

Rev 1
28.4
MR 4.0
BOS 2.0

To: Operations
From: T. H. Van Vleck
Date: April 23, 1976
Subject: Operational changes for MR 4.0

INTRODUCTION

A new version of the Multics storage system has been installed with system MR 4.0. This version of the system will be much more reliable and be able to handle more storage.

Definitions

Several terms are used often in describing the new storage system's operation:

physical volume	A disk pack. Sometimes we speak of the combination of pack and disk drive as the volume.
volume label	Each physical volume has a label which identifies it to the system.
VTOC	Each physical volume contains a Volume Table of Contents (VTOC), which lists information about the segments on the volume.
root physical volume (RPV)	The physical volume which contains the root directory. It also contains the hardcore partition.
partition	A block of disk storage used for a special purpose.
hardcore partition	A partition used for paging those segments which come in off the system tape or are created during bootload.

Multics Project internal working documentation. Not to be reproduced or distributed outside the Multics Project.

logical volume A set of physical volumes always used together.

root logical volume (RLV) The only logical volume which contains directories.

record A 1024-word contiguous block of disk. Each volume has Multics records numbered starting with zero to a maximum which depends on the model number of the disk pack and drive.

sector A 64-word block of disk. A sector is the smallest addressable unit of disk with the current hardware. Each Multics record contains 16 sectors. Field engineering test and diagnostic programs and some other special applications may use sector addressing, but its use is rare in normal operation. Since no Multics record is split across disk cylinders, there may be some sectors on a disk pack which are not part of any record, depending on the model number of the disk.

The term "area" will not be used any more to describe a particular disk drive; all disk drives will be identified by subsystem name and drive number. Drive numbers start at 1.

Major changes

There are four new rules imposed by the new storage system:

1. All the pages of a segment are always on the same physical volume. In older versions of the supervisor, any page of a segment might be found on any disk unit.
2. All segments in the same directory are always in the same logical volume. This attribute of a directory is its "sons logical volume ID" or "sons_lvid." Usually, when a directory is created, the new directory gets the same sons_lvid as the parent directory. A directory whose sons_lvid is different from its parent directory's sons_lvid is called a master directory.
3. All directories are always on the root logical volume.
4. All logical volumes are registered. The registration information for a logical volume includes the list of physical volumes in the

logical volume and the access control rules for use of the logical volume.

These changes mean that if a single disk unit has a head crash it will be much easier to find out what segments were lost and whether the system can continue to run. Often, after a crash, the operator will be able to reload only those volumes whose contents were damaged. Under the old system, any head crash was bound to destroy a few pages each of many directories; in the new system only those physical volumes which are part of the root logical volume have directories on them.

Contents of a Physical Volume

The contents of a physical volume look like this:

```
-----  
low | Volume Label |  
-----  
| Volume Map |  
-----  
| VTOC header and error data |  
-----  
| |  
| VTOC |  
| |  
-----  
| |  
| Pages of segments |  
| |  
| |  
-----  
| Partitions (optional) |  
-----
```

Not all volumes have partitions, and the partitions may exist either at the top of the pages of segments (as shown in the picture), or between the VTOC and the pages of segments. Partitions must always reside completely on a single physical volume.

BOS CHANGES

Many changes have been made to BOS. Most of these are a result of the new volume organization.

Config Deck Changes

The most important change in BOS is that the old notion of a partition, defined by a PART card, has been replaced.

The PART MULT card used to define the total space available for Multics segments. There is no PART MULT card under the new storage system and BOS does not know which volumes are to be used for Multics paging. Only Multics has this information.

PART cards are still used to tell BOS about the special areas of the disk storage which it uses for FDUMP, and for BOS itself, and to tell Multics where the FDUMP, SYSERR log, and HC (hardcore) partitions are. The new format of a PART card looks like this:

PART <partname> <subsystem> <drive>
Example: PART LOG DSK7 5

This card says that the LOG partition is to reside on the volume mounted on drive 5 of disk subsystem DSK7. The system will consult the label of that volume to find where the records of the LOG partition are to reside. During a cold boot, additional information is specified on the PART card for any partitions to reside on the RPV.

Disk subsystems used to be defined by D191 or E191 or whatever cards in the CONFIG deck. These subsystems are now defined by the PRPH card, which looks like this:

PRPH <subsystem> <iom> <chn> <nchn> <model1> <n1> ...
Example: PRPH DSK7 A 30 4 400. 9. 450. 2

All disk subsystems have names of the form DSKx, x arbitrary. This card describes the DSK7 subsystem. It is connected to IOM A, on channels 30-33. Drives 1 through 9 are model MSU0400 drives, and drives 10 and 11 are model MSU0450 drives.

The valid model numbers are:

181.	DSU181	
191.	DSS191	(same as MSU0400)
400.	MSU0400	
450.	MSU0450	

The ROOT card is new. it points out the location of the root physical volume.

ROOT <subsystem> <drive>
 Example: ROOT DSK7 1

by specifying the subsystem name and drive number. The disk unit pointed to must be the same as that on the PART HC card.

A new configuration card has been provided to specify the sizes of certain wired-down tables. There may be more than one WTBL card; each card may specify up to 6 pairs of the form <ld> <length> where the length is in words.

WTBL <ld1> <length1> ...
 Example: WTBL VTB 1024. PVT 1024.
 WTBL TTYB 2048.

The following is a table of the valid arguments to the WTBL card:

ld	Segment Name	Default Lth
D355	dn355_data	360.
DISK	disk_seg	500.
DST	dst_seg	768.
I0BM	lobm_data	152.
I0MD	lom_data	532.
PRTB	printer_trace_buffer	0.
PVT	pvt	2048.
SYST	system_trace_seg	0.
TTYB	tty_buf	6144.
VTB	vtoc_buffer_seg	2048.

The TTYB and PRTB cards become obsolete, and must not be used. If present in the CONFIG deck, they must be changed to WTBL cards.

The SST card must be adjusted because the size of an AST entry has increased from 8 words to 12. Either the segment size must be increased, or the pool sizes must be decreased. For example, if a site has the following SST card

SST 36. 576. 370. 110. 10.

then the increase in words for the new system is $(576+370+110+10)*4/1024$, or 5 pages. An alternative would be to decrease the sizes of the AST pools and keep the SST size constant. So either of the following SST cards would be reasonable:

SST 41. 576. 370. 110. 10.
 SST 36. 430. 317. 104. 10.

