT command mounts a disk volume making it accessible to DISCUTIL. If an LDEV represents a disk and the device is responding, DISCUTIL attempts to mount the volume. Once a new volume has been mounted, it is available to DISCUTIL. Only volumes that have been initialized as a volume MASTER or MEMBER can be mounted.

## Task

Mounts a volume.

## Capability

No special capability required.

## Syntax

```
MOUNT [LDEV=] ldev
```

## Parameter

| | |
|---|---|
| *ldev* | A number from 1 to 100 specifying the logical device number of the volume to be mounted. **Required**. |

## Example

This example shows how to make a volume accessible to DISCUTIL.

1. Use the DSTAT command to determine the state of the volume.
2. Use the MOUNT command to mount the volume.

```
discutil>DSTAT


LDEV-TYPE   STATUS   VOLUME (VOLUME SET - GEN)
---------   ------   -------------------------
11-079350   MEMBER   BVOL1          (BSET-0)
12-079350   MEMBER   BVOL2          (BSET-0)

discutil> MOUNT 13

*Note: VOLUME MOUNTED ON LDEV 13

discutil>DSTAT


LDEV-TYPE   STATUS   VOLUME (VOLUME SET - GEN)
---------   ------   -------------------------
11-079350   MEMBER   BVOL1          (BSET-0)
12-079350   MEMBER   BVOL2          (BSET-0)
13-079350   MEMBER   BVOL3          (BSET-0)
```

# PDEV

PDEV displays the volume set name, volume number, and device physical path for all the disks known to DISCUTIL. A disk becomes known to DISCUTIL either by the CONFIGURE command or when DISCUTIL is first invoked.

## Task

Displays information about all configured disks.

## Capability

No special capability required.

## Syntax

```
PDEV
```

## Example

This example shows how to display volume information. The volume number in the example below refers to the number of the volume in the volume set.

```
discutil> PDEV

CURRENT DEVICE CONFIGURATION:

   LDEV : 1

       VOLUME SET NAME : MPEXL_SYSTEM_VOLUME_SET
       VOLUME NUMBER   : 1
       PATH            : 8.0.0

   LDEV : 11

       VOLUME SET NAME : ACCOUNTING_VOL_SET
       VOLUME NUMBER   : 1
       PATH            : 8.0.4

   LDEV : 12

       VOLUME SET NAME : ACCOUNTING_VOL_SET
       VOLUME NUMBER   : 2
       PATH            : 8.0.5
```

# REDO

The REDO command allows you to edit a command from the command history stack and then reexecute the command.

## Task

Allows for editing and reexecution of a command from the command history stack.

## Capability

No special capability required.

## Syntax

```
REDO [[CMD=]cmdid]
```

## Parameters

*cmdid* Identifies a particular command in the command history stack. It can be one of the following:

- A relative command number (executes the *n*th number in the stack, starting with the most recent).
- An absolute number (executes number *n* in the stack).
- A string (executes the most recent command starting with string).
- Omitted (executes last command).

**Optional.**

## Example

The example on the following page shows how to reexecute a command from the command history stack.

1. List the commands in the command history stack using the LISTREDO command.
2. Reexecute command 30.

```
discutil> LISTREDO
    24) LIST REDO
    25) LISTREDO
    26) REDO 24
    27) LIST REDO
    28) LISTREDO
    29) LIST REDO
    30) HELP
    31) HELP RECOVER
    32) HELP LISTREDO
    33) LISTREDO

discutil> REDO 30
```

# SAVE

SAVE is used with the VOLUTIL RECOVER command for recovering disk files if there is a system failure, a directory corruption, or a disk becomes inaccessible to the operating system. The SAVE function retrieves the files from disk and copies them to magnetic tape for later recovery (via the VOLUTIL RECOVER command).

For more information on VOLUTIL, refer to chapter 4, "Troubleshooting".

## Task

Retrieves files from disk and copies them to tape.

## Capability

No special capability required.

## Syntax

```
SAVE
```

## Example

This example shows how to save disk files to tape.

