

# Hierarchical Storage Management for OpenVMS

---

## Command Reference Guide

Order Number: AA-R8EXJ-TE

This manual contains the command reference information for HSM and Media, Device and Management Services (MDMS).

**Required Operating System** OpenVMS V6.2 or higher

**Required Software** Storage Library System for OpenVMS V2.9B or higher, **or** Media, Device and Management Services for OpenVMS Version 2.9C, 2.9D, 3.1, 3.1A, 3.2, 3.2A, V4.0 or V4.0A

DECnet (Phase IV) or DECnet-Plus (Phase V)

TCP/IP Services for OpenVMS

**Revision/Update  
Information:**

This manual replaces version AA-R8EXH-TE

Software Version: HSM Version V4.0A

**Compaq Computer Corporation  
Houston, Texas**

---

**April 2002**

© 2002 Compaq Information Technologies Group, L.P

Compaq, the Compaq logo, OpenVMS, VAX and Tru64 are trademarks of Compaq Information Technologies Group, L.P. in the U.S. and/or other countries. UNIX is a trademark of The Open Group in the U.S. and/or other countries. All other product names mentioned herein may be trademarks of their respective companies.

Confidential computer software. Valid license from Compaq required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. government under vendor's standard commercial license.

Compaq shall not be liable for technical or editorial errors or omissions contained herein. The information in this document is provided "as is" without warranty of any kind and is subject to change without notice. The warranties for Compaq products are set forth in the express limited warranty statements accompanying such products. Nothing herein should be construed as constituting an additional warranty.

Compaq service tool software, including associated documentation, is the property of and contains confidential technology of Compaq Computer Corporation. Service customer is hereby licensed to use the software only for activities directly relating to the delivery of, and only during the term of, the applicable services delivered by Compaq or its authorized service provider. Customer may not modify or reverse engineer, remove, or transfer the software or make the software or any resultant diagnosis or system management data available to other parties without Compaq's or its authorized service provider's consent. Upon termination of the services, customer will, at Compaq's or its service provider's option, destroy or return the software and associated documentation in its possession.

Printed in the U.S.A.

---

# Contents

Preface .....	vii
---------------	-----

## 1 The HSM Command Environment

HSM .....	1-2
DCL Commands .....	1-5
DIRECTORY .....	1-6
HSM\$BACKUP .....	1-7
PRESHELVE .....	1-9
SET FILE .....	1-12
SET PROCESS .....	1-13
SHELVE .....	1-14
UNPRESHELVE .....	1-18
UNSHELVE .....	1-21
The Shelf Management Utility .....	1-25
Shelf Management Utility Commands .....	1-26
SMU ANALYZE .....	1-27
SMU CHECKPOINT .....	1-30
SMU COPY .....	1-32
SMU DETECT .....	1-34
SMU EXIT .....	1-36
SMU HELP .....	1-37
SMU LOCATE .....	1-38
SMU LOGICAL NAMES .....	1-42
SMU RANK .....	1-44
SMU REPACK .....	1-46
SMU SET ARCHIVE .....	1-50
SMU SET CACHE .....	1-53
SMU SET DEVICE .....	1-56
SMU SET FACILITY .....	1-60
SMU SET POLICY .....	1-62
SMU SET SCHEDULE .....	1-68
SMU SET SHELF .....	1-71
SMU SET VOLUME .....	1-75
SMU SHOW ARCHIVE .....	1-79
SMU SHOW CACHE .....	1-81
SMU SHOW DEVICE .....	1-82
SMU SHOW FACILITY .....	1-84
SMU SHOW POLICY .....	1-85
SMU SHOW REQUESTS .....	1-87
SMU SHOW SCHEDULE .....	1-89
SMU SHOW SHELF .....	1-91
SMU SHOW VERSION .....	1-93
SMU SHOW VOLUME .....	1-94

SMU SHUTDOWN .....	1-96
SMU SPAWN .....	1-97
SMU STARTUP .....	1-98

## 2 MDMS DCL Commands

MDMS ALLOCATE DRIVE .....	2-2
MDMS ALLOCATE VOLUME .....	2-6
MDMS BIND VOLUME .....	2-11
MDMS CANCEL REQUEST .....	2-13
MDMS CREATE DRIVE .....	2-15
MDMS CREATE GROUP .....	2-20
MDMS CREATE JUKEBOX .....	2-23
MDMS CREATE LOCATION .....	2-29
MDMS CREATE MAGAZINE .....	2-32
MDMS CREATE MEDIA_TYPE .....	2-36
MDMS CREATE NODE .....	2-39
MDMS CREATE POOL .....	2-44
MDMS CREATE SCHEDULE .....	2-47
MDMS CREATE VOLUME .....	2-52
MDMS DEALLOCATE DRIVE .....	2-61
MDMS DEALLOCATE VOLUME .....	2-62
MDMS DELETE DRIVE .....	2-65
MDMS DELETE GROUP .....	2-66
MDMS DELETE JUKEBOX .....	2-67
MDMS DELETE LOCATION .....	2-68
MDMS DELETE MAGAZINE .....	2-69
MDMS DELETE MEDIA_TYPE .....	2-70
MDMS DELETE NODE .....	2-71
MDMS DELETE POOL .....	2-72
MDMS DELETE SCHEDULE .....	2-73
MDMS DELETE VOLUME .....	2-74
MDMS INITIALIZE VOLUME .....	2-75
MDMS INVENTORY JUKEBOX .....	2-78
MDMS LOAD DRIVE .....	2-81
MDMS LOAD VOLUME .....	2-83
MDMS MOVE MAGAZINE .....	2-86
MDMS MOVE VOLUME .....	2-89
MDMS REPORT VOLUME .....	2-93
MDMS SET DOMAIN .....	2-99
MDMS SET DRIVE .....	2-105
MDMS SET GROUP .....	2-109
MDMS SET JUKEBOX .....	2-111
MDMS SET LOCATION .....	2-116
MDMS SET MAGAZINE .....	2-118
MDMS SET MEDIA_TYPE .....	2-121
MDMS SET NODE .....	2-123
MDMS SET POOL .....	2-127
MDMS SET SCHEDULE .....	2-130
MDMS SET SERVER .....	2-134
MDMS SET VOLUME .....	2-135
MDMS SHOW DOMAIN .....	2-143
MDMS SHOW DRIVE .....	2-145

MDMS SHOW GROUP .....	2-148
MDMS SHOW JUKEBOX .....	2-150
MDMS SHOW LOCATION .....	2-152
MDMS SHOW MAGAZINE .....	2-154
MDMS SHOW MEDIA_TYPE .....	2-156
MDMS SHOW NODE .....	2-158
MDMS SHOW POOL .....	2-160
MDMS SHOW REQUEST .....	2-162
MDMS SHOW SCHEDULE .....	2-164
MDMS SHOW SERVER .....	2-166
MDMS SHOW VERSION .....	2-168
MDMS SHOW VOLUME .....	2-169
Start_Session .....	2-173
MDMS UNBIND VOLUME .....	2-175
MDMS UNLOAD DRIVE .....	2-177
MDMS UNLOAD VOLUME .....	2-179

## **A MDMS Files and Logical Names**

A.1 MDMS File Names .....	A-1
A.2 MDMS Logical Names .....	A-4

## **B MDMS Messages**

## **C MDMS Rights and Privileges**

C.1 MDMS Rights - Types .....	C-1
C.1.1 High Level Rights .....	C-1
C.1.2 Low-level rights .....	C-2
C.2 Default High-Level to Low-Level Mapping .....	C-4
C.2.1 MDMS_USER: .....	C-4
C.2.2 MDMS_OPERATOR Rights: .....	C-5
C.2.2.1 Domain Commands for Mapping Privileges .....	C-6



---

# Preface

## Document Structure

This document is organized in the following manner and includes the following information:

- Part I HSM Commands
- Part II MDMS Commands
- Appendix A - MDMS Files and Logical Names
- Appendix B - MDMS Messages
- Appendix C - MDMS Rights and Privileges

## Related Documents

The following documents are related to this documentation set or are mentioned in this manual. The lower case x in the part number indicates a variable revision letter.

---

<b>Document</b>	<b>Order No.</b>
<i>HSM for OpenVMS Installation Guide</i>	AA-QUJ1x-TE
<i>HSM for OpenVMS Guide to Operations</i>	AA-PWQ3x-TE
<i>HSM for OpenVMS Command Reference Guide</i>	AA-R8EXx-TE
<i>HSM for OpenVMS Software Product Description</i>	AE-PWNTx-TE
<i>HSM Hard Copy Documentation Kit</i> <i>(consists of the above HSM documents and a cover letter)</i>	QA-0NXAA-GZ
<i>OpenVMS System Management Utilities Reference Manual: A-L</i>	AA-PV5Px-TK
<i>OpenVMS DCL Dictionary: A-M</i>	AA-PV5Kx-TK
<i>OpenVMS DCL Dictionary: N-Z</i>	AA-PV5Lx-TK
<i>OpenVMS License Management Utility Manual</i>	AA-PVXUx-TK
<i>OpenVMS User's Manual</i>	AA-PV5Jx-TK

---

## Related Products

The following related products are mentioned in this document:

---

<b>Product</b>	<b>Description</b>
HSM	Refers to HSM for OpenVMS™ software.
MDMS	Refers to Media, Device and Management Services for OpenVMS™ software.

---

OpenVMS	Refers to the OpenVMS operating system
SMF	Refers to Sequential Media File System for OpenVMS software
SLS	Refers to Storage Library System for OpenVMS software

## Conventions

The following conventions are used in this guide:

Convention	Description
{ }	In the command description format, braces indicate required elements. You must include one of the elements.
[ ]	Square brackets show optional elements in command syntax. You can omit these elements if you wish to use the default response.
...	Horizontal ellipsis indicate the omission of information from a sentence or paragraph that is not important to the topic being discussed.
.	Vertical ellipsis indicate the omission of information from an example or command format. The information has been omitted because it is not important to the topic being discussed.
<b>boldface type</b>	Boldface type in text indicates the first instance of that term in the text or in the Glossary.
<i>italic type</i>	Italic type emphasizes important information and indicates: variables, complete titles of manuals and parameters for system information.
<code>Starting test . . .</code>	This type font denotes system response to commands, user input, command and other code examples.
Ctrl/x	Indicates that the key labeled Ctrl (Control) and the specified key (such as Ctrl/Z) must be held down simultaneously.
PF1 x	PF1 x indicates that you press and release the PF1 key, and then you press and release another key (indicated here by x)
<i>n</i> (lowercase italic n)	Indicates the generic use of a number. For example, 19nn indicates a four-digit number in which the last two digits are unknown.
<i>x</i> (lowercase italic x)	Indicates the generic use of a letter. For example, xxx indicates any combination of three alphabetic characters.

## Determining and Reporting Problems

If you encounter a problem while using HSM, report it to Compaq through your usual support channels.

Review the Software Product Description (SPD) and Warranty Addendum for an explanation of the warranty. If you encounter a problem during the warranty period, report the problem as indicated or follow alternate instructions provided by Compaq for reporting SPD nonconformance problems.



# Part I

---

## HSM Commands

This part contains information about the HSM SMU Commands.



---

# The HSM Command Environment

## HSM

This chapter describes the commands for the HSM for OpenVMS software. This chapter contains the following sections:

- Digital command language (DCL) commands
- The shelf management utility (SMU) commands

In each section, the commands are listed in alphabetical order. Table 1–1 and Table 1–1 contain summaries of the DCL and SMU commands respectively.

**Table 1–1 Summary of DCL Commands**

Command	Purpose
DIRECTORY	Selects files based on shelving attributes and displays the shelved state of files.
HSM\$BACKUP	Backs up headers of HSM files.
PRESHELVE	Copies a file's data from online storage to the shelf or online cache. The file data remains online and accessible to users.
SET FILE	Sets the file to be (pre)shelvable or not using the /[NO]SHELVABLE qualifier. Files set /NOSHELVABLE cannot be shelved or preshelved.
SET PROCESS	Used with the /[NO]AUTO_UNSHELVE qualifier to enable or disable automatic unshelve operations on access to shelved files.
SHELVE	Copies a file's data from online storage to the shelf or online cache, and deletes the online file data. Once shelved, the file must be unshelved prior to any user access. However, its header can be read and modified as needed.
UNPRESHELVE	<p>Converts a formerly shelved or preshelved file to an online file and optionally deletes the cache file and removes the file's HSM catalog entry according to the following criteria set on cache.</p> <p><b>In case of a Delete operation on a preshelved file (entry):</b></p> <p>If /HOLD is set:      ▶ and /SAVETIME is used: The entry is not deleted                                  ▶ and /NOSAVE is used:    The entry is not deleted</p> <p>If /NOHOLD is set:   ▶ and /SAVETIME is used: The entry is deleted                                  ▶ and /NOSAVE is used:    The entry is deleted</p>
UNSHELVE	Restores a file's data from the shelf to online disk storage. The file acquires all characteristics defined by its current online header.

**Table 1–2 Summary of SMU Commands**

<b>Command</b>	<b>Purpose</b>
ANALYZE	Compares the online file metadata for shelved files on the specified volume(s) with information in the catalog, and optionally repairs the catalog to reflect the online information.
CHECKPOINT	Dismounts the current shelving tape volume associated with a specified archive class and directs subsequent shelving operations to the next volume in the archive class. The dismounted tapes may then be removed to a remote location for safer storage.
COPY	Copies files, including shelved files, from one location to another without unshelving the files, and updates the HSM catalog to reflect the new location.
EXIT	Exits the SMU utility. QUIT and Ctrl /Z also can be used to exit.
HELP	Displays information on SMU commands and qualifiers.
LOCATE	Searches the HSM catalog to locate shelved file data, which can be manually recovered using OpenVMS BACKUP.
RANK	Displays an ordered list of files that would be shelved if the specified policy were to run on the specified volume.
REPACK	Copies valid shelved data from one shelf volume to another, while eliminating deleted or obsolete data.
SET ARCHIVE	Defines an archive class. For HSM Plus mode, must identify a media type and density defined in TAPESTART.COM. For HSM Basic mode, can be used in special situations to modify the volume label for the shelving tape volume associated with an archive class.
SET CACHE	Specifies a shelf cache disk or magneto-optical device that can be used by HSM.
SET DEVICE	Defines a nearline or offline tape device, magazine loader, or large tape juke-box for use by HSM, and the archive classes supported by the device.
SET FACILITY	Provides VMScluster™-wide control of HSM operations, defines eligible shelf servers, controls event logging, and changes the HSM mode from Basic to Plus.
SET POLICY	Defines preventative and reactive policies used for automatic shelving of files by HSM.
SET SCHEDULE	Schedules preventative policy execution for volumes on a periodic basis.
SET SHELF	Defines attributes of a shelf and associates archive classes with the shelf.
SET VOLUME	Defines permitted HSM operations on an online disk and associates the disk with a shelf and reactive policies.
SHOW ARCHIVE	Displays information about an archive class, its associated media type and shelving volume information.
SHOW CACHE	Displays information about an online cache-disk or magneto-optical device.
SHOW DEVICE	Displays information about a nearline or offline tape-device and its associated media type and archive classes.

**Table 1–2 Summary of SMU Commands**

<b>Command</b>	<b>Purpose</b>
SHOW FACILITY	Displays information about VMScluster™-wide HSM operations, designated shelf servers, event logging, and the HSM mode (Basic or Plus).
SHOW POLICY	Displays the characteristics of preventative and reactive policies.
SHOW REQUESTS	Returns a count of the number of active HSM requests and optionally dumps request information to an activity log.
SHOW SCHEDULE	Displays information about a scheduled preventative policy and its associated online disk volume.
SHOW SHELF	Displays information about a shelf and its associated archive classes.
SHOW VERSION	Displays the version identifiers of four components of the HSM software.
SHOW VOLUME	Displays information about an online disk volume, and its associated shelf and reactive policies.
SHUTDOWN	Shuts down HSM and optionally terminates all pending shelving operations.
SPAWN	Creates a subprocess to execute DCL commands without exiting SMU.
STARTUP	Starts up HSM. This command is usually only executed during SYSTEM startup.

---

## DCL Commands

This section lists the DCL commands that support file shelving. The PRESHELVE, SHELVE and UNSHELVE commands are part of the HSM for OpenVMS product.

---

### Note

**The DIRECTORY, SET FILE and SET PROCESS commands are part of the OpenVMS system. As such only OpenVMS extensions that support shelving are described in this document. Description in this section includes both user and system administrator commands.**

---

---

## DIRECTORY

The DIRECTORY command includes two additional qualifiers that select files based on shelving attributes and display the shelved state of files. These qualifiers are described here for reference.

---

**Note**

---

**For a full description of the DIRECTORY command, refer the OpenVMS DCL Dictionary or online help.**

---

### Format

DIRECTORY [file\_spec[,...]]

### Parameters

**[file\_spec[,...]]**

Specifies one or more files. To specify two or more files, separate them with commas. Wildcard characters are allowed.

### Qualifiers

**/SELECT=keyword**

The /SELECT qualifier accepts the following additional keywords:

[NO]ONLINE The ONLINE keyword selects files that are online or preshelved.  
 The NOONLINE keyword selects files that are shelved.

[NO]SHELVABLE The SHELVABLE keyword selects files that are shelvable.  
 The NOSHELVABLE keyword selects files that are not shelvable, as set by the SET FILE /NOSHELVABLE command.

**/SHELVED\_STATE**

Displays the state of the file as shelved or online. Preshelved files are displayed as online.

### Example

The following example shows the DIRECTORY /SHELVED\_STATE command:

```

$ DIRECTORY /SHELVED_STATE

DIRECTORY MYDISK:[SMITH]

MYFILE.TXT;2      Online
NOT_SHELVED.TXT;1 Online
SHELVED.TXT;1     Shelved
    
```



---

## HSM\$BACKUP

The HSM\$BACKUP command is a special version of OpenVMS BACKUP that supports consolidated Backup with HSM.

---

### Note

**For a full description of the BACKUP command and qualifiers that can also be applied to HSM\$BACKUP, refer the OpenVMS DCL Dictionary or online help.**

---



---

### Warning

**HSM\$BACKUP should not be made available to general users. It should be restricted to system managers performing regular system backups when using the consolidated backup with HSM paradigm. Normal OpenVMS BACKUP should be used for all other purposes.**

---

HSM\$BACKUP is enabled with the SET COMMAND as shown in the set of examples below.

### Format

HSM\$BACKUP *input\_spec output\_spec*

### Parameters

#### **input\_spec**

Specifies files to be backed up, or source saveset for a restore operation.

#### **output\_spec**

Specifies destination output spec for files to be backed up, or destination location for files on a restore.

### Qualifiers

#### **/PRESHELVED (default)**

#### **/NOPRESHELVED**

/NOPRESHELVED backs up only the headers of preshelved files. /PRESHELVED (default) backs up the data of preshelved files. This is applicable to IMAGE and file backup operations and not applicable to restore.

#### **/SHELVED (default)**

#### **/NOSHELVED**

#### **/NOSHELVED**

/NOSHELVED backs up only the headers of shelved files. SHELVED (default) unshelves the file and backs up the data. Applicable to non-image backup operations; not applicable to restore.

## HSM\$BACKUP

### Examples

The following example shows how to use HSM\$BACKUP to perform image and incremental backups, which back up only the headers of shelved and preshelved files.

```
$ SET COMMAND SYS$SYSTEM:HSM$BACKUP.CLD
$ HSM$BACKUP/IMAGE/IGNORE=INTERLOCK/RECORD/LOG/NOPRESHELVED -
$ $1$DKA100: $1$MUA100:DKA100_AUG23_IMAGE.BCK/SAVESET
$ HSM$BACKUP/RECORD/SINCE=BACKUP/LOG/NOSHELVED/NOPRESHELVED -
$ $1$DKA100: $1$MUA100:DKA100_AUG24.BCK/SAVESET
```

The following example shows how to use HSM\$BACKUP to copy shelved file data from one disk to a saveset on another using /IMAGE. The /INITIALIZE qualifier on the output disk is required.

```
$ MOUNT/FOREIGN $12$DKA400:
$ HSM$BACKUP/IMAGE/IGNORE=INTERLOCK/RECORD/NOPRESHELVED/LOG -
$ $1$DKA100: $12$DKA400:DKA100_AUG30_IMAGE.BCK/SAVESET/INITIALIZE
```

---

## PRESHELVE

The PRESHELVE command copies file data from online storage to shelf storage or cache, but leaves the original file data online and accessible.

The PRESHELVE command requires READ and WRITE access to the file, or an appropriate file access privilege.

### Format

**PRESHELVE file\_spec[,...]**

### Parameters

**file\_spec[,...]**

Specifies one or more files to be preshelled. If two or more files are specified, separate them with commas. Wildcard characters are allowed.

### Qualifiers

#### **/BACKUP**

Uses the backup date for selection criteria when using the /BEFORE, /SINCE or /ELAPSED qualifiers. If a file has no backup date, it is not selected for the operation.

#### **/BEFORE[=time]**

Selects only those files dated prior to the specified time. Specify the time as an absolute time, a delta time, or a combination time, or as one of the following keywords:

TODAY (default), TOMORROW, or YESTERDAY.

Specify one of the following qualifiers with /BEFORE to indicate the time attribute to be used as the basis for selection: /BACKUP, /CREATED, /EXPIRED, or /MODIFIED.

Time must be specified using the OpenVMS formats for absolute and delta times.

#### **/BY\_OWNER[=uic]**

Selects only those files whose owner user identification code (UIC) matches the specified owner UIC. The default UIC is that of the current process. If not specified, all UICs are eligible.

#### **/CANCEL**

Cancels any outstanding command requests for the files.

#### **/CONFIRM**

##### **/NOCONFIRM (Default)**

Controls whether a question is asked before each PRESHELVE operation to confirm that the operation should be performed on that file. The following responses are valid:

YES	NO	QUIT
TRUE	FALSE	Ctrl/Z
1	0	ALL

Refer the OpenVMS DCL Concepts Manual for more information on the /CONFIRM qualifier.

#### **/CREATED**

Uses the creation date for selection criteria when using the /BEFORE, /SINCE or /ELAPSED qualifiers.

**/ELAPSED=[ delta time]**

Selects only those files for which the specified time has elapsed after the file was subjected to any one of the file events. Specify any one of the following qualifiers with /ELAPSED to indicate the file event that is to be used for selection of the files: /BACKUP, /CREATED, /EXPIRED, or /MODIFIED

The file modification time is considered for selection as default when /ELAPSED is specified without any of the file event qualifiers.

The default time for /ELAPSED qualifier is 30 days. /ELAPSED qualifier is mutually exclusive with qualifiers /BEFORE and /SINCE.

Time must be specified using the OpenVMS format for delta times.

**/EXCLUDE=(file\_spec[,...])**

Excludes the specified files from the command operation. Wildcard characters are allowed in the file specification. However, you cannot use relative version numbers to exclude a specific version. If you provide only one file specification, you can omit the parentheses.

**/EXPIRED**

Uses the expiration date for selection criteria when using the /BEFORE, /SINCE or /ELAPSED qualifiers. The file expiration date is used in conjunction with the volume retention functionality. When volume retention has been enabled for a volume, the expiration date is updated whenever a file is accessed (read or write). Each time a file is accessed, the current time is added to the minimum retention time. If the sum is greater than the current expiration date, a new expiration date is computed. Updating the expiration date depends on the settings of the retention time for the volume.

Refer the OpenVMS SET VOLUME/RETENTION command for further details.

If a file has no expiration date, it is not selected for the operation.

**/KEEP[=number\_of\_versions]**

Specifies that the latest n versions of the file are not processed. If /KEEP is specified without a *number of versions* value, then all versions of the file except the highest version are processed.

**/LOG****/NOLOG (Default)**

Specifies that a command acknowledgment is displayed. If the command is specified with the /WAIT qualifier or if the shelf handler can process the command immediately, the log output contains the final result of the operation. If the command is specified with the /NOWAIT qualifier, a message indicating that the request is queued may be displayed on success for long-running operations.

**/MODIFIED (Default)**

Uses the modification date for selection criteria when using the /BEFORE, /SINCE or /ELAPSED qualifiers.

**/NOTIFY****/NONOTIFY (Default)**

Specifies whether a message is sent to the user upon completion of the command operation for each file in the request. The message is sent to all terminals on which the user is logged-in that have BROADCAST enabled.

**/SAVE**

When issued with the shelve command, HSM will not allow the save sets to span from one tape to another on encountering an end of tape during the BACKUP operation.

**/SELECT=(keyword[,...])**

Allows selection of files according to size or shelved state. Choose one of the following keywords:

- SIZE=MAXIMUM=n** Selects files that have fewer blocks than the value of n, which defaults to 1,073,741,823. Use with **MINIMUM=n** to specify a size range for files to be selected.
- SIZE=MINIMUM=n** Selects files that have blocks equal to or greater than the value of n, which defaults to 0. Use with **MAXIMUM=n** to specify a size range for files to be selected.
- NOONLINE** Selects only files that are shelved. Use this keyword to copy shelved file data to additional archive classes.
- ONLINE (Default)** Selects only files that are online or preshelved. This is the default file selection.

(**ONLINE, NOONLINE**)Selects files regardless of their shelved state.

By default, file selection is not based on file size criteria.

**/SINCE[=time]**

Selects only those files dated after the specified time. Specify the time as an absolute time, a delta time, or a combination time, or as one of the following keywords:

- **TODAY** (default),
- **TOMORROW**, or
- **YESTERDAY**.

Specify one of the following qualifiers with **/SINCE** to indicate the time attribute to be used as the basis for selection:

**/BACKUP, /CREATED, /EXPIRED, or /MODIFIED.**

Time must be specified using the OpenVMS formats for absolute and delta times.

**/WAIT (Default)****/NOWAIT**

Specifies whether to wait for the operation to complete. The default is **/WAIT**. Files moved to the cache typically take the same amount of time as a similar **COPY** command. Files shelved directly to nearline/offline storage may take anywhere from tens of seconds to tens of minutes. Multiple files take correspondingly longer.

**Example**

The following example shows the **PRESHELVE/NOTIFY** command:

```
$ PRESHELVE/NOTIFY JLRM_DATA.DAT
%HSM-S-PRESHELVED, file $1$DKA0:[BUTLER.DB_FILES]JLRM_DATA.DAT;1 preshelved
```

---

## SET FILE

### FILE

The SET FILE command includes an additional qualifier to control the eligibility of a file for shelving. This qualifier is described here for reference. For a full description of the SET FILE command, refer the OpenVMS DCL Dictionary or online help.

### Format

```
SET FILE file_spec[,...]
```

### Parameters

***file\_spec*[,...]**

Specifies one or more files to be modified. If two or more files are specified, separate them with commas. Wildcard characters are allowed.

### Qualifiers

**/SHELVABLE**

**/NOSHELVABLE**

The /SHELVABLE qualifier sets the file named by *file\_spec* to be shelvable. By default, all user files are shelvable. Use the /NOSHELVABLE qualifier to prevent the file from being shelved or preshelved.

### Example

The following example shows the SET FILE /NOSHELVABLE command, which ensures that the file cannot be shelved or preshelved:

```
$ SET FILE /NOSHELVABLE AARDVARKS.TXT;3
```

---

## SET PROCESS

### PROCESS

The SET PROCESS command includes an additional qualifier that specifies the default action to be taken when a shelved file is accessed by the process. This qualifier is described here.

---

#### Note

---

**For a full description of the SET PROCESS command, refer the OpenVMS DCL Dictionary or online help.**

---

### Format

```
SET PROCESS process-name[,...]
```

### Parameters

***process-name*[,...]**

Specifies the name of the process that will have its characteristics changed. You must own the process or have GROUP privileges to use this command.

