

To: Distribution
From: D.R. Vinograd
Date: 08/20/76
Subject: Volume Backup Documentation

The attached draft documentation comprises the documentation package, less a PLM, for the Volume Backup Subsystem.

This includes MPM command and reference section documentaion, MOH command and reference section documentation and MAM reference section documentation.

Additional design documentation can be found in MTB-232.

Your observations and comments will be appreciated.

SAM OVERVIEW-VOLUME BACKUP

The Multics volume backup system consists of three separate subsystems: the volume dumping, volume reloading, and volume retrieval subsystems. In combination, these subsystems provide a low-cost, easily controlled method of assuring the safety of online information and recovery of segments and directories that may be lost or damaged.

VOLUME DUMPING

The volume dumping subsystem operates in one of three ways:

incremental volume dump--

For each physical volume specified, those segments or directories that reside on it are dumped if they have been modified since the last incremental volume dump.

consolidated volume dump--

For each physical volume specified, those segments or directories that have been incrementally dumped since the last consolidated volume dump are dumped.

complete volume dump--

For each physical volume specified, all segments and directories are dumped.

Volumes are specified via a control segment as either physical volumes or logical volumes. Logical volumes are converted into a list of physical volumes.

The incremental volume dumper operates cyclically using a default time interval of one hour unless otherwise specified. That is, the dumper process is activated at one-hour intervals, incrementally dumps all physical volumes specified in the dump control segment, and then waits for the next wakeup. If the real time required to complete a dump cycle exceeds the wakeup interval, the dump cycle repeats immediately. The incremental volume dumper should be run whenever the Multics system is operational. Incremental volume dumping may be partitioned among several different processes, if desired, by specifying different physical volumes in different control segments. The dump control

segment should specify at least all public volumes and may specify any registered volume. If the volume is not mounted, it is skipped.

The consolidated volume dumper normally operates as a single pass dump, using the same control segment as the incremental volume dump. It is effectively a merge operation of all the incremental volume dumps produced since the last consolidated volume dump. It is not a system requirement that it be run, but its operation significantly reduces the amount of input that must be scanned during a recovery operation.

The complete volume dumper also operates as a single pass dump but is nonselective in what it dumps; that is, it dumps everything.

The incremental and consolidated volume dumpers are controlled by bit maps that the system maintains on a per physical volume basis. These bit maps designate the VTOC entries on a physical volume that are to be dumped. The complete dumper constructs a temporary bit map of the VTOC entries currently in use before it starts dumping. In both cases, the object described by the VTOC entry is accessed in cooperation with the supervisor in a manner that bypasses the storage system access control list and ring control.

What is dumped is controlled by the volume specifications in the dump control segment, and whether the owner of the segment has enabled/disabled the incremental and complete volume dumping switches. If both of these switches are off, the segment is not dumped and cannot be recovered.

As the volume dumper operates, it records information about its dumping in a segment(s) referred to as the volume log. There is one valid volume log for each physical volume that has been dumped. If the volume log cannot be found, it is created. The volume log contains a record of every tape that has information on it that was dumped from this physical volume. The set of tapes necessary to logically reconstruct the physical volume is referred to as a reload group. Normally, a volume log contains two reload groups. The `display_volume_log` and `purge_volume_log` commands (see MPM XXX) can be used to display the contents of a volume log and set the number of reload groups it contains, respectively. The uses of the volume log segment are more fully explained during the discussion of the volume reloader and retriever later in this section.

As each tape is created, the unique identifiers of the objects written on it are stored in a segment referred to as the

contents segment. The contents segment is written on the next dump tape (of the same dump mode) and can then be deleted if necessary. Its use is explained more fully during the discussion of volume retrieval.

In order to determine (at any time) the name of a previous contents segment, a history of previous dump tapes is maintained in the `prev_volume_log` segment.

The structure of the dumper's output is determined by the I/O module used to create it. In the default case, it is a magnetic tape written in Multics standard tape format. The order of data on a tape is as follows: dump info segment, contents segment of previous tape in dump series, and a set of objects dumped from some specified physical volume preceded and followed by the volume log of that physical volume (this last item is repeated for each physical volume dumped).

VOLUME RELOADING

The volume reloader is invoked to reconstruct the contents of a physical volume. It uses as input some of the tapes created by all three modes of volume dumping and the volume log segment. When invoked, the volume reloader determines from the volume log the set of tapes that defines a reload group. Input is taken in reverse chronological order so that once an object is recovered, subsequent copies can be skipped. When a volume reload has completed, the resultant physical volume is a logical image of its former self, less any changes that were introduced by the operator (as described below), changes that were not recorded by the dumper, or changes that were unreadable because of input errors.