A new parameter is allowed on the DEBG card, for system debugging use. If the parameter CRWL is put on the DEBG card, the system will return to BOS on every attempt to crawl out of ring 0, with the message

verify_lock: crawlout stop specified on debug card

so that the system staff can take a dump. Typing GO will cause the system to continue operation. Another parameter which can be specified on the DEBUG card is the DIRW flag, which causes modified directory pages to be written from core whenever a directory is unlocked. Specifying this parameter increases the safety of the system at some cost in increased paging.

A new card is used when multiple IOM's are used, to indicate which channels can reach a disk subsystem from each IOM:

CHNL <subsystem> <iom2> <chn2> <iom3> <chn3> ...
Example: CHNL DSK7 B 24

This card is used only when more than one IOM is in the configuration.

To indicate that a group of disk units are to be used for user I/O disk packs the former PRPH DISK card has been changed to the following:

UDSK <subsystem> <nchn> <drive1> <count1> ...
Example: UDSK DSK7 2 8. 2

This card specifies the maximum number of channels which may be used to support user peripheral disk I/O; if the channels are not required for user I/O they are available for storage system I/O. Up to six pairs of drive number and number of drives may be specified to indicate the particular units which are to be used for user peripheral disk I/O. A disk drive can be used for either storage system volumes or user peripheral packs; the system must be shut down to change the status of a drive from one to the other.

To change the default options for all salvaging operations, another card is used.

SALV <options>
Example: SALV CNSL RBLD

The following options are recognized:

CNSL	All output via SYSERR console
RBLD	Rebuild all directories
PATH	Print pathname of all directories salvaged
DUMP	Dump damaged objects
DEBG	Debugging use only

The CNSL option is used when all printers at an installation are down. All other options should be used only at the direction of

the system staff.

Sample Configuration Deck

Here is a sample configuration deck for a small machine.

```

CPU B 6
MEM A 128. ON
MEM B 128. ON
IOM A 0 4 14 24 34
CLOK 5 EST
PRPH DSK7 A 24 4 400. 7 * NEW
ROOT DSK7 1 * NEW
PART BOS DSK7 1 * NEW
PART HC DSK7 1 * NEW
PART LOG DSK7 2 * NEW
PART DUMP DSK7 2 * NEW
UDSK DSK7 1 7 1 * NEW
BULK 0 512. 1 2
PAGE BULK 0 1000
SST 30. 220. 110. 40. 6.
INT 230 231 232 233 22 223
OPTY OC_
SCHD 400000 10 10 100
TCD 6. 88. 196. 192.
TBLS STR 6
WTBL TTYB 2048. * REPLACES TTYB 2
PRPH OPC A 20 IBM 40.
PRPH PRTA A 30 1600. 600. 136.
PRPH TAPE A 22 500. 2 0 2 1 3 0 0
PRPH RDRA A 32 301.
PRPH PUNA A 33 301.
PRPH IMPW A 11 ABSI 6
PRPH IMPR A 10 ABSI
MPC A 30 2 A 32 2
MPC A 24 3
MPC A 22 1 A 23 1
D355 A 21 A
LSLA A 0 1 300. 1 133. 2 150. 1 300. 8.
HSLA A 0 0 1200. 2 300. 2 4800. 8.
HSLA A 0 20 1200. 4 4800. 4

```

This setup defines a system with 7 MSU0400 drives.

Changes to DUMP and PATCH

The device addresses typed to DUMP and PATCH when dumping specified disk addresses have been changed in a manner similar to the config deck changes. Requests to DUMP have the form

DEV <volume-designator> <device-position> <count>

where <volume-designator> may be one of

<subsystem> <drive>
PART <partname>
ROOT
BULK

and <device-position> may be one of

<Multics record number>
RECORD <Multics record number>
SECTOR <hardware disk sector number>

Example: DEV DSK7 4 19170. 3.

and a device address request to PATCH has the form

DEV <volume-designator> <device-position> <wordno>
Example: SET DEV DSK7 5 10000 0 767676767676

Changes to SAVE and RESTOR

The elimination of the PART MULT card changes SAVE and RESTOR drastically. The command line now specifies only the SAVE medium and options, like this:

SAVE <output-medium> <options>
Example: SAVE TAPE 1 2 ALL BRIEF

The keywords ALL and BRIEF can also be given, and instead of TAPE 1 2 one could say DISK DSK7 3 4 or COPY DSK7 5. SAVE (or RESTOR) then reads request lines from the BOS input source until an END request is encountered. Each request specifies one block of a physical volume to be dumped. Possible forms are:

<volume-designator> <extent> <pv_name>

where <extent> may be one of

<device-position> FOR <count>
<device-position> TO <lrec>
<device-position> ONLY
ENTIRE

Example: DSK7 3 0 FOR 6 ROOT2
 DSK7 3 12 TO 17 LDD3
 DSK7 1 0 ONLY
 DSK7 3 ENTIRE CHARLIE
 ROOT ENTIRE RPV
 PART LOG ENTIRE RPV

The PVNAME specified at SAVE time will be checked at RESTOR time, so that physical volumes need not be remounted on the same device addresses. Unless ALL is specified, dumping an ENTIRE pack will dump only those records in the volume which are actually used, plus the label information, the volume map, and the VTOC.

For example, to save the volumes on units 1 and 2 onto disk packs on units 3 and 4 in the sample configuration above, the operator might type

```
SAVE DISK DSK7 3 4
      DSK7 1 ENTIRE RPV
      DSK7 2 ENTIRE CISL2
      END
```

(Assuming that the physical volume name of the pack on unit 1 is RPV and the name of the pack on unit 2 is CISL2.) The indented lines are request lines read by the SAVE command; each input request is acknowledged by a message of the form

DSK7 1 (M400 A 30) 0. TO 19269.

but the SAVE does not actually start until all input has been typed and the END request given. When SAVE or RESTOR begin a new volume of the SAVE medium, they type out a message describing what volume address from the Multics data is on the SAVE medium, in the form

RPV (DSK7 1) 12349. ON TAPE 3

As each volume of the SAVE medium (tape reel or disk pack) is finished, SAVE or RESTOR will type out the error count.

SAVE and RESTOR do some checking of the volume labels of disk packs. If an attempt is made to RESTOR the contents of one volume onto a volume with a valid label but a different volume name, RESTOR will print

```
PHYSICAL VOLUME NAME DOES NOT MATCH REQUEST
REQUEST IS FOR FOOBAR, BUT VOLUME NAME IS RPV
```

and refuse to proceed. Similarly, if an attempt is made to use a storage system volume as a SAVE pack, the message

```
NDISK: DSK7 4 IS MULTICS STORAGE SYSTEM VOLUME CISL3
```

will be printed and no SAVE or RESTOR will be done. Instead of typing END the operator should type the QUIT request to return to BOS command level; he should then check the assignment of packs to disk units very carefully. In very rare cases, a pack will have an incorrect label left over from a previous use, which causes one of the above messages and prevents SAVE and RESTOR from operating. The TEST command should be used to clear the label in this case.