### Qualifiers

**/AUTO\_UNSHELVE (default)**

**/NOAUTO\_UNSHELVE**

The /AUTO\_UNSHELVE qualifier enables the process to generate a file fault upon read, write, extend or truncate access to a shelved file. The process stalls while the file is unshelved, then proceeds normally.

The /NOAUTO\_UNSHELVE qualifier disables file faults: accessing a shelved file returns an error message instead. By default, all processes are started with /AUTO\_UNSHELVE enabled.

### Example

The following example shows the SET PROCESS /AUTO\_UNSHELVE command:

```
$ SET PROCESS /AUTO_UNSHELVE SMITH_1
```

---

## SHELVE

The SHELVE command copies file data from online storage to the shelf or the online cache, and deletes the online file data.

Once the file is shelved either to the cache or the shelf, the user cannot read, write, extend, or truncate the file without causing a file fault. However, the user retains access to the file header, and can enter commands such as DIRECTORY and SET FILE without causing a file fault. The user does not have direct access to either the cache or shelf version of the file.

The SHELVE command requires READ and WRITE access to the file, or an appropriate file access privilege.

### Format

SHELVE file\_spec[,...]

### Parameters

***file\_spec[,...]***

Specifies one or more files to be shelved. If two or more files are specified, separate them with commas. Wildcard characters are allowed.

### Qualifiers

**/BACKUP**

Uses the backup date for selection criteria when using the /BEFORE, /SINCE or /ELAPSED qualifiers.

If a file has no backup date, it is not selected for the operation.

**/BEFORE[=time]**

Selects only those files dated prior to the specified time. Specify the time as an absolute time, a delta time, or a combination time, or as one of the keywords: TODAY (default), TOMORROW, or YESTERDAY.

Specify one of the following qualifiers with /BEFORE to indicate the time attribute to be used as the basis for selection: /BACKUP, /CREATED, /EXPIRED, or /MODIFIED.

---

**Note**

**Time must be specified using the OpenVMS formats for absolute and delta times.**

---

**/BY\_OWNER[=uic]**

Selects only those files whose owner user identification code (UIC) matches the specified owner UIC. The default UIC is that of the current process. If not specified, all UICs are eligible.

**/CANCEL**

Cancels any outstanding command requests for the files.

**/CONFIRM**

**/NOCONFIRM (Default)**

Controls whether a request is issued before each SHELVE operation to confirm that the operation should be performed on that file.



The following responses are valid:

YES	NO	QUIT
TRUE	FALSE	Ctrl/Z
1	0	ALL

Refer the OpenVMS DCL Concepts Manual for more information on the /CONFIRM qualifier.

### **/CREATED**

Uses the creation date for selection criteria when using the /BEFORE, /SINCE or /ELAPSED qualifiers.

### **/ELAPSED=[ delta time]**

Selects only those files for which the specified time has elapsed after the file was subjected to any one of the file events. Specify any one of the following qualifiers with /ELAPSED to indicate the file event that is to be used for selection of the files: /BACKUP, /CREATED, /EXPIRED, or /MODIFIED

The file modification time is considered for selection as default when /ELAPSED is specified without any of the file event qualifiers.

The default time for /ELAPSED qualifier is 30 days. /ELAPSED qualifier is mutually exclusive with qualifiers /BEFORE and /SINCE.

Time must be specified using the OpenVMS format for delta times.

### **/EXCLUDE=(file\_spec[,...])**

Excludes the specified files from the command operation. Wildcard characters are allowed in the file specification. However, you cannot use relative version numbers to exclude a specific version. If you provide only one file specification, you can omit the parentheses.

### **/EXPIRED**

Uses the expiration date for selection criteria when using the /BEFORE, /SINCE or /ELAPSED qualifiers. The file expiration date is used in conjunction with the volume retention functionality. When volume retention has been enabled for a volume, the expiration date is updated whenever a file is accessed (read or write).

Each time a file is accessed, the current time is added to the minimum retention time. If the sum is greater than the current expiration date, a new expiration date is computed. Updating the expiration date depends on the settings of the retention times for the volume. Refer the OpenVMS SET VOLUME/RETENTION command for further details.

If a file has no expiration date, it is not selected for the operation.

### **/HIGH\_PRIORITY**

Specifies a request priority as HIGH. HSM pushes the request to the top of the queue of requests at that time. In case of multiple /HIGH requests, each request is moved to the top of the queue and execution starts with the latest /HIGH request. This prioritizing of requests does not affect urgent requests like:

- file faults
- make space requests

### **/KEEP[=number\_of\_versions]**

Specifies that the latest n versions of the file are not processed. If /KEEP is specified without a number of versions value, then all versions of the file except the highest version are processed.

**/LOG****/NOLOG (Default)**

Specifies that a command acknowledgment is displayed. If the command is specified with the /WAIT qualifier or if the shelf handler can process the command immediately, the log output contains the final result of the operation. If the command is specified with the /NOWAIT qualifier, a message indicating that the request is queued may be displayed on success for long-running operations.

**/MODIFIED (Default)**

Uses the modification date for selection criteria when using the /BEFORE, /SINCE or /ELAPSED qualifiers.

**/NOTIFY****/NONOTIFY (Default)**

Specifies whether a message is sent to the user upon completion of the command operation for each file in the request. The message is sent to all terminals on which the user is logged-in that have BROADCAST enabled.

**/SAVE**

When issued with the shelve command, HSM will not allow the save sets to span from one tape to another on encountering an end of tape during the BACKUP operation.

**/SELECT=(keyword[,...])**

Allows selection of files according to size or shelved state. Choose one of the following keywords:

SIZE=MAXIMUM=n	Selects files that have fewer blocks than the value of n, which defaults to 1,073,741,823. Use with MINIMUM=n to specify a size range for files to be selected.
SIZE=MINIMUM=n	Selects files that have blocks equal to or greater than the value of n, which defaults to 0. Use with MAXIMUM=n to specify a size range for files to be selected.
NOONLINE	Selects only files that are shelved. Use this keyword to copy shelved file data to additional archive classes.
ONLINE (Default)	Selects only files that are online or preshelved. This is the default file selection.

(ONLINE, NOONLINE) Selects files regardless of their shelved state.

By default, file selection is not based on file size criteria.

**/SINCE****/SINCE[=time]**

Selects only those files dated after the specified time.

Specify the time as an absolute time, a delta time, or a combination time, or as one of the keywords: TODAY (default), TOMORROW, or YESTERDAY.

Specify one of the following qualifiers with /SINCE to indicate the time attribute to be used as the basis for selection: /BACKUP, /CREATED, /EXPIRED, or /MODIFIED.

Time must be specified using the OpenVMS formats for absolute and delta times.

**/WAIT (Default)****/NOWAIT**

Specifies whether to wait for the operation to complete. The default is /WAIT. Files moved to the cache typically take the same amount of time as a similar COPY command. Files shelved directly to nearline/offline storage may take anywhere from tens of seconds to tens of minutes. Multiple files take correspondingly longer.

**Example**

The following example shows the SHELVE/NOTIFY command:

```
$ SHELVE/NOTIFY JLRM_DATA.DAT  
%HSM-S-SHELVED, file $1$DKA0:[BUTLER.DB_FILES]JLRM_DATA.DAT;1 shelved
```

---

## UNPRESHELVE

Converts a formerly shelved or preshelved file to an online file and optionally deletes the cache file and removes the file's HSM cache catalog entry if /NOHOLD is set on the cache.

If the file is subsequently preshelved or shelved, a new copy is made to the HSM subsystem. The UNPRESHELVE command can be issued on preshelved files only; shelved files cannot be explicitly unpreshelved unless an unshelve is performed first.

This function is automatically invoked if a preshelved file is modified, invalidating its HSM copy of the data.

The UNPRESHELVE command requires READ and WRITE access to the file, or an appropriate file access privilege.

### Format

```
UNPRESHELVE file_spec[,...]
```

### Parameters

***file\_spec[,...]***

Specifies one or more files to be unshelved. If two or more files are specified, separate them with commas. Wildcard characters are allowed.

### Qualifiers

#### **/BACKUP**

Uses the backup date for selection criteria when using the /BEFORE, /SINCE or /ELAPSED qualifiers.

If a file has no backup date, it is not selected for the operation.

#### **/BEFORE[=time]**

Selects only those files dated prior to the specified time. Specify the time as an absolute time, a delta time, or a combination time, or as one of the keywords: TODAY (default), TOMORROW, or YESTERDAY.

Specify one of the following qualifiers with /BEFORE to indicate the time attribute to be used as the basis for selection: /BACKUP, /CREATED, /EXPIRED, or /MODIFIED.

Time must be specified using the OpenVMS formats for absolute and delta times.

#### **/BY\_OWNER[=uic]**

Selects only those files whose owner user identification code (UIC) matches the specified owner UIC. The default UIC is that of the current process. If not specified, all UICs are eligible.

#### **/CANCEL**

#### **/CANCEL**

Cancels any outstanding command requests for the files.

#### **/CONFIRM**

#### **/NOCONFIRM (Default)**

Controls whether a request is issued before each UNSHELVE operation to confirm that the operation should be performed on that file.

The following responses are valid:

YES	NO	QUIT
TRUE	FALSE	Ctrl/Z
1 0	ALL	

Refer the OpenVMS DCL Concepts Manual for more information on the /CONFIRM qualifier.

### **/CREATED**

Uses the creation date for selection criteria when using the /BEFORE, /SINCE or /ELAPSED qualifiers.

### **/ELAPSED=[ delta time]**

Selects only those files for which the specified time has elapsed after the file was subjected to any one of the file events. Specify any one of the following qualifiers with /ELAPSED to indicate the file event that is to be used for selection of the files: /BACKUP, /CREATED, /EXPIRED, or /MODIFIED

The file modification time is considered for selection as default when /ELAPSED is specified without any of the file event qualifiers.

The default time for /ELAPSED qualifier is 30 days. /ELAPSED qualifier is mutually exclusive with qualifiers /BEFORE and /SINCE.

Time must be specified using the OpenVMS format for delta times.

### **/EXCLUDE=(file\_spec[,...])**

Excludes the specified files from the command operation. Wildcard characters are allowed in the file specification. However, you cannot use relative version numbers to exclude a specific version. If you provide only one file specification, you can omit the parentheses.

### **/EXPIRED**

Uses the expiration date for selection criteria when using the /BEFORE, /SINCE or /ELAPSED qualifiers. The file expiration date is used in conjunction with the volume retention functionality. When volume retention has been enabled for a volume, the expiration date is updated whenever a file is accessed (read or write). Each time a file is accessed, the current time is added to the minimum retention time. If the sum is greater than the current expiration date, a new expiration date is computed. Updating the expiration date depends on the settings of the retention times for the volume. Refer the OpenVMS SET VOLUME/RETENTION command for further details.

If a file has no expiration date, it is not selected for the operation.

### **/LOG**

#### **/NOLOG (Default)**

Specifies that a command acknowledgment is displayed. If the command is specified with the /WAIT qualifier or if the shelf handler can process the command immediately, the log output contains the final result of the operation. If the command is specified with the /NOWAIT qualifier, a message indicating that the request is queued may be displayed on success for long-running operations.

#### **/MODIFIED (Default)**

Uses the modification date for selection criteria when using the /BEFORE, /SINCE or /ELAPSED qualifiers.

### **/NOTIFY**

#### **/NONOTIFY (Default)**

Specifies whether a message is sent to the user upon completion of the command operation for each file in the request. The message is sent to all terminals on which the user is logged-in that have BROADCAST enabled.

## UNPRESHELVE

### **/SELECT=keyword[,...]**

Allows selection of files according to size or shelved state. Choose one of the following keywords:

**SIZE=MAXIMUM=n** Selects files that have fewer blocks than the value of n, which defaults to 1,073,741,823. Use with **MINIMUM=n** to specify a range for files to be selected.

**SIZE=MINIMUM=n** Selects files that have blocks equal to or greater than the value of n, which defaults to 0. Use with **MAXIMUM=n** to specify a size range for files to be selected.

By default, file selection is not based on file size criteria.

### **/SINCE[=time]**

Selects only those files dated after the specified time. Specify the time as an absolute time, a delta time, or a combination time, or as one of the following keywords: **TODAY** (default), **TOMORROW**, or **YESTERDAY**.

Specify one of the following qualifiers with **/SINCE** to indicate the time attribute to be used as the basis for selection: **/BACKUP**, **/CREATED**, **/EXPIRED**, or **/MODIFIED**.

Time must be specified using the OpenVMS formats for absolute and delta times.

### **/WAIT (Default)**

### **/NOWAIT**

Specifies whether to wait for the operation to complete. The default is **/WAIT**. Files moved from the cache typically take the same amount of time as a similar **COPY** command. Files unshelved from nearline/offline storage may take anywhere from tens of seconds to tens of minutes. Multiple files take correspondingly longer.

## Example

The following example shows the **UNPRESHELVE/NOTIFY** command:

```
$ UNPRESHELVE/NOTIFY JLRM_DATA.DAT
%HSM-S-UNPRESHELVED, file $1$DKA0:[BUTLER.DB_FILES]JLRM_DATA.DAT;1
unpreshelved
```

---

## UNSHELVE

Restores a file's data from the shelf to the online disk and makes the file accessible to user operations.

Modifications to the file header may take place between the shelve and unshelve operations. If this occurs, the current (potentially modified) online file header is in effect when the file is unshelved. If, for example, the access protection is changed, the same user that shelved the file may not be able to unshelve it.

Once a file is unshelved, the user may read, write, extend, or truncate the file as usual.

After a file is unshelved, it is left in the preshelved state. If a subsequent request to shelve the file is received and the file has not been modified since the unshelve operation, it is reshelved without data movement.

The UNSHELVE command requires READ and WRITE access to the file, or an appropriate file access privilege.

### Format

```
UNSHELVE file_spec[,...]
```

### Parameters

***file\_spec[,...]***

Specifies one or more files to be unshelved. If two or more files are specified, separate them with commas. Wildcard characters are allowed.

### Qualifiers

#### **/BACKUP**

Uses the backup date for selection criteria when using the /BEFORE, /SINCE or /ELAPSED qualifiers. If a file has no backup date, it is not selected for the operation.

#### **/BEFORE[=time]**

Selects only those files dated prior to the specified time. Specify the time as an absolute time, a delta time, or a combination time, or as one of the keywords: TODAY (default), TOMORROW, or YESTERDAY.

Specify one of the following qualifiers with /BEFORE to indicate the time attribute to be used as the basis for selection: /BACKUP, /CREATED, /EXPIRED, or /MODIFIED.

Time must be specified using the OpenVMS formats for absolute and delta times.

#### **/BY\_OWNER[=uic]**

Selects only those files whose owner user identification code (UIC) matches the specified owner UIC. The default UIC is that of the current process. If not specified, all UICs are eligible.

#### **/CANCEL**

Cancels any outstanding command requests for the files.

#### **/CONFIRM**

#### **/NOCONFIRM (Default)**

Controls whether a request is issued before each UNSHELVE operation to confirm that the operation should be performed on that file.

The following responses are valid:

YES	NO	QUIT
TRUE	FALSE	Ctrl/Z
1	0	ALL

Refer the OpenVMS DCL Concepts Manual for more information on the /CONFIRM qualifier.

### **/CREATED**

Uses the creation date for selection criteria when using the /BEFORE, /SINCE or /ELAPSED qualifiers.

### **/ELAPSED=[ delta time]**

Selects only those files for which the specified time has elapsed after the file was subjected to any one of the file events. Specify any one of the following qualifiers with /ELAPSED to indicate the file event that is to be used for selection of the files: /BACKUP, /CREATED, /EXPIRED, or /MODIFIED

The file modification time is considered for selection as default when /ELAPSED is specified without any of the file event qualifiers.

The default time for /ELAPSED qualifier is 30 days. /ELAPSED qualifier is mutually exclusive with qualifiers /BEFORE and /SINCE.

Time must be specified using the OpenVMS format for delta times.

### **/EXCLUDE=(file\_spec[,...])**

Excludes the specified files from the command operation. Wildcard characters are allowed in the file specification. However, you cannot use relative version numbers to exclude a specific version. If you provide only one file specification, you can omit the parentheses.

### **/EXPIRED**

Uses the expiration date for selection criteria when using the /BEFORE, /SINCE or /ELAPSED qualifiers. The file expiration date is used in conjunction with the volume retention functionality. When volume retention has been enabled for a volume, the expiration date is updated whenever a file is accessed (read or write). Each time a file is accessed, the current time is added to the minimum retention time. If the sum is greater than the current expiration date, a new expiration date is computed. Updating the expiration date depends on the settings of the retention times for the volume. Refer the OpenVMS SET VOLUME/RETENTION command for further details.

If a file has no expiration date, it is not selected for the operation.

### **/HIGH\_PRIORITY**

Specifies a request priority as HIGH. HSM pushes the request to the top of the queue of requests at that time. In case of multiple /HIGH requests, each request is moved to the top of the queue and execution starts with the latest /HIGH request. This prioritizing of requests does not affect urgent requests like:

- file faults
- make space requests

### **/KEEP[=number\_of\_versions]**

Specifies that the latest n versions of the file are not processed. If /KEEP is specified without a number of versions value, then all versions of the file except the highest version are processed.



**/LOG****/NOLOG (Default)**

Specifies that a command acknowledgment is displayed. If the command is specified with the /WAIT qualifier or if the shelf handler can process the command immediately, the log output contains the final result of the operation. If the command is specified with the /NOWAIT qualifier, a message indicating that the request is queued may be displayed on success for long-running operations.

**/MODIFIED (Default)**

Uses the modification date for selection criteria when using the /BEFORE, /SINCE or /ELAPSED qualifiers.

**/NOTIFY****/NONOTIFY (Default)**

Specifies whether a message is sent to the user upon completion of the command operation for each file in the request. The message is sent to all terminals on which the user is logged-in that have BROADCAST enabled.

**/OVERRIDE**

Overrides the consistency checking performed by HSM and forces the file(s) to be unshelved. This qualifier requires BYPASS privilege on files with inconsistent HSM metadata.

Following an UNSHELVE/OVERRIDE, the file's data should be carefully checked for accuracy, because HSM detected an inconsistency between the online file state and the contents of the HSM catalog, and the file data may not be correct.

**/SELECT=*keyword*[,...]**

Allows selection of files according to size or shelved state. Choose one of the following keywords:

**SIZE=MAXIMUM=*n*** Selects files that have fewer blocks than the value of *n*, which defaults to 1,073,741,823. Use with **MINIMUM=*n*** to specify a size range for files to be selected.

**SIZE=MINIMUM=*n*** Selects files that have blocks equal to or greater than the value of *n*, which defaults to 0. Use with **MAXIMUM=*n*** to specify a size range for files to be selected.

By default, file selection is not based on file size criteria.

**/SINCE[=*time*]**

Selects only those files dated after the specified time. Specify the time as an absolute time, a delta time, or a combination time, or as one of the following keywords: TODAY (default), TOMORROW, or YESTERDAY.

Specify one of the following qualifiers with /SINCE to indicate the time attribute to be used as the basis for selection: /BACKUP, /CREATED, /EXPIRED, or /MODIFIED.

Time must be specified using the OpenVMS formats for absolute and delta times.

**/WAIT (Default)****/NOWAIT**

Specifies whether to wait for the operation to complete. The default is /WAIT. Files moved from the cache typically take the same amount of time as a similar COPY command. Files unshelved from nearline/offline storage may take anywhere from tens of seconds to tens of minutes. Multiple files take correspondingly longer.

UNSHELVE

## Example

The following example shows the UNSHELVE/NOTIFY command:

```
$ UNSHELVE/NOTIFY JLRM_DATA.DAT  
%HSM-S-UNSHELVED, file $1$DKA0:[BUTLER.DB_FILES]JLRM_DATA.DAT:1 unshelved
```

## The Shelf Management Utility

The Shelf Management Utility (SMU) is the component of HSM that gives you control over shelving operations. SMU supports SET and SHOW operations for the following managed entities:

- Archive
- Cache
- Device
- Facility
- Policy
- Schedule
- Shelf
- Volume

SMU provides additional commands to control and monitor the HSM environment.

Online help is available for all SMU commands.

## Shelf Management Utility Commands

SMU commands may be entered directly on the DCL command line using the SMU keyword. Alternatively, a set of commands may be entered from an SMU prompt after entering SMU on the command line.

Use of SMU commands requires SYSPRV, TMPMBX, and SYSLCK privileges for all commands. Additional privileges required for specific commands are noted under that command description.

---

## SMU ANALYZE

### ANALYZE

The SMU ANALYZE command compares the online file metadata for shelved and preshelved files on the specified volume(s) with the information in the catalog, and optionally repairs the catalog to reflect the online information. When a file is found that is of interest, it's HSM metadata (ACE information) is compared against entries in the catalog and any discrepancies are reported. If the /REPAIR qualifier is used, the discrepancy can be repaired. If /CONFIRM is not used, then the default repair action will be applied.

### Format

```
SMU ANALYZE volume_name[,...]
```

### Parameters

#### **volume\_name[,...]**

This specifies the volume name of an online disk volume. Wildcards are not allowed, but logical names are supported. The volume can be a member of a volume set. If it is, all volumes in the set will be analyzed.

### Qualifiers

#### **/CACHE={CACHE\_DEVICE\_LIST}**

The /CACHE qualifier specifies that the SMU ANALYZE operation should run only on those files of the specified input volume that were cached to the device specified in /CACHE.

Different cache devices can be used at different points of time for a given volume.

For Analyze to run, files that have been cached to devices in the volume set list specified with the /CACHE qualifier, should be present in the device specified as the volume list.

/CACHE supports a comma separated volumeset list. A single volume is treated as a volumeset with one volume.

#### **/CONFIRM**

#### **/NOCONFIRM (Default)**

When the /REPAIR option is given, /CONFIRM prompts the user to allow repair of each file.

A default of "YES" or "NO" is provided for each confirmation, and this varies on the type of repair. The user can confirm each operation with (YES, 1, TRUE) to proceed for the specific repair, (NO, 0, FALSE) to inhibit the specific repair, (ALL) to proceed with the repair and subsequent repairs with the default confirm answer, and (^Z, QUIT) to stop the analysis/repair.

/NOCONFIRM does not prompt the user on repair and applies repairs with a default confirmation answer of "YES", and does not apply repairs with a default confirmation of "NO". The default is /NOCONFIRM. The following responses are valid:

YES	NO	QUIT
TRUE	FALSE	Ctrl/Z
1	0	ALL

**/OUTPUT[=file\_specification]**

The /OUTPUT qualifier specifies where to write the output of the command.

The default is /OUTPUT=SYS\$OUTPUT. The output can optionally be sent to a file.

If /REPAIR/CONFIRM is specified with /OUTPUT=file\_spec, the output is also sent to SYS\$OUTPUT in order for the system manager perform the confirmation.

/CONFIRM questions are not sent to the output file, but always to SYS\$OUTPUT.

By default, the output is written to the current SYS\$OUTPUT device. No Wildcard characters are allowed.

**/REPAIR****/NOREPAIR (Default)**

The /REPAIR qualifier repairs the catalog entries and/or HSM metadata for consistency. Repairs are applied when the user specifies “YES” to a repair confirm question, or according to the default confirm answer when /NOCONFIRM or /CONFIRM answer “ALL” is entered. /NOREPAIR simply scans the files and reports errors, but no updates are made. The default is /NOREPAIR.

SMU ANALYZE/REPAIR command now repairs entries corresponding to PRESHELVED files whose online allocation size and the allocation size in the catalog do not match. If there is an allocation size mismatch and if the revision date of the online file and that recorded in the catalog are same then a new entry with the online allocation size is created for the file. Else the entry is marked unrepairable.

**Examples**

This example analyses volume \$1\$DKB500 with no repair:

```
$ SMU ANALYZE $1$DKB500:
```

This example analyses volume \$1\$DKB500 and applies the default repair. Note that this means that some errors are not repaired.

```
$ SMU ANALYZE/REPAIR $1$DKB500:
```

This example analyses volume \$1\$DKB500 and prompts the user for each repair. The default repair can be applied by entering <return> to each prompt. This is the only way that non-default repairs can be applied.

```
$ SMU ANALYZE/REPAIR/CONFIRM $1$DKB500:
```

The following examples demonstrate the use of /cache qualifier with SMU ANALYZE.

```
$ SMU ANALYZE dkb200,dka0 /cache=(dka100,dkb300)
```

```
$ smu show cache
Cache device _$1$DUAL15: is enabled, Cache flush is held until after
08-Apr-2002 16:30:52.70, Backup is performed at flush intervals,
Cached files are held on delete of online file
Block size:          50000
Highwater mark:      80%
Flush interval:      <none>
Cache device _$1$DUAL17: is enabled, Cache flush is held until after
08-Apr-2002 16:28:28.64, Backup is performed at flush intervals,
Cached files are held on delete of online file
Block size:          50000
Highwater mark:      80%
Flush interval:      <none>
$ shelve Y.Y:1
$ smu locate y.y:1/full
$1$DUAL15:[000000]

Y.Y:1 (21,2,0)
File Shelved:
```

```

Y.Y;1 5/0 08-Apr-2002 19:23:50.19
Cache copy: _$1$DUA17:
Y.Y$9291950631;1 5/8 08-Apr-2002 19:23:50.19
Shelved on 08-Apr-2002 13:44:28.83

```

In the above example the file y.y;1 is cached in \$1\$dua17:

In the following example SMU ANALYZE without /CACHE is used if the header information for this file is corrupted.

```

$ smu analyze $1$dua15
%SMU-I-PROCESSING, processing input device $1$DUA15
%SMU-I-STARTSCAN, scanning for shelved files on disk volume _$1$DUA15:
File (19,2,0) "$1$DUA15:[000000]Z.Z;16"
No catalog entry found - file not repairable
File (21,2,0) "$1$DUA15:[000000]Y.Y;1"
Allocation size mismatch - entry not repairable
Current: 0
Catalog: 8
%SMU-I-ENDSCAN, completed scan for shelved files on disk volume _$1$DUA15:
%SMU-I-ERRORS, 2 error(s) detected, 0 error(s) repaired
$ smu analyze $1$dua15/cache=$1$dua15
%SMU-I-PROCESSING, processing input device $1$DUA15
%SMU-I-STARTSCAN, scanning for shelved files on disk volume _$1$DUA15:
No catalog entry for Cached file...$1$DUA15:[000000]Z.Z;16
%SMU-I-ENDSCAN, completed scan for shelved files on disk volume _$1$DUA15:
%SMU-I-ERRORS, 0 error(s) detected, 0 error(s) repaired

```

Also note that in the above case y.y;1 is not analyzed since it is not cached in \$1\$dua15:

```

$ smu analyze $1$dua15/cache=($1$dua15,$1$dua17)
SMU-I-PROCESSING, processing input device $1$DUA15
%SMU-I-STARTSCAN, scanning for shelved files on disk volume _$1$DUA15:
No catalog entry ...$1$DUA15:[000000]Z.Z;16
File $1$DUA15:[000000]Y.Y;1 is cached in _$1$DUA17: will be analyzed
File (21,2,0) "$1$DUA15:[000000]Y.Y;1"
Allocation size mismatch - entry not repairable
Current: 0
Catalog: 8
%SMU-I-ENDSCAN, completed scan for shelved files on disk volume _$1$DUA15:
%SMU-I-ERRORS, 1 error(s) detected, 0 error(s) repaired

```

In the above example y.y;1 is analyzed since it is present in one of the cache devices specified in the list.

---

## SMU CHECKPOINT

### CHECKPOINT

The SMU CHECKPOINT command allows system managers to use the next tape volume for subsequent shelving operations in an archive class, so that the current volume can be stored, possibly in a remote location. Invoking this command executes a DISMOUNT/UNLOAD command on a tape device if the device contains the current shelving volume. The checkpoint operation can be confirmed using an SMU SHOW ARCHIVE command.