The disk pack used by the volume reload facility is accessed as an I/O disk and is assumed to have been initialized via the `init_vol` initializer command. The parameters that describe the organization of the pack are compared with those stored in the volume log and any mismatches are reported to the operator. Volume reloading of any physical volume can be carried out while the system remains operational for users, although the logical volume that contains the physical volume to be reloaded is not available. Volume reloading of a physical volume that is part of the root logical volume (RLV), but not the root physical volume (RPV), is accomplished by bringing the system up to ring 1 initializer command level, prior to accepting any volume, and issuing the `reload_volume` command. If the volume to be reloaded

is the RPV, the system must be cold booted to ring 1 command level using a spare disk pack and the latest copy of the RPV volume log must be recovered using the `recover_volume_log` command (see MPM XXX). Once this is done, the volume reload can proceed.

When a volume reload is complete, in all cases except that of the RPV, the physical volume can normally be used as a direct replacement by remounting the logical volume, adjusting the disk table, or putting the new disk pack on the old disk drive. In the RPV case, additional steps that must be taken are: shut the system down; either move the disk pack to the drive specified on the root config card, or change the card; and reboot BOS if it is was in the RPV.

MOH OVERVIEW-VOLUME BACKUP

The Multics volume backup system consists of three separate subsystems: the volume dumping, volume reloading, and volume retrieval subsystems.

VOLUME DUMPING

The volume dumping subsystem produces dumps, usually on magnetic tape, which are used by the volume reloading and retrieval subsystems. The volume dumping subsystem operates in one of three ways:

incremental volume dump--

For each physical volume specified, those segments or directories that reside on it are dumped if they have been modified since the last incremental volume dump.

consolidated volume dump--

For each physical volume specified, those segments or directories that have been incrementally dumped since the last consolidated volume dump are dumped.

complete volume dump--

For each physical volume specified, all segments and directories are dumped.

As the dumper runs, it records information about its operation for use by the volume reloader and retriever subsystems. This information consists of the following segments (in the working directory):

previous output volume log segment--

Contains information defining the previous tape identifiers used by the incremental, consolidated, or complete volume dumper. This segment is named `prev_volume_log`.

volume log segment--

Contains a record of all tapes containing information copied from the physical volume. This segment is referred to as the volume log segment. A volume log segment is created for each physical volume dumped; the name of this segment is the name of the physical volume.

contents segment--

Contains a unique identifier for each segment or directory written on a tape. This segment is referred to as the contents segment. A contents segment is created for each tape used; the segment name is the tape identifier.

The volume log segment can be displayed using the `display_volume_log` command (see MPM XXX). If lost, it can be recovered using the `recover_volume_log` command (see MPM XXX).

The volume log segment is used by both the volume reloader and retriever subsystems to determine which tapes created by the dumper should be used as input. The contents segment is used by the volume retriever to bypass searching a dump tape. The `prev_volume_log` segment is used by the volume dumper to locate the contents segment of the previous tape.

VOLUME RELOADING

The volume reloader creates a control segment (in the working directory) for each physical volume that it reloads. The control segment is given the name of the physical volume plus the suffix, "control"; e.g., `dska_1.control`. This control segment contains information about already reloaded objects and allows the reload operation to be restarted.

The disk pack used by the reloader must have been initialized by the `init_vol` command (see MPM XXX). If the disk pack log differs in any way that would preclude its use or damage existent information, the operator will be warned.

MPM OVERVIEW-VOLUME BACKUP

The Multics volume backup system provides the user with a controlled, low-cost method for the preservation and recovery of segments or directories that may be lost or damaged.

The volume backup system is run by system operations as a system service. The volume dumping subsystem operates in one of three ways:

incremental volume dump--

For each physical volume specified, those segments or directories that reside on it are dumped if they have been modified since the last incremental volume dump.

consolidated volume dump--

For each physical volume specified, those segments or directories that have been incrementally dumped since the last consolidated volume dump are dumped.

complete volume dump--

For each physical volume specified, all segments and directories are dumped.

Dumping is done on a physical volume basis and is normally not done for private volumes unless special arrangements have been made.

Segment dumping is controlled by two user-settable switches: the incremental and complete dump switches, indicating the segment is to be dumped if encountered during that type of dump. These switches are set/reset using the `dump_switch_on` and `dump_switch_off` commands (see MPM XXX).