Changes to TEST and FMT

The TEST and FMT commands now operate on only one pack or part of a pack at a time. One may say

TEST <subsystem> <drive> <extent> <options>

Example: TEST DSK7 3 ENTIRE WRITE PATTERN 525252525252

FMT <subsystem> <drive> <extent>

Example: FMT DSK7 1 0 FOR 10

and so forth. The keywords CHECK, WRITE, and PATTERN still work for TEST.

Changes to BOOT

The BOOT command now allows the operator to say any single command to Multics ring-1 as an argument to the BOS command. The most common form will be to say

BOOT STAR

to cause the system to start up without pausing in ring 1. The interface between BOS and Multics limits the commands which can be typed to four-character items. The following commands are valid:

STAR	startup
MULT	multics
SALV	salvage_dirs
STAN	standard
RPVS	(see below)

The printer should be ready and paper synchronized when bootloading the system, since the salvager may be invoked automatically. If no printer is available, a SALV CNSL card should be used.

COLD Card Changes

The new format of the COLD card and WARM card for a BOS bootload deck looks like this:

<warm_or_cold> <model> <chn> <drive>

Example: COLD 400. 30 1

When BOS is bootloaded, the loader will read the label of the disk pack at channel 30, unit number 1, as if it were a MSU0400

drive. The label of this pack will be checked for existence of a BOS partition, and the extents of that partition found in the label will be used. The BOS loader will type out a message of the form

BOS PARTITION AT 19170. FOR 170.

giving the first record address and size of the partition. If no BOS partition can be found, one of the following messages will be printed:

CANNOT READ VOLUME LABEL
VOLUME DOES NOT HAVE A MULTICS LABEL
BOS PARTITION NOT DEFINED IN VOLUME LABEL

these messages usually result from an incorrect COLD or WARM card, or leaving an incorrect pack mounted.

BOS Loader Changes

For the first bootload of BOS on an empty residence volume, a special method must be used, since the pack has no label. In this case, the COLD card is replaced by the following card:

NLABEL <model> <chn> <drive> <device-position> <nrec>
Example: NLABEL 400. 30 1 19100. 170.

The 170-record BOS partition occupies addresses 19100-19269 of the MSUG400 pack on drive 1, channel 30 of the bootload IOM. The next step is to initialize the label of the BOS residence volume, by means of a BOOT COLD. This rare situation is the only time that an NLABEL card should be used, since it can destroy the contents of the storage system if packs have been moved.

OPERATOR_COMMAND_CHANGES

The salvager is now built into the regular Multics system, and if the system detects an inconsistency while bootloading, the salvager will be invoked automatically. The operator may also request the salvage of the directory hierarchy or of any single volume by a command.

The disk configuration -- that is, which disk drives contain which physical volumes -- is not expressed on the CONFIG cards, and so must be conveyed to the system in another way. The system keeps a table of which volume is on which disk unit, called the disk_table, and on each bootload assumes that no volumes have been moved until told otherwise. In the normal case, the operator may start the system up without typing any commands which pertain to the disk configuration. For a cold boot, however, the operator must tell the system about the position of all volumes with the "add_vol" and "add_lv" commands, which tell the system to accept the volume if its label matches the registered attributes for the volume. Once the system has been told where a volume is, it assumes that it will not be moved.

Volume Registration

Before a storage system volume can be used it must be registered. There is a registration file for each logical volume. The registration file contains the data necessary to mount the logical volume correctly and to control its use. For example, the registration file of the logical volume "root" might look like this:

```
lvname: root
lvld:      220531524466
owner:     Initializer.SysDaemon
public:    yes
min_access_class: 0:000000
max_access_class: 7:777777
acs_path:  >lv>root.acs
```

```
npv:      2
```

```
pvname: rpv
pvid:     220531524345
serial:   2733-58
model:    400
location: online
```

```
pvname: root2
pvid:     570144045160
serial:   2733-59
model:    400
location: online
```

This is a sample of the output of the "list_vol_registration" command described in appendix B.

Since the registration file contains the name and model number for each physical volume in the logical volume, the system can tell whether a logical volume is completely mounted, or if there are enough free drives available to mount all its physical volumes.

ACCESS CONTROL

In each volume registration file, several pieces of information about access control are kept.

Public and Private Volumes

A volume may be either PUBLIC or PRIVATE. Public volumes are like pre-4.0 disk storage: segments on public volumes can be referenced by user programs subject to the normal ACL and ring checking. To access a segment on a private volume, the user must still have appropriate permission to the segment; but in addition, the user must have completed a "mount" operation for the private logical volume.

Access Control Segment

In order to mount a private volume, the user must have appropriate access to a special segment called the Access Control Segment (ACS) for the logical volume. The location of the ACS is specified in the volume registration file.

Access Isolation

For sites which use the Access Isolation Mechanism, each logical volume may have a range of sensitivity levels specified. Segments outside the given range cannot exist on any physical volume of the logical volume.

Master Directory Control

A user may create a master directory for a logical volume only if he has a Quota Account for the volume, registered in the volume's Master Directory Control Segment (MDCS). The MDCS lists all quota accounts for its logical volume, the master directories charged to them, and their quotas.

Volume Executive

Certain users may modify the MDCS for a logical volume. These users are called Volume Executives for the logical volume. Volume executives have "e" access to the ACS for the volume.

DATA BASES

The volume registration file and the MDCS are ring-1 segments stored in the directory >lv. The ACS for private volumes is stored in a directory named by the volume owner. For public volumes, the ACS is kept in >lv.

Because the system must be able to access the registration file of the root logical volume before it can accept the other physical volumes of the root, a special arrangement has been made. The registration data for the root logical volume is actually kept in segments in the root directory itself. Links are placed in >lv which point to the segments in the root directory.

If the system cannot locate a registration file for the root logical volume at startup time it recreates one, since the system requires this segment in order to operate.

REGISTERING VOLUMES

A new operator command, "add_vol_registration" (avr) is used to register all volumes except the RPV. The add_vol_registration command can be executed in ring 1 or ring 4. A sample call to it might look like this:

```
avr pv pub04 -lv public
```

In this example the physical volume named "pub04" is being registered as a member of the logical volume named "public."