### Format

```
SMU CHECKPOINT archive_id[,...]
```

### Parameters

***archive\_id[,...]***

A comma-separated list of the identifiers of the archive classes.

### Restrictions

The SMU CHECKPOINT command should not be entered during a repack operation, since the repack is controlling the archive class shelving volume. An attempt to do so results in an error on the checkpoint.

### Examples

The following example shows the SMU CHECKPOINT command used for HSM Basic mode. Assuming that the current shelving volume label is HS1005 for archive class 2, the command directs future (pre)shelve operations to volume HS1006, and dismounts and unloads the volume HS1005 from any device on which it is loaded.

```
$ SMU SHOW ARCHIVE 2

HSM$ARCHIVE02 has been used
Identifier:      2
Media type:     CompactTape III
Label:          HS1005
Position:       1293
Device refs:    1
Shelf refs:     2

$ SMU CHECKPOINT 2

$ SMU SHOW ARCHIVE 2

HSM$ARCHIVE02 has been used
Identifier:2
Media type:CompactTape III
Label:HS1006
Position:0
Device refs:1
Shelf refs:1
```



The following example shows the SMU CHECKPOINT command as used for HSM Plus mode.

```
$ SMU SHOW ARCHIVE 1

HSM$ARCHIVE01 has been used
Identifier:1
Media type:TZ87
Density:<NONE>
Label:PLS400
Position:963
Device refs:1
Shelf refs:1
Current pool:HSM-POOL
Enabled pools:HSM-POOL

$ SMU CHECKPOINT 1

$ SMU SHOW ARCHIVE 1

HSM$ARCHIVE01 has been used
Identifier:1
Media type:TZ87
Density:<NONE>
Label:PLS500
Position:0
Device refs:1
Shelf refs:1
Current pool:HSM-POOL
Enabled pools:HSM-POOL
```

---

## SMU COPY

### COPY

The SMU COPY command copies files, including shelved files, from one location to another without unshelving the files, and updates the appropriate HSM catalog to reflect the new location. The input files can be optionally deleted after a successful copy.

The SMU COPY command requires SYSPRV, TMPMBX, SYSLCK, and BYPASS privileges.

### Format

SMU COPY source destination

### Parameters

#### source

A full file specification (including disk and directory) of files to be copied.

- Wildcards are allowed
- A file list is not allowed

#### destination

A disk and directory optional specification for the output location of the files. The original filenames are preserved. If wildcards are used in the destination directory, the semantics of processing are the same as BACKUP, rather than OpenVMS COPY. In particular, if a whole disk is being copied, the output directory specification should be [\*...] rather than [000000...].

### Restrictions

- The source and destination devices must be different devices. Attempts to SMU COPY files to the same device will fail. Use the DCL RENAME command to “copy” files on the same device without unshelving files.
- Nodenames are not allowed in the source or destination specifications.
- The source and destination devices must exist.
- You cannot specify a filename or extension in the destination specification, just the device and directory.
- If the catalog server is enabled on the facility, then the SMU COPY command must be entered from the shelf server node. Use the SMU SHOW FACILITY command to determine the current server node before entering SMU COPY.

### Qualifiers

#### **/DELETE**

#### **/NODELETE**

Specifies whether the input file(s) should be deleted after the copy. If so, deleted shelved files are marked in the catalog for later delete.

#### **/LOG**

#### **/NOLOG**

Specifies if a log of each file copied is made to SYSS\$OUTPUT.

## Examples

The following example shows how to perform a group move of files from one location to another while preserving shelved files across the move. The source files are deleted after the copy.

```
$ SMU COPY/DELETE DISK$USER1:[JONES...]*.*;* DISK$USER10:[JONES...]
```

The following example copies all files from one disk to another, and retains the original files.

```
$ SMU COPY $1$DKA100:[000000...]*.*;* $12$DKA400:[*...]
```

---

## SMU DETECT

This command has been provided to detect if files were not properly archived as a result of the occurrence of a potential shelving error during use of the following versions of HSM:

- HSM 3.0
- HSM 3.0A
- HSM 3.1

The following is an indication of Potential Shelving Errors:

- NULL ACE, on the file header accompanied with an absence of catalog entry, for the file.  
For more information on Interpretation of Log files, please read the DESCRIPTION.

FORMAT

SMU DETECT VOLUME\_NAME[,...]

---

### Note

**Recovery: On running the tool, if the customer concludes that files were not properly saved during a shelving operation, the only known recourse to recover those files is from previous backups of the shelved files prior to the suspect shelving operations.**

---

## DESCRIPTION

The SMU DETECT command compares the online file's metadata for shelved and preshelved files on the specified volume(s). It checks for the above condition and reports the affected files.

It also checks for shelved/preshelved files with a missing ACE, and suggests to run the command SMU ANALYZER/REPAIR.

Interpretation of Log file:

Message 1:

<File Specification>

"No catalog entry found - file not repairable - Check Possible DATA LOSS"

The above message indicates a possible data loss i.e. possible loss of file.

Message 2:

<File Specification>

"ACE corrupted, Catalog Entry found"

The above message is generated for a file, with a NULL ACE, but a proper catalog entry.

Such files should be recoverable with an Unshelve command.

Message 3:

<File Specification>

```
"ACE corrupted, Catalog Entry found - Possible Manual Recovery"
```

The above message is generated for files with a NULL ACE, and a catalog entry with some discrepancy. These files require a manual recovery.

Message 4:

```
"Error ACE not found for"
```

```
<File Specification>
```

```
"Run SMU ANALYZE/REPAIR to fix catalog entry"
```

The above message is generated for files that do not have an ACE. SMU ANALYZE/REPAIR will add an ACE for these files, if a catalog entry is found.

## Qualifiers

### **/OUTPUT**

### **/OUTPUT[=*file\_specification*]**

The /OUTPUT qualifier specifies where to write the output of the command. The default is /OUTPUT=SYS\$OUTPUT. The output can optionally be sent to a file. By default, the output is written to the current SYS\$OUTPUT device. No wildcard characters are allowed.

## EXAMPLE

The following example performs a DETECT on the specified device and directs the output to the LOG file:

```
$SMU DETECT $1$DK100:/OUTPUT=EXAMPLE.LOG
```

## SMU EXIT

### EXIT

The SMU EXIT command exits the SMU utility. Alternatively, Ctrl/Z or the QUIT command can be used to exit.

### Format

SMU EXIT

### Example

The following example shows the SMU EXIT command:

```
SMU> EXIT
$
```

---

## SMU HELP

### HELP

The SMU HELP command displays information about an SMU command or topic. In response to the “Topic?” prompt, you can perform the following:

- Enter the name of the command or topic for which you need help.
- Enter HINTS if you are not sure of the name of the command or topic for which you need help.
- Enter a question mark (?) to redisplay the most recently requested text.
- Press the Return key one or more times to exit from Help.

You can abbreviate any topic name, although ambiguous abbreviations result in all matches being displayed. You also can enter HELP SMU from the DCL command line.

### Format

SMU HELP

### Example

The following example shows the SMU HELP command:

```
$ SMU HELP SHOW CACHE
```

```
SHOW
```

```
CACHE
```

```
The SHOW CACHE subcommand displays the configuration associated with the
cache on the specified volumes.
```

```
Note:
```

```
The SMU SHOW CACHE command requires SYSPRV privilege.
```

```
Format
```

```
SMU SHOW CACHE [volume_name,...]
```

```
Additional information available:
```

```
Additional information available:
```

```
Parameters Qualifiers
```

```
/OUTPUT
```

```
SMU SHOW CACHE Subtopic?
```

---

## SMU LOCATE

### LOCATE

The SMU LOCATE command finds information on shelved files and can be used when the normal unshelve command does not work. Usually, the file to be located cannot be found due to one of the following reasons:

- disk crash
- deleted file
- corrupted file header

It summarizes the information at the end of the output as follows:

```

$ SMU LOCATE
.
.
.
Total of 16766 entries found
Total of 10522 files found

```

Entries refer to the catalog entries and files refer to the unique filenames. Hence there may be a difference in these numbers. A Single file can have multiple catalog entries.

SMU LOCATE accepts standard RMS file specification conventions. You can specify either Wildcard characters or, when known, the file identifier (FID). When specifying a FID, any file specification entered as a parameter is ignored.

When entering file specifications, these apply to the file at the time of shelving. If the file has been renamed since the time of shelving, the online filename will not match the name in the catalog, and will not be located. In these situations, you may wish to search by FID instead.

If you use this command and do not provide a file specification, it displays the entire contents of the catalog.

### Format

```
SMU LOCATE [file_spec[,...]]
```

### Parameters

#### **[file\_spec[,...]]**

Specifies one or more files to locate. If two or more files are specified, separate them with commas. Wildcard characters are allowed.

The file specifications (including wildcards) apply to the file names as they are stored in the catalog at the time of shelving, which is not necessarily the same as the current online file name.

If you do not provide a file specification, the default `*:[000000...]*.*;*` is used. Any part of a file specification limits the Wildcard specification from `*:[000000...]*.*;*`. The concept of default device and directory do not apply.

### Restrictions

- NODE specifications are ignored because this information is not stored in the HSM catalog.
- Version specifications for latest versions or related versions do not work. For example, `[mydir]*.*;` or `[mydir]*.*;-1` do not work.



## Qualifiers

***/ARCHIVE=archive\_id[,...]***

Displays catalog entries of the specified archive\_id(s).

***/BRIEF (Default)***

Displays the catalog file name and file identifier (FID) of the specified files.

***/CACHE=device\_name[,...]***

Displays catalog entries of the specified device name(s).

***/CATALOG=file\_spec[,...]***

Displays catalog entries of the specified catalog file name(s).

***/DUMP***

Displays all fields of the catalog entries.

***/FID=file\_identifier***

Locates a file by file identifier instead of file specification. The /FID qualifier overrides any file specification on the command line. The specified FID may find a match on more than one file if the files are located on different devices.

***/FULL***

Displays a complete list of information about the specified file. Once the file is located, it can be restored manually using a BACKUP command.

***/OUTPUT[=filespec]***

***/NOOUTPUT***

Controls where to write output of the command. By default, the display is written to the current SYS\$OUTPUT device. No Wildcard characters are allowed.

***/VOLUME=volume\_label [,...]***

Displays catalog entries of the specified tape volume label(s).

## Examples

The following example shows how to display the contents of the entire HSM catalog using the SMU LOCATE command:

```
$ SMU LOCATE

$1$DKA0:[000000]
CF5.DAT;1(6542,14636,0)
CF6.DAT;1(6542,14637,0)
FURROW.DAT;1(5989,24877,0)
FOOL.DAT;1(7293,49830,0)

$1$DKA0:[HSM]
LAURA.TXT;1(6386,55397,0)

$1$DKA100:[BAILEY.CODE.EXE]
BIG.DAT;1(503,99,0)
F10.DAT;1(265,56,0)
F11.DAT;1(267,76,0)
F12.DAT;1(269,41,0)
F15.DAT;1(276,72,0)

$1$DKA100:[SAMPLE]
X.C;1(1435,38,0)

$1$DKA200:[BUCKLEY]
LOGIN.COM;1(2316,1,0)
MYFILE.TXT;1(18117,25594,0)
X.TXT;1(18051,5,0)
X.X;1(18116,25591,0)
```

## SMU LOCATE

```
$1$DKA200:[JENSEN.HSM.CLASS]
SHP_REQUEST.LIS;2 (18600,1,0)
```

```
$1$DKA200:[JENSEN.HSM]
X.X;4(18792,1,0)
```

```
$1$DKA200:[JENSEN]
TESTFILE.TXT;1 (12551,9,0)
X.X;3 (18539,1,0)
X.X;7 (18119,25600,0)
```

```
$1$DKA200:[WELLS.HSM.WORK]
LOGIN.COM;83(5038,5,0)
LOGIN.COM;84 (13147,1,0)
UNSHELVE_FILE.TMP;1(108,270,0)
UNSHELVE_FILE.TMP;1(102,150,0)
UNSHELVE_FILE.TMP;2(173,42,0)
UNSHELVE_FILE.TMP;3(241,44,0)
```

The following example shows how use SMU LOCATE to locate and recover a deleted file from the shelf copy:

```
$ DIRECTORY SMU_SRC.MMS
%DIRECT-W-NOFILES, no files found

$ SMU LOCATE SMU_SRC.MMS/FULL

$1$DKA200:[WELLS.HSM.WORK.TEMP]

SMU_SRC.MMS;1 (2538,39,0)
<online file not found>
Shelf copy:
Volume: HS0001 Saveset: 2DDB74A67.BCK Position: 4116
14/15 08-Apr-2002 14:11:53.16
Media type: CompacTape III, Loader
Shelved on 08-Apr-2002 14:11:53.16

$ MOUNT/FOREIGN $1$MUA0: HS0001
%MOUNT-I-MOUNTED, HS0001 mounted on _$1$MUA0:
$ BACKUP $1$MUA0:2DDB74A67.BCK/SAVE_SET SMU_SRC.MMS
$ DIRECTORY SMU_SRC.MMS

Directory $1$DKA200:[WELLS.HSM.WORK.TEMP]

SMU_SRC.MMS

Total of 1 file.
```

The following examples show how SMU LOCATE interprets file specifications with respect to the Wildcard default.

1.

```
$ SMU LOCATE X.TMP
```

Is interpreted as:

```
$ SMU LOCATE *:[000000...]X.TMP;*
```

2.

```
$ SMU LOCATE DISK1:X.TMP;5
```

Is interpreted as:

```
$ SMU LOCATE DISK1:[000000...]X.TMP;5
```

3.

```
$ SMU LOCATE [DIR1]X
```

Is interpreted as:

```
$ SMU LOCATE *:[DIR1]X.*;*
```

---

## SMU LOGICAL NAMES

HSM officially supports logical names as follows:

- HSM\$BLOCK\_SIZE - can be defined to the block size that can be used by HSM\$BACKUP. Users should use a block size that is supported by all tape drives in their environment. This will be made an attribute of a device in the next version of HSM.
- HSM\$CACHE\_FILE\_THRESHOLD\_SIZE - Defining this logical to some numeric value, would by-pass shelving of files with size less than the value specified to the cache and would get directly shelved to tape.
- HSM\$CATALOG - Points to the device and directory that contain the default HSM catalog. This logical is not used for shelf catalogs that are defined in the SMU SHELF database. Required.
- HSM\$CACHE\_RENAME - This logical when set to 1 in the system table will allow SMU ANALYZE/RENAME to remove the name discrepancy for files shelved only to cache.
- HSM\$FAULT\_AFTER\_OPEN - When defined to a list of process names, the shelf handler initiates a "background" restore request for the file when an informational request (on file open) is issued from one of the listed processes - it then waits 3 seconds and either completes the request as a file fault normally (e.g after a restore from cache), or returns  
Designed for use with PATHWORKS.
- HSM\$FAULT\_ON\_OPEN - When defined to a list of process names, the shelf handler performs a file fault on Open when the request is issued from one of the listed processes -  
Designed for NFS access.
- HSM\$INIT\_TAPE - On defining this logical to anything system wide, HSM will initialize the newly allocated tape, before writing to it. The logical should be defined in the HSM server node for internal checkpoint to initialize the tapes. In an cluster environment, the logical should also be defined in the client node if the user wishes to give the SMU CHECKPOINT<archive> from the client node.
- HSM\$LOG - Points to the device and directory that contain the HSM startup, audit and error logs, and the location of the activity log generated by SMU SHOW REQUESTS/FULL.  
Required.
- HSM\$MANAGER - Points to the device and directory that contain the SMU database files, the (Basic Mode) tape device/magazine database and the request log files.  
Required.
- HSM\$NO\_FAST\_TAPE - When defined to anything, indicates that the shelf handler uses OpenVMS skipfile functions for tape positioning, rather than fast SCSI and DSSI positioning algorithms.
- HSM\$POLICY\_PFQUOTA - This logical can be used to specify the page file quota for the detached process created for executing the primary and secondary policy script files. In the absence of this logical HSM uses the system default page file quota as specified in by the system parameter PQL\_DPFLQUOTA. In the following example HSM will set 200000 pages in VAX and 200000 pagelets in ALPHA as the pagefile quota for the newly created process.

```
DEFINE/SYS HSM$POLICY_PFQUOTA 200000
```

- HSM\$TEMP\_RESTORE\_DISK - is introduced to enable users to select the device where the temporary files are created during an unshelve operation.

---

## SMU RANK

### RANK

The SMU RANK command displays an ordered list of files that would be shelved if the specified policy were to run on the specified volume. Use this command to analyze the impact of specifying various policy attributes, the amount of shelving for each policy run, or detect files that should not be shelved in advance. You can apply the RANK command to both preventative and reactive policies. The RANK command does not actually shelve any files.

### Format

```
SMU RANK volume_name policy_name
```

### Parameters

***volume\_name***

Specifies the online disk volume on which the policy would be applied. You can specify only one volume, and no wildcards are allowed.

***policy\_name***

***policy\_name***

Specifies the preventative or reactive policy to run on the volume. You can specify only one policy, and no wildcards are allowed.

### Qualifiers

***/OUTPUT[=filespec]***

***/NOOUTPUT***

Controls where to write output of the command. By default, the display is written to the current SYS\$OUTPUT device. No Wildcard characters are allowed.

***/UIC=[uic]***

Selects only those files belonging to the specified owner user identification code (UIC).

You must specify a valid UIC qualifier value, enclosed in brackets, as described in the VMS DCL Concepts Manual. Use this qualifier to emulate a policy triggered by an exceeded quota event.

### Examples

The following examples show usage of the SMU RANK command.  
The first example shows the /UIC qualifier:

```
$ SMU RANK SYS$DISK HSM$DEFAULT_OCCUPANCY/UIC=[SMITH]
```

```
Volume capacity:           649040 blocks
Current utilization:       396380 blocks
Volume lowwater mark:     324520 blocks
Blocks to be reclaimed:   71860
```

```
Executing primary policy definition
DISK$USER1:[SMITH.V12]TEST.EXE;1
  date: 08-Apr-2002    size: 13281
```

```
DISK$USER1:[SMITH]LSTARV032.A;1
  date: 08-Apr-2002    size: 3564
```

```
DISK$USER1:[JONES.TMS.LIB]TEST.OLB;1
```

```
DISK$USER1:[SMITH.CODE.LIB]PROD_V2.OLB;1
  date: 08-Apr-2002  size: 16218
```

```
DISK$USER1:[SMITH.CHECKOUT]TMSA012.B;1
  date: 08-Apr-2002  size: 7020
  date: 08-Apr-2002  size: 7020
```

```
DISK$USER1:[JONES.LST]LST$MAIN_DEBUG.EXE;1
  date: 08-Apr-2002  size: 3306
```

```
DISK$USER1:[SMITH]TMSA012.C;1
  date: 08-Apr-2002  size: 10386
```

Total of 8 files ranked which will recover 80205 blocks

The following example shows the SMU RANK command with no qualifiers:

```
$ SMU RANK SYS$DISK HSM$DEFAULT_OCCUPANCY
```

Policy HSM\$DEFAULT\_OCCUPANCY is enabled for shelving

Policy History:

```
  Created:      08-Apr-2002 11:01:56.85
  Revised:     08-Apr-2002 11:01:56.85
```

```
Selection Criteria:  State: Enabled
Action:             Shelving
File Event:        Modification date
Elapsed time:      180 00:00:00
Before time:       <none>
Since time:        <none>
Low Water Mark:    80 %
Primary Policy:    Space Time Working Set (STWS)
Secondary Policy:  Least Recently Used (LRU)
```

Verification:

```
  Mail notification: <none>
  Output file:       <none>
```

```
Quota capacity:      10000 blocks
Current utilization:  5745 blocks
Quota lowwater mark: 5000 blocks
Blocks to be reclaimed: 745
```

Executing primary policy definition

```
DISK$USER2:[SMITH]AUDIT.C960214-390;1
  date: 08-Apr-2002  size: 20568
```

Total of 1 files ranked which will recover 20568 blocks

Quota lowwater mark can be reached

---

## SMU REPACK

### REPACK

The SMU REPACK command copies valid shelved file data from one shelf volume to another, while eliminating deleted or obsolete data. It can also be used to recover lost or destroyed shelf volumes. The appropriate catalog(s) are updated to reflect the file data's new shelf location. Use SMU REPACK in one of the following ways:

- To repack volumes in the same archive class to eliminate obsolete data and recycle the original volumes, specify the archive class in the `archive_id` parameter, and omit both the `/TO_ARCHIVE` and `/FROM_ARCHIVE` qualifiers.
- To repack volumes from one archive class to a new archive class, specify the original archive class in the `archive_id` parameter, and the `/TO_ARCHIVE` qualifier to specify the new archive class.
- To recover lost shelving volumes using data from another archive class, specify the archive class to recover in the `archive_id` parameter, and use the `/FROM_ARCHIVE` to specify the class from which to retrieve the data.

In each case, you may restrict the repack to a set of input volumes using the `/VOLUME` qualifier. If you do not specify volumes using the `/VOLUME` qualifier, all volumes in the archive class are repacked.

HSM does not repack obsolete files. Obsolete files are determined by the SHELF attributes previously set. For example, if the SHELF `/SAVE_TIME` was previously set to 60 days, HSM does not repack any files for which it has been longer than 60 days since the online file was deleted. Similarly, if the SHELF attribute was set to `/UPDATES=5`, HSM will not repack any file updates that are not among the latest 5 file updates.

If `/UPDATES=0` was previously set, it specifies that all file updates or file revisions, will be retained during repack. For further clarification on obsolete files refer to the SMU SET SHELF command and its `/UPDATES` qualifier.

SMU REPACK requires a disk staging area of at least 100,000 blocks in order to consolidate savesets. You may define a system-wide logical name of `HSM$REPACK` to point to a device and directory containing this amount of space. If this logical name is not defined, the staging area will be `HSM$MANAGER` instead. You should ensure that sufficient staging space is available prior to starting the repack. The staging area is cleaned up following the repack.

Repack also requires the use of two tape drives, compatible with the archive class(es) specified in the repack command. One drive must be compatible with the `archive_id` parameter, and the second must be compatible with any `/TO_ARCHIVE` or `/FROM_ARCHIVE` specified, or with the `archive_id` parameter if neither are specified.

The repack operation starts writing savesets to either the current shelving volume for the output archive class, or to a new shelving volume. A new volume is needed only for the case where the current shelving volume itself is eligible to be repacked, either explicitly or implicitly.

The repack function is performed on the shelf server node, but the command may be entered from any node.

The following are the logical names under REPACK.

- `HSM$F_ARP_ERROR_LIMIT` - While performing repack, if the volume from which repack is being performed is corrupt, the repack process would abort after a definite number of retries to read the volume. The number of retries it should make can be set by defining the system wide logical `HSM$F_ARP_ERROR_LIMIT` to an integer. The default is 3.



- **HSM\$F\_ARP\_PRINT\_DEBUG** - Controls the output of repack debug and tracing information. It must be defined as either a valid file specification or as `SYSS$OUTPUT`.
- **HSM\$F\_SHP\_BYPASS\_REPACK\_CONFIG** - When defined to anything, causes the shelf handler to disregard any drive deficiencies during Repack configuration validation.
- **HSM\$REPACK** - Points to a device and directory used as a staging area for a repack operation. It is recommended that at least 100,000 blocks are available on the `HSM$REPACK` disk. If not defined, HSM uses `HSM$MANAGER` for this purpose. The logicals `HSM$CATALOG`, `HSM$LOG` and `HSM$MANAGER` must be correctly defined for HSM to operate. All other logical names are optional.

## Format

SMU REPACK *archive\_identifier*

## Parameters

### ***archive\_identifier***

This specifies the identifier of the archive class to repack. Only one archive class can be repacked at a time.

## Restrictions

This command requires `BYPASS` privilege. The `/FROM_ARCHIVE` and `/TO_ARCHIVE` qualifiers are mutually exclusive. If the `/FROM_ARCHIVE` qualifier is specified, the `/THRESHOLD` qualifier is ignored.

## Qualifiers

### **/CANCEL**

Cancels the current repack operation.

### **/CORRUPT**

This qualifier has been added to the repack command and is used in conjunction with `/FROM`. It takes the name of a tape as input and deletes all catalog entries pertaining to this tape volume. This helps in removing unwanted references for a corrupt tape.

### **Example:**

If `tape1` belonging to archive 1 is corrupt and if its contents need to be repacked from `tape2`, belonging to archive 2, then the catalog entries for `tape1` remain useless in the HSM catalog after the repack is done. These entries can now be removed from the HSM CATALOG by adding the corrupt qualifier with the corrupt tape's name whose corresponding entries need to be purged.

### **Example:**

```
$ SMU REPACK 1 /FROM 2 /VOL=TAPE1 /NOTHRESHOLD /CORRUPT=TAPE1
```

### **/FROM\_ARCHIVE=archive\_id**

You can repack an archive class using the data from a different (but duplicate) archive class. Use this option to restore lost or destroyed shelf volumes.

### **/LIST[=*file\_specification*]**

### **/NOLIST**

List all files repacked in the specified file, or to the default `HSM$LOG:HSM$REPACK.LIS` if no file specification is supplied. The supplied file specification must not exceed 140 characters.

**/PURGE**

This new to the repack command deletes all obsolete entries with reference to the archive class specified. Actual repack operation is not attempted with this operation and only the HSM catalog is cleaned of obsolete entries.

**Example:**

```
$ SMU REPACK 1 /PURGE
```

**/REPORT**

If REPORT option is specified, Repack will only perform the analysis phase of a repack and not actual repacking. This feature would be extremely useful for a system manager to:

- analyze repacking requirements/benefits
- select the most useful threshold values
- and schedule repacks at convenient times

If used with the /SAVE option, the resultant candidates file will be saved and can be used in subsequent repack/s if the system manager wants the entire repack, as analyzed, to proceed.

**/RESTART**

Since repacks can take several hours/days to complete, it would be useful to allow the continuation of a repack that had been interrupted for any reason. The /RESTART qualifier would utilize the current candidates file and restart the repack from where it left off, without a further analysis or repacking files/volumes that had already been repacked.

**/SAVE**

Repack deletes the candidates file when it completes normally and in case of error (or on /CANCEL). With a /SAVE qualifier, the candidates file would be retained, and a subsequent repack could continue the repack from where it left off with a REPACK/RESTART. In addition, the candidates file could be retained from a REPACK/REPORT/SAVE command, which performs just the analysis phase. Subsequently the data moving phase could be initiated by a REPACK/RESTART.

**/THRESHOLD[=percent]****/NOTHRESHOLD**

Defines a threshold of deleted and obsolete data for consideration on whether to repack a particular volume or volume set. The default is a threshold of 50%, meaning that a volume (set) will only be repacked if its percentage of obsolete data (determined by the number of obsolete files) is 50% or more of the total number of files on the tapes.

The /NOTHRESHOLD qualifier (or /THRESHOLD=0) repacks all associated volumes in the archive class. A threshold of 100% repacks only volumes that contain no valid data.

**/TO\_ARCHIVE=archive\_id**

Defines a different archive class, possibly containing different media types, as the output of the repack. Use this qualifier to copy data from one archive class to another, or to upgrade to new tape technology.

**/VOLUME[=volume\_name[,...]]**

Specifies specific volumes in an archive class to be repacked. A volume list is supported but wildcards are not allowed. If a specified volume is a member of a volume set, the entire volume set is eligible for packing. The volume list can contain up to 9 volumes.