1. Enter the SAVE command at the discutil> prompt.

```
discutil> SAVE

WARNING BLOCK

ENTER FILE SET TO BE SAVED: MYFILE.JOHN.SMITH

TRAVERSE DIRECTORIES UNDER FILESET (Y/N)?

ENTER THE LDEV: 17

ENTER THE MODIFICATION DATE (MM/DD/YYYY): 07/01/1986

ENTER THE TAPE LDEV:

MYFILE.JOHN.SMITH - LDEV 17 - ADDR $0002CA0 - FOUND
MYFILE.JOHN.SMITH - LDEV 17 - ADDR $0002CA0 - SAVED

WARNING BLOCK

ENTER FILE SET TO BE SAVED:
```

The SAVE command prompts you for a file set, LDEV, and modification date.

```
ENTER FILE SET TO BE SAVED: MYFILE.JOHN.SMITH
```

Use the *filename.groupname.acctname* syntax where:

*filename* is the file designator

The file(s) must reside on a disk available when DISCUTIL was started or one that was made available with the DISCUTIL CONFIGURE command.

*groupname* is an optional group designator.

*acctname* is an optional account designator.

If you press Return without entering a file set, DISCUTIL ends the SAVE command.

The parameters *filename*, *groupname*, and *acctname* may be replaced by @ to signify "all members of the set" (for example, @.@.@).

The prompt is repeated after the file set is saved. To terminate SAVE, press Return.

2. Enter RETURN at the following prompt:

```
TRAVERSE DIRECTORIES UNDER FILESET (Y/N)?
```

3. The SAVE command prompts you for an LDEV.

```
ENTER THE LDEV: 17
```

To save files from a particular logical device (disk), enter the LDEV now. SAVE searches only the specified disk for the file(s) you want to save. You may also press Return at this prompt to indicate you want SAVE to search each LDEV for file(s). The LDEV entered must be one that is known to DISCUTIL. Use the PDEV command to determine if an LDEV is valid. For more information on DISCUTIL, refer to chapter 4 of this manual, and to the *System Utilities Manual* (32650-90081).

4. The SAVE command prompts you for a volume set name if you did not specify an LDEV number.

```
ENTER VOLUME SET NAME: PROD_SET
```

Enter a volume set name if you want SAVE to search for file(s) on a particular volume set. Use the PDEV command to determine valid volume set names. Press Return to indicate you want SAVE to search all disks for the file(s).

5. The SAVE command prompts you for a modification date.

```
ENTER THE MODIFICATION DATE (MM/DD/YYYY):07/01/1986
```

Only files modified since this date are saved to tape. DD is a two-digit number for the day of the month, MM is a two-digit number for the month, and YYYY is a four-digit number for the year. Omit the date specification and press Return at the prompt to save all files requested.

6. Enter the tape LDEV.

# SHOWDEV

SHOWDEV displays information about configured devices.

## Task

Displays information about configured devices.

## Capability

No special capability required.

## Syntax

```
SHOWDEV
```

## Example

This example shows how to display information about configured devices using the SHOWDEV command.

```
discutil> SHOWDEV

    1      DISC    8.0.0
    4  TERMINAL    8.1.3
    7      TAPE    8.2.3
   17      DISC    8.0.1
   18      DISC    8.0.2
   19      DISC    8.0.5
```

# TAPE

The `TAPE` command displays the current tape drive's logical device number (LDEV) or selects another tape drive as the current tape drive.

If the LDEV parameter is omitted, the current tape drive's LDEV is displayed. If the LDEV parameter is supplied, that LDEV becomes the current tape drive. The LDEV must have been configured as a tape drive.

## Task

Displays the current tape drive's LDEV or selects another LDEV as the current tape drive.

## Capability

No special capability required.

## Syntax

`TAPE [[LDEV=]` *ldev* `]`

## Parameter

*ldev* — A number from 1 to 100 specifying the logical device number of the tape drive that will become the current tape drive. **Optional**.

## Example

This example shows how to change the current tape drive's LDEV by using the `TAPE` command.

```
discutil> TAPE 15

*Note: TAPE DRIVE SWITCHED TO LDEV 15
```

# UNCONFIG

UNCONFIG unconfigures a currently configured device. This command is used dynamically to remove a device from the system's configuration. This releases the corresponding LDEV and physical path, making them available for reassignment with the CONFIGURE command.