Each physical volume must be registered before it is initialized. When the physical volume is initialized, registration information from the volume registration data is copied into the physical volume label. Whenever the physical volume is mounted, the physical volume label is checked against the registration data.

Many control arguments may be specified to the add_vol_registration command. A complete description of the command is provided in appendix B.

The "change_vol_registration", "del_vol_registration", and "list_vol_registration" ("cvr", "dvr", and "lvr") commands are also provided to change, delete, and list the volume registration data. These commands are also described in appendix B.

A special command is provided for regenerating the registration data for a physical volume if the registration data was destroyed in a crash. the command

```
reregister <pv_name> <drive>  
Example: reregister george dskb_04
```

will cause the system to read the label of the physical volume on the specified drive, and if its volume name is correct, to regenerate any missing parts of the registration data. This operation is performed automatically as part of startup for all volumes which are part of the RLV.

VOLUME INITIALIZATION

The "init_vol" command takes a blank disk pack and initializes it to be a Multics storage system physical volume by writing a label, an empty volume map, and an empty VTOC on the pack. Each volume must be initialized before it can be used by the storage system.

Only registered volumes can be initialized.

The init_vol command reads the label of the physical volume it is supposed to register before writing on it. If the label is a valid-appearing Multics label, for any other volume, init_vol will print an error message and refuse to proceed. This check is a safety feature to prevent the accidental destruction of the contents of a valid physical volume. In rare cases a pack may have a valid-appearing label from some previous application which prevents it from being relabeled. In such a case the pack should be erased first before init_vol is invoked.

Mounting Volumes

When the system is bootloaded, the hardcore system runs in the hardcore partition defined on the RPV by the PART HC card in the configuration deck. During file system initialization, the root directory is located and that section of the directory hierarchy which resides on the RPV is made available for the system. The root logical volume is defined at this time. Other PART cards may point to other disk units; if so, the partitions will be located by looking at the labels on the packs on these units but segments on these volumes are not made available for Multics use during hardcore initialization.

This is the situation when the system comes up in ring 1. Only the root logical volume is defined, and the only physical volume being used is the RPV. This is the only time that the system operates with a partially mounted logical volume.

The operator's normal action is to issue a series of add_vol commands to cause the other volumes of the root logical volume to be mounted, and then to issue add_lv commands and add_vol commands for additional logical volumes and the physical volumes that make them up.

The operator may list the disk table to find out which disk drives are mounted by issuing the "list_disks" ("ld") command. Its output looks like this:

```
dsk_a_01 root2      root
dsk_a_02 new        (blue)      *
dsk_a_03 (io drive)
dsk_a_04
dsk_a_05 (deleted)
dsk_a_06 true       (blue)      ***
dsk_a_07
dsk_a_08
dsk_a_09
dsk_a_10
dsk_a_11
dsk_a_12 rpv        root
dsk_a_13
dsk_b_01
dsk_b_02
dsk_b_03
dsk_b_04
dsk_b_05
```

In the example, the operator has issued the command

```
add_vol root2 dsk_a_01
```

to complete the root volume. The one star after the line for dsk_a_02 indicates that the pack on dsk_a_02 is assumed to be the physical volume "new" of the logical volume "blue." Since it is assumed, the system has not yet read its label. The three stars after the line for dsk_a_06 indicate that the pack on dsk_a_06 is known to be the physical volume "true" of logical volume "blue" -- the system has read the label of dsk_a_06 and it matches the registration for physical volume "true." The add_vol command makes a volume known, but does not make it available for paging unless an add_lv is pending and this add_vol completes the logical volume. The parentheses around the logical volume name indicate that the logical volume "blue" is unavailable for paging.

To cause all assumed and known physical volumes of a logical volume to be accepted for paging, the operator issues the add_lv command. If this command is issued when some physical volumes of the logical volume are not listed in the disk table, the system will type out mount messages for each physical volume and will then assume that the physical volumes will be mounted on the specified volumes. The operator will then type add_vol commands

for each physical volume which has been requested. When the last `add_vol` command is typed, the system will respond with a message saying that the logical volume is mounted.

In the example above, if the logical volume "blue" were registered to contain only the two physical volumes "new" and "true," the operator could issue the command

```
add_lv blue
```

to cause the label of `dska_02` to be read and checked, and to cause both physical volumes to be accepted for paging and the logical volume to be made available for user use. The system would type the message

```
public lv blue mounted
```

and the output of `list_disks` would then look like this:

```
dska_01 root2    root
dska_02 new      blue
dska_03 (io drive)
dska_04
dska_05 (deleted)
dska_06 true     blue
dska_07
dska_08
dska_09
dska_10
dska_11
dska_12 rpv      root
dska_13
dskb_01
dskb_02
dskb_03
dskb_04
dskb_05
```

physical volumes which are being used have no stars after their name.

If the system has been brought up after packs have been moved, the disk table may show certain assumptions which are no longer true. The `del_vol` (dv) command can be used to erase the entries for assumed or known physical volumes from the disk table. The `del_vol` command does not do anything to any disk drive; it just changes the table.

Default Startup Operations

At this point it may appear that the operator has to type an add_vol command for each physical volume and a add_lv command for each logical volume, in order to get the system started. Not so: default assumptions have been built into the system to eliminate extra commands when they are not necessary.

Whenever the Initializer process crosses from ring 1 to ring 4, as a result of the "startup", "multics", or "standard" commands, the system does an automatic "add_lv root" to insure that the root logical volume is complete. If the root cannot be completed, the system remains in ring 1, since otherwise it might attempt to access a directory which was not mounted.

Whenever the Initializer process attempts to start the answering service, as a result of the "startup" or "multics" commands, or when a "reload" command is issued, an automatic "add_lv -all" command is executed to attempt to mount all logical volumes which are not in use because one or more of their physical volumes is known or assumed. If any logical volume cannot be completed, the answering service startup is not attempted. The operator may then use the del_vol command to correct the disk table and try startup again, or may use add_vol to obtain the missing volumes.

These default assumptions mean that if the operator just types "startup" in ring 1, as he is used to doing, all volumes which were in use at the time of shutdown will be re-accepted and the system will resume operation. If the operator issues the command "BOOT STAR" to BOS, and if all physical volumes are where they were at the time of the last crash or shutdown, the system will not even pause in ring 1. But if there is any trouble, the system will wait for an operator command instead of plunging ahead blindly. As described below, the salvager has been integrated into this scheme so that if salvaging is necessary it is invoked automatically.