## Examples

In this example, perform a repack of volumes containing files shelved to archive class 1. Select from all volumes in the archive class, and repack only those in which the percentage of records belonging to obsolete files exceeds the default threshold of 50%. New volumes are allocated to the archive class and repacked volumes are released from the archive class automatically.

```
$ SMU REPACK 1
```

In this example, perform a repack of volumes containing files shelved to archive class 2. Select from all volumes in the archive class, and repack only those in which the percentage of obsolete files exceeds 80%. New volumes are allocated to the archive class and repacked volumes are released from the archive class automatically. List all files which are moved to new volumes in HSM\$LOG:HSM\$REPACK.LIS.

```
$ SMU REPACK 2/THRESHOLD=80/LIST
```

In this example, perform a repack of volumes containing files shelved to archive class 11. Select from volumes in the volume set which contains volume AFA200, and repack all volumes in the volume set, regardless of the percentage of records that belong to obsolete files. New volumes are allocated to the archive class and repacked volumes are released from the archive class automatically.

```
$ SMU REPACK 11/VOLUME=AFA200/NOTHRESHOLD
```

In this example perform a repack of volumes containing files shelved to archive class 21 onto volumes compatible with those in archive class 22. Repack all volumes in the archive class, regardless of the percentage of records which belong to obsolete files. New volumes are allocated to archive class 22 while repacked volumes are released from the archive class 21 automatically. List all files which are moved to new volumes in REPACK\_21.LOG in the current directory.

```
$ SMU REPACK 21/TO_ARCHIVE=22/LIST=REPACK_21.LOG/NOTHRESHOLD
```

In this example, replace the non-obsolete shelved file data on volumes in archive class 16 which are members of the volume set which contains volume HSM001, using shelved file data on volumes in archive class 32. New volumes are allocated to the destination archive class; no volumes are released from the archive class.

```
$ SMU REPACK 16/FROM_ARCHIVE=32/VOLUME=HSM001
```

---

## SMU SET ARCHIVE

### SET

### ARCHIVE

The SMU SET ARCHIVE command defines and modifies information about an archive class. You must use this command to tell HSM Basic mode what archive classes are available and to associate archive classes with media type definitions for HSM Plus mode.

### Format

SMU SET ARCHIVE *archive\_id[,...]*

### Parameters

***archive\_id[,...]***

Specifies the archive classes to be set.

### Qualifiers

***/ADD\_POOL=(pool\_name[,...]) (Plus mode only)***

Adds a volume pool to the list of pools for the archive class. The pool\_name must be assigned in MDMS. This qualifier is valid only for HSM Plus mode.

***/CONFIRM***

***/NOCONFIRM (Default)***

Controls whether a question is asked before each SET ARCHIVE operation to confirm that the operation should be performed on that archive class. The following responses are valid:

YES	NO	QUIT
TRUE	FALSE	Ctrl/Z
1	0	ALL

Refer the OpenVMS DCL Concepts Manual for more information on the /CONFIRM qualifier.

***/DELETE***

Deletes an archive class. This qualifier is valid only if no files have been shelved to this archive class.

***/DENSITY=density (Plus mode only)***

If a value is assigned to the DENS\_n value in TAPESTART.COM for the media type to be used for this archive class, then you must use this qualifier and specify the exact same string as is specified in TAPESTART.COM. This qualifier is valid only for HSM Plus mode.

Once an archive class has been used, you cannot modify the density.

***/LABEL=volume\_label (Basic mode only)***

Explicitly sets the desired volume label for the HSM Basic mode archive class. Use this command only if there has been a system failure, or if HSM cannot determine the correct volume label. This command provides an emergency reset for recovery purposes only. The next shelving operation for the archive class is performed on the specified volume. The volume label must conform to the HSM Basic mode volume label convention of HSxnnn.

Compaq recommends you use the SMU CHECKPOINT command instead of SET ARCHIVE /LABEL for removal of shelving volumes under normal circumstances. When you reset the volume label with SET ARCHIVE /LABEL, the shelving tape position is reset to zero.

***/LOG******/NOLOG (Default)***

Specifies that a command acknowledgment is displayed. If the command is specified with the /WAIT qualifier or if the shelf handler can process the command immediately, the log output contains the final result of the operation. If the command is specified with the /NOWAIT qualifier, a message indicating that the request is queued may be displayed on success for long-running operations.

***/MEDIA\_TYPE=media\_type (Plus mode only)***

Required qualifier for HSM Plus mode that identifies the media type defined in TAPESTART.COM to use for this archive class. The media type must exactly match a defined media type in TAPESTART.COM. This qualifier is valid only for HSM Plus mode.

Once an archive class has been used, you cannot modify the media type.

***/REMOVE\_POOL=pool\_name (Plus mode only)***

Removes a volume pool from the list of pools for the archive class. This qualifier is valid only for HSM Plus mode.

***/SAVESET\_SIZE=SaveSet\_size in MB***

Specifies the maximum SAVESET\_SIZE that will be used in a shelving operation. Multiple files upto the maximum SAVESET\_SIZE specified or 1000 files will be combined to make a single saveset to improve performance.

---

**Note**

---

**Note that increasing the SAVESET\_SIZE will also increase the unshelving time.  
The default maximum SAVESET\_SIZE is 40 MB.**

---

## SMU SET ARCHIVE

### Examples

The following example shows the SMU SET ARCHIVE command for HSM Basic mode:

```
$ SMU SHOW ARCHIVE 1

%SMU-W-ARCHID_NF, archive class id 1 not found
%RMS-E-RNF, record not found

$ SMU SET ARCHIVE 1

$ SMU SHOW ARCHIVE 1

HSM$ARCHIVE01 has not been used
Identifier: 1
Media type: Unknown Type
Label: HS0001
Position: 0
Device refs: 0
Shelf refs: 0
SAVESET_SIZE:40 MB
```

The following example shows the SMU SET ARCHIVE command for HSM Plus mode:

```
$ SMU SHOW ARCHIVE 2

%SMU-W-ARCHID_NF, archive class id 2 not found
%RMS-E-RNF, record not found

$ SMU SET ARCHIVE 2 /MEDIA_TYPE=TZ87

$ SMU SHOW ARCHIVE 2

HSM$ARCHIVE02 has not been used

Identifier: 2
Media type: TZ87
Density: <NONE>
Label: <NONE>
Position: 0
Device refs: 0
Shelf refs: 0
Current pool: <NONE>
Enabled pools:<NONE>
```

---

## SMU SET CACHE

### SET

### CACHE

The SMU SET CACHE command creates or modifies an existing HSM cache volume or a magneto-optical volume. Any device that appears as a system-mounted FILES-11 disk volume can be used as a cache volume, including magneto-optical devices.

You can use cache volumes for one of two purposes:

- As a temporary online staging area to speed shelving operations. A cache used for this purpose is set up with a limited block size, and a regular flush interval. Shelving operations are initially directed to the cache and complete in a similar amount of time as a normal file copy. At a later time, the cache is flushed to the archive classes defined for nearline or offline storage, and files in the cache are deleted.
- As an alternative shelf, using magneto-optical devices or excess online disk devices. A cache used for this purposes usually uses the entire device for caching, but does not flush the files to nearline or offline storage. Rather, additional copies are made to the nearline or offline archive classes at shelving time, using the /BACKUP qualifier.
- HSM optionally periodically flushes the cache to the shelf archive classes. The interval at which the flush occurs is controlled by the /INTERVAL and /AFTER qualifiers.

By default, caching is disabled.

### Format

```
SMU SET CACHE device_name[,...]
```

### Parameters

***device\_name[,...]***

Specifies the online system-mounted devices that are to be used as shelf cache. If two or more devices are specified, separate them with commas. Wildcard characters are only allowed for updates to existing cache devices.

### Qualifiers

***/AFTER[time]***

Requests that the cache flush operation be held until after a specific time. If no time value is specified, an immediate cache flush begins. Use this qualifier without a value to trigger an immediate cache flush at any time.

Specify the time as either an absolute time, a delta time, or a combination time. For more information on specifying time values, see the OpenVMS DCL Concepts Manual.

**/BACKUP**  
**/NOBACKUP (Default)**

Controls whether copies to the shelf archive classes are made when the file is moved to the cache at shelving time (/BACKUP), or when the cache is flushed at a later time (/NOBACKUP). The default is /NOBACKUP, which provides fast shelving and unshelving, but only provides one copy of the shelved data until a cache flush is performed.

The /BACKUP qualifier creates the shelf copies concurrently with the cache copy, resulting in slow shelving but fast unshelving. The shelf operation is not completed until all copies are made. Use this option when a cache device is used as a permanent repository rather than a staging area.

**/BLOCK\_SIZE[=BLOCK\_SIZE]**

The maximum number of blocks HSM can use for caching on the volume. The default is 50,000 blocks. If any shelf operation would exceed the block size of the cache, the file is shelved directly to the shelf archive classes instead.

If the /BLOCK\_SIZE is set to zero the entire disk may be used for the cache.

**/CONFIRM (Default)**  
**/NOCONFIRM**

Controls whether a request is issued before each SET CACHE operation to confirm that the operation should be performed on that disk volume. The following responses are valid:

YES	NO	QUIT
TRUE	FALSE	Ctrl/Z
1	0	ALL

Refer the OpenVMS DCL Concepts Manual for more information on the /CONFIRM qualifier.

**/DELETE**

Removes the cache entry from the database and causes an immediate flush of any files in the cache to the shelf archive classes.

**/DISABLE**

Disables the devices for use as an online shelf cache. However, no cache flush is triggered when a device is disabled.

**/ENABLE (Default)**

Enables the devices for use as an online shelf cache.

**/HIGHWATER\_MARK=percentage**

Specifies a high water mark usage capacity for the cache, at which point cache flushing is initiated. Files can continue to be shelved to the cache above the high water mark as long as the /BLOCK\_SIZE value is not exceeded. The default for the /HIGHWATER\_MARK qualifier for the cache is 80 percent of the block size.

If you wish to disable cache flushing based upon capacity, set the high water mark to 100 percent.

**/HOLD (Default)**  
**/NOHOLD**

Specifies whether the cache file is deleted when the online file is deleted or modified. /HOLD is the default and keeps the file in the cache when the online file is deleted or modified. /NOHOLD allows the file to be deleted from cache when the online file is deleted or modified. This reduces cache space in permanent caches and avoids unnecessary flushes of deleted or modified files. If you use the /NOHOLD option, HSM cannot recover deleted or modified files from cache.



**/INTERVAL[=*delta\_time*]****/NOINTERVAL**

Requests that the cache flush operation be performed at regular intervals. If no interval qualifier or value is specified, the default interval is six hours.

If /NOINTERVAL is specified, the cache is not flushed at regular intervals.

If a /HIGHWATER\_MARK value of less than 100 percent is specified, the cache is flushed based on usage only.

If the high water mark is set to 100 percent and /NOINTERVAL is specified, the cache is never flushed. Use this option when using the cache as an alternative shelf.

Delta time is expressed as dddd-hh:mm:ss.cc. The value used cannot be less than one minute.

**/LOG****/NOLOG (Default)**

Displays the device specification of each cache device modified as the command executes.

**Examples**

The following examples show the SMU SET CACHE command.

The first command sets up an online disk volume as a staging area, with a maximum usage of 100,000 blocks, and a flush interval of one day to begin at 4:00 am. By default, the cache is enabled, and a high water mark of 80 percent applies.

```
$ SMU SET CACHE _$15$DKA100: /BLOCK=100000/INTERVAL=1-
$ /AFTER=04:00:00
$ SMU SHOW CACHE _$15$DKA100:
```

```
Cache device _$15$DKA100: is enabled, Cache flush is held until
after 08-Apr-2002 04:00:00.00, Backup is performed at flush
intervals, Cached files are held on delete of online file
Block size: 100000
Highwater mark: 80%
Flush interval: 1 00:00:00.00
```

The second example sets up a magneto-optical volume as an alternative shelf. Shelf archive copies are made at shelving time, the entire volume may be used for caching, and there is no flushing.

```
$ SMU SET CACHE _$1$JKA0: /BLOCK=0/BACKUP/NOINTERVAL -
$ /HIGHWATER_MARK=100
$ SMU SHOW CACHE _$1$JKA0:
```

```
Cache device _$1$JKA0: is enabled, Cache flush is held until
after 08-Apr-2002 09:53:42.55, Backup is performed at
shelving time, Cached files are held on delete of online
file Block size: 0 Highwater mark: 100%
Flush interval: <none>
```

---

## SMU SET DEVICE

### SET

### DEVICE

The SMU SET DEVICE command specifies the nearline or offline storage devices that HSM can use to perform copies to and from the shelf archive classes. Devices specified in SET DEVICE must be tape devices, and must not be allocated or mounted by any user while HSM is using them. Devices can include tape magazine loaders, or large tape jukeboxes for HSM Plus mode. HSM uses OpenVMS BACKUP on foreign-mounted volumes for all shelf copy operations.

The devices can be either shared with other users, or exclusively dedicated to HSM. When a device is shared, HSM relinquishes control of the device after a short period of inactivity after the last shelving operation and no pending operations for the device.

Each device supports one or more archive classes, which must be associated with the device by using this command. Devices can be associated with any archive class with a compatible media type. For HSM Basic mode, the media type for an archive class is established implicitly when the first device is associated with it, and assumes the media type of that device. For HSM Plus mode, the media type and density are assigned explicitly to the archive classes which are associated with the device. Once a media type is established, a device with an incompatible media type cannot be associated with the archive class, unless both of the following are true:

- The archive class has not been used (no shelf copies have ever been made)
- No other device is associated with the archive class (device reference count is zero)

You can use the SMU SHOW ARCHIVE command to verify these conditions.

In addition, devices associated with an archive class must all be magazine loaders, or not be magazine loaders, at any one time.

Each device can be enabled for shelving, unshelving, or both operations. Assigning specific operations for a device can result in more optimal operations, fewer tape mounts and operator interaction, at the expense of hardware resources.

When specified for unshelving only, the association of device with archive class additionally includes media types that can read (but not write) the device. For example, a TK70 device (media CompacTape II) can be specified for unshelving an archive class written by a TK50 device (media CompacTape I), but not vice versa.

Refer the HSM Guide to Operations for a complete discussion of magazine loaders and compatible media types for HSM Basic mode.

When you set a device and do not use either the /SHARE or the /DEDICATE qualifiers, HSM defaults the device to be shared with other applications (/SHARE=ALL). If you want HSM to dedicate use of the device for shelving or unshelving operations, you must specify the /DEDICATE qualifier.

Compaq recommends you use the SMU SHOW DEVICE command after you input an SMU SET DEVICE command to confirm your device definitions.

## Format

SMU SET DEVICE *device\_name*[,...]

## Parameters

### ***device\_name*[,...]**

Specifies the nearline or offline devices that are to be used. If two or more devices are specified, separate them with commas. Wildcard characters are only allowed for updates to existing shelf devices.

If specifying a remote device in Plus mode, a node name or a logical that maps to a node name must precede the device name.

## Restrictions

In HSM Plus mode, for any devices that are not visible across the VMScluster (for example, SCSI devices that are not TMSCP served), SMU SET DEVICE must be entered on the node from which the devices are visible.

## Qualifiers

### ***/ARCHIVE\_ID=(archive\_id*[,...])**

### ***/NOARCHIVE\_ID***

Defines the identifiers for archive classes that are supported by this device. Up to 36 archive class identifiers may be listed. If only one archive class identifier is provided, you can omit the parentheses. The archive classes must already be defined prior to entering this qualifier.

The */NOARCHIVE\_ID* qualifier means there are no archive classes defined for the device.

### ***/CONFIRM***

### ***/NOCONFIRM (Default)***

Controls whether a request is issued before each SET DEVICE operation to confirm that the operation should be performed on that device. The following responses are valid:

YES	NO	QUIT
TRUE	FALSE	Ctrl/Z
1	0	ALL

Refer the OpenVMS DCL Concepts Manual for more information on the */CONFIRM* qualifier.

### ***/DEDICATE=keyword***

Allows you to dedicate a nearline or offline device exclusively to HSM. A device can be selected to service shelve, unshelve, or all operations. The possible keywords are as follows:

ALL	Dedicates the device for shelve, (Default) preshelve and unshelve operations. This is the default if you do not specify either <i>/DEDICATE</i> or <i>/SHARE</i> .
SHELVE	Dedicates the device exclusively for shelve and preshelve operations.
UNSHELVE	Dedicates the device exclusively for unshelve operations.

The */DEDICATE* and */SHARE* qualifiers are mutually exclusive.

The */DEDICATE* qualifier cannot be used with the */REMOTE* qualifier.

**/DEFAULT**

Updates the default device record HSM\$DEFAULT\_DEVICE.

**/DELETE**

Removes the storage device entry from the database. Pending requests for the device are requeued to other devices. If a volume is mounted on the device, it will take about one minute for that volume to be dismounted after you issue this directive.

**/DISABLE**

Disables the devices for use by HSM.

**/ENABLE (Default)**

Enables the devices for use by HSM.

**/LOG**

**/NOLOG (Default)**

Displays the device specification of each device modified as the command executes.

**/REMOTE (Plus Mode only)**

Required to tell HSM the device is remote (not directly connected to the VMScluster). The device specification must include the node name or the node name must be included in a logical name assignment for the device. The device must be identified through MDMS with the same exact specification as is used here.

The /DEDICATE qualifier cannot be used with the /REMOTE qualifier. You cannot dedicate remote devices for HSM use.

**/ROBOT\_NAME=robot\_name (Basic mode only)**

**/NOROBOT\_NAME (default)**

In Basic mode, if the device is a robot-controlled loader (for example, a TZ877 loader), you must specify a robot name for the device to be treated as a loader.

The robot name must be specified for the loader to be robotically controlled. If /NOROBOT\_NAME (the default) is specified, any loaders will be treated as sequentially loadable stackers. Only non-loader standalone tapes should be specified with /NOROBOT\_NAME.

This qualifier is not supported in Plus mode.

**/SHARE=keyword**

Allows the system administrator to share a nearline or offline device between HSM and other users. A device can be selected to share shelf, unshelve, or all operations with system users. Sharing devices with HSM allows the system administrator to allow multiuser access to the device. When enabled, the shelf handler relinquishes control of the device after the series of operations is completed.

The possible keywords are as follows:

ALL (Default)	Shares the device on shelf, preshelve, and unshelve operations with other system users.
SHELVE	Shares the device on shelf and preshelve operations with other system users.
UNSHELVE	Shares the device on unshelve operations with other system users.

The /DEDICATE and /SHARE qualifiers are mutually exclusive.

## Examples

The following example sets the device `$11$MUA0:`, a magazine loader, for shared use with archive classes `HSM$ARCHIVE01` and `HSM$ARCHIVE02` for HSM Basic mode. The device is connected to an HSJ controller with the robot name equal to the HSJ command disk name:

```
$ SMU SET DEVICE/ARCHIVE_ID=(1,2)/LOG/SHARE=ALL $11$MUA0:-
/ROBOT_NAME=$1$DUA204:
%SMU-I-DEVICE_CREATED, device _$11$MUA0: created
$ SMU SHOW DEVICE
```

```
HSM drive _$11$MUA0: is enabled.
Shared access: < shelve, unshelve >
Drive status: Configured
Media type: CompactTape III, Loader
Robot name: _$1$DUA204:
Enabled archives: HSM$ARCHIVE01 id: 1
HSM$ARCHIVE02 id: 2
```

The following Plus mode example sets the device `NODE1$MKB300` for dedicated shelving and unshelving, using archive classes 4 and 5:

```
$ SMU SET DEVICE/ARCHIVE_ID=(4,5)
/DEDICATE=ALL NODE1$MKB300:
$ SMU SHOW DEVICE NODE1$MKB300:
```

```
HSM drive _NODE1$MKB300: is enabled.
Dedicated access: <shelve, unshelve>
Drive status: Configured
Enabled archives: HSM$ARCHIVE04 id: 4
HSM$ARCHIVE05 id: 5
```

The following Plus mode example sets the remote device `$1$MUA23` on a remote node for shared shelving and unshelving, using archive classes 2 and 3.

```
$ SMU SET DEVICE/ARCHIVE_ID=(2,3)/REMOTE YOURNODE:::$1$MUA23:
$ SMU SHOW DEVICE YOURNODE:::$1$MUA23:
```

```
HSM remote drive YOURNODE::_$1$MUA23: is enabled.
Shared access: < shelve, unshelve >
Drive status: Configured
Enabled archives: HSM$ARCHIVE02 id: 2
HSM$ARCHIVE03 id: 3
```

---

## SMU SET FACILITY

### SET

### FACILITY

The SMU SET FACILITY command controls HSM operations on a VMScluster-wide basis, allows specification of designated shelf servers, controls event logging, and allows you to convert from HSM Basic mode to HSM Plus mode without reinstalling the HSM software.

### Format

SMU SET FACILITY

### Qualifiers

**/CATALOG\_SERVER****/NOCATALOG\_SERVER (Default)**

Determines whether all cache operations and catalog updates are performed by the shelf server node. /CATALOG\_SERVER restricts all cache operations and catalog updates to the current shelf server node. /NOCATALOG\_SERVER (default) allows cache operations and catalog updates to be performed by the requesting client node. This does not affect tape operations which are always performed on the shelf server node.

**/DISABLE=keyword**

Disables HSM from performing specified operations on the VMScluster. The command applies to all volumes on all shelves on all nodes. This qualifier also disables in-progress cache flushing.

ALL            Disables the facility for preshelve, shelve, unshelve and cache flush operations.  
SHELVE        Disables the facility for preshelve, shelve, and cache flush operations.  
UNSHELVE     Disables the facility for unshelve operations.

**/ENABLE=keyword**

Enables HSM to perform specified shelving operations. If neither /ENABLE nor /DISABLE are specified, HSM defaults to /ENABLE=ALL. The possible keywords are as follows:

ALL            Enables the facility for preshelve, shelve, unshelve and cache flush operations.  
SHELVE        Enables the facility for preshelve, shelve and cache flush operations.  
UNSHELVE     Enables the facility for unshelve operations.

**/EVENT\_LOGGING=keyword**

Enables and disables HSM event logging. The possible keywords are as follows:

[NO]ALL        Enables/disables the facility for all types of logging.  
[NO]AUDIT     Enables/disables audit logging.  
[NO]ERROR     Enables/disables error logging.  
[NO]EXCEPTION Enables/disables exception logging.

By default, all logging is enabled.

**/NOLOG (Default)**

Displays the facility that was modified as the command executes.

**/MODE=mode\_type**

Tells HSM the mode in which to operate. Valid keywords for this qualifier are:

- Basic     Operate HSM in Basic mode, which provides limited device support.
- Plus      Operate HSM in Plus mode, which provides enhanced device support through the Media, Device and Management Services for Open VMS (MDMS) software.

The initial default value for the mode depends on the mode in which HSM is installed. Once you run in HSM Plus mode, you cannot go back to HSM Basic mode. For information on converting from HSM Basic mode to HSM Plus mode, see HSM for OpenVMS Guide to Operations.

---

**Note**


---

**Changing the HSM mode through this qualifier does not affect HSM operations until you shut down and restart the shelf handler.**

---

**/REENABLE**

Reenables all operations for the facility. If the shelf handler is failing too often, HSM may disable the facility for shelving, unshelving, disk full, or quota error conditions. This command can be used to reenable the facility once the error condition is corrected.

**/RESET\_LOGS**

Closes the current audit and error logs, then starts writing to a new version. The closed logs can be subsequently reviewed, shelved or deleted.

**/SERVER=(server\_name[,...])****/NOSERVER**

Specifies which nodes can function as shelf servers.

Nodes are entered as a comma-separated list. Up to ten nodes can be entered as potential servers. Shelf server nodes must have access to all nearline/offline tape devices in Basic mode.

If /NOSERVER is entered, all nodes in the VMScluster are eligible to be a shelf server.

**Example**

The following example enables HSM for all error logging, and designates nodes SYS001 and SYS004 as shelf servers:

```
$ SMU SET FACILITY/EVENT_LOGGING=ALL/SERVER=(SYS001,SYS004)

$ SMU SHOW FACILITY

HSM for OpenVMS is enabled for Shelving and Unshelving
Facility History:
Created:    08-Apr-2002 10:36:34.96
Revised:   08-Apr-2002 13:49:37.50
Designated servers:  SYS001
SYS004
Current server:SYS001
Catalog server:Disabled
Event logging:  Audit
Error
Exception
HSM mode:     Basic
Remaining license: 500 gigabytes
```

---

## SMU SET POLICY

### SET

### POLICY

The SMU SET POLICY command defines a preventative or reactive policy that can be executed to maintain a desired capacity usage on online disk volumes.

### Format

SMU SET POLICY policy\_name[,...]

### Parameters

***policy\_name[,...]***

Specifies one or more policy names. If two or more policy names are specified, separate them with commas. Wildcard characters are allowed.

Policy names are limited to 31 alphanumeric characters, including the hyphen and underscore.

### Qualifiers

***/BACKUP***

Uses the backup date for selection criteria when using the */BEFORE* and */SINCE* qualifiers and the STWS and LRU policy algorithms. If a file has no backup date, it is not selected for the operation.

The */BACKUP*, */CREATED*, */EXPIRED*, and */MODIFIED* qualifiers are mutually exclusive. The default is */EXPIRED* if none of the above are entered. If you specify one of these qualifiers without the */BEFORE*, */ELAPSED*, or */SINCE* qualifiers, */BEFORE*, */ELAPSED*, and */SINCE* values are unaffected.

If a file has no backup date, it is not selected for shelving by the predefined algorithms.

***/BEFORE[=time] /NOBEFORE***

Selects only those files dated prior to the specified time. Specify the time as an absolute time, a combination time, or as one of the following keywords: TODAY (default), TOMORROW, or YESTERDAY.

Specify one of the following qualifiers with */BEFORE* to indicate the time attribute to be used as the basis for selection:

*/BACKUP*, */CREATED*, */EXPIRED*, or */MODIFIED*.

Time must be specified using the OpenVMS formats for absolute and delta times.

The values specified by the */BEFORE* qualifier are translated to absolute values that do not change between policy runs. To specify a relative before time, use the */ELAPSED* qualifier instead. The */BEFORE* qualifier is mutually exclusive of the */ELAPSED* qualifier.

The */NOBEFORE* qualifier clears any previously issued SET POLICY/*BEFORE* values.



**/CONFIRM****/NOCONFIRM (Default)**

Controls whether a request is issued before each SET POLICY operation to confirm that the operation should be performed on that policy.

The following responses are valid:

YES	NO	QUIT
TRUE	FALSE	Ctrl/Z
1	0	ALL

Refer the OpenVMS DCL Concepts Manual for more information on the /CONFIRM qualifier.

**/CREATED**

Uses the creation date for selection criteria when using the /BEFORE and /SINCE qualifiers and the STWS and LRU policy algorithms.

The /BACKUP, /CREATED, /EXPIRED, and /MODIFIED qualifiers are mutually exclusive. The default is /EXPIRED if none of the above are entered. If you specify one of these qualifiers without the /BEFORE, /ELAPSED, or /SINCE qualifiers, /BEFORE, /ELAPSED, and /SINCE values are unaffected.

**/DEFAULT**

Sets or modifies the system policy defaults. When a new volume definition is created, all default attributes are set from this default volume definition.

The /DEFAULT qualifier cannot be used with a parameter value.

**/DELETE**

Deletes the entry from the policy database.

**/DISABLE**

The policy is disabled from execution until it has been reactivated by the /ENABLE qualifier.

**/ELAPSED=delta\_time****/NOELAPSED**

Selects only those files that are equal to or older than the specified delta time. The time is expressed as a relative value and applied as such at each policy run.