## Task

Unconfigures a device.

## Capability

No special capability required.

## Syntax

UNCONFIG [LDEV=] *ldev*

## Parameter

*ldev* The logical device number of the device to be unconfigured. **Required.**

## Example

This example shows how to use the UNCONFIG command.

```
discutil> UNCONFIG 20

*NOTE: LDEV 20 UNCONFIGURED
```

# 6

# VOLUTIL/DISCUTIL Program

## (dialog,shell,discutil,volutil)

## VOLUTIL Messages

### dialog

| | | |
|---|---|---|
| 1 | MESSAGE | *Error: Unexpected error returned from execution of FFILEINFO intrinsic. (dialog 1) |
| 2 | MESSAGE | *Error: Unable to close INPUT file. (dialog 2) |
| 3 | MESSAGE | *Error: Unable to close OUTPUT file. (dialog 3) |
| 4 | MESSAGE | *Error: Unexpected error occurred during INPUT file input/output. (dialog 4) |
| 5 | MESSAGE | *Error: Unexpected error occurred during OUTPUT file input/output. (dialog 5) |
| 6 | MESSAGE | *Error: Unexpected error occurred during $STDIN input operation. (dialog 6) |
| 7 | MESSAGE | *Error: Unexpected error occurred during $STDLIST output operation. (dialog 7) |
| 8 | MESSAGE | *Error: Unexpected error from FCONTROL intrinsic with INPUT file. (dialog 8) |

| | | |
|---|---|---|
| 9 | MESSAGE | *Error: Unexpected error from execution of FGETINFO intrinsic. (dialog 9) |
| 10 | MESSAGE | *Error: File type not ASCII. (dialog 10) |
| 11 | MESSAGE | *Error: Invalid file record size. (dialog 11) |
| 12 | MESSAGE | *Error: Unexpected error from execution of PRINTOPREPLY intrinsic. (dialog 12) |
| 13 | MESSAGE | *Error: End-of-file encountered on OUTPUT file. (dialog 13) |
| 14 | MESSAGE | *Error: End-of-file encountered on INPUT file. (dialog 14) |
| 15 | MESSAGE | *Error: End-of-file encountered on $STDLIST. (dialog 15) |
| 16 | MESSAGE | *Error: Unexpected error from FPOINT intrinsic. (dialog 16) |

## parserr

| | | |
|---|---|---|
| 15 | MESSAGE | *Error: The VNAME parameter is required. (parserr 15) |
| | CAUSE | When you entered the NEWVOL command, you forgot to specify a name for the volume. |
| | ACTION | Reenter the NEWVOL command and remember to specify a volume name. |

## shell

| | | |
|---|---|---|
| 1 | MESSAGE | *Error: Incomplete command pending on end-of-file. (shell 1) |
| 2 | MESSAGE | *Error: Unable to process command. (shell 2) |

| | | |
|---|---|---|
| 3 | MESSAGE | *Error: Invalid parameter length. (shell 3) |
| 4 | MESSAGE | *Error: Command history stack is empty. (shell 4) |
| 5 | MESSAGE | *Error: Command not found in command history stack. (shell 5) |
| 6 | MESSAGE | *Error: Invalid parameter length or composition. (shell 6) |
| 7 | MESSAGE | *Error: Unrecognized command. (shell 7) |
| 8 | MESSAGE | *Error: GET_NAME error. (shell 8) |
| 9 | MESSAGE | *Error: This command has no associated functionality. (shell 9) |
| 10 | MESSAGE | *Error: Error in accessing message catalog. (shell 10) |
| 11 | MESSAGE | *Error: Invalid numeric index into command history stack. (shell 11) |
| 12 | MESSAGE | *Error: Command name length exceeds 16 characters. (shell 12) |
| 13 | MESSAGE | *Error: Insufficient capabilities to execute this command. (shell 13) |
| 14 | MESSAGE | *Error: MiUssing required parameter not specified in command. (shell 14) |
| 15 | MESSAGE | *Error: Parameter can consist of numeric characters only. (shell 15) |
| 16 | MESSAGE | *Error: Depth of command stack can only be in the range 1 to 100. (shell 16) |

| | | |
|---|---|---|
| 17 | MESSAGE | *Error: Invalid key option specified for set command. (shell 17) |
| 18 | MESSAGE | *Error: Command only valid in interactive mode. (shell 18) |
| 19 | MESSAGE | *Error: Subsys !, Info: !. (shell 19) |
| 20 | MESSAGE | *Error: Invalid command edit operation. (shell 20) |