These changes are part of a general plan which will allow the system to be left in unattended mode, with a BOS runcom which will reboot the system if it crashes.

Demounting Volumes

A physical volume may be in one of the following states:

1. Not mounted
2. On the drive but drive not ready.
3. Drive ready, system unaware of contents.

4. Known (system has read label).

5. In use for storage system or I/O.

The add_vol and add_lv commands can move a physical volume to higher-numbered states. There are also commands for moving physical volumes back down.

The command

```
del_lv <lv_name>  
Example: del_lv blue
```

causes the system to make all segments on the logical volume <lv_name> unavailable to users, and then causes the orderly shutdown and unloading of each physical volume contained in the logical volume. As each disk unit is unloaded a message of the form

```
demounted dsk_a_02
```

will be printed. Since segments which users may be using are made unavailable by the del_lv command, the operator should use this command only when he is instructed to do so.

Since the system will encounter errors if a disk drive is stopped manually when it shouldn't have been stopped, the software is designed to support the following rule:

Never stop a disk drive manually while
Multics is in operation.

When the system demounts a disk pack, either a storage system physical volume or an I/O pack, it will always unload the disk drive by program control.

Changes to Old Operator Commands

adddev, addd	Now allowed in ring 1 as well as ring 4. This command may specify a storage system disk drive which has been deleted by the deldev command.
deldev, deid	Now allowed in ring 1 as well as ring 4. This command may specify a storage system disk drive. If the drive is currently in use for paging, an error occurs. Otherwise, the drive is made unavailable for storage system use.
force_reset	Name changed from "force".

storage, stor	Abbreviation changed from "dvm". Output format changed.
multics, mult	Now mounts volumes if necessary.
standard, stan	Now mounts volumes if necessary.
startup, star	Now mounts volumes if necessary.

NEW_SALVAGER

The storage system structure repair procedures known collectively as the salvager have been extensively modified in MR 4.0, both for automating crash recovery and for handling the structural changes in the storage system data. The salvager programs are now part of the standard system tape; new salvaging modes have been introduced and some old modes redefined; and control of the salvager is now from command options rather than from the processor switches. There are now four kinds of salvaging operation which the system can perform. These are emergency shutdown, volume salvage, directory hierarchy salvage, and on-line salvage. The system has been modified to invoke the salvager automatically whenever data in the storage hierarchy may have been damaged.

Emergency Shutdown

Emergency shutdown has been reworked completely and greatly improved. It is much more reliable than in pre-4.0 systems.

Sometimes the system crashes before the storage system has been turned on. Previously, an attempt to emergency shutdown in such a situation would lead to another crash, and might do serious damage to the directory hierarchy. This has been changed so that if a premature emergency shutdown is attempted, the following message is printed:

```
ESD BEFORE FS ENABLED  
SHUTDOWN COMPLETE
```

and the directory hierarchy is untouched.

It is always safe to attempt an ESD, and if emergency shutdown crashes it may be retried.

Volume Salvaging

Volume salvaging insures that there are no reused addresses or ill-formed VTOC entries in the VTOC and volume map of a single physical volume.

If a volume was in use by Multics and was not demounted or shut down by normal or emergency shutdown, it will be salvaged automatically the next time that the physical volume is accepted for paging.

The operator may force the volume salvage of a particular volume by typing the command

```
salvage_vol <vol_name> <drive> <options>  
Example: salvage_vol new dsk02 -check_dir
```

This operation cannot be done once the volume has been accepted for paging. Options may come from control arguments or from the SALV card. The "-check_dir" option causes the connection between the VTOC entry and the segment's directory entry to be checked; if there is no directory entry for the VTOC entry, the VTOC entry will be deleted.

To cause the volume salvage of all physical volumes which are assumed present or known but not yet in use, issue the command

```
salvage_vol -all <options>
```

A special interface is available to force the volume salvage of the Root Physical Volume. Typing

```
BOOT RPVS
```

will force the RPV to be salvaged during bootload. (Part of the directory hierarchy on the RPV will also be salvaged, as described below.) The RPVS option may be combined with any other command, so that

```
BOOT RPVS SALV
```

is a valid command, which calls for a volume salvage of the RPV, a salvage of the directory hierarchy existing on the RPV, and then a directory hierarchy salvage of all directories.

The "-check_dir" option for the salvage_vol command cannot be used on volumes which are part of the RLV. Volume salvaging can only be done on un-accepted volumes; but in order to do the -check_dir operation, the the RLV must be completely accepted so that all directories can be looked at. If the directory salvager destroyed a branch for a segment on the RLV, the "orphan" VTOC entry would never get cleaned up if Multics were unable to connection salvage. A special scheme is provided to do a salvage_vol of a copy of any physical volume of the RLV. Using BOS, the operator copies the physical volume or physical volumes to be checked onto scratch packs. After bootloading the system, the operator then types

```
salvage_vol <pv_name> <drive> -copy <drive2>  
Example: salvage_vol root2 dska_01 -copy dskb_05
```

changes, if any, are made to the copy, not to the version of the RLV pack that the system is running on. After all salvage operations have been done, the system can be shut down and the salvaged copies used to replace the old versions.

Hierarchy Salvaging

Hierarchy salvaging is the most like the pre-4.0 salvager: it corrects all errors in directories, rebuilding the directories if necessary. The directory hierarchy salvager is invoked by operator command by typing

```
salvage_dirs <options>  
Example: salvage_dirs -check_vtoce
```

The "-check_vtoce" option specifies that connection from the directory entry to the VTOC is to be checked; if a directory entry has no VTOC entry it will be deleted. The "-rebuild" option specifies that all directories should be rebuilt. Both of these options increase running time substantially.

RPV SALVAGING

Whenever a volume salvage is done on the RPV, the directory hierarchy salvager is invoked to salvage the root directory and every directory at up to two levels below the root if all the directory's parents are on the RPV. This operation is done in order to make sure that the system can complete bootload and come up to ring 1 command level.

RLV SALVAGING

Before leaving ring 1, the system automatically salvages directories down to level three of the directory hierarchy if any volume of the root logical volume was automatically volume salvaged. This action, called RLV salvaging, is taken because such a volume salvage may have detected a reused address and introduced a page of zeroes into a crucial directory.

On-Line Salvager

The on-line salvager is invoked by recovery mechanisms when certain software errors or hardware failures occur. It can also be invoked manually by a system programmer who calls

```
hphcs_$salvage_directory (path)
```

The MR 4.0 version of the on-line salvager has been substantially improved over previous versions.

Paging Device Flush

The paging device, if present in a configuration, will be flushed in stages as each volume is accepted. Except in cases of emergency, the following rule should be observed:

After a crash, try ESD.