Delta time is an offset (a time interval) from the current date and time to a time in the past. The general format for a delta time is as follows:

```
" [ dddd- ] [ hh:mm:ss.cc ] "
```

The fields are as follows:

dddd	Number of days	an integer in the range 0 to 9999
hh	Number of hours	an integer in the range 0 to 23
mm	Number of minutes	an integer in the range 0 to 59
ss	Number of seconds	an integer in the range 0 to 59
cc	Number of hundredths of seconds	in the range 0-99

The following table shows some examples of delta time specifications:

Time, Specification	Result
"30-"	30 days in the past (the dash is required)
"7-:25"	7 days and 25 minutes in the past
"90-12:15"	90 days, 12 hours, and 15 minutes in the past

The /ELAPSED qualifier is mutually exclusive of the /SINCE and /BEFORE qualifiers.  
The /NOELAPSED qualifier clears any previously issued SET POLICY/ELAPSED values.

**/ENABLE (Default)**

The policy is enabled for execution. The policy remains enabled until it is explicitly disabled.

**/EXPIRED (Default)**

Uses the effective date of last access for selection criteria when using the /BEFORE, /SINCE and /ELAPSED qualifiers, and the STWS and LRU policy algorithms. This qualifier uses the file's expiration date and the volume's retention time to determine the date of last access according to the following formula:

$$\text{date\_of\_last\_access} = \text{expiration\_date} - \text{maximum\_volume\_retention}$$

While specifying date and time values in the associated /BEFORE, /SINCE and /ELAPSED qualifiers, you should specify the date of last access rather than simply the expiration date, which is likely to be in the future.

If volume retention is not enabled on the volume, HSM will select no files for shelving when the policy is run. Therefore, to use file selection based on expiration date (the default), you must first enable volume retention on the volume (see HSM for OpenVMS Guide to Operations). HSM will automatically initialize expiration dates on all files on the volume to the current date plus the maximum retention time when the policy is first run.

The /BACKUP, /CREATED, /EXPIRED, and /MODIFIED qualifiers are mutually exclusive. The default is /EXPIRED if none of the above is entered. If you specify one of these qualifiers without the /BEFORE, /ELAPSED, or /SINCE qualifiers, /BEFORE, /ELAPSED, and /SINCE values are unaffected.

**/LOG****/NOLOG (Default)**

Displays a message for each schedule definition as the command executes.

**/LOWWATER\_MARK=[percentage]**

The /LOWWATER\_MARK qualifier establishes a threshold where policy execution terminates. It is set to the desired percentage of volume capacity that HSM attempts to reach while executing a policy.

Possible percentages range from zero to 100 percent. The shelving policy continues to shelve dormant files until the low water mark is reached.

For instance, if the low water mark is set to 20 percent, then HSM shelves dormant files until the disk usage is 20 percent or less. The default low water mark is 80 percent.

**/MAIL=destination****/NOMAIL (Default)**

Sets notification upon completion of the operation through user mail. The destination can be a specific user, such as SOCCR::SMITH, or a distribution list contained in a command procedure, such as @POLICY.DIS.

The mail header indicates success or failure. The message body contains a brief summary of shelving operations that were executed.

**/MODIFIED**

Uses the modification date for selection criteria when using the /BEFORE and /SINCE qualifiers and the STWS and LRU policy algorithms.

The /BACKUP, /CREATED, /EXPIRED, and /MODIFIED qualifiers are mutually exclusive. The default is /EXPIRED if none of the above are entered. If you specify one of these qualifiers without the /BEFORE, /ELAPSED, or /SINCE qualifiers, the /BEFORE, /ELAPSED, and /SINCE values are unaffected.

**/OUTPUT[=*file\_spec*]****/NOOUTPUT (Default)**

Records all shelving activities caused by the execution of the volume policy definition.

The /OUTPUT qualifier controls where the output record of the volume policy execution is sent. By default, no output file is created. No Wildcard characters are allowed.

If you enter the /OUTPUT qualifier with a partial file specification (for example, /OUTPUT=[JONES]), the policy\_name parameter is used as the file name with a default file extension of .LOG. The output is suppressed if you enter the /NOOUTPUT qualifier.

**/PRESHELVE****/NOPRESHELVE**

If /PRESHELVE is specified, the LRU and STWS algorithms preshelve files instead of shelving them. The /PRESHELVE qualifier is applicable only to periodically scheduled policies and should not be applied to volume occupancy full, high water mark exceeded or exceeded disk quota policies, as preshelving does not make disk space available.

**/PRIMARY\_POLICY=*keyword*****/NOPRIMARY\_POLICY**

The /PRIMARY\_POLICY qualifier sets the primary policy for selecting dormant files to be shelved. The primary policy is executed until the low water mark is reached or until no further candidates are found. If the low water mark is not reached, then the secondary policy is executed to select additional candidates. Typically, the secondary policy is set to a different algorithm than the primary. This algorithm is then executed until the low water mark is reached.

Setting the /NOPRIMARY\_POLICY qualifier effectively disables policy execution. For example, this can be used to disable any action on volume occupancy full or disk quota exceeded events.

The possible policy keywords are as follows:

STWS	Space-Time Working Set-Selects files based on the time since the last reference and the size of the file. Large files are selected with a higher priority than small files.
LRU	Least Recently Used-Selects files with the greatest time since the last reference.
SCRIPT= <i>filename</i>	A filename containing DCL shelving commands. This file is then used to select files for shelving, preshelving or unshelving. Note that scripts are run to completion and are not terminated when the low water mark has been reached.

When you use a script, HSM ignores the file selection qualifiers /BACKUP, /CREATED, /MODIFIED, or /EXPIRED, /BEFORE, /ELAPSED, or /SINCE.

SMU does not check for the existence of the script file until the policy execution time is reached.

If the specified date field has no value (for example, no backup or expiration date), the STWS and LRU algorithms do not select the file for shelving.

**/SECONDARY\_POLICY=keyword****/NOSECONDARY\_POLICY**

Secondary policy is used whenever the primary policy fails to reach the low water mark for the volume. It should be set to a different policy algorithm than the primary so different file selection criteria are used.

If no secondary policy is specified, the secondary policy from the default periodic policy is used.

The keywords for the /SECONDARY\_POLICY qualifier are the same as those used for the /PRIMARY\_POLICY qualifier.

**/SINCE=time****/NOSINCE**

Selects only those files dated after the specified time. Specify the time as an absolute time, as a combination of absolute and delta times, or as one of the following keywords: TODAY (default), TOMORROW, or YESTERDAY.

Specify one of the following qualifiers with /SINCE to indicate the time attribute to be used as the basis for selection: /BACKUP, /CREATED, /EXPIRED, or /MODIFIED.

Time must be specified using the OpenVMS formats for absolute and delta times.

The /SINCE qualifier is mutually exclusive of the /ELAPSED qualifier.

The /NOSINCE qualifier clears any previously issued SET POLICY/SINCE values.

The /BACKUP, /CREATED, /EXPIRED, and /MODIFIED qualifiers are mutually exclusive. The default is /EXPIRED if none of the above are entered. If you specify one of these qualifiers without the /BEFORE, /ELAPSED, or /SINCE qualifiers, /BEFORE, /ELAPSED, and /SINCE values are unaffected.

**Example**

The following example changes the default occupancy policy to shelve files that have not been accessed for read or write for 100 days. The shelving algorithms to be used are STWS as the primary policy and a user-defined script as the secondary policy, with a low water mark goal of 70 percent. At completion of the policy, mail confirmation is sent to the SYSTEM account. The example also illustrates that the attributes can be specified in multiple commands, as well as in a single command.

```

$ SMU SET POLICY HSM$DEFAULT_OCCUPANCY - _
$ /ELAPSED=100-0 -
$ /MAIL=SYSTEM -
$ /LOWWATER_MARK=70 -
$ /EXPIRED

$ SMU SET POLICY HSM$OCCUPANCY_POLICY -
$ /PRIMARY_POLICY=STWS -
$ /SECONDARY_POLICY=SCRIPT=DISK$USER1:[SMITH]SECPOL.COM -

$ SMU SHOW POLICY HSM$DEFAULT_OCCUPANCY
Policy HSM$DEFAULT_OCCUPANCY is enabled for shelving
Policy History:
Created: 08-Apr-2002 10:36:36.45
Revised: 08-Apr-2002 11:26:21.09
Selection Criteria:
State: Enabled
Action: Shelving
File Event: Expiration date
Elapsed time: 100 00:00:00
Before time: <none>
Since time: <none>
Low Water Mark: 70 %
Primary Policy: Space Time Working Set(STWS)
Secondary Policy: SCRIPT

```

```
DISK$SLOPER1:[SMITH]SECPOL.COM;  
Verification:  
Mail notification:  
SYSTEM  
Output file: <none>
```

---

## SMU SET SCHEDULE

### SET

### SCHEDULE

The SMU SET SCHEDULE command is used to schedule preventative policy execution for a volume. The policy can be scheduled to execute immediately or at periodic intervals.

### Format

SMU SET SCHEDULE volume\_name[,...], policy\_name

### Parameters

**volume\_name[,...]**

The name of one or more online volumes on which the shelving policy is to be executed.

If two or more volumes are specified, separate them with commas.

**policy\_name**

The name of an existing policy definition.

### Qualifiers

**/ALL**

Used in conjunction with wildcard volume names to select all matching volumes found on the system. If /ALL is not specified, Wildcard matching applies only to volumes defined with SET VOLUME in the SMU database.

**/AFTER[=time]**

Requests that the shelving policy execution be held until after a specific time. If no time value is specified, execution begins immediately.

Specify the time as either an absolute time or as a combination of absolute and delta times. For more information on specifying time values, see the OpenVMS DCL Concepts Manual.

**/CONFIRM****/NOCONFIRM (Default)**

Controls whether a request is issued before each schedule operation to confirm that the operation should be performed. The following responses are valid:

YES	NO	QUIT
TRUE	FALSE	Ctrl/Z
1	0	ALL

Refer the OpenVMS DCL Concepts Manual for more information on the /CONFIRM qualifier.

**/DELETE**

Deletes the entry from the scheduling database.

**/INTERVAL=*delta\_time***

Requests that the shelving policy be executed at regular intervals. This qualifier specifies the minimum time between shelving policy executions for the online volume. Delta time is expressed as dddd-hh:mm:ss.cc. If no interval value is specified, the default interval is one day.

If the /AFTER qualifier is specified with the /INTERVAL qualifier, the first shelving policy execution begins at or after the /AFTER time. All subsequent policy executions occur at the specified time.

Specify the time as either an absolute time or as a combination of absolute and delta times. For more information on specifying time values, see the OpenVMS DCL Concepts Manual.

The policy can be executed only if the policy definition has been enabled.

**/ENTRY=*number***

When a policy is scheduled for execution, an entry number is associated with it. Use the entry number when modifying an existing policy schedule event. Use SHOW SCHEDULE to determine the entry number.

**/LOG****/NOLOG (Default)**

Displays the policy specification of each policy modified as the command executes.

**/OUTPUT[=*filespec*]****/NOOUTPUT (Default)**

The /OUTPUT qualifier controls where the output record of the set schedule activity is sent. By default, no output file is created. No Wildcard characters are allowed.

**/REQUEUE=*server\_name***

In conjunction with the /ENTRY qualifier, the /REQUEUE qualifier requeues the policy for execution on a different shelf server node.

**/SERVER=*node\_name***

Specify the node name of the HSM shelf server on which to execute the policy. This does not have to be the current shelf server, but any eligible shelf server. If you do not specify a server node name, the default server becomes the first entry listed under Designated Servers in the SHOW FACILITY command display.

## SMU SET SCHEDULE

### Example

The following example shows the SMU SET SCHEDULE command. In this example, a preventative policy DAILY\_OCCUPANCY is run on a daily basis on volumes \$1\$DKA100: and \$1\$DKA200:, starting at 2:00 am, on shelf server node SYS001.

```
$ SMU SET SCHEDULE $1$DKA100:, $1$DKA200: DAILY_OCCUPANCY -  
$ /INTERVAL=1-0/AFTER=02:00:00/SERVER=SYS001
```

```
$ SMU SHOW SCHEDULE/FULL
```

<u>Volume</u>	<u>Policy</u>	<u>Entry</u>	<u>Status</u>
_\$1\$DKA100:	DAILY_OCCUPANCY	517	PENDING
Server node:	SYS001		
Scheduled after:	08-Apr-2002 02:00:00.00		
Scheduled interval:	1 00:00:00.00		
Output file	<none>		

<u>Volume</u>	<u>Policy</u>	<u>Entry</u>	<u>Status</u>
_\$1\$DKA200:	DAILY_OCCUPANCY	517	PENDING
Server node:	SYS001		
Scheduled after:	08-Apr-2002 02:00:00.00		
Scheduled interval:	1 00:00:00.00		
Output file	<none>		



---

## SMU SET SHELF

### SET

### SHELF

The SMU SET SHELF command creates a new shelf configuration or modifies an existing shelf configuration. A shelf describes the set of attributes associated with a set of online volumes, including the number of shelf data copies and the archive classes to be used. HSM operations also can be enabled and disabled on a per-shelf basis, affecting all volumes associated with the shelf.

HSM provides predefined default shelf definitions that are used when no shelf definitions are associated with a volume.

If the shelf definition already exists, it is modified with the new definition. When a new shelf is created, all default attributes are set from the default shelf definition. The default shelf definition can be modified by using the /DEFAULT qualifier.

Online disk volumes are associated with the shelf using the SMU SET VOLUME command.

### Format

```
SMU SET SHELF shelf_name[,...]
```

### Parameters

#### ***shelf\_name***

One or more shelf names. A shelf name can be a user-defined name, or the default shelf name HSM\$DEFAULT\_SHELF.

### Qualifiers

#### **/ARCHIVE\_ID=(archive\_id[,...])**

#### **/NOARCHIVE\_ID**

Identifies archive class identifiers to be used by this shelf. Archive classes must be defined using SMU SET ARCHIVE before they can be identified for use with the shelf. HSM transfers the file's data to each archive class in the list. This can be used to create multiple copies of the shelved file's data content.

This archive list specifies the archive classes to be written during shelving operations. The restore archive list specifies the archive classes to be read during unshelving operations. When a shelf is first created, the restore archive list is initialized to the archive list specified by this qualifier. Thereafter, the two lists are maintained separately.

Up to ten archive classes may be specified in this list.

/NOARCHIVE\_ID removes all existing archive class references.

#### **/CANCEL**

Cancels a split-merge operation for the specified catalog.

#### **/CATALOG=file\_spec**

Specifies the file specification of the catalog file to be used for this shelf. A full specification can be supplied, otherwise default elements are taken from the logical name HSM\$CATALOG. A logical name may also be supplied.

**/CONFIRM****/NOCONFIRM (Default)**

Controls whether a request is issued before each SET SHELF operation to confirm that the operation should be performed on that shelf. The following responses are valid:

YES	NO	QUIT
TRUE	FALSE	Ctrl/Z
1	0	ALL

Refer the OpenVMS DCL Concepts Manual for more information on the /CONFIRM qualifier.

**/DEFAULT**

Sets or modifies the system shelf defaults. When a new shelf definition is created, all default attributes are set from this default shelf definition.

The /DEFAULT qualifier cannot be used with a parameter value.

**/DELETE**

Deletes the entry from the shelf database.

**/DISABLE=keyword**

Disables shelving operations on all volumes associated with the shelf. The shelf can be configured to perform individual or all shelving operations. The possible keywords are as follows:

ALL	Disables the shelf for shelve, preshelve, and unshelve operations.
SHELVE	Disables the shelf for shelve and preshelve operations.
UNSHELVE	Disables the shelf for unshelve operations.

**/ENABLE[=keyword]**

Enables operations on volumes associated with the shelf. The shelf can be configured to perform only individual or all shelving operations. By default, the shelf is enabled for all shelve, preshelve, and unshelve operations. The possible keywords are as follows:

ALL (Default)	Enables the shelf for shelve, preshelve, and unshelve operations (Default).
SHELVE	Enables the shelf for shelve and preshelve operations.
UNSHELVE	Enables the shelf for unshelve operations.

**/LOG****/NOLOG (Default)**

Displays the shelf specification of each shelf modified as the command executes.

**/RESTORE\_ARCHIVE=(archive\_id[,...])****/NORESTORE\_ARCHIVE**

Identifies which archive classes are used by HSM on a restore operation. Archive classes must be defined using SMU SET ARCHIVE before they can be identified for use with the shelf. The order of this list is significant. On an UNSHELVE operation the first archive class in restore list is used to attempt to unshelve the file. If the restore attempt is unsuccessful, the next archive class in the restore list is attempted, and so on. Up to 36 archive classes may be specified in this list. This list should normally include the current archive list, plus any archive classes that have ever been defined for this shelf.

If you don't specify a restore archive ID when the shelf is created, the save archive class specified by the /ARCHIVE\_ID command is used as the restore archive. Thereafter, the two lists must be maintained separately.

**/NORESTORE\_ARCHIVE**

removes all restore archive class references.

**/SAVE\_TIME[=*delta\_time*]**

**NOSAVE\_TIME**

Specifies that the shelf file data and catalog entries are saved in the HSM subsystem for the specified *delta\_time* after the online file is deleted. If no *delta\_time* is specified, a default value of 30 days is applied. With /NOSAVE\_TIME, no delay is applied.

While repacking obsolete files are determined based on the value of SAVE\_TIME.

When a delete is attempted on a cache file if cache is set as /NOHOLD, the catalog entry is always deleted irrespective of SAVE\_TIME. Similarly, if the cache is set as /HOLD, the catalog entry is not deleted irrespective of SAVE\_TIME.

During a preshelve operation, if /HOLD is set on the Cache and /SAVE\_TIME is used

**If /HOLD is set:**

and /SAVETIME is used: The entry is not deleted  
and /NOSAVE is used: The entry is not deleted.

**If /NOHOLD is set:**

and /SAVETIME is used: The entry is deleted  
and /NOSAVE is used: The entry is deleted.

**/VERIFY**

**/NOVERIFY (Default)**

Specifies whether shelving and unshelving operations to tape volumes are performed using BACKUP/VERIFY. The /VERIFY option slows down shelving operations, but provides increased reliability for shelving.

**/UPDATES=*n***

Specifies the number of updates of a shelved file that should be saved on shelf media and in the catalog. For example, for /UPDATES=2, a file that has been: shelved /unshelved/modified/shelved 10 times will retain the latest two updates on shelf media and in the catalog.

Use /UPDATES=0 to specify that all updates should be retained. Note that this has nothing to do with the OpenVMS version of the file, but rather files that are updated in place.

## Example

The following example creates the shelf named CRITICAL\_SHELF for use with archive classes 1 and 2, and associates a separate catalog with the shelf.

```
$ SMU SET SHELF CRITICAL_SHELF -  
_$_ /ARCHIVE_ID=(1,2) -  
_$_ /CATALOG=DISK$USER1:[HSM.CATALOG]HSM$CAT1.SYS  
$ SMU SHOW SHELF CRITICAL_SHELF
```

```
Shelf CRITICAL_SHELF is enabled for Shelving and Unshelving  
Catalog File:   DISK$USER1:[HSM.CATALOG]HSM$CAT1.SYS  
Shelf history:  
  Created:    08-Apr-2002 12:32:24.70  
  Revised:   08-Apr-2002 12:32:24.70
```

```
Backup Verification: Off  
Save Time      <none>  
Updates Saved: All  
Archive Classes:  
  Archive list: HSM$ARCHIVE01 id: 1  
                HSM$ARCHIVE02 id: 2  
  Restore list: HSM$ARCHIVE01 id: 1  
                HSM$ARCHIVE02 id: 2
```

---

## SMU SET VOLUME

### SET

### VOLUME

The SMU SET VOLUME command defines shelving attributes for a specific online disk volume. This command controls HSM operations on the volume, associates the volume with a shelf, defines operations on certain types of files, and defines the reactive policies for the volume.

If an online disk volume does not have a specific entry in the SMU database, the attributes defined in the default volume HSM\$DEFAULT\_VOLUME are applied to it.

Specific HSM operations can be enabled or disabled on a per-volume basis using this command. For an operation to be performed on the volume, it must be enabled on the volume, the associated shelf, and the facility.

Each volume must be associated with exactly one shelf. By default, volumes are associated with the default shelf HSM\$DEFAULT\_SHELF.

The SET VOLUME command also defines the reactive shelving policy for the online volume. This policy is executed by HSM when the device's high water mark has been reached, the volume maximum occupancy has been reached (device full), or a process disk quota exceeded error. Files are then selected by using the criteria defined by this policy.

When a new volume definition is created, all attributes not entered in the command line are set from the default volume definition HSM\$DEFAULT\_VOLUME. The default volume definition itself may be modified using the SMU SET VOLUME /DEFAULT command.

If the volume definition already exists, it is modified with the new attributes entered on the command line.

### Format

```
SMU SET VOLUME volume_name[,...]
```

### Parameters

***volume\_name[,...]***

Sets the attributes for the specified online volume or volumes. Either the logical name for the disk device or the physical device name can be used. If two or more volumes are specified, separate them with commas.

Wildcard characters are allowed for updates to volumes that exist in the database, unless used with the /ALL qualifier. Wildcard applies only to the physical device

## Qualifiers

### **/CANCEL**

Cancels a split-merge operation for the specified shelf.

### **/ALL**

Used in conjunction with wildcard volume names to match with all online disks on the system. If /ALL is not specified, the wildcard only matches with the volumes already defined in the SMU database.

### **/CONFIRM**

#### **/NOCONFIRM (Default)**

Controls whether a request is issued before each SET VOLUME operation to confirm that the operation should be performed on that volume. The following responses are valid:

YES	NO	QUIT
TRUE	FALSE	Ctrl/Z
1	0	ALL

Refer the OpenVMS DCL Concepts Manual for more information on the /CONFIRM qualifier.

### **/CONTIGUOUS**

#### **/NOCONTIGUOUS (Default)**

Controls the eligibility of files marked contiguous on the volume to be shelved. The default is /NOCONTIGUOUS, meaning that contiguous files are never shelved. If /CONTIGUOUS is specified and contiguous files are shelved, then they are always restored contiguously. If a file cannot be restored contiguously, an error occurs. This qualifier does not affect preshelving.

### **/DEFAULT**

Sets or modifies the system volume defaults. When a new volume definition is created, all default attributes are set from this default volume definition.

This qualifier cannot be used with a volume\_name parameter value.

### **/DELETE**

Deletes the entry from the volume database. HSM attributes for the specified online disk volume reverts to those defined in the default volume record HSM\$DEFAULT\_VOLUME. The volume must be first set to use the shelf specified by the default volume before a delete is allowed.

### **/DISABLE [=keyword]**

The volume is disabled from specified HSM operations until it is reactivated by the /ENABLE qualifier. The volume can be configured to disable individual or all shelving operations.

By default, the volume is enabled for preshelving, shelving and unshelving operations, and disabled for high water mark detection, occupancy full and exceeded quota events.

The possible keywords are as follows:

ALL	Disables the volume for all HSM operations.
HIGHWATER_MARK	Disables the volume for handling high water mark conditions.
OCCUPANCY	Disables the volume for handling volume occupancy full conditions.
QUOTA	Disables the volume for handling disk quota exceeded conditions.
UNSHELVE	Disables the volume for UNSHELVE operations.

**/ENABLE[=*keyword*]**

Enables HSM operations on the volume. The volume can be configured to perform individual or all shelving operations. By default, the volume is enabled for preshelving, shelving and unshelving operations, and disabled for high water mark detection, occupancy full and exceeded quota events.

The possible keywords are as follows:

HIGHWATER_MARK	Enables the volume for handling high water mark conditions.
OCCUPANCY	Enables the volume for handling volume occupancy full conditions.
QUOTA	Enables the volume for handling disk quota exceeded conditions.
SHELVE	Enable the volume for SHELVE and PRESHELVE operations.
UNSHELVE	Enables the volume for UNSHELVE operations.
ALL	Enables the volume for all HSM operations.

**/HIGHWATER\_MARK=*percent***

The /HIGHWATER\_MARK qualifier establishes a threshold where reactive occupancy policy execution is triggered. It is set to a usage percentage of the volume capacity. Possible percentages range from 0 to 100 percent. HSM periodically checks the device capacity and compares it to the high water mark. If it has been exceeded, and high water mark operations are enabled, a volume occupancy full event is triggered, and the volume's occupancy policy is executed. If no high water mark processing is desired, specify the high water mark as 100 percent, or disable high water mark operations.

**/LOG****/NOLOG (Default)**

Displays the volume specification of each volume modified as the command executes.

**/OCCUPANCY[=*policy\_name*]****/NOOCCUPANCY**

This policy is executed whenever the online device has insufficient allocatable space or the volume high water mark has been exceeded. HSM then executes this occupancy policy. Policy execution is terminated when enough space has been freed to satisfy the request, or no candidates were found.

- If no occupancy full policy name is specified, HSM applies a predefined default occupancy policy definition (HSM\$DEFAULT\_OCCUPANCY).
- If /NOOCCUPANCY is specified, then no policy is run on volume occupancy full and high water mark conditions.

**/PLACEMENT (Default)****/NOPLACEMENT**

Defines the eligibility of files marked with placement attributes to be shelved.

The default is /PLACEMENT, meaning that placed files can be shelved. Files that have been shelved are not guaranteed to be restored with identical placement as before. This qualifier does not affect preshelving.

**/QUOTA[=*policy\_name*]****/NOQUOTA**

This policy is executed whenever a disk quota exceeded event is encountered. When a disk quota exceeded event is encountered, HSM then executes the volume's quota policy.

If no quota policy name is specified, HSM applies a predefined default quota policy definition (HSM\$DEFAULT\_QUOTA).

If /NOQUOTA is specified, then no policy is run on disk quota exceeded conditions.

**/SHELF\_NAME=*shelf\_name***

The shelf to associate the volume with. If no shelf qualifier is entered, the default shelf HSM\$DEFAULT\_SHELF is used. The shelf must exist prior to associating a volume with it.

**Example**

The following example shows the SMU SET VOLUME command. In this example, volumes \$1\$DKA100: and \$1\$DKA200: are assigned to the default shelf. All operations are enabled for the volume, and user-defined reactive policies are set up. The high water mark is set to 95 percent.

```
$ SMU SET VOLUME $1$DKA100:,$1$DKA200: -
$ /ENABLE=ALL /HIGHWATER_MARK=95 -
$ /OCCUPANCY_POLICY = CRITICAL_OCC_POL -
$ /QUOTA_POLICY = CRITICAL_QUOT_POL

$ SMU SHOW VOLUME $1$DKA100:,$1$DKA200 /FULL
```

```
Volume _$1$DKA100: on Shelf HSM$DEFAULT_SHELF, Shelving is enabled,
Unshelving is enabled, Highwater mark detection is enabled, Occupancy full
detection is enabled, Disk quota exceeded detection is enabled
Created: 08-Apr-2002 12:37:07.50
Revised: 08-Apr-2002 12:37:07.50
Ineligible files: <CONTIGUOUS>
High Water Mark: 95%
OCCUPANCY Policy: CRITICAL_OCC_POL
QUOTA Policy: CRITICAL_QUOT_POL
```

```
Volume _$1$DKA200: on Shelf HSM$DEFAULT_SHELF, Shelving is enabled,
Unshelving is enabled, Highwater mark detection is enabled, Occupancy full
detection is enabled, Disk quota exceeded detection is enabled
Created: 08-Apr-2002 12:37:07.64
Revised: 08-Apr-2002 12:37:07.64
Ineligible files: <CONTIGUOUS>
High Water Mark: 95%
OCCUPANCY Policy: CRITICAL_OCC_POL
QUOTA Policy: CRITICAL_QUOT_POL
```



---

## SMU SHOW ARCHIVE

### SHOW

### ARCHIVE

The SMU SHOW ARCHIVE command displays information associated with archive classes. The displayed information includes:

- The archive class name
- An indication whether the archive class has been used - it is flagged as used after the first shelving operation on the archive class
- The archive class identifier from 1 to 36 for HSM Basic mode or 1 to 9999 for HSM Plus mode - used in archive commands
- For HSM Basic mode, the media type used to write the archive class, and an indication whether a magazine loader is used
- For HSM Plus mode, the media type used to write the archive class, as defined in TAPE-START.COM
- For HSM Plus mode, the density for the media type, as defined in TAPESTART.COM
- The tape volume label of the current shelving volume
- The tape position of the next file to be shelved on the current shelving volume
- For HSM Plus mode, the pools associated with the archive class and the current pool being used
- The number of devices associated with this archive class (device refs)
- The number of shelves associated with this archive class (shelf refs)

### Format

SMU SHOW ARCHIVE *archive\_id[,...]*

### Parameters

***archive\_id[,...]***

Specifies the archive classes to be displayed. If two or more archive identifiers are specified, separate them with commas. If no archive identifiers are specified, HSM lists all defined archive classes.