## volutil

| | | |
|---|---|---|
| 53 | MESSAGE | *Error: Volume already exists. (volutil 53) |
| | CAUSE | When you entered the NEWVOL command, you specified a volume name that already exists. |
| | ACTION | Use the DSET command to examine the names and characteristics of all the volumes currently on the system. Then reenter the NEWVOL command and specify a volume name that does not exist for the current volume set. |
| 54 | MESSAGE | *Error: Volume does not exist. (volutil 54) |
| | CAUSE | When you entered the INITVOL command, you specified a volume name associated with a particular volume set, but a volume of that name does not exist. |
| | ACTION | Use the SHOWSET command to examine the names of all volumes in the current volume set. Then reenter the INITVOL command and specify a volume name that does exist in the current volume set. |
| 70 | MESSAGE | *Error: Ldev is not a valid disk type for mirroring (volutil 70) |
| | CAUSE | Mirrored disks are not the correct type (HP7937FL) for volume initialization. |
| | ACTION | Use correct disk type. |

| | | |
|---|---|---|
| 71 | MESSAGE | *Error: Mirrored volumes must be specified as a pair. (volutil 71) |
| | CAUSE | Only one volume was specified in NEWMIRRSET or NEWMIRRVOL. |
| | ACTION | Specify correct number of disks. |
| 72 | MESSAGE | *Error: Mirrored volumes must be unique. (volutil 72) |
| | CAUSE | LDEV's specified a duplicate number. |
| | ACTION | Specify a unique LDEV. |
| 73 | MESSAGE | *Error: Cannot add mirrored volumes to non-mirrored volume set. (volutil 73) |
| | CAUSE | Mirrored volumes cannot be added to a non-mirrored volume set. |
| | ACTION | Specify a mirrored volume set. |
| 74 | MESSAGE | *Error: Cannot add non-mirrored volumes to a mirrored volume set. (volutil 74) |
| | CAUSE | Non-mirrored volumes cannot be added to a mirrored volume set. |
| | ACTION | Specify a mirrored volume. |
| 75 | MESSAGE | *Error: Specified volume set is not mirrored. (volutil 75) |
| | CAUSE | An attempt was made to perform a mirrored operation on a non-mirrored volume set. |
| | ACTION | Cause indicates appropriate action. |
| 76 | MESSAGE | *Error: Specified volume does not exist on specified ldev. (volutil 76) |
| | CAUSE | Either the wrong volume name or the wrong LDEV was specified. |
| | ACTION | Cause indicates appropriate action. |
| 77 | MESSAGE | *Error: Command only valid for disabled mirrored volumes. (volutil 77) |
| | CAUSE | REPLALCEMIRRVOL was issued for a non-disabled volume. |
| | ACTION | Cause indicates appropriate action. |

| | | |
|---|---|---|
| 78 | MESSAGE | *Error: Illegal replace. Mirrored partner is not available. (volutil 78) |
| | CAUSE | REPLALCEMIRRVOL was issued but good volume was not mounted as MEMBER or MASTER. |
| | ACTION | Verify that volume used is good. |
| 79 | MESSAGE | *Error: Mirrored volume must be in pending state. (volutil 79) |
| | CAUSE | The volume specified in the SUSPENDMIRROR command was not in a pending state. |
| | ACTION | Be sure specified volume is pending before issuing command. |
| 80 | MESSAGE | *Error: Command not supported for mirrored volume sets. (volutil 80) |
| | CAUSE | The COPYSET command is not supported for mirrored volume sets. |
| | ACTION | Use a different command. |
| 83 | MESSAGE | *Error: Illegal replace. MIrrored pair is already in operation. (volutil 83) |
| | CAUSE | REPLACEMIRRORVOL was issued on an already functioning mirrored pair. |
| | ACTION | CAUSE indicates appropriate action. |
| 85 | MESSAGE | *Error: User volumes incomplete, cannot use as repair source. (volutil 85) |
| | CAUSE | The JOINMIRRSET command with SOURCE=USER was specified, and one or more user volumes are missing. |
| | ACTION | CAUSE indicates appropriate action. |
| 86 | MESSAGE | *Error: Backup volumes incomplete, cannot use as repair source. (volutil 86) |
| | CAUSE | The JOINMIRRSET command with SOURCE=BACKUP was specified, and one or more backup volumes are missing. |
| | ACTION | CAUSE indicates appropriate action. |