If emergency shutdown does not complete, make sure that all volumes in use at the time of the crash are accepted during salvage.

Observing this rule insures that all information on the paging device will be flushed back onto the volume where it belongs.

The system will not exit from ring 1 until the paging device has been completely flushed.

NEW ERROR MESSAGESMultics Disk Error Messages

These messages have been changed to show the subsystem and drive number. The general error will look like this:

```
disk_control: dev attention for dsk7 4 (iom 1 chn 30) cmd 42
               stat 422000000000.
```

```
disk_control: dsk7 4 sect=765, cyl=14, hd=0, addr=13370.
```

```
disk_control: dsk7 4 detailed status: 40 00 00 80 00 00
```

The detailed status message will only be printed in cases where it is useful. If the disk unit is unusable, for example if it has gone into standby state, the message

```
disk_control: dsk7 4 requires intervention.
```

will be printed and the alarm will sound. The operator may ready the drive, after correcting the problem; the system will then type

```
disk_control: dsk7 4 now operative.
```

and attempt to use the disk unit again.

The other possible error messages about disk will not, in general, require operator intervention.

```
disk_control: IOM system fault for dsk7 4.
```

```
disk_control: Ignored special on dsk7 4.
```

```
disk_control: Interrupt from unknown devx 7, sx 7.
               status = 422000000000.
```

```
disk_control: Lost IOM interrupt for dsk7 4 (iom 1 chn 30).
```

```
disk_control: Unexpected IOM status 422000000000777000000000
               for dsk7 4 (iom 1 chn 30).
```

```
disk_control: dsk7 4 tally residue = 0001.
```

BOS Disk Error Message

The error message printed by BOS for a disk error is similar to the Multics message. It looks like this:

```
ERROR ON M400 A 30 4 (DSK7)
```

```
  CMD=32, S=3445, ADDR=666000, STAT= 422000 777000
DETAILED STATUS: 22F0000000004000
```

The subsystem name (DSK7 in the example) may be replaced by stars if BOS has not yet read the CONFIG deck.

Bootload Error Messages

If the operator attempts to boot the system with the clock set to an obviously bad value, the system will crash with one of these messages:

```
init_pvt: clock malfunctioning
init_pvt: incorrect clock setting
init_pvt: clock reads time before rpv label
```

this crash occurs before any changes are made to the disk contents.

An attempt to boot the system with a non-root volume where the RPV should be will usually result in a crash with one of the messages

```
init_pvt: no root
init_pvt: dska 3 has no Multics label
init_pvt: no part hc on root dska 3
```

the ROOT card must be corrected, or the contents of the RPV restored. Once again, no changes have been made to the disk contents.

CHANGES TO OPERATIONAL PROCEDURES

Several of the standard operational procedures are affected by the new storage system.

Standard SAVE

The system programmers will have set up some new RUNCOM files for the standard SAVE procedures. At least the following will be provided:

SAVET	Save all volumes to tape
RESTT	Restore all volumes from tape

When a new volume is added to the system, or any volume is moved from one disk unit to another, these RUNCOM files must be updated.

Standard Dumps

The regular incremental, catchup, and complete dumping procedures will not change too much with MR 4.0, but the installation management may choose to rearrange the dump control files.

Changing Disk Configuration

If the PRPH DSKx card is changed, the system is unable to assume anything about the location of any volumes on the next bootload. So after a change to the PRPH DSKx card, the operator will have to type an "add_vol" command for every volume in the system.

In order to add a new volume to an existing logical volume, the operator must shut down and reboot, and type an "initialize_disk" command for the new volume when ring 1 command level is reached.

Crash Recovery

Many internal changes have been made to Multics so that it will crash less often and make crashes which do occur easier to recover from.

PAGING DEVICE RECOVERY

After a system crash, the standard procedure is to attempt an ESD first. If ESD fails or cannot be attempted because core has been lost, do a

BOOT SALV

to cause the salvager to be run. If the ESD was not done, the following message will be printed when the salvager is bootloaded:

init_pvt: paging device will be flushed: 2040 records

If this message occurs, The paging device records for each volume will be flushed onto the volume as it is accepted for paging. (This operation takes a few extra seconds for each volume.) At the end of salvaging, if the operator types "startup", "standard", or "multics", the system will attempt to enable the paging device, and if successful will print the message

enabled pd: 2040 records

giving the size of the paging device. If some volumes, which were mounted at the time of the crash, don't get accepted during the next bootload (salvager or regular), the paging device may contain pages which must be flushed before the paging device can be used for regular paging. The system will refuse to enter normal operation if this is so, and will print

error: paging device unflushed. need 2 vols for 78 recs
Command:

The system will not enter standard mode (ring 4) until the paging device is completely flushed. In some cases, for example if a pack has a head crash, it may be impossible to flush the paging device completely; in other cases, for instance if a disk MPC goes down, several bootloads with different volumes on-line will be required to get all PD records flushed. Once all flushable records have been flushed, the operator must issue the command

force_pd_abandon

to clear the paging device of records which cannot be flushed.

DISK UNIT FAILURE

Often, only one disk pack or disk unit is involved in a system failure. The symptom that an operator will notice is a burst of write error messages for the drive, and then a message like

disk_control: dsk7 4 requires intervention.

with an audible alarm. The error messages may repeat periodically. In some cases, the system crashes when a disk unit fails; but often the system will continue operating although without the availability of segments on the device affected. If a disk drive has dropped into standby, and the system is still operating, the operator should attempt to ready it. If the drive can be made ready, the system will print

disk_control: dsk7 4 now operative.

and attempt to use segments on the drive again. If there has been a head crash or the disk unit has become unusable, the operator must shut down, substitute a spare or repaired disk pack and drive, and use RESTOR to restore the contents of the volume from the last SAVE by typing

```
RESTOR input_medium
      DSK7 4 ENTIRE USER_PACK_8
      END
```

and mounting the appropriate tapes or SAVE packs. The next step is to reboot the system, and reload the segments which have changes since the SAVE by issuing the following command:

```
reload -pvname user_pack_8
```

and allowing reload to scan all catchup and incremental tapes made since the SAVE. If the site has a paging device, and there are pages for the damaged volume on the paging device, it would not be safe to flush these pages back onto the new pack, even after reloading; so the system staff will probably have to zero the paging device as described above.

If several volumes are damaged, and some are from the RLV, these volumes should be reloaded before non-RLV volumes are attempted.