## Qualifiers

**/OUTPUT[=*filespec*]**  
**/NOOUTPUT**

Controls where to write output of the command. By default, the display is written to the current SYSS\$OUTPUT device. No Wildcard characters are allowed.

## Examples

The following example shows the SMU SHOW ARCHIVE command for Basic mode on a system where only two archive classes are defined:

```
$ SMU SHOW ARCHIVE
HSM$ARCHIVE01 has been used
Identifier:      1
Media type:     CompactTape III, Loader
Label:         HS0004
Position:       1350
Device refs:    1
Shelf refs:     1

HSM$ARCHIVE02 has not been used
Identifier:      2
Media type:     CompactTape III,
Loader Label:   HS1001
Position:       0
Shelf refs:     0
SAVESET_SIZE:  40 MB
```

The following example shows the SMU SHOW ARCHIVE command for a specific archive class for Plus mode:

```
$ SMU SHOW ARCHIVE 1

HSM$ARCHIVE01 has been used
Identifier:      1
Media type:     TK85K
Density:        COMP
Label:          ACW401
Position:       21
Device refs:    2
Shelf refs:     3
Current pool:   HSM-POOL
Enabled pools:  HSM-POOL
```

---

## SMU SHOW CACHE

### SHOW

### CACHE

The SMU SHOW CACHE command displays the configuration associated with an online cache disk volume or magneto- optical device. The displayed information includes:

- The cache volume name and an indication whether it is currently enabled
- The time set in the SMU SET CACHE command for cache flush to occur; this is not necessarily the next cache flush time
- An indication whether copies to the shelf archive classes are made at shelving time or at cache flush time
- The maximum block size available for caching (a 0 means the entire volume may be used)
- The high water mark percentage of the block size, which if exceeded causes a cache flush
- The cache flush interval

### Format

SMU SHOW CACHE volume\_name[,...]

### Parameters

**volume\_name[,...]**

Specifies the online system mounted volumes that are specified as shelf cache devices. If no volumes are entered, all cache volumes are displayed.

If two or more volumes are specified, separate them with commas. Wildcard characters are allowed.

### Qualifiers

**/OUTPUT[=file\_spec]**

**/NOOUTPUT (Default)**

Controls where to write the output of the command. By default, the display is written to the current SYS\$OUTPUT device. No Wildcard characters are allowed.

If you enter the /OUTPUT qualifier with a partial file specification (for example, /OUTPUT=[JONES]), HSM\$CACHE is the default file name and .LIS is the default file extension. The output is suppressed if you enter the /NOOUTPUT qualifier.

### Example

The following example shows the SMU SHOW CACHE command display.

```
$ SMU SHOW CACHE _$15$DKA100:
```

```
Cache device _$15$DKA100: is enabled, Cache flush is held until
after 08-Apr-2002 04:00:00.00, Backup is performed at flush
intervals, Cached files are held on delete of online file
Block size: 100000
Highwater mark: 80%
Flush interval: 1 00:00:00.00
```

---

## SMU SHOW DEVICE

### SHOW

### DEVICE

The SMU SHOW DEVICE command displays the information associated with the specified nearline or offline device. The displayed information includes:

- The device name
- An indication of whether the device is currently enabled for HSM use
- Shared or dedicated access indication
- The operations that are supported by the device
- An indication of whether the device is correctly configured and usable by HSM
- If the information displayed is Not configured, this means one of the following things:
  - The media type defined for the SMU SET DEVICE command does not match any of the media types defined in TAPESTART.COM (Plus mode).
  - The device does not match the DRIVES\_n specification in TAPESTART.COM for the identified media type (Plus mode).
  - MDMS is not running, so HSM cannot verify the information provided (Plus mode).
  - The robot name is not specified, or the specified robot name does not exist or is not accessible (Basic mode).
  - The default device is being displayed.
- The media type the device supports in Basic mode
- The robot name controlling a loader device in Basic mode, or <none> for a non-loader device
- The archive classes supported by the device

### Format

```
SMU SHOW DEVICE device_name[,...]
```

### Parameters

***device\_name*[,...]**

Specifies the nearline/offline devices to be displayed. If two or more devices are specified, separate them with commas. Wildcard characters are allowed. If no device name is specified, information about all devices is displayed.

### Qualifiers

***/OUTPUT[=*filespec*]***

***/NOOUTPUT***

Controls where to write output of the command. By default, the display is written to the current SYSS\$OUTPUT device. No Wildcard characters are allowed.

## Examples

The following example shows the SMU SHOW DEVICE display for Basic mode:

```
$ SMU SHOW DEVICE _$10$MIA5:

HSM drive _$10$MIA5: is enabled.
Shared access: < shelve, unshelve >
Drive status:   Configured
Media type:     CompactTape III, Loader
Robot name:     _$1$DUA406:
Enabled archives: HSM$ARCHIVE01 id: 1
                  HSM$ARCHIVE02 id: 2
```

The following example shows the SMU SHOW DEVICE display for Plus mode:

```
$ SMU SHOW DEVICE _NODE2$MKA500:
HSM drive _NODE2$MKA500: is enabled.
Shared access: <shelve, unshelve>
Drive status:   Configured
Enabled archives: HSM$ARCHIVE01 id: 1
                  HSM$ARCHIVE03 id: 3
```

---

## SMU SHOW FACILITY

### SHOW

### FACILITY

The SMU SHOW FACILITY command displays information about the HSM facility, whose attributes have VMScluster-wide impact on all shelves, volumes and nodes. The displayed information includes:

- Operations that are enabled across the HSM facility
- The list of designated shelf servers
- The current shelf server node
- An indication of whether cache operations and all catalog updates are performed by the shelf server node
- The state of individual types of event logging
- The HSM operations mode (Plus or Basic)
- The amount of shelving storage capacity remaining; if you have acquired the unlimited license, this shows as “unlimited”.

### Format

SMU SHOW FACILITY

### Qualifiers

**/OUTPUT[=filespec]**  
**/NOOUTPUT**

Controls where to write output of the command. By default, the display is written to the current SYS\$OUTPUT device. No Wildcard characters are allowed.

If you enter the /OUTPUT qualifier with a partial file specification (for example, /OUTPUT=[JONES]), HSM\$FACILITY is the default file name and .LIS is the default file extension. The output is suppressed if you enter the /NOOUTPUT qualifier.

### Example

The following example shows the SMU SHOW FACILITY command display.

```
$ SMU SHOW FACILITY

HSM for OpenVMS is enabled for Shelving and Unshelving
Facilityhistory:
  Created:                08-Apr-2002 10:36:34.96
  Revised:                08-Apr-2002 12:27:51.87
  Designated servers:    SYS001
                        SYS002
  Current server:        SYS001
  Catalogserver:         Disabled
  Event logging:         Audit
                        Error
                        Exception
  HSMmode:               Plus
  Remaining license:     500 gigabytes
```

---

## SMU SHOW POLICY

### SHOW

### POLICY

The SMU SHOW POLICY command displays a shelving policy definition. The displayed information includes:

- The name of the policy
- An indication of whether the policy is currently enabled
- The action to be taken by the policy: either shelving or preshelving
- The creation and modification dates of the record (/FULL only)
- The date used for selecting files for shelving: either the Backup, Creation, Modification, or Expiration dates (/FULL only)
- The period of relative elapsed time at policy execution that files will not be selected for shelving, applied to the specified date field (/FULL only)
- A before or since absolute time that files will not be selected for shelving, applied to the specified date field (/FULL only)
- The low water mark capacity goal that the policy attempts to reach before completing execution (/FULL only)
- The policy algorithms used for primary and secondary policy, or the locations of script files (/FULL only)
- Account to which mail is delivered on policy completion (/FULL only)
- File to which policy information is logged (/FULL only)

### Format

```
SMU SHOW POLICY [policy_name[,...]]
```

### Parameters

***[policy\_name[,...]]***

Specifies one or more policy names. If two or more policy names are specified, separate them with commas. Wildcard characters are allowed.

If no policy\_name is specified, all policy definitions are displayed.

## Qualifiers

### **/BRIEF (Default)**

Displays one line of information for each specified policy.

### **/DEFAULT**

Shows the default policy record HSM\$DEFAULT\_POLICY. When a new policy definition is created, all default attributes are set from this default policy definition.

The /DEFAULT qualifier cannot be used with a parameter value.

### **/FULL**

Displays complete information about each specified policy.

### **/OUTPUT[=*filespec*]**

### **/NOOUTPUT**

Controls where to write output of the command. By default, the display is written to the current SYS\$OUTPUT device. No Wildcard characters are allowed.

If you enter the /OUTPUT qualifier with a partial file specification (for example, /OUTPUT=[JONES]), HSM\$POLICY is the default file name and .LIS is the default file extension. The output is suppressed if you enter the /NOOUTPUT qualifier.

## Examples

The following examples show the SMU SHOW POLICY display.

```
$ SMU SHOW POLICY

Policy HSM$DEFAULT_OCCUPANCY is enabled for shelving
Policy HSM$DEFAULT_POLICY is enabled for shelving
Policy HSM$DEFAULT_QUOTA is enabled for shelving

$ SMU SHOW POLICY/FULL HSM$DEFAULT_QUOTA

Policy HSM$DEFAULT_QUOTA is enabled for shelving
PolicyHistory:
  Created:      08-Apr-2002 10:36:36.47
  Revised:     08-Apr-2002 10:36:36.47
Selection Criteria:
  State:       Enabled
  Action:      Shelving
  FileEvent:   Expirationdate
  Elapsedtime: 18000:00:00
  Before time: <none>
  Since time:  <none>
  Low Water Mark: 80%
  Primary Policy: Space Time Working Set (STWS)
  SecondaryPolicy: Least Recently Used (LRU)
Verification:
  Mail notification: <none>
  Output file:      <none>
```



---

## SMU SHOW REQUESTS

### SHOW

### REQUESTS

The SMU SHOW REQUESTS command displays the number of HSM operations in progress, and optionally dumps information about the requests to an activity log. The log is located at HSM\$LOG:HSM\$SHP\_ACTIVITY.LOG.

The SMU SHOW REQUESTS command will only display the requests on the current node you are using. To show the requests on all nodes in the cluster, use the OpenVMS SYSMAN command as shown in the example below.

### Format

SMU SHOW REQUESTS

### Qualifiers

**/BRIEF (Default)**

Writes a line to the display device to indicate the number of requests in progress.

**/FULL**

Writes a line to the display device to indicate the number of requests in progress. In addition, if there is at least one active request, the requests are dumped to the file HSM\$LOG:HSM\$SHP\_ACTIVITY.LOG.

**/OUTPUT[=filespec]****/NOOUTPUT**

Controls where to write output of the command. By default, the display is written to the current SYS\$OUTPUT device. No Wildcard characters are allowed.

---

**Note**

---

**This qualifier affects only the one line display of the number of requests in progress. The activity log is always placed in a new version of HSM\$LOG:HSM\$SHP\_ACTIVITY.LOG.**

---

## SMU SHOW REQUESTS

### Example

The following example shows the SMU SHOW REQUESTS command:

```
$ SMU SHOW REQUESTS/FULL
%HSM-S-DMPACTREQS, shelving facility active with 3 requests
%HSM-I-DMPFILE, active requests dumped to file
HSM$LOG:HSM$SHP_ACTIVITY.LOG
```

The following example shows how to use the OpenVMS SYSMAN command to show the requests on all nodes in the cluster.

```
$ MCR SYSMAN  SYSMAN> SET ENVIRONMENT/CLUSTER
%SYSMAN-I-ENV, current command environment:
Clusterwide on local cluster
Username PETER will be used on nonlocal nodes

SYSMAN> DO SMU SHOW REQUESTS
%SYSMAN-I-OUTPUT, command execution on node HERSHY
%HSM-S-DMPACTREQS, shelving facility active with 1 requests
%SYSMAN-I-OUTPUT, command execution on node MINT
%HSM-S-DMPACTREQS, shelving facility active with 1 requests
%SYSMAN-I-OUTPUT, command execution on node CHOCO
%HSM-S-DMPACTREQS, shelving facility active with 4 requests
```

---

## SMU SHOW SCHEDULE

### SHOW

### SCHEDULE

The SMU SHOW SCHEDULE command displays information about existing preventative policy schedules. The displayed information includes:

- The volume name for which the policy is scheduled - each volume is listed separately
- The policy name
- The entry number, which can be used in subsequent modifications to the schedule entry
- The status of the schedule - the status values are the same as those for OpenVMS batch queue entries
- The server node on which the policy is scheduled (/FULL only)
- The scheduled time of the policy execution (/FULL only)
- The scheduling interval (/FULL only)
- The output file containing the policy execution information (/FULL only)

The SMU SHOW SCHEDULE command may be entered with no parameters to show all HSM schedules or to show a particular entry using /ENTRY. For all other cases, at least one volume name must be entered.

### Format

```
SMU SHOW SCHEDULE [volume_name[,...] [policy_name]]
```

### Parameters

***volume\_name[,...]***

The name of the online volume(s) on which the scheduling information is to be displayed. If no volume names and policy names are entered, the schedule for all volumes is displayed, unless the /ENTRY qualifier is specified.

If two or more volumes are specified, separate them with commas. Wildcard characters are not allowed.

***[policy\_name]***

The name of an existing policy definition. If no policy name is entered, the schedules for all policies are displayed.

## SMU SHOW SCHEDULE

### Qualifiers

#### **/BRIEF (Default)**

Displays a one-line summary for each scheduled entry.

#### **/ENTRY=*entry\_number***

Displays a specific scheduled entry only, and should be entered with no parameters on the command line.

#### **/FULL**

Displays complete information for each scheduled entry.

#### **/OUTPUT[=*filespec*]**

#### **/NOOUTPUT**

Controls where to write output of the command. By default, the display is written to the current SYS\$OUTPUT device. No Wildcard characters are allowed.

#### **/SERVER=*server\_name***

Displays only those entries scheduled on a specific server node.

### Examples

The following examples show the SMU SHOW SCHEDULE command:

```
$ SMU SHOW SCHEDULE _$1$DKA100: , _$1$DKA200:
```

Volume	Policy	Entry	status
-----	-----	-----	-----
_\$1\$DKA100:	HSM\$DEFAULT_OCCUPANCY	517	PENDING

Volume	Policy	Entry	status
-----	-----	-----	-----
_\$1\$DKA200:	HSM\$DEFAULT_OCCUPANCY	518	PENDING

```
$ SMU SHOW SCHEDULE/ENTRY=518/FULL
```

Volume	Policy	Entry	status
-----	-----	-----	-----
_\$1\$DKA200:	HSM\$DEFAULT_OCCUPANCY	518	PENDING
Servernode:	SLIPPER		
Scheduled after:	08-Apr-2002 02:00:00.00		
Scheduled interval:	1 00:00:00.00		
Output file	<none>		

---

## SMU SHOW SHELF

### SHOW

### SHELF

The SMU SHOW SHELF command displays information about a shelf object. The displayed information includes:

- The shelf name and enabled operations
- The name of the catalog supporting this shelf
- The creation and modification date of the shelf record
- An indication whether verification operations are to be performed on BACKUP operations
- The amount of time to wait before a REPACK operation removes shelf data after a file is deleted
- The number of in-place updates that can be performed on a file before a REPACK operation removes shelf data for the file
- The archive classes defined for the shelf - both the archive classes used for shelving, and those used for unshelving

### Format

SMU SHOW SHELF shelf\_name[,...]

### Parameters

***shelf\_name[,...]***

The name of the shelf. Wildcard characters are allowed.

If no shelf name is specified, all shelves are displayed.

## Qualifiers

**/OUTPUT[=filespec]**

**/NOOUTPUT**

Controls where to write output of the command. By default, the display is written to the current SYSS\$OUTPUT device. No Wildcard characters are allowed.

If you enter the /OUTPUT qualifier with a partial file specification (for example, /OUTPUT=[JONES]), HSM\$\$SHELF is the default file name and .LIS is the default file extension. The output is suppressed if you enter the /NOOUTPUT qualifier.

## Example

The following example shows the SMU SHOW SHELF command display.

```
$ SMU SHOW SHELF HSM$DEFAULT_SHELF

Shelf HSM$DEFAULT_SHELF is enabled for Shelving and Unshelving
Catalog File:$1$DKA0:[HSM$SERVER.CATALOG]HSM$CATALOG.SYS
Shelfhistory:
  Created:          08-Apr-2002 12:39:30.86
  Revised:         08-Apr-2002 13:17:58.87
Backup Verification: Off
SaveTime          <none>
Updates Saved:    All
Archive Classes:
  Archivelist:     HSM$ARCHIVE01 id: 1
                  HSM$ARCHIVE02 id: 2
  Restorelist:    HSM$ARCHIVE01 id: 1
                  HSM$ARCHIVE02 id: 2
```

---

## SMU SHOW VERSION

### SHOW

### VERSION

The SMU SHOW VERSION command displays the version information about the various HSM components. The displayed information includes:

- The HSM component name
- The version and interval base level of the component
- The date on which the component was built

In a properly configured system, all version attributes should be the same for all components.

### Format

```
SMU SHOW VERSION
```

### Qualifiers

**/OUTPUT[=filespec]**

**/NOOUTPUT**

Controls where to write output of the command. By default, the display is written to the current SYSS\$OUTPUT device. No Wildcard characters are allowed.

If you enter the /OUTPUT qualifier with a partial file specification (for example, /OUTPUT=[JONES]), HSM\$VERSION is the default file name and .LIS is the default file extension. The output is suppressed if you enter the /NOOUTPUT qualifier.

### Example

The following example shows the SMU SHOW VERSION command display.

```
$ SMU SHOW VERSION

HSM Shelf Handler version - V4.0A(BL42), April 2002
HSM Shelving Driver version - V4.0A(BL42), April 2002
HSM Policy Execution version - V4.0A(BL42), April 2002
HSM Shelf Management version - V4.0A(BL42), April 2002
```

---

## SMU SHOW VOLUME

### SHOW

### VOLUME

The SMU SHOW VOLUME command displays the shelving attributes of an online disk volume. The displayed information includes:

- The volume name
- The shelf with which the volume is associated
- Operations that are enabled on the volume
- Creation and revision date of the volume record (/FULL only)
- Specific files types ineligible for shelving (contiguous and placement)
- The high water mark capacity for the volume (/FULL only)
- The name of the reactive policy to be executed when a volume (/FULL only) occupancy full or high water mark exceeded event occurs (/FULL only)
- The name of the reactive policy to be executed when an exceeded disk quota event occurs (/FULL only)

### Format

SMU SHOW VOLUME volume\_name[...]

### Parameters

***volume\_name[...]***

Shows the event policy for the online volume or volumes. Specify the logical volume name. If two or more volumes are specified, separate them with commas. Wildcard characters are allowed.

If no volume\_name is specified, all volumes are displayed.



## Qualifiers

### **/BRIEF (Default)**

Displays brief information about the specified volume.

### **/DEFAULT**

Shows the system volume defaults. When a new volume definition is created, all default attributes are set from this default volume definition.

The /DEFAULT qualifier cannot be used with a parameter value.

### **/FULL**

Displays a complete list of information about the volume.

### **/OUTPUT[=*filespec*]**

### **/NOOUTPUT**

Controls where to write output of the command. By default, the display is written to the current SYSS\$OUTPUT device. No Wildcard characters are allowed.

If you enter the /OUTPUT qualifier with a partial file specification (for example, /OUTPUT=[JONES]), HSM\$VOLUME is the default file name and .LIS is the default file extension. The output is suppressed if you enter the /NOOUTPUT qualifier.

## Examples

The following examples show the SMU SHOW VOLUME command display.

```
$ SMU SHOW VOLUME
```

```
Volume HSM$DEFAULT_VOLUME on Shelf HSM$DEFAULT_SHELF, Shelving is enabled,
Unshelving is enabled, Highwater mark detection is disabled,Occupancy full
detection is disabled, Disk quota exceeded detection is disabled
```

```
Volume _$1$DKA100: on Shelf HSM$DEFAULT_SHELF, Shelving is enabled,
Unshelving is enabled, Highwater mark detection is enabled, Occupancy full
detection is enabled, Disk quota exceeded detection is enabled
```

```
Volume _$1$DKA200: on Shelf HSM$DEFAULT_SHELF, Shelving is enabled,
Unshelving is enabled, Highwater mark detection is enabled, Occupancy full
detection is enabled, Disk quota exceeded detection is enabled
```

```
$ SMU SHOW VOLUME /DEFAULT /FULL
```

```
Volume HSM$DEFAULT_VOLUME on Shelf HSM$DEFAULT_SHELF, Shelving is enabled,
Unshelving is enabled, Highwater mark detection is disabled, Occupancy full
```

```
detection is disabled, Disk quota exceeded detection is disabled
```

```
Created: 08-Apr-2002 12:39:33.07
```

```
Revised: 08-Apr-2002 12:39:33.07
```

```
Ineligible files: <CONTIGUOUS>
```

```
High Water Mark: 90%
```

```
OCCUPANCY Policy: HSM$DEFAULT_OCCUPANCY
```

```
QUOTA Policy: HSM$DEFAULT_QUOTA
```

---

# SMU SHUTDOWN

## SHUTDOWN

The SMU SHUTDOWN command disables and shuts down HSM in an orderly fashion. SMU SHUTDOWN and SMU SHUTDOWN/NOW take effect on all nodes in the VMScluster, whereas SMU SHUTDOWN /FORCE affects only the issuing node.

This command is typically executed during SYSTEM shutdown (SMU SHUTDOWN/FORCE). It should be placed in the appropriate command files for the system. The other two variants are used to shut down the HSM subsystem across the VMScluster, for perhaps an HSM upgrade or some other reason.

SMU SHUTDOWN waits for all pending requests to complete, whereas SMU SHUTDOWN/NOW and SMU SHUTDOWN/FORCE shut down immediately; any pending requests are recovered when HSM is restarted.

In any variant of SMU SHUTDOWN, there may be a delay of up to five minutes if a BACKUP or tape positioning operation is in progress.

The SMU SHUTDOWN command requires SYSPRV, TMPMBX, and WORLD privileges.

## Format

SMU SHUTDOWN

## Qualifiers

**/FORCE**

**/FORCE**

Shut down HSM immediately on the requesting node only. HSM will continue to run on other nodes in the VMScluster. Pending requests are not processed but will be recovered when HSM starts up later.

**/NOW**

**/NOW**

Shut down HSM immediately across the VMScluster. Pending requests are not processed but will be recovered when HSM starts up later.

## Example

The following example shows the SMU SHUTDOWN command as should be applied in the system shutdown command procedure, to shut HSM down on one node:

```
§ SMU SHUTDOWN/FORCE
```

---

## SMU SPAWN

### SPAWN

The SMU SPAWN command creates a subprocess to execute DCL commands without leaving the SMU session.

SPAWN returns you to the DCL level in a subprocess. You can then enter DCL commands. To terminate the subprocess and resume the SMU session, log out of the subprocess.

Alternatively, you can enter a single DCL command line as a parameter to the spawn command.

### Format

SMU SPAWN

### Examples

The following examples show the SMU SPAWN command:

```
$ SMU
SMU> SPAWN SET PROCESS/NOAUTO_UNSHELVE
SMU>
```

alternatively, you could use the following commands:

```
S SMU
SMU> SPAWN
$ SET PROCESS/NOAUTO_UNSHELVE
$ LOGOUT
SMU>
```

---

## SMU STARTUP

### STARTUP

The SMU STARTUP command starts up HSM.

This command is typically executed during SYSTEM startup. It should be placed in the appropriate command files for the system.

The SMU STARTUP command requires the following privileges:

BYPASS	CMKRNL	DETACH
DIAGNOSE	EXQUOTA	LOG_IO
NETMBX	OPER	PHY_IO
PRMMBX	SHARE	SYSLCK
SYSNAM	SYSPRV	TMPMBX
VOLPRO	WORLD	

### Format

SMU STARTUP

### Examples

The following example shows the SMU STARTUP command:

```
$ SMU STARTUP
%SMU-S-SHP_STARTED, shelf handler process started 22E0193B
%SMU-S-PEP_STARTED, policy execution process started 22E011BC
```

# Part II

---

## MDMS Commands

This part contains information about the MDMS STORAGE Commands.



---

## MDMS DCL Commands

MDMS is the Media, Device and Management Services Component of the Archive Backup System (ABS) and Hierarchical Storage Management (HSM) products.

You can use MDMS commands to manage all the databases for ABS. Alternatively, the pre-V4 ABS DCL commands are also available.

MDMS commands are also used to manipulate the media management database for HSM, whereas SMU manages the HSM database.

---

## MDMS ALLOCATE DRIVE

The MDMS ALLOCATE DRIVE command selects and allocates a drive based on one of the following:

- A specified drive
- A specified volume
- A specified media type and optionally location, node, or group
- A specified jukebox

The selected drive name is assigned to the optional process logical name, which may be used in subsequent commands by that process.

Equivalent STORAGE Command: STORAGE SELECT

### Format:

MDMS ALLOCATE DRIVE [drive\_name]

### Parameters

drive\_name

The name of a specific drive to allocate, which must already have been created with a CREATE DRIVE command. This parameter is optional: drives can also be allocated on the basis of volume, media type, location or node, and jukebox, or certain combinations thereof.

The maximum length of the drive name is 31 characters.

Valid characters are "\$,-,0-9,A-Z,\_,a-z".

### Description

The MDMS ALLOCATE DRIVE command selects and allocates a drive based on one of the following:

- A specified drive
- A specified volume
- A specified media type and optionally location, node, or group
- A specified jukebox

The selected local OpenVMS device name and the drive name are assigned to the optional logical name in a search list. Both MDMS and non-MDMS DCL commands may be issued using the logical name. The logical name remains assigned until it is deassigned, the allocating process terminates, or the MDMS DEALLOCATE DRIVE command specifying the logical name is issued.

The MDMS ALLOCATE DRIVE command allocates the selected drive to the current process. After the MDMS ALLOCATE DRIVE command is issued, the name of the selected drive is displayed on the terminal screen. The DCL SHOW LOGICAL command also displays the OpenVMS local device name and the drive name.

MDMS supports allocation of both local drives and remote drives (using the RDF software). However, allocation of remote drives through RDF is not available if you are running with the ABS-OMT license.



MDMS attempts to allocate a drive local to the node performing the allocation, if one matches the selection criteria. Failing that, a TMSCP-accessible drive is chosen next. If that fails, an RDF-accessible remote drive is chosen.

## Privileges

The request requires MDMS\_ALLOCATE\_ALL or MDMS\_ALLOCATE\_OWN.

MDMS\_ALLOCATE\_OWN requires the specification of an owned volume for selection. All other selections including the drive name parameter require MDMS\_ALLOCATE\_ALL.

MDMS\_ASSIST is also required unless /NOASSIST is specified.

## Restrictions

The drive\_name parameter cannot be used with the /MEDIA\_TYPE, /JUKEBOX, /LOCATION, /NODE, /GROUP, or /VOLUME qualifiers.