| | | |
|---|---|---|
| 87 | MESSAGE | *Error: Volume set close failed during join. (volutil 87) |
| | CAUSE | SOURCE=BACKUP was specified for the join and it failed. One of the reasons could be that users are still logged onto the destination volume set. The JOINMIRRSET command with SOURCE=BACKUP was specified, and one or more backup volumes are missing. |
| | ACTION | Log off all users from the volume set and retry the JOINMIRRSET. If SOURCE=USER was specified and this message is seen, a split-volume backup is taking place on the backup volumes. |
| 88 | MESSAGE | *Error: New volumes cannot be added to a split volume set. (volutil 88) |
| | CAUSE | The NEWVOL command was used to attempt to add volumes to a split-volume set. |
| | ACTION | CAUSE indicates appropriate action. |
| 89 | MESSAGE | *Error: Mirrored partner is already in suspend mirror state. (volutil 89) |
| | CAUSE | A SUSPENDMIRRVOL command was issued against a volume whose partner is in the SUSPENDMIRR state. |
| | ACTION | CAUSE indicates appropriate action. |
| 122 | MESSAGE | *Error: Volume mounted in invalid state for desired operation. (volutil 122) |
| | CAUSE | When you entered the INITVOL command, you specified a value for the state parameter which is not valid. |
| | ACTION | CAUSE indicates appropriate action. |
| 130 | MESSAGE | *Error: Error encountered while reading first file label. (volutil 130) |
| 131 | MESSAGE | *Error: No tape label on the beginning of tape. Cannot recover. (volutil 131) |
| 132 | MESSAGE | *Error: Recover aborted because of tape label error. (volutil 132) |

| | | |
|---|---|---|
| 133 | MESSAGE | *Error: Recover aborted because of tape error. (volutil 133) |
| 134 | MESSAGE | *Error: Recover aborted because of tape positioning error. (volutil 134) |
| 135 | MESSAGE | *Error: Recover aborted because of file closing error. (volutil 135) |
| 136 | MESSAGE | *Error: Recover aborted by user. (volutil 136) |
| 137 | MESSAGE | *Error: Recover aborted because of incorrect capability. (volutil 137) |
| | MESSAGE | *Error: Destination volume on ldev ! not mounted in valid state. |
| | MESSAGE | *Error: Invalid response. Please answer with a 'Y', 'YES', 'N', or 'NO'. |
| | MESSAGE | *Error: Source volume on ldev ! not a member of set being copied. |
| | MESSAGE | *FATAL ERROR: OUTER-MOST RECOVER BLOCK INVOKED. PROGRAM WILL TERMINATE. |
| | MESSAGE | Subsys !, Info: !. |

## DISCUTIL Messages

| | |
|---|---|
| MESSAGE | CONFIGURATION INFO FOR LDEV NN CORRECTED |
| MESSAGE | CONFIGURATION INFO FOR LDEV NN NOT CORRECTED |