The default settings for these switches are as follows:

For directories--
Does not apply.

For segments--

Both switches on except for object and listing segments created by the following commands (in which case both switches are off):

pl1
fortran
~~basic~~
alm
~~runoff~~
bind

.list only

These defaults are chosen to decrease the number of re-createable segments that are dumped, and thus decrease the system resources used by backup. The user can override the defaults if he chooses.

If a segment or directory is lost or damaged, the user can request recovery via the `enter_retrieval_request` command (see MPM XXX). If the object recovered is not correct, an earlier copy can be recovered by reissuing the command.

MOH VOLUME RELOADER

VOLUME RELOADER

The volume reloader facility uses as input the tapes produced by the volume dumper facility. It is used to reconstruct the contents of a physical volume. This can normally be done while the system is available to users except when the physical volume to be reconstructed is part of the RLV. For more information about this special case, refer to the XXX.

The volume reloader requires as input a preinitialized disk pack. For more information on how to initialize a disk pack, see the description of the `init_vol` initializer command in "XXXX". The disk pack must be initialized with the name of the physical volume that is to be rebuilt.

The volume reloader uses the I/O disk facility of the system to read and write the disk pack it is rebuilding. Thus, an IO disk drive must be configured and available (see "XXXX").

The volume reloader normally specifies the name and order of tapes to be used. If the reload is to succeed, the order must be followed.

The volume reloader can run in any user ring, at any AIM level, but it is normally run at a system high AIM level in ring 1. The volume reloader requires re access to the following special gates:

```
hc_backup_  
rcp_sys
```

The volume reloader uses the `volume_log` for the specified physical volume. It searches for the volume log in its working directory; if not found, the reload aborts. For more information, refer to the `recover_volume_log` command (see MPM XXX).

The volume reloader is invoked via the reload_volume command as follows:

```
reload_volume -pvname <name> -operator <operator name> {-control_args}
```

where:

-pvname pname
specifies the name of the physical volume to be reloaded. It must be specified.

-operator <operator name>
specifies the name of the operator doing the reload. It must be specified.

-disk dtype
where dtype is the type of disk being reloaded. Type must be one of the following:

```
m500  
m400  
m451  
d191  
d181
```

If not specified, the registration information for physical volume pname is used.

-debug, -db
specifies that the reloader is operating in debug mode. The default is off.

-error_on
specifies that error messages are written to the switch error output as well as the file:

```
error_file.mm/dd/yy.hhmm.s
```

The default is off.

-output_volume_desc o_att_desc
where o_att_desc is an attach description for the output media the reloader writes to. The default is:

```
"rdisk_ dtype pname dtype -write-sys"
```

-input_volume_desc i_att_desc
where i_att_desc is an attach description for the input media the reloader reads from. The default is:

```
"tape_mult_ medium_id -sys"
```

- no_detach
specifies that at the completion of the reload neither the input or output media are detached. The default is off.
- no_object
specifies that data objects are not read from the input media and thus not written to the output media. In this case only the VTOC is reloaded. The default is off.
- restart, -rt
specifies that reload is to be restarted using control information contained in the segment pname.control in the working directory. This control argument should only be used if a system failure occurs during a reload sequence. The default is off.
- manual
specifies that the input media will be selected by the operator. To terminate the media request, the operator should type a period (.).
- rpv
specifies that the pack being reloaded is the root physical volume and certain fields in the volume label should be set.

MOH VOLUME DUMPER

VOLUME DUMPER

The volume dumper creates copies of data objects in the storage system hierarchy on an output medium, normally magnetic tapes.

The volume dumping subsystem operates in one of three ways:

incremental volume dump--

For each physical volume specified, those segments or directories that reside on it are dumped if they have been modified since the last incremental volume dump.

consolidated volume dump--

For each physical volume specified, those segments or directories that have been incrementally dumped since the last consolidated volume dump are dumped.

complete volume dump--

For each physical volume specified, all segments and directories are dumped.

What is dumped is controlled by a segment that contains the names of logical or physical volumes. The format of this segment is as follows:

lv, <logical volume name> or
pv, <physical volume name>

Each line must specify no more than one name and may not have any blanks. Logical volume names are translated into a list of physical volume names. Physical volumes are dumped in the order that they appear in the control list.

The incremental dumper operates in cyclic manner. The cycle time is specified as a control argument (see below), or defaults to one hour. At completion of a volume dump cycle, the incremental volume dumper goes inactive unless it has not completed within the cycle time, whereupon it immediately restarts. The complete and consolidated dumpers are single pass operations.