The /JUKEBOX qualifier cannot be used with /GROUP, /LOCATION, /NODE, /VOLUME, or the drive\_name parameter.

The /MEDIA\_TYPE qualifier cannot be used with /VOLUME or the drive\_name parameter.

The VOLUME qualifier cannot be used with /GROUP, /LOCATION, /MEDIA\_TYPE, /NODE, /JUKEBOX or the drive\_name parameter.

The /GROUP, /JUKEBOX, /LOCATION and /NODE qualifiers are mutually exclusive. If one is specified, then /MEDIA\_TYPE must also be specified (except for /JUKEBOX).

The /[NO]PREFERRED qualifier is ignored if /VOLUME is not supplied.

## Qualifiers

### **/ASSIST (D)**

#### **/NOASSIST**

The default /ASSIST qualifier outputs an operator request to allocate a drive if there is a problem allocating a drive immediately. If the operator responds positively to the operator message, the request is retried, otherwise the request fails. The /NOASSIST qualifier performs the operation without operator assistance, and uses the /RETRY and /INTERVAL qualifiers to perform retries. If all retries are exhausted the command fails.

The right MDMS\_ASSIST is required unless /NOASSIST is specified.

#### **/DEFINE=logical\_name**

Specifies the logical name to be assigned for the drive. The logical name is a process logical name.

The maximum length of the logical name is 31 characters.

The logical name is assigned to an equivalence string containing the drive name and allocated VMS device name in a search list - as such, both MDMS and non-MDMS commands (e.g. MOUNT) can be issued on the logical name.

#### **/GROUP=group\_name**

When used with /MEDIA\_TYPE, you can specify the name of the group (of nodes supporting the drive) from which to select and allocate the drive. Only one group may be specified.

This qualifier requires the right MDMS\_ALLOCATE\_ALL.

#### **/INTERVAL=delta\_time**

Specifies the interval between retries when no drives are available. If not specified, the default interval is one minute.

**/JUKEBOX=jukebox\_name**

The name of the jukebox from which the drive will be selected.

The maximum length of the jukebox name is 31 characters.

Valid characters are "\$,-,0-9,A-Z,\_,a-z".

If /MEDIA\_TYPE is also specified, the drive must support the specified media type as well as being in the jukebox.

This qualifier requires the right MDMS\_ALLOCATE\_ALL.

**/LOCATION=location**

When used with /MEDIA\_TYPE, this qualifier specifies the location of the node from which to select and allocate the drive. Only one location may be specified.

This qualifier requires the right MDMS\_ALLOCATE\_ALL.

**/MEDIA\_TYPE=media\_type**

The distinct media type that the drive must support. This media type must match one defined for a drive in the media type attribute (for read/write allocation), or in the read only media type attribute (for read-only allocation).

This qualifier requires the right MDMS\_ALLOCATE\_ALL.

**/NODE=node\_name**

When used with /MEDIA\_TYPE, you can specify the name of the node from which to select and allocate the drive. Only one node may be specified.

This qualifier requires the right MDMS\_ALLOCATE\_ALL.

**/NOPREFERRED (D)**

The /PREFERRED qualifier tries to allocate the preferred drive for a volume, when /VOLUME is entered; this is the last drive that the volume was loaded in, if that drive is available. The default /NOPREFERRED forces a round-robin drive selection.

**/REPLY=symbol**

The name of the symbol to receive the operator's reply when operator intervention is needed.

The symbol will contain the operator reply to the DCL REPLY/TO or DCL REPLY/ABORT commands. The maximum length of a symbol name is 31 characters. This qualifier is only applicable when /ASSIST is specified.

**/RETRY\_LIMIT=number**

**/NORETRY\_LIMIT**

This qualifier specifies if MDMS should retry the operation if no drive is available and /ASSIST is not specified. /NORETRY\_LIMIT means that there is no limit on the number of retries.

Retries are performed at the delta time specified by /INTERVAL. If all retries are exhausted, the operation fails.

The default is /RETRY\_LIMIT=0 which means that no retries are performed.

**/VOLUME=volume\_id**

The volume ID of the volume for which an appropriate drive will be selected that can support the volume's media type and placement. If a volume is specified that currently resides in a jukebox, the drive selected will be from the same jukebox if the drives in the jukebox support the media type of the volume.

The maximum length of the volume ID is 6 characters.

Valid characters are " ,! ,% -?,A-Z,\_" .

**/WRITE (D)****/NOWRITE**

Specifies that the drive is to be allocated for read-only operations only. This potentially makes a larger pool of drives available for allocation for certain media types.

The default is WRITE.

**Examples**

```
$ MDMS ALLOCATE DRIVE $1$MUA5 /DEFINE=MYDRIVE
```

This command allocates the drive \$1\$MUA5 and assigns the drive name to the MYDRIVE logical name. Note that the drive name is the same as the VMS device name in this case.

```
$ MDMS ALLOCATE DRIVE JUKE_2_DRIVE_1 /ASSIST -
_$ /REPLY=ALLOC_REPLY
```

This command allocates drive JUKE\_2\_DRIVE\_1, and requests operator assistance if the allocation fails, and stores the operator's reply message in symbol ALLOC\_REPLY.

```
$ MDMS ALLOCATE DRIVE /VOLUME=LAB003 /NOWRITE -
_$ /NOPREFERRED
```

This command allocates a drive that supports volume LAB003 for read-only operations, and requests a round-robin drive selection rather than use the preferred drive.

```
$ MDMS ALLOCATE DRIVE /JUKEBOX=TESTJUKE -
_$ /DEFINE=MYDRIVE
```

This command allocates one of the drives in jukebox TESTJUKE and assigns the drive name to the MYDRIVE logical name.

```
$ MDMS ALLOCATE DRIVE /MEDIA_TYPE=TK85K -
_$ /NODE=CRUMBS /INTERVAL=00:00:10 /NORETRY_LIMIT
```

This command allocates a drive that supports the TK85K media type on node CRUMBS, and specifies a retry interval of 10 seconds if a drive is not immediately available with no limit on retries.

---

## MDMS ALLOCATE VOLUME

The MDMS ALLOCATE VOLUME command allocates available free volumes to the requesting user or a specified user.

Equivalent STORAGE Command: STORAGE ALLOCATE

### Format:

MDMS ALLOCATE VOLUME [volume\_id]

### Parameters

volume\_id

Specifies the volume to allocate. This parameter is optional, and volumes can be allocated on the basis of /BIND, /JUKEBOX, /LIKE\_VOLUME, /LOCATION, /MEDIA\_TYPE or /POOL instead.

The maximum length of the volume\_id is 6 characters. Valid characters are " ,! , % - ? , A - Z , \_ " .

### Description

The MDMS ALLOCATE VOLUME command allocates available free volumes to the requesting user or a specified user. The newly allocated volume(s) can also be appended to the end of a volume set. A volume set is defined to be one or more allocated volumes.

If the volume\_id parameter is used, that specific volume will be allocated if it is in the Free state. If the volume\_id parameter is not used, a Free volume will be selected based on one or more of the following selection criteria.

- Bind volume
- Jukebox
- Like volume
- Location
- Pool
- Media type

### Privileges

The request requires MDMS\_ALLOCATE\_ALL or MDMS\_ALLOCATE\_POOL.

MDMS\_ALLOCATE\_POOL requires the volume to be allocated from a named pool to which the calling user is authorized. All other allocations, including allocation from the scratch (unnamed) pool and the specification of /USER\_NAME require MDMS\_ALLOCATE\_ALL.

The /BIND qualifier requires MDMS\_BIND\_ALL if the /USER\_NAME qualifier is specified, or MDMS\_BIND\_OWN.

The /BLOCK\_FACTOR, /DESCRIPTION, /FORMAT, /RECLENGTH, [NO]SCRATCH\_DATE and /[NO]TRANSITION\_TIME qualifiers require MDMS\_SET\_ALL or MDMS\_SET\_VOLUME if the /USER\_NAME qualifier is specified, or MDMS\_SET\_OWN or MDMS\_SET\_POOL.

MDMS\_ASSIST is also required unless /NOASSIST is specified.

## Restrictions

The `volume_id` parameter cannot be used with the `/JUKEBOX`, `/LIKE_VOLUME`, `/LOCATION`, `/POOL` or `/QUANTITY` qualifiers.

`/BIND` cannot be used with the `/LIKE_VOLUME`, `/JUKEBOX`, `/LOCATION`, `/MEDIA_TYPE`, or `/POOL` qualifiers.

`/LIKE_VOLUME` cannot be used with the `/BIND`, `/JUKEBOX`, `/LOCATION`, `/MEDIA_TYPE` or `/POOL` qualifiers.

`/MEDIA_TYPE` is required when `/JUKEBOX`, `/LOCATION`, or `/POOL` are specified.

`/MEDIA_TYPE` is also required when allocating a specific volume that has multiple media types.

## Qualifiers

### **/ASSIST (D)**

### **/NOASSIST**

The default `/ASSIST` qualifier outputs an operator request to allocate a volume if there is a problem allocating a volume. If the operator responds positively to the operator message, the request is retried, otherwise the request fails. The `/NOASSIST` qualifier performs the operation without operator assistance, and fails if there is a problem allocating a volume.

The right `MDMS_ASSIST` is required unless `/NOASSIST` is specified.

### **/BIND=volume\_id**

The new allocated volume or volume set is appended to the volume set specified by the `volume_id`. The `volume_id` specifies the selection criteria for allocating the new volume(s).

The maximum length of the `volume_id` is 6 characters. Valid characters are " ,! ,% -? ,A-Z ,\_ ".

When the `/BIND` qualifier is used, the following attributes must match:

- Media type
- Pool
- Placement (in same jukebox, magazine, or location)

The `BIND` volume must be allocated

This qualifier requires the right `MDMS_BIND_OWN`, or `MDMS_BIND_ALL` if the `/USER_NAME` qualifier is used.

### **/BLOCK\_FACTOR=number**

Modifies the block factor attribute of the allocated volume record(s). If not specified, the block factor attribute is not changed.

This qualifier requires the right `MDMS_SET_OWN` or `MDMS_SET_POOL`, and `MDMS_SET_ALL` or `MDMS_SET_VOLUME`

if the `/USER_NAME` qualifier is used.

### **/DEFINE=logical\_name**

Specifies a logical name to be assigned for the volume allocated. The logical name is a process logical name. If the `/QUANTITY` qualifier is used, the returned volume will be the first volume in the volume set. The maximum length of the logical name is 31 characters.

### **/DESCRIPTION="text"**

Modifies comments about the object in the volume record. If the text contains spaces, then it must be enclosed within quotation marks. The length of the description can range from 0 to 255 characters.

## MDMS ALLOCATE VOLUME

To clear the existing description, specify "". If not specified, the volume description is not changed.

This qualifier requires the right MDMS\_SET\_OWN or MDMS\_SET\_POOL, and MDMS\_SET\_ALL or MDMS\_SET\_VOLUME if the /USER\_NAME qualifier is used.

### **/FORMAT=keyword**

This qualifier modifies the format field in the volume record. If not specified, the format attribute is not changed. Valid values are:

- ASCII
- BACKUP
- EBCDIC
- NONE
- RMUBACKUP

This qualifier requires the right MDMS\_SET\_OWN or MDMS\_SET\_POOL, and MDMS\_SET\_ALL or MDMS\_SET\_VOLUME if the /USER\_NAME qualifier is used.

### **/JUKEBOX=jukebox\_name**

When specified with /MEDIA\_TYPE, allocates a volume that resides in the specified jukebox.

### **/LIKE\_VOLUME=volume\_id**

This qualifier directs that the allocated volume has attribute values identical to those specified to the volume\_id. The following attributes define a "like volume".

- Media type
- Pool
- Placement (in the same jukebox, magazine, or location)

### **/LOCATION=location**

When specified with /MEDIA\_TYPE, specifies the location from which to allocate volume(s). If a volume's placement is OFFSITE then the offsite location is used. Otherwise the volume's ONSITE location is used. If not specified, other selection criteria are used to allocate the volume(s).

### **/MEDIA\_TYPE=media\_type**

Specifies that the volume(s) are allocated with the specified media type. If not specified, other selection criteria are used to allocate the volume(s). This qualifier is required when allocating a specific volume that currently has multiple media types defined for it.

### **/POOL=pool\_name**

The pool from which the volume(s) will be selected.

The maximum length of the pool name is 31 characters. Valid characters are "\$,-,0-9,A-Z,\_,a-z".

The user must be authorized for the pool unless he has MDMS\_ALLOCATE\_ALL.

### **/QUANTITY=number**

The number of volumes to allocate. Use this qualifier to allocate more than one volume. The default value is 1. If you enter a number greater than 1, the allocated volumes are bound together in one volume set. If you use the /BIND qualifier, these allocated volumes are bound to the end of the volume (set) specified in the /BIND qualifier.

**/RECLENGTH=number**

Modifies the volume's record length attribute. If not specified, the record length attribute is not changed.

This qualifier requires the right MDMS\_SET\_OWN or MDMS\_SET\_POOL, and MDMS\_SET\_ALL or MDMS\_SET\_VOLUME if the /USER\_NAME qualifier is used.

**/REPLY=symbol**

The name of the symbol to receive the operator's reply when operator intervention is needed. The symbol will contain the operator reply to the DCL REPLY/TO or DCL REPLY/ABORT commands. The maximum length of a symbol name is 31 characters. This qualifier is only applicable when /ASSIST is specified.

**/SCRATCH\_DATE=date****/NOSCRATCH\_DATE**

Modifies the scratch date in the volume record. The scratch date is the date that the volume is placed in the TRANSITION state (or FREE state if the volume has no transition time). If /NOSCRATCH\_DATE is specified, the volume will never be automatically deallocated. If not specified, the volume's scratch date is not changed.

This qualifier requires the right MDMS\_SET\_OWN or MDMS\_SET\_POOL, and MDMS\_SET\_ALL or MDMS\_SET\_VOLUME if the /USER\_NAME qualifier is used.

**/TRANSITION\_TIME=delta\_time****/NOTRANSITION\_TIME**

Modifies the amount of time, as a delta time, that will be applied to the current scratch date to form the length of time the volume(s) will remain in the TRANSITION state before going into the FREE state.

Use the standard OpenVMS delta time format to specify a delta time for the duration.

If /NOTRANSITION\_TIME is specified, the volume goes directly into the FREE state on deallocation. If not specified, the transition time in the volume record is not changed.

This qualifier requires the right MDMS\_SET\_OWN or MDMS\_SET\_POOL, and MDMS\_SET\_ALL or MDMS\_SET\_VOLUME if the /USER\_NAME qualifier is used.

**/USER\_NAME=username**

The user for whom the volume is being allocated. The username must exist on the system where the command is entered.

The maximum length of the username is 31 characters.

This qualifier requires the right MDMS\_ALLOCATE\_ALL.

## MDMS ALLOCATE VOLUME

### Examples

```
$ MDMS ALLOCATE VOLUME /QUANTITY=3 /MEDIA_TYPE=TK85K
```

This command allocates three TK85K volumes to the current user. The volumes are bound in a new volume set.

```
$ MDMS ALLOCATE VOLUME /MEDIA=TK85K /LOCATION=CXO
```

This command allocates one TK85K volume at location CXO to the current user.

```
$ MDMS ALLOCATE VOLUME /BIND=TEST01
```

This command allocates one volume with the same attributes as TEST01 to the current user. The new volume is bound to the end of the volume set containing TEST01.

```
$ MDMS ALLOCATE VOLUME USER30 /DESCRIPTION="MAY REPORTS"
```

This command allocates volume USER30 to the current user and modifies the volume's description to "MAY REPORTS".

```
$ MDMS ALLOCATE VOLUME /LIKE_VOLUME=AGW500 -  
/USER_NAME=SYSTEM
```

This command allocates a volume with similar attributes to volume AGW500, for user SYSTEM



---

## MDMS BIND VOLUME

The MDMS BIND VOLUME command binds a volume to another volume or volume set or binds a volume set to another volume set.

Equivalent STORAGE Command: STORAGE APPEND

### Format:

MDMS BIND VOLUME [volume\_id]

### Parameters

volume\_id

Specifies the volume ID of the volume to bind. If an existing volume set is to be bound to another volume or volume set, then specify the volume ID of the first member of the volume set to bind. If the volume\_id parameter is omitted, a volume with similar characteristics to those in the set will be allocated and bound to the end of the volume set.

The maximum length of a volume ID is 6 characters. Valid characters are " ,! ,% -? ,A-Z ,\_ ".

### Description

The MDMS BIND VOLUME command binds a volume to another volume or volume set or binds a volume set to another volume set. When binding to a volume or volume set, all involved volumes must be allocated and of the same media type.

When binding a volume set to another volume or volume set, the volume\_id parameter must be the first member of the volume set. To append to the end of a volume set, use the /TO\_SET qualifier.

The volumes in the volume set must already be allocated to the same user (username and UIC). When a new volume is bound to a volume set the scratch dates of all volumes in the set are set to the scratch date of the newly bound volume.

### Privileges

The request requires MDMS\_BIND\_ALL or MDMS\_BIND\_OWN.

If the /USER\_NAME qualifier is not specified, MDMS\_BIND\_OWN allows the user to bind volumes which are allocated to him. Binding on behalf of another user with the /USER\_NAME qualifier requires MDMS\_BIND\_ALL.

### Restrictions

None

### Qualifiers

**/TO\_SET=volume\_id**

The volume ID of a volume or volume set member. The volume (set) specified in the volume\_id parameter is appended to the end of the volume set containing this volume.

The maximum length of the volume ID is 6 characters. Valid characters are " ,! ,% -? ,A-Z ,\_ ".

If the volume\_id command parameter is not specified, then a volume similar to the volume in the /TO\_SET qualifier is allocated and bound to the end of that volume set.

## MDMS BIND VOLUME

### **/USER\_NAME=username**

The user for whom the volume is being bound. The username must exist on the system where the command is entered. The maximum length of the username is 31 characters.

This qualifier requires the right MDMS\_BIND\_ALL.

## **Examples**

```
$ MDMS BIND VOLUME VOL001/TO_SET=VOL006
```

Volume set one contains VOL001, VOL002 and VOL003.

Volume set two contains VOL004, VOL005 and VOL006.

This command binds the volume set containing VOL001 to the volume set containing VOL006. The created volume set will contain volumes VOL004, VOL005, VOL006, VOL001, VOL002, VOL003. Volumes in both volume sets must have compatible attributes.

```
$ MDMS BIND VOLUME VOL002 /TO_SET=VOL005 -  
/USER_NAME=SYSTEM
```

Volume VOL002 is a single volume and VOL005 is part of a volume set that contains VOL004, VOL005 and VOL006. This command binds the volume VOL002 to the volume set containing VOL005. The created volume set will contain volumes VOL004, VOL005, VOL006, VOL002. All the volumes involved are allocated to user SYSTEM.

---

## MDMS CANCEL REQUEST

The MDMS CANCEL REQUEST command cancels a previously issued request. The request may have been issued either synchronously or asynchronously.

Equivalent STORAGE Command: None.

### Format:

MDMS CANCEL REQUEST request\_id [...]

### Parameters

request\_id

Specifies the request ID(s) of the request(s) to cancel. If the request ID is not known, the user can issue a SHOW REQUESTS command, which displays the request ID's of all outstanding requests.

### Description

The MDMS CANCEL REQUEST command cancels the specified outstanding request. The following types of request can be cancelled using this command:

- ALLOCATE DRIVE
- ALLOCATE VOLUME
- INITIALIZE VOLUME
- INVENTORY JUKEBOX
- LOAD DRIVE
- LOAD VOLUME
- MOVE MAGAZINE
- MOVE VOLUME
- UNLOAD DRIVE
- UNLOAD VOLUME
- Any operation involving OPCOM

### Privileges

The request requires MDMS\_CANCEL\_ALL, MDMS\_CANCEL\_OWN or MDMS\_CANCEL\_POOL.

If you are canceling your own request, MDMS\_CANCEL\_OWN or MDMS\_CANCEL\_POOL is required. If you are canceling any other request, MDMS\_CANCEL\_ALL is required.

### Restrictions

None

### Qualifiers

None

## MDMS CANCEL REQUEST

### **Example**

```
$ MDMS CANCEL REQUEST 812
```

This command cancels the request with ID 812.

---

## MDMS CREATE DRIVE

The MDMS CREATE DRIVE command creates a new drive definition in the MDMS database.  
Equivalent STORAGE Command: None

### Format:

MDMS CREATE DRIVE drive\_name [...]

### Parameters

drive\_name

Specifies the name of the drive. The OpenVMS device name must also be specified with the /DEVICE qualifier if it is different from the drive name. The drive name must be unique in the MDMS domain. You should not specify a node in the drive name; rather you should specify either a list of nodes or groups that have access to the drive in the /NODES or /GROUPS qualifiers.

The maximum length of a drive name is 31 characters. Valid characters are "\$,-,0-9,A-Z,\_,a-z".

A comma-separated list of drive names may be entered.

### Description

The MDMS CREATE DRIVE command creates a new drive definition in the MDMS database.

When creating a new object record, MDMS supplies default values on attributes you do not specify. Alternatively, they can be inherited from a specified drive using the /INHERIT qualifier.

### Privileges

The request requires MDMS\_CREATE\_ALL.

The /STATE qualifier also requires MDMS\_SET\_PROTECTED.

Since this attribute is normally managed by MDMS. You should not modify this attribute unless you are trying to recover from an abnormal situation.

### Restrictions

The /ADD, /REMOVE and /REPLACE qualifiers are mutually exclusive. If none are specified, attributes are added to list attributes by default.

The /JUKEBOX and /STACKER qualifiers are mutually exclusive.

The MDMS server will refuse this command if mutually exclusive qualifiers are used in the affirmative forms.

## Qualifiers

### **/ACCESS=keyword**

This qualifier defines the type of access to the drive, which can be one of the following keywords:

ALL (D)	Supports local node/cluster/fibre access and remote (RDF) access
LOCAL	Supports local node/cluster/fibre access only
REMOTE	Supports remote (RDF) access only

Access to drives is restricted on allocate requests - for example, it is not possible to allocate a drive designated as local access remotely using RDF. However, with the proper rights, it is possible to issue other MDMS commands (such as LOAD) both locally and remotely.

The default is ALL.

### **/ACCESS\_CONTROL=(USERNAME=user\_id, ALLOW=(option[,...]))**

#### **/NOACCESS\_CONTROL (D)**

Access control allows the system administrator to restrict access to the object to all or specific users on specific nodes or groups. Select one or more of the following options to allow the operation:

- SET - Allows setting the attributes of the object
- SHOW - Allows showing the object
- DELETE - Allows deleting the object
- READ - Allows restoring data using the object
- WRITE - Allows saving data using the object
- EXECUTE - Allows execution operations using the object
- CONTROL - Allows setting the access control on the object

By default, access controls are added to the list of access controls - only one user can be added per command. However, you can remove users using the /REMOVE qualifier, and replace the entire access control using the /REPLACE qualifier. You can remove all access controls on the object by specifying /NOACCESS\_CONTROL.

The user\_id should be in the format node::username or group::username. Wildcards are supported in the node, group and username portions.

Access control restrictions are in addition to the normal MDMS rights allowing operations on an object. If no access control is defined, then the default domain access control is applied.

### **/ADD (D)**

The /ADD qualifier works in conjunction with certain qualifiers that accept lists and adds the specified attributes to the existing list. This is the default when specifying list items.

### **/AUTOMATIC\_REPLY (D)**

#### **/NOAUTOMATIC\_REPLY**

Specifies that MDMS automatically replies to all OPCOM messages that can be polled for completion on requests for this particular drive.

The default is /AUTOMATIC\_REPLY.

### **/DESCRIPTION="text"**

Comments about the drive. If the text contains spaces, then it must be enclosed within quotation marks. The length of the description can range from 0 to 255 characters. To clear the description, specify "".

**/DEVICE=vms\_device\_name**

Specifies the OpenVMS device name of the drive. This qualifier is required if the OpenVMS device name is different from the drive name. Do not specify a node specification (NODE::name) in the vms\_device\_name. Instead use the /NODES or /GROUPS qualifier.

The default device name is the drive name.

**/DISABLED**

Places the drive in the disabled state. This prevents the drive from being selected and allocated for use. This takes effect immediately. However, if the drive is already in use, operations on that drive will continue until the drive is deallocated.

The default is /ENABLED.

**/DRIVE\_NUMBER=number**

This qualifier defines the drive number for robot commands if the drive is in a jukebox. This qualifier must be specified for multi-drive MRD-controlled jukeboxes.

The default is drive number 0.

**/ENABLED (D)**

Places the drive in the enabled state. This allows the drive to be selected and allocated for use. This takes effect immediately.

The default is /ENABLED.

**/GROUPS=(group\_name[,...])****/NOGROUPS**

Specifies the names of groups of nodes that share common access to this device. Usually, only one group is specified. If neither /NODES or /GROUPS are specified, the node from which the command was issued is used as the node name. Groups can also be specified with the /NODES qualifier.

**/INHERIT=drive\_name**

This qualifier allows you to specify an existing drive record from which the new object record inherits attribute values. MDMS supplies default values if you specify none. All attributes maybe inherited with the exception of the following:

Drive Name

Device name

State

**/JUKEBOX=jukebox\_name**

If the drive is in a jukebox, this qualifier specifies the jukebox name.

**/MEDIA\_TYPE=(media\_type[,...])****/NOMEDIA\_TYPE**

Specifies one or more distinct media types that the drive can support for read-write access. The /REMOVE or /REPLACE qualifiers can be used to remove or replace objects in the list, rather than adding them by default. The /NOMEDIA\_TYPES qualifier removes all media types. When a drive is created with no media types, the default media type from the domain is used.

**/NODES=(node\_name[,...])**

**/NONODES**

Specifies one or more distinct nodes that have direct access to the drive. The /REMOVE or /REPLACE qualifiers can be used to remove or replace objects in the list, rather than adding them by default. The /NONODES qualifier removes all nodes. If neither /NODES nor /GROUPS is specified, the node from which the command was issued is used as the node name.

**/OWNER\_NAME=node::username**

**/OWNER\_NAME=group::username**

**/NOOWNER\_NAME**

The /OWNER\_NAME qualifier specifies the owner of the object. Specify an owner as node::username or group::username. The node, group and username portions may be wild-carded.

The owner of an object has default access control of all allowances to the object, regardless of whether a specific access control entry is present or the state of the default domain access control. If you wish to remove this unconditional owner access, specify /NOOWNER\_NAME. Note that being an owner of an object overrides only the access control; the appropriate MDMS rights for a request on the object must still be honored.

By default, the owner of an object is the group\_name::username of the user entering the CREATE command on the object from a node in the group.

**/READONLY=(media\_type[,...])**

**/NOREADONLY**

Specifies one or more distinct media types that the drive can support for read-only access. The /REMOVE or /REPLACE qualifiers can be used to remove or replace objects in the list, rather than adding them by default. The /NOREADONLY qualifier removes all read-only media types.

**/REMOVE**

The /REMOVE qualifier works in conjunction with certain qualifiers that accept lists and removes specified attributes from them.

**/REPLACE**

The /REPLACE qualifier works in conjunction with certain qualifiers that accept lists and replaces the existing attributes list with the specified list. By default, attributes are added to the existing list.

**/SHARED**

**/NOSHARED (D)**

The /SHARED qualifier defines that the drive may be used by non-MDMS clients and that the drive is only partially managed. If the drive is set to the default /NOSHARED, the MDMS server allocates the drive at all times it is not used by an application or user. Setting the drive to /SHARED clears this allocation.

The default is /NOSHARED.

**/STACKER**

**/NOSTACKER (D)**

The /STACKER qualifier indicates that the drive is to be treated as a stacker gravity loader. The default /NOSTACKER indicates that the drive is to be treated as a standalone drive, or a robotically-controlled jukebox, as appropriate.