Checking Number of Records Left

The system handles out-of-disk much better; running out of disk on most volumes is no longer a system crash, although it may annoy users with segments on the volume.

The device_meters command no longer tells how many records are left. To get a summary of what volumes are on-line and how much room is left on them, type

```
list_vols
```

while in admin mode. The output is sorted by logical volume name. If the argument "-lv NAME" or "-pv NAME" is given, only

that physical or logical volume will be listed. The output looks like this:

Drive	Records	Left	VTOCs	Left	PV name	LV name
dsk_a_02 P	17752	3688	2817	1850	rpv	root
dsk_a_04 P	17752	3333	2817	1865	cisl2	root
dsk_a_03	18322	4102	4135	2438	ldd	libraries

The "P" flag above indicates those drives containing partitions. If the "-total" argument is given, the output is

Records	Left	VTOCs	Left	LV Name
18322	4102	4135	2438	libraries
35504	7021	5634	3715	root

and the operator "storage" command now does a "list_vols -tt".

APPENDIX A

LIST OF OPERATOR COMMANDS

The following list shows all ring-1 and ring-4 commands. Unless otherwise noted in the "Ring" column, the command is for ring-4 only.

<u>Command</u>	<u>Abbrev</u>	<u>Ring</u>
abs	abs	
accept	acce	
add_iv	alv	NEW both
add_vol	av	NEW both
add_vol_registration	avr	NEW both
addcpu	addc	
adddev	addd	CHG both
addmem	addm	
addpage	addp	
admin		
attach	atta	
bos		both
bump		
change_vol_registration	cvr	NEW both
cripple	crip	
define	defl	
del_iv	div	NEW both
del_vol	dv	NEW both
del_vol_registration	dvr	NEW both
delcpu	delc	
deldev	deld	CHG both
delmem	delm	
delpage	delp	
deroute	dero	
detach	deta	
develop	deve	
down		
drop		
exec	x	
force_reset		CHG
force_pd_abandon		NEW ring-1
go		
help		ring-1
hmu		
init_vol		NEW both
intercom	lc	
list_disks	ld	NEW both
list_vol_registration	lvr	NEW both
login	logi	
logout	logo	
maxunits	maxu	
mc_list		
mc_on	mc	

message	mess	
multics	mult	CHG both
no_start	ns	
quit		
rcp		
redefine	rede	
reload		both
reload_system_release		ring-1
remove	remo	
reply	r	
reregister		NEW ring-1
reroute	rero	
reset	rese	
route	rout	
salvage_dirs	salv	NEW ring-1
salvage_vol	sv	NEW both
shutdown	shut	both
standard	stan	CHG ring-1
startup	star	CHG both
stop		
storage	stor	CHG
substty	subs	
sysid	sysi	
terminate	term	
unbump	unbu	
undefine	unde	
warn	w	
who		
word		

APPENDIX B

COMMAND DOCUMENTATION

Only new commands are written up in this section. Consult the Multics Operator's Handbook for other commands.

Name: add_vol_registration, avr

Function: Register a new physical volume. If the logical volume does not already exist, register the new logical volume also.

Syntax: add_vol_registration pv pv_name control_args

Arguments:

pv	is the literal string "pv"
pv_name	is the physical volume name

Control Arguments:

-lv LVNAME	Specifies the logical volume name. This argument must be provided.
-model MODNO	Specifies the Honeywell model number for the disk pack. If omitted, the default is 400. The following are valid values for MODNO: 181 DSU181 191 DSS191 400 MSU0400 450 MSU0450
-serial SER	Specifies the manufacturer's serial number for the pack. If this argument is omitted, the default is "none."

The remaining arguments are valid only when a new logical volume is being registered. Otherwise they are ignored.

-owner Person.Project	Specifies the user identifier of the volume owner. If omitted the Person.Project of the user executing the command is used.
-access_low xxx	Specifies the minimum sensitivity level for information on the volume.
-access_high xxx	Specifies the maximum sensitivity level for information on the volume.
-public YESNO	Specifies whether the volume is public. If omitted the default is "yes."
-acs PATH	Specifies the path name for the ACS.

Name: change_vol_registration, cvr

Function: Modify logical volume or physical volume registration.

Syntax: change_vol_registration pv pvname control_args
change_vol_registration lv lv_name control_args

Arguments:

pv	is the literal string "pv"
pvname	is a physical volume name
lv	is the literal string "lv"
lv_name	is a logical volume name

Control Arguments: The following control arguments are valid when "pv" is specified:

-model MODNO Specifies the Honeywell model number for the disk pack. The following are valid values for MODNO:

181	DSU181
191	DSS191
400	MSU0400
450	MSU0450

-serial SER Specifies the manufacturer's serial number for the disk pack.

-location LOC Specifies the current location of the disk pack.

-date_registered DATE Specifies the date the physical volume was registered.

-name PVNAME Specifies the physical volume name.

-pvid PVID Specifies the unique identifier of the physical volume.

The following arguments are valid when "lv" is specified:

-owner Person.Project Specifies the user identifier of the volume owner.

-access_low xxx Specifies the minimum sensitivity level for information on the volume.

-access_high xxx Specifies the maximum sensitivity level for information on the volume.

-public YESNO Specifies whether the volume is public.

-acs PATH Specifies the path name for the ACS.

-name LVNAME specifies the logical volume name.

-lvid LVID Specifies the unique identifier of the volume.

Notes: Changing the name or unique ID attributes of a physical volume or logical volume is an operation which is only done to correct a wrong registration file. The volume label is not affected by the change to the registration file, and simply changing the registration file will usually have only the effect of making the volume or logical volume unmountable

Name: del_vol_registration, dvr

Function: Unregister logical volume or physical volume.

Syntax: del_vol_registration pv pvname
del_vol_registration lv lv_name

Arguments:

pv	is the literal string "pv"
pvname	is a physical volume name
lv	is the literal string "lv"
lv_name	is a logical volume name

Notes:

When "pv" is specified, the command will refuse to leave a logical volume with no physical volumes in it.

When unregistering a logical volume, the physical volumes being released will be listed.

Name: list_vol_registration, lvr

Function: Print registration information for logical volume or physical volume.

Syntax: list_vol_registration pv pvname
list_vol_registration lv lv_name control_arg

Arguments:

pv	is the literal string "pv"
pvname	is a physical volume name
lv	is the literal string "lv"
lv_name	is a logical volume name

Control Arguments: If "lv" is specified, the "-brief" ("-bf") argument lists only physical volume names.