The volume dumper can operate in any user ring and at any AIM level, although it normally operates in ring 1 at an AIM "system high" level. The volume dumper requires re access to the following gates:

```
hc_backup_  
hcs_  
rcp_sys_  
mdc_  
mdc_priv_  
rcp_
```

Access to phcs_ and hphcs_ is not required.

The three types of volume dumping--incremental, consolidated, and complete--are invoked as separate commands with similar control arguments. Thus, to start an incremental volume dump one might issue the command:

```
incremental_volume_dump -control <control_segment_path_name>  
-operator <operator name> {-control_args}
```

To start a complete or consolidated volume dump, the command name is changed to complete_volume_dump or consolidated_volume_dump.

The control arguments that can be used are as follows:

- control <name>
where name is the pathname of the control segment. The suffix "dump" is assumed if not specified. This must be specified.
- operator <name>
where name is the name or initials of the operator. This must be specified.
- wakeup DT
where DT is a wakeup interval in minutes for the incremental dumper. The default is 60 minutes.

- error_on
specifies that all error messages are to be written to the switch error_output as well as to the file error_file dd/mm/yy/hhmm.s. The default is off.
- debug, -db
specifies that the dump is to be run in debugging mode. The default is off.
- output_volume_desc atl_desc
where atl_desc is used instead of the default attach description.

"tape_mult_ <reel_id> -wrt -sys"
- restart pvname, -rt pvname
where pvname is a physical volume name, implicitly or explicitly defined in the control file, where the dump should start from. In the incremental mode, this only occurs in the first pass. The default is off.
- mod_after DT
where DT is a time acceptable to the convert_date_to_binary_ subroutine. In the complete volume dump mode, if -modafter is specified, those data objects whose dtm exceed DT are volume dumped.
- no_detach
specifies that at the completion of a dump the output medium is not to be detached. The default is off.
- no_object
specifies that data objects are not to be dumped. Only the header information that describes them is dumped. The default is off.
- no_update, -nud
specifies that the vtoce of data objects dumped is not to be updated. This speeds up dumping at the cost of invalidating the dtd and volume id fields of the vtoce. The default is off.
- reset, -rs
specifies that for incremental and consolidated dumping, the object dumped is not to be removed from the list of objects to be dumped next time. This provides cumulative dumping if desired. The default is off.

display_volume_log

display_volume_log

Name: display_volume_log

The display_volume_log command displays a summary of the output volumes, usually magnetic tape reels, that contain information dumped from a specified physical volume.

Usage

display_volume_log pvname {-control_args}

where:

1. pvname
is the name of a requested physical volume.
2. control_args
are chosen from the following:
 - no_header, -nhe
specifies that the output header is not to be printed. The default prints the output header.
 - valid valid
specifies that only the information pertinent to the tape valid is to be printed.
 - mode, -md (incr| cons| comp)
specifies that information about tapes of the dump mode indicated is to be displayed. If not specified, information about all dump modes is displayed.
 - from DT, -fm DT
specifies that only information about tapes created after time DT is to be printed. Time DT must be a time acceptable to the convert_date_to_binary_subroutine. If not specified, the earliest information is used.
 - to DT
specifies that only information about tapes completed before time DT is to be printed. Time DT must be a time acceptable to the convert_date_to_binary_subroutine. If not specified, the latest information is used.

dump_switch_off

dump_switch_off

Name: dump_switch_off, dsf

The dump_switch_off command turns off the specified dump switch of a segment. If this switch is off, the segment is not dumped during that mode of dumping and thus may not be available for reloading and retrieval. The user is cautioned not to turn off both dump switches unless the segment is easily re-createable, is a temporary segment that exists outside the process directory, or is privately backed up by some other means.

Usage

dump_switch_off paths {-control_args}

where:

1. paths
are the pathnames of segments whose dump switches are to be turned off. The star convention is accepted.
2. control_args
are chosen from the following:
 - incremental, -incr
specifies that the incremental dump switch is to be turned off.
 - complete, -comp
specifies that the complete dump switch is to be turned off.

dump_switch_on

dump_switch_on

Name: dump_switch_on, dsn

The dump_switch_on command turns on the specified dump switch of a segment. If this switch is not on, the segment is not dumped during that mode of dumping, and thus cannot be reloaded or retrieved.