| | |
|---|---|
| MESSAGE | DISCUTIL HAS ENCOUNTERED A TAPE ERROR: REWINDING TAPE |
| MESSAGE | END OF TAPE #NN; PLEASE WAIT |
| MESSAGE | ENTER "HELP" FOR A LIST OF VALID COMMANDS |
| MESSAGE | ERROR - CONFIGURATION INFO FOR LDEV NN IS NOT VALID |
| MESSAGE | ERROR - ENCOUNTERED A MISSING VOLUME; UNABLE TO CONTINUE SAVING FILE |
| MESSAGE | ERROR - FOUND UNEXPECTED VOLUME ON LDEV NN; UNABLE TO CONTINUE SAVING FILE |
| MESSAGE | ERROR - LABEL TABLE CORRUPTED ON LDEV NN; UNABLE TO CONTINUE ON THIS LDEV |
| MESSAGE | ERROR - LABEL TABLE OFFSET MISMATCH IN EXTENT BLOCK ENTRY |
| MESSAGE | ERROR - OFFSET MISMATCH IN FILE LABEL; SKIPPING TO NEXT LABEL |
| MESSAGE | ERROR - UNABLE TO COMPLETE DISK READ ON LDEV NN |
| MESSAGE | ERROR - UNABLE TO COMPLETE TAPE EOF WRITE ON LDEV NN |
| MESSAGE | ERROR - UNABLE TO COMPLETE TAPE WRITE ON LDEV NN |
| MESSAGE | ERROR - UNABLE TO CONVERT LDEV TO PHYSICAL PATH |
| MESSAGE | ERROR - UNABLE TO OBTAIN DEVICE STATUS FROM LDEV NN |
| MESSAGE | ERROR - UNABLE TO READ THE HPE VOLUME LABEL ON LDEV NN |
| MESSAGE | ERROR - UNABLE TO READ THE LABEL TABLE ON LDEV NN; UNABLE TO SAVE FILE |

| | |
|---|---|
| MESSAGE | ERROR - UNABLE TO READ THE LABEL TABLE ON LDEV NN |
| MESSAGE | ERROR - UNABLE TO READ THE LABEL TABLE; SKIPPING TO NEXT LABEL |
| MESSAGE | ERROR - UNABLE TO REWIND TAPE ON LDEV NN |
| MESSAGE | ERROR - UNABLE TO SET TAPE DENSITY ON LDEV NN |
| MESSAGE | LAST FILE ON PREVIOUS TAPE WILL BE RESAVED |
| MESSAGE | MOUNT TAPE #NN |
| MESSAGE | PRESS RETURN WHEN THE TAPE IS READY |
| MESSAGE | SAVE FOR LDEV NN HAS BEEN ABORTED |
| MESSAGE | SAVE WILL RESTART WITH TAPE NN; MOUNT A TAPE |
| MESSAGE | TAPE DENSITY SET TO DDDD ON LDEV NN |
| MESSAGE | THE TAPE ON LDEV NN IS READY |
| MESSAGE | WARNING - LDEV NN IS A TAPE DRIVE |
| MESSAGE | WARNING - LDEV NN IS A TERMINAL |
| MESSAGE | WARNING - LDEV NN IS NOT A VALID DISK LDEV |
| MESSAGE | WARNING - LDEV NN IS NOT CONFIGURED |
| MESSAGE | WARNING - "XXXX" IS NOT A VALID DISCUTIL COMMAND |

| | |
|---|---|
| MESSAGE | WARNING - "XXXX" IS NOT A VALID DISCUTIL COMMAND; LDEV NN IS A PRINTER |
| MESSAGE | WARNING - LDEV NN IS NOT VALID |
| MESSAGE | WARNING - NO LDEVS HAVE BEEN SELECTED FOR SAVING |
| MESSAGE | WARNING - THE FILE SET ENTERED IS NOT VALID |
| MESSAGE | WARNING - THE MODIFICATION DATE ENTERED IS NOT VALID |
| MESSAGE | WARNING - THE TAPE DRIVE DOOR ON LDEV NN IS OPEN |
| MESSAGE | WARNING - THE TAPE DRIVE ON LDEV NN IS OFFLINE |
| MESSAGE | WARNING - THE TAPE ON LDEV NN DOES NOT HAVE A WRITE ENABLE RING |
| MESSAGE | WARNING - THE TAPE ON LDEV NN IS NOT AT BOT |
| MESSAGE | WARNING - THE VOLUME SET NAME ENTERED IS NOT VALID |

# Index

## U



## V

## W