Example:

```
list_vol_registration lv root
  lvname: root
  lvid:      220531524466
  owner:     Initializer.SysDaemon
  public:    yes
  min_access_class: 0:000000
  max_access_class: 7:777777
  acs_path:  >lv>root.acs

  npv:      2

  pvname:   rpv
  pvid:     220531524345
  serial:   2733-58
  model:    400
  location: online

  pvname:   root2
  pvid:     570144045160
  serial:   2733-59
  model:    400
  location: online

list_vol_registration lv root -bf
  rpv
  root2
```

Name: add_vol, av

Function: Tell the system that a physical volume is on a disk drive. Read and check the label.

Syntax: add_vol pvname drive_name
add_vol -all

Arguments:

pvname	is the name of a physical volume
drive_name	has the form <subsys>_<nn>

Control Arguments:

-all	Mount all assumed physical volumes
------	------------------------------------

Notes: The registration file for <pvname> is consulted to find out the logical volume name and to validate the label.

The volume label is read and checked. If the label is correct the drive is left in the known state.

If an add_lv command for the logical volume containing this physical volume has been previously issued, and if this add_vol command completes the logical volume, the logical volume is accepted for paging and a message is printed.

Name: add_lv, alv

Function: Cause a logical volume to be accepted for paging.

Syntax: add_lv lv_name
add_lv -all

Arguments:

lv_name is the name of a registered logical volume

Control Arguments:

-all Mount all incomplete logical volumes.

Notes: If all physical volumes for the logical volume are already accepted, known, or assumed, add_lv will insure that all labels have been read and checked, and will then accept the logical volume for paging.

If one or more volumes are missing, add_lv will assign drives for the missing volumes, print mount messages for each missing volume in the form

```
mount pv k202 on dska_07
```

and set the assigned drives into the assumed state. The operator may mount the physical volumes where requested or use his own judgement. As each physical volume is mounted, the operator will issue an add_vol command to the system; the last add_vol command will cause the logical volume to be accepted with a message of the form

```
private lv cp22 mounted
```

When the add_lv command is issued, all assumed drives should have the correct contents; otherwise an error message will be typed.

The list_disks command lists all outstanding add_lv's.

The del_lv command can cancel an outstanding add_lv.

Name: del_vol, dv

Function: Tell the system that a physical volume is no longer on a drive.

Syntax: del_vol drive_name

Arguments:

drive_name has the form <subsys>_<nn>

Notes: The drive specified may not be in use. See the del_lv command for demounting logical volumes which are in use.

Name: del_iv, div

Function: Tell the system to force the demounting of a logical volume.

Syntax: del_iv lvname

Arguments:

lvname specifies the name of a logical volume

Notes: If the logical volume is in the process of being mounted, this command cancels the request.

If the logical volume is already mounted, this command makes the segments on the volume unavailable to all users, and shuts down and unloads all physical volumes in the logical volume with a message of the form

demounted dska_02

Name: init_vol

Function: Write the label of a new physical volume. Set up its VTOC and volume map. This operation destroys any previous contents of the physical volume.

Syntax: init_vol pvname drive_name control_arg

Arguments:

pvname	is the name of a physical volume
drive_name	has the form <subsys>_<nn>

Control Arguments:

-special	Enter dialogue with the operator.
----------	-----------------------------------

Special Mode:

In special mode the operator may specify the position and size of partitions on the physical volume.

The following request lines may be typed

part NAME HILOW SIZE Where NAME is a 4-character partition name, HILOW is either "high" or "low", and SIZE is the partition size in records.

avg FFF.FF Specifies the average segment length for segments on the physical volume. The default length is 5.0. This number is used to calculate the number of VTOC entries on the volume.

list lists the partitions defined so far.

end causes the physical volume to be initialized. This operation takes about 90 seconds.

quit causes an exit without doing anything.

Name: list_disks, ld

Function: List the usage of storage system disk drives and all pending add_lv's.

Syntax: list_disks arg

Arguments:

drive_name is the name of a drive to be listed

Control Arguments:

-lv LVNAME list only drives pertaining to LVNAME
-pv PVNAME list only drives pertaining to PVNAME
-brief, -bf list only nonempty drives
-mount, -mt list only pending add_lv's

Example:

list_disks

```
dsk_a_01 root2      root
dsk_a_02 new        (blue)      *
dsk_a_03 (lo drive)
dsk_a_04
dsk_a_05 (deleted)
dsk_a_06 true       (blue)      ***
dsk_a_07
dsk_a_08
dsk_a_09
dsk_a_10
dsk_a_11
dsk_a_12 rpv        root
dsk_a_13
dsk_b_01
dsk_b_02
dsk_b_03
dsk_b_04
dsk_b_05
public lv blue mount in progress
```

Name: reregister

Function: Regenerate volume registration information.

Syntax: reregister pvname drive_name

Arguments:

pvname is the name of a physical volume
drive_name has the form <subsys>_<nn>

Notes: This command reads the label of the volume on the specified drive name. If it is a valid label for the volume whose name is given, the registration data is refabricated from the label and the drive is left in the known state.

Name: force_pd_abandon

Function: Discard all data on the paging device.

Syntax: force_pd_abandon

Arguments: none

Notes: This command destroys information. It should be used only after a system catastrophe which has left pages on the paging device which cannot be flushed; and the command should not be invoked until all physical volumes which can accept data from the paging device have been flushed.

Name: salvage_dirs, salv

Function: Salvage the directory hierarchy.

Syntax: salvage_dirs control_args

Arguments: none

Control Arguments:

-check_vtoce	Delete branch if VTOCE missing
-rebuild	Rebuild all directories
-pathname	Print pathname being processed
-long	-check_vtoce -rebuild -pathname
-console	Output to SYSERR console instead of printer
-dump	Dump damaged objects
-debug	System programmer use only
-level N	Salvage levels 0-N only

Notes:

The SALV card may specify some default options.

Name: salvage_vol, sv

Function: Volume salvage a physical volume.

Syntax: salvage_vol pvname drive_name control_args
salvage_vol -all control_args

Arguments:

pvname	is the name of a physical volume
drive_name	has the form <subsys>_<nn>

Control Arguments:

-console	Output to SYSERR console instead of printer
-check_dir	Delete VTOCE if no branch
-copy DRIVE	Salvage inactive copy on DRIVE
-dump	Dump damaged objects
-debug	System programmer use only

Notes: If -all is specified, all known and assumed physical volumes are salvaged.

An accepted physical volume cannot be salvaged.

The -copy option is used to salvage a copy of an active volume, usually because the operator wishes to use the -check_dir option on a volume of the RLV.

The SALV card may supply some default options.