Usage

dump_switch_on paths {-control_args}

where:

1. paths
are the pathnames of segments whose dump switches are to be turned on. The star convention is accepted.
2. control_args
are chosen from the following:
 - incremental, -incr
specifies that the incremental dump switch is to be turned on.
 - complete, -comp
specifies that the complete dump switch is to be turned on.
 - all, -a
specifies that both the incremental and complete dump switches are to be turned on.

Name: enter_retrieval_request, err

The enter_retrieval_request command queues retrieval requests for specific segments, directories and subtrees. The user must have effective write access or modify permission to an object in order to retrieve it. If an object has been deleted, then append permission on the containing directory is also required.

Usage

enter_retrieval_request path {-control_args}

where:

1. path
is the pathname of the segment, directory or node of a subtree that is to be retrieved. The star convention is not supported.
2. control_args
are chosen from the following:
 - all, -a
specifies that the subtree inferior to the directory specified in path as well as the directory is to be retrieved. If, after the directory is retrieved the subtree is found intact, it is not retrieved. The default is off. See "Notes" for more information.
 - queue n, -q n
queues requests in priority queue n. The default is queue 3.
 - newest DT
specifies that the search for path and all inferior branches, if specified, proceeds from time DT backwards. Thus, objects dumped later than time DT are not recovered. Time DT must be acceptable to the convert_date_to_binary_subroutine. If the control argument is not specified, time DT is assumed to be the start of the retrieval operation.

enter_retrieval_request

enter_retrieval_request

-oldest DT

specifies that the search for path and all inferior branches, if specified, stops at time DT. Thus, objects dumped before time DT are not recovered. Time DT must be acceptable to the `convert_date_to_binary_subroutine`. If the control argument is not specified, all valid dump tapes are searched.

-notify

specifies that the user wants to be notified of the success or failure of the request. The secure mail facility is used. The default is not to notify.

Notes

In certain cases where a directory is damaged, the inferior subtree may be unavailable until the directory is recovered. When a directory is recovered, and the `-all` control argument is specified, a check is made to see if the subtree is available, and if so, retrieval is assumed complete.

The pathname of the segment or directory to be retrieved need not be specified as a set of primary names. Any set of valid entrynames is acceptable.

merge_volume_log

merge_volume_log

Name: merge_volume_log, mvl

The merge_volume_log command merges the contents of two volume logs that describe the same physical volume. It should only be used to merge the segment produced by a recover_volume_log operation with the volume log already in existence.

Usage

merge_volume_log path1 path2

where:

1. path1
is the pathname of the old volume log. After path1 and path2 are combined, the merged copy is given the name of the old volume log (path1).
2. path2
is the pathname of the new volume log. After path1 and path2 are combined, path2 is deleted.

purge_volume_log

purge_volume_log

Name: purge_volume_log

The purge_volume_log command is used to remove, in a consistent manner, those entries in a volume log that represent out-of-date information. Each entry in a volume log is part of a reload group. A reload group consists of a set of tapes that would be used during a reload of the physical volume. Normally, only two reload groups are kept in a volume log, although this can be changed (see below).

Usage

purge_volume_log pname {-control arg}

where:

1. pname is the name of the volume log.
2. control_arg can be -group n to specify that n reload groups are to be retained. The default is 2.

recover_volume_log

recover_volume_log

Name: recover_volume_log, rvl

The recover_volume_log command recovers a volume log from a dump tape. It should be invoked only if the volume log for the physical volume to be reloaded cannot be found. Its input is the latest tape for the physical volume in question as specified by the caller.

Usage

recover_volume_log pvname

where pvname is the name of the physical volume whose volume log is missing.

retrieve_volume

retrieve_volume

Name: retrieve_volume

The retrieve_volume command is used to process queued retrieval requests. It can be operated in an automatic or manual mode. It must be run from a process that has access to retrieval queues located in >system_control_1>backup and to the ring 0 gate hc_backup_.

Usage

retrieve_volume {-control_args}

where control_args are:

- queue n, -q n
specifies the queue that requests are to be taken from. If not specified, the default is queue 3.
- manual
specifies that each request is to be printed on the terminal before processing so that the operator can decide if he wishes to proceed. If not specified, each request is processed until the queue is exhausted. See "Notes" for more information.
- list
lists all requests in the specified queue.

Notes

If the -manual control argument is specified, each retrieval request is printed and the retriever waits for the operator to type one of three commands:

- proceed, p do the request and delete request from queue.
- skip, s skip the request but leave in queue.
- cancel, c do not process the request and delete it from the queue.