The default is /NOSTACKER.



**/STATE=state**

This is a protected field that should be modified only to recover on error. Use the **LOAD** and **UNLOAD** commands to manipulate the state field under normal operation. The **/STATE** qualifier sets the current drive state. The valid keywords are:

Empty

Full

Loading

Unloading

This qualifier requires the right **MDMS\_SET\_PROTECTED**.

**Examples**

```
$ MDMS CREATE DRIVE $1$MUA5:
```

This command creates a local drive called **\$1\$MUA5** with default parameters supplied by MDMS. In this example, the drive name is equivalent to the OpenVMS device name and the current node is used as the drive's node name.

```
$ MDMS CREATE DRIVE DRIVE_1 /DEVICE=$1$MUA5:/NODE=FARLFT
```

This command creates a local drive called **DRIVE\_1** for OpenVMS device **\$1\$MUA5** on node **FARLFT**.

```
$ MDMS CREATE DRIVE JUKE_1_DRIVE_1 -
```

```
/MEDIA_TYPES=(TK85K, TK88K) /DEVICE=$1$MKA500 -
```

```
/GROUPS=COOKIE /SHARE /JUKEBOX=JUKE_1/DRIVE_NUMBER=1 /ACCESS=ALL
```

This command creates a drive with drive name **JUKE\_1\_DRIVE\_1** and OpenVMS device name **\$1\$MKA500** in group **COOKIE**, for local and remote access, shared by non-MDMS users, and supporting media types **TK85K** and **TK88K**. The drive is part of jukebox **JUKE\_1** with drive number 1.

```
$ MDMS CREATE DRIVE FRED /ACCESS=REMOTE
```

This command creates a drive object record named **FRED** for remote access. MDMS supplies all default attribute values.

```
$ MDMS CREATE DRIVE SPARKY_5 /DEVICE=$1$MUA5 -
```

```
/NODE=SPARKY /DISABLE /MEDIA_TYPES=TK85K /ACCESS=ALL
```

This command defines drive **SPARKY\_5**, device **\$1\$MUA5** on node **SPARKY**, that supports media type **TK85K**, supports both local and remote access, and is located on remote node **SPARKY**. The drive is not available for immediate use.

---

## MDMS CREATE GROUP

The MDMS CREATE GROUP command creates a definition of a group of nodes in the MDMS domain.

Equivalent STORAGE Command: None

### Format:

MDMS CREATE GROUP group\_name [...]

### Parameters

group\_name

Specifies the name of the group. A group includes nodes that share a common storage device, or some other relationship. There is no limit to the number of groups that you may specify, and any node may appear in any number of groups. Groups may or may not be equivalent to clusters in your environment. Groups may be used instead of nodes in drive and jukebox definitions, and also as authorized or default users in pool definitions.

The maximum length of the group name is 31 characters.  
Valid characters are "\$,-,0-9,A-Z,\_,a-z".

A comma-separated list of group names may be entered.

### Description

The MDMS CREATE GROUP command creates a new group definition in the MDMS database. When creating a new object record, the user can specify attribute values or allow for MDMS default value assignments. Alternatively, values can be inherited from a specified group using the /INHERIT qualifier.

### Privileges

The request requires MDMS\_CREATE\_ALL.

### Restrictions

The /ADD, /REMOVE and /REPLACE qualifiers are mutually exclusive. If none are specified, attributes are added to list attributes by default.

The MDMS server will refuse this command if mutually exclusive qualifiers are used in the affirmative forms.

### Qualifiers

**/ACCESS\_CONTROL=(USERNAME=user\_id, ALLOW=(option[,...]))**

**/NOACCESS\_CONTROL (D)**

Access control allows the system administrator to restrict access to the object to all or specific users on specific nodes or groups. Select one or more of the following options to allow the operation:

- SET - Allows setting the attributes of the object
- SHOW - Allows showing the object
- DELETE - Allows deleting the object
- READ - Allows restoring data using the object

- WRITE - Allows saving data using the object
- EXECUTE - Allows execution operations using the object
- CONTROL - Allows setting the access control on the object

By default, access controls are added to the list of access controls - only one user can be added per command. However, you can remove users using the /REMOVE qualifier, and replace the entire access control using the /REPLACE qualifier. You can remove all access controls on the object by specifying /NOACCESS\_CONTROL.

The user\_id should be in the format node::username or group::username. Wildcards are supported in the node, group and username portions.

Access control restrictions are in addition to the normal MDMS rights allowing operations on an object. If no access control is defined, then the default domain access control is applied.

#### **/ADD (D)**

The /ADD qualifier works in conjunction with certain qualifiers that accept lists and adds the specified attributes to the existing list. This is the default when specifying list items.

#### **/DESCRIPTION="text"**

Comments about the group. If the text contains spaces, then it must be enclosed within quotation marks. The length of the description can range from 0 to 255 characters. To clear the description, specify "".

#### **/INHERIT=group\_name**

This qualifier allows you to specify an existing group record to inherit attribute values. All attributes may be inherited except for group name.

#### **/NODES=(node\_name[,...])**

#### **/NONODES**

Specifies one or more distinct nodes that are members of the group. The /REMOVE or /REPLACE qualifiers can be used to remove or replace objects in the list, rather than adding them by default. The /NONODES qualifier removes all nodes.

When specifying nodes, the node names are the MDMS node names, which should have an equivalent node object (not the DECnet-Plus (Phase V) or TCP/IP fullnames).

#### **/OWNER\_NAME=node::username**

#### **/OWNER\_NAME=group::username**

#### **/NOOWNER\_NAME**

The /OWNER\_NAME qualifier specifies the owner of the object. Specify an owner as node::username or group::username. The node, group and username portions may be wild-carded.

The owner of an object has default access control of all allowances to the object, regardless of whether a specific access control entry is present or the state of the default domain access control. If you wish to remove this unconditional owner access, specify /NOOWNER\_NAME. Note that being an owner of an object overrides only the access control; the appropriate MDMS rights for a request on the object must still be honored.

By default, the owner of an object is the group\_name::username of the user entering the CREATE command on the object from a node in the group.

#### **/REMOVE**

The /REMOVE qualifier works in conjunction with certain qualifiers that accept lists and removes specified attributes from them.

## MDMS CREATE GROUP

### **/REPLACE**

The **/REPLACE** qualifier works in conjunction with certain qualifiers that accept lists and replaces the existing attributes list with the specified list. By default, attributes are added to the existing list.

### **Examples**

```
$ MDMS CREATE GROUP HOSER /NODES=(TOOKUS, GREAT, NORTH)
```

This command creates a group named **HOSER** and includes the nodes **TOOKUS**, **GREAT**, and **NORTH**.

```
$ MDMS CREATE GROUP MOUNTIE /INHERIT=HOSER -
```

```
/NODES=LABATT /ADD
```

This command creates a new group that includes the nodes in the group **HOSER** and adds node **LABATT** to the node list. You might use this command if the group **HOSER** includes all nodes in an OpenVMS Cluster, and you want to manage a device shared between that cluster and node **LABATT** (which could be a member of a different OpenVMS Cluster)

---

## MDMS CREATE JUKEBOX

The MDMS CREATE JUKEBOX command creates a new jukebox definition in the MDMS database.

Equivalent STORAGE Command: None

### Format:

MDMS CREATE JUKEBOX jukebox\_name [...]

### Parameters

jukebox\_name

Specifies the name of the jukebox. The jukebox must be unique in the MDMS domain.

The maximum length of the jukebox name is 31 characters.

Valid characters are "\$,-,0-9,A-Z,\_,a-z".

A comma-separated list of jukebox names may be entered.

### Description

The MDMS CREATE JUKEBOX command creates a new jukebox definition in the MDMS database. MDMS supplies attribute values when creating a new object record. Alternatively, they can be inherited from a specified jukebox using the /INHERIT qualifier.

An MDMS jukebox may be controlled by the MRD subsystem if it accepts direct SCSI robotic operations. Each MRD-controlled jukebox contains a set of slots, drives and robotics configured with a single robotic device name, regardless of whether the the jukebox is a loader, a single-tower library or a multi-tower library. A robot name must be associated with each MRD-controlled jukebox. For multi-tower configurations, a topology field may optionally be associated with the jukebox if the jukebox supports direct magazine moves.

Certain other types of jukeboxes, specifically silos manufactured by StorageTek, require the DCSC subsystem for control. With this model, each MDMS jukebox object maps to a Library Storage Module (LSM) containing a carousel of cells, drives and robotics. One or more LSMs are contained in an Automated Cartridge System (ACS) and one or more ACS's are managed by a UNIX-based system called a library. For each DCSC-controlled jukebox, the library, ACS ID and LSM ID must be specified. In addition, each LSM contains one or more Cartridge Access Points (CAPs) that are used to move volumes into and out of the jukebox (LSM). Each CAP may contain a different number of cells, so the CAP size should be specified for each CAP for optimal performance. A value of 40 is used if the CAP size is not specified for a particular CAP.

### Privileges

The request requires MDMS\_CREATE\_ALL.

The /STATE qualifier also requires MDMS\_SET\_PROTECTED since this attribute is normally managed by MDMS. You should not modify this attribute unless you are trying to recover from an abnormal situation.

## Restrictions

The /ADD, /REMOVE and /REPLACE qualifiers are mutually exclusive. If none are specified, attributes are added to list attributes by default.

The MDMS server will refuse this command if mutually exclusive qualifiers are used in the affirmative forms.

## Qualifiers

### **/ACCESS=keyword**

This qualifier defines the type of access to the jukebox, which can be one of the following keywords:

ALL (D) - supports local node/access and remote access - default

LOCAL - supports local node/cluster access only

REMOTE - supports remote access only

### **/ACCESS\_CONTROL=(USERNAME=user\_id, ALLOW=(option[,...]))**

#### **/NOACCESS\_CONTROL (D)**

Access control allows the system administrator to restrict access to the object to all or specific users on specific nodes or groups. Select one or more of the following options to allow the operation:

- SET - Allows setting the attributes of the object
- SHOW - Allows showing the object
- DELETE - Allows deleting the object
- READ - Allows restoring data using the object
- WRITE - Allows saving data using the object
- EXECUTE - Allows execution operations using the object
- CONTROL - Allows setting the access control on the object

By default, access controls are added to the list of access controls - only one user can be added per command. However, you can remove users using the /REMOVE qualifier, and replace the entire access control using the /REPLACE qualifier. You can remove all access controls on the object by specifying /NOACCESS\_CONTROL.

The user\_id should be in the format node::username or group::username. Wildcards are supported in the node, group and username portions.

Access control restrictions are in addition to the normal MDMS rights allowing operations on an object. If no access control is defined, then the default domain access control is applied.

### **/ACS=acs\_id**

The /ACS qualifier specifies the Automated Cartridge System (ACS) identifier. The default value is zero. Each MDMS jukebox maps to one Library Storage Module (LSM), and requires the specification of the library, ACS and LSM identifiers. Valid for DCSC-controlled jukeboxes only.

### **/ADD (D)**

The /ADD qualifier works in conjunction with certain qualifiers that accept lists and adds the specified attributes to the existing list. This is the default when specifying list items.

**/AUTOMATIC\_REPLY (D)****/NOAUTOMATIC\_REPLY**

Specifies that MDMS automatically replies to all OPCOM messages that can be polled for completion on requests for this particular jukebox.

**/CAP\_SIZE=(number[,...])**

For DCSC-controlled jukeboxes equipped with Cartridge Access Points (CAPs), this attribute specifies the number of cells for each CAP. The first number is the size for CAP 0, the next for CAP 1 etc. If a size is not specified for a CAP, a default value of 40 is used. Specifying the CAP size optimizes the movement of volumes to and from the jukebox by filling the CAP to capacity for each move operation. When specifying CAP sizes, the specified numbers always replace any previous sizes in the database. Valid for DCSC-controlled jukeboxes only.

**/CONTROL=keyword**

This qualifier specifies the robot control facility used to control the jukebox's robot. The valid keywords are:

MRD (D) - The robot is controlled with the MRD facility

DCSC - The jukebox is a silo controlled by the DCSC facility - for StorageTek(R) Silos only. This option is not available when running with the ABS-OMT license: control is hard-coded to MRD.

**/DESCRIPTION="text"**

Defines comments about the object in the record. If the text contains spaces, then it must be enclosed within quotation marks. The length of the description can range from 0 to 255 characters. To clear the existing description, specify "".

**/DISABLED**

Places the jukebox in the disabled state. This prevents all drives in the jukebox from being selected and allocated for use. This takes effect immediately. However, any drives in use will continue to be used until they are deselected.

**/ENABLED (D)**

Places the jukebox in the enabled state. This allows drives in the jukebox to be selected and allocated for use. This takes effect immediately. This is the default.

**/GROUPS=(group\_name[,...])****/NOGROUPS**

Specifies the names of groups of nodes that share common access to this device. Normally, only one group is specified. The /REMOVE or /REPLACE qualifiers can be used to remove or replace objects in the list, rather than adding them by default. The /NOGROUPS qualifier removes all groups. If neither /NODES nor /GROUPS is specified, the node from which the request was issued is used as the node name. Groups can also be specified with the /NODES qualifier.

**/INHERIT=jukebox\_name**

This qualifier allows you to specify an existing jukebox record to inherit attributes. MDMS supplies values for attributes you do not specify on creation. All attributes may be inherited except, for the following:

Jukebox name

Robot name

State

**/LIBRARY=library\_id**

This qualifier specifies the library identifier for a silo. Valid values are 1,2,3,4, and the default is 1 when the jukebox is controlled by DCSC and 0 (not applicable) when controlled by MRD. Each MDMS jukebox maps to one Library Storage Module (LSM), and requires specification of the library, ACS and LSM identifiers.

**/LOCATION=location\_name****/NOLOCATION**

This qualifier specifies the location of the jukebox, which is used when moving volumes into and out of the jukebox. If not specified, or /NOLOCATION is specified, the default onsite location from the domain record is used as the jukebox location.

**/LSM=lsn\_id**

The /LSM qualifier specifies the Library Storage Module (LSM) identifier. The default value is zero. Each MDMS jukebox maps to one Library Storage Module (LSM), and requires the specification of the library, ACS and LSM identifiers. Valid for DCSC-controlled jukeboxes only.

**/NODES=(node\_name[,...])****/NONODES**

Specifies one or more nodes that can directly access the jukebox. The /REMOVE or /REPLACE qualifiers can be used to remove or replace objects in the list, rather than adding them by default. The /NONODES qualifier removes all nodes.

If neither /NODES nor /GROUPS is specified, the node from which the request was issued is used as the node name.

**/OWNER\_NAME=node::username****/OWNER\_NAME=group::username****/NOOWNER\_NAME**

The /OWNER\_NAME qualifier specifies the owner of the object. Specify an owner as node::username or group::username. The node, group and username portions may be wildcarded.

The owner of an object has default access control of all allowances to the object, regardless of whether a specific access control entry is present or the state of the default domain access control. If you wish to remove this unconditional owner access, specify /NOOWNER\_NAME. Note that being an owner of an object overrides only the access control; the appropriate MDMS rights for a request on the object must still be honored.

By default, the owner of an object is the group\_name::username of the user entering the CREATE command on the object from a node in the group.

**/REMOVE**

The /REMOVE qualifier works in conjunction with certain qualifiers that accept lists and removes specified attributes from them.

**/REPLACE**

The /REPLACE qualifier works in conjunction with certain qualifiers that accept lists and replaces the existing attributes list with the specified list. By default, attributes are added to the existing list.

**/ROBOT=robot\_name****/NOROBOT**

The /ROBOT qualifier defines the OpenVMS device name of the robot. Required for, and applicable to MRD-controlled jukeboxes only. Do not specify a node name in the robot name.

**/SLOT\_COUNT=number**

The /SLOT\_COUNT qualifier specifies the total number of slots in the entire jukebox. For any jukebox, either the slot count or topology must be specified. Valid for MRD-controlled jukeboxes only.



**/STATE=keyword**

This is a protected field that should be modified only to recover on error. Use the LOAD, UNLOAD or MOVE commands to manipulate the state field under normal operation. The /STATE qualifier specifies the usage state of the jukebox. The keyword values are:

Available - The jukebox is available for use

In\_use - The jukebox is in use

This qualifier requires the right MDMS\_SET\_PROTECTED.

**/THRESHOLD=number**

Specifies that an OPCOM message is output when the number of free volumes in the jukebox falls below the specified number. The default value is zero, which disables the feature. The OPCOM message is output on a node that can directly access the jukebox.

**/TOPOLOGY=(TOWERS=(number[,...]), FACES=(number[,...]), LEVELS=(number[,...]), SLOTS=(number[,...]))**

Specifies topology of jukebox, when a TL820-class jukebox is being used as a magazine. Valid for MRD-controlled jukeboxes only. The topology specification allows OPCOM messages to move magazines to be specified with TOWER, FACE, LEVEL rather than slot range. The specification of topology is optional. For each tower in the configuration, a corresponding entry must also be placed in FACES LEVELS and SLOTS that reflects the configuration of that tower.

The tower numbers start at zero, and additional towers must be the next number in sequence (i.e. 0,1,2 etc). Other specifications are absolute counts of the entity being specified for each tower (i.e. the total number of faces, levels and slots in each tower).

For example, for a three-tower jukebox, each tower having 8 faces, the first tower having two levels and the other two towers having three levels, and support of 11-slot bin-packs, the topology specification would be:

```
/TOPOLOGY=(TOWERS=(0,1,2), /FACES=(8,8,8), LEVELS=(2,3,3),
```

```
SLOTS=(11,11,11)
```

**/USAGE=[NO]MAGAZINE**

The /USAGE=MAGAZINE qualifier specifies that the jukebox is configured for magazines, and that the movement of volumes may be performed using magazines. The /USAGE=NOMAGAZINE qualifier does not support magazine use. The default is NOMAGAZINE. You must specify /USAGE=MAGAZINE when defining the /TOPOLOGY attribute. Note that you can use the jukebox for non-magazine moves even when the usage is magazine, but the reverse is not true. Valid for MRD-controlled jukeboxes only.

## Examples

```
$ MDMS CREATE JUKEBOX JUKE_1 /SLOT_COUNT=7 -
```

```
/ROBOT=$1$DUA512: /USAGE=MAGAZINE
```

This command creates a jukebox JUKE\_1 with 7 slots and controlled by robot \$1\$DUA512: and supports magazines. The control type is MRD by default, and the current node is used as the node name supporting the jukebox.

```
$ MDMS CREATE JUKEBOX JUKE_2 /SLOT_COUNT=140 -
```

```
/ROBOT=$1$DUA600 /USAGE=NOMAGAZINE /GROUPS=COOKIE
```

This command creates a jukebox JUKE\_2 with 140 slots and controlled by robot \$1\$DUA600 in group COOKIE. The control type is MRD by default and magazines are not supported.

```
$ MDMS CREATE JUKEBOX JUKE_3 /CONTROL=DCSC /LIBRARY=1 -
```

```
/ACS=0 /LSM=0 /CAP_SIZE=(20,30)
```

This command creates a DCSC-controlled jukebox JUKE\_3 with a library number of 1, ACS ID of 0 and LSM ID of 0. In addition, there are two CAPS with sizes of 20 for CAP 0, and 30 for CAP 1. The control type is DCS meaning this is a silo, does not support magazines, and other default attribute values are supplied by MDMS.

```
$ MDMS CREATE JUKEBOX JUKE_4 /CONTROL=MRD -
```

```
/ROBOT=$1$DUA510: /TOPOLOGY=(TOWERS=(0,1),FACES=(8,8), -
```

```
LEVELS=(3,2),SLOTS=(11,11))
```

This command creates a multi-tower jukebox, controlled by MRD with robot name \$1\$DUA510: Its topology is:

TOWER 0 with 8 faces, 3 levels and 11 slots; TOWER 1 with 8 faces, 2 levels and 11 slots.

---

## MDMS CREATE LOCATION

The MDMS CREATE LOCATION command creates a new location definition in the MDMS database.

Equivalent STORAGE Command: None

### Format:

MDMS CREATE LOCATION location [...]

### Parameters

location

Specifies the name of the location.

The maximum length of the location is 31 characters. Valid characters are "\$,-,0-9,A-Z,\_,a-z".

A comma-separated list of locations may be entered.

### Description

The MDMS CREATE LOCATION command creates a new location definition in the MDMS database. MDMS supplies attribute values you do not define on creation. Alternatively, they can be inherited from a named location object using the /INHERIT qualifier.

### Privileges

The request requires MDMS\_CREATE\_ALL.

### Restrictions

None

### Qualifiers

**/ACCESS\_CONTROL=(USERNAME=user\_id, ALLOW=(option[,...]))**

**/NOACCESS\_CONTROL (D)**

Access control allows the system administrator to restrict access to the object to all or specific users on specific nodes or groups. Select one or more of the following options to allow the operation:

- SET - Allows setting the attributes of the object
- SHOW - Allows showing the object
- DELETE - Allows deleting the object
- READ - Allows restoring data using the object
- WRITE - Allows saving data using the object
- EXECUTE - Allows execution operations using the object
- CONTROL - Allows setting the access control on the object

By default, access controls are added to the list of access controls - only one user can be added per command. However, you can remove users using the /REMOVE qualifier, and replace the entire access control using the /REPLACE qualifier. You can remove all access controls on the object by specifying /NOACCESS\_CONTROL.

## MDMS CREATE LOCATION

The user\_id should be in the format node::username or group::username. Wildcards are supported in the node, group and username portions.

Access control restrictions are in addition to the normal MDMS rights allowing operations on an object. If no access control is defined, then the default domain access control is applied.

### **/DESCRIPTION="text"**

Defines comments about the object in the record. If the text contains spaces, then it must be enclosed within quotation marks. The length of the description can range from 0 to 255 characters. To clear the existing description, specify "".

### **/INHERIT=location**

This qualifier allows you to specify an existing location record to inherit default attributes. The default is that MDMS supplies the default values. All attributes may be inherited except for location name.

### **/LOCATION=location**

#### **/NOLOCATION (D)**

The /LOCATION qualifier allows you to specify a parent location, thus creating a location hierarchy. If there is no parent location, specify /NOLOCATION. Use parent locations to allow selection of volumes or drives in compatible locations. One location is compatible with another if it has a common parent location in the hierarchy. If you do not wish to utilize the compatible location feature, do not specify parent locations. Locations with common parents are most useful where the parents and siblings are in close proximity to one another (e.g. rooms 101 and 102, with parent location floor 1), and selection of volumes or drives from any of the locations is desired. Do not use parent locations across larger distances.

### **/OWNER\_NAME=node::username**

### **/OWNER\_NAME=group::username**

### **/NOOWNER\_NAME**

The /OWNER\_NAME qualifier specifies the owner of the object. Specify an owner as node::username or group::username. The node, group and username portions may be wildcarded.

The owner of an object has default access control of all allowances to the object, regardless of whether a specific access control entry is present or the state of the default domain access control. If you wish to remove this unconditional owner access, specify /NOOWNER\_NAME. Note that being an owner of an object overrides only the access control; the appropriate MDMS rights for a request on the object must still be honored.

By default, the owner of an object is the group\_name::username of the user entering the CREATE command on the object from a node in the group.

### **/SPACES=(range)**

### **/NOSPACES**

The /SPACES qualifier defines individual spaces for volumes or magazines at the location. Spaces are alphanumeric strings of up to 8 characters. The spaces can be specified as a range - only a single range is supported.

When specifying a range, the first and last spaces in the range must have the same number of characters (as in the example), and there is a limit of 1000 spaces per location. The /NOSPACES qualifier removes all spaces.

**Examples**

```
$ MDMS CREATE LOCATION SHELF_40 /SPACES=(40:S001-40:S100)
```

This command creates a shelf location for volumes with 100 spaces, named 40:S001 to 40:S100.

```
$ MDMS CREATE LOCATION VAULT_1
```

This command creates a location named VAULT\_1 with no spaces.

---

## MDMS CREATE MAGAZINE

The MDMS CREATE MAGAZINE command creates a new magazine definition in the MDMS database.

Equivalent STORAGE Command: STORAGE ADD MAGAZINE

### Format:

MDMS CREATE MAGAZINE magazine\_name [...]

### Parameters

magazine\_name

Specifies the name of the magazine.

The maximum length of the magazine name is 31 characters.

Valid characters are "\$,-,0-9,A-Z,\_,a-z".

A comma-separated list of magazine names may be entered.

### Description

The MDMS CREATE MAGAZINE command creates a new magazine definition in the MDMS database. Magazines are used for two reasons:

- For compatibility with previous versions of MDMS, where loader-type jukeboxes could only be supported using magazines.
- To support a set of volumes in a physical magazine whose placement is the same for all volumes in the magazine. In other words, if a volume is in a magazine, it is moved with all the other volumes in the magazine.

It is important to note that you no longer have to use magazines for loader-type jukeboxes. You may instead choose to treat volumes separately and move them into and out of jukeboxes individually.

### Privileges

The request requires MDMS\_CREATE\_ALL.

Several protected qualifiers also require MDMS\_SET\_PROTECTED since these attributes are normally managed by MDMS. You should not modify these attributes unless you are trying to recover from an abnormal situation.

The /JUKEBOX, /PLACEMENT, /POSITION and /START\_SLOT are qualifiers that also require MDMS\_SET\_PROTECTED.

### Restrictions

None

## Qualifiers

**/ACCESS\_CONTROL=(USERNAME=user\_id, ALLOW=(option[,...]))**

**/NOACCESS\_CONTROL (D)**

Access control allows the system administrator to restrict access to the object to all or specific users on specific nodes or groups. Select one or more of the following options to allow the operation:

- SET - Allows setting the attributes of the object
- SHOW - Allows showing the object
- DELETE - Allows deleting the object
- READ - Allows restoring data using the object
- WRITE - Allows saving data using the object
- EXECUTE - Allows execution operations using the object
- CONTROL - Allows setting the access control on the object

By default, access controls are added to the list of access controls - only one user can be added per command. However, you can remove users using the /REMOVE qualifier, and replace the entire access control using the /REPLACE qualifier. You can remove all access controls on the object by specifying /NOACCESS\_CONTROL.

The user\_id should be in the format node::username or group::username. Wildcards are supported in the node, group and username portions.

Access control restrictions are in addition to the normal MDMS rights allowing operations on an object. If no access control is defined, then the default domain access control is applied.

**/DESCRIPTION="text"**

Defines comments about the object in the record. If the text contains spaces, then it must be enclosed within quotation marks. The length of the description can range from 0 to 255 characters. To clear the existing description, specify "".

**/INHERIT=magazine\_name**

This qualifier allows you to specify an existing magazine record to inherit default attributes. The default is that MDMS supplies values you do not specify. All attributes may be inherited except for the following:

Magazine name

Jukebox name

Placement

Position

Start slot

**/JUKEBOX=jukebox\_name**

This is a protected field that should only be modified to recover on error. Use the MOVE MAGAZINE command to set up the jukebox name under normal operations. This qualifier specifies the name of the jukebox in which the magazine resides. The maximum length of the jukebox name is 31 characters.

This qualifier requires the right MDMS\_SET\_PROTECTED.

**/OFFSITE=(*[LOCATION=location]* [, *[NO]DATE[=date]*])**

**/NOOFFSITE**

This qualifier specifies the date that the magazine is to be taken offsite and the offsite location. The location field is required when using the /OFFSITE qualifier if no location has been previously specified. The LOCATION keyword cannot be negated and if specified must point to a valid location object. Specify a VMS absolute or delta time in the date field. The NODATE keyword may be used to remove the date. This has the effect of disabling the schedule for the magazine, while retaining the location. To clear the offsite date specify /NOOFFSITE.

The default offsite location is that stored in the domain object.

**/ONSITE=(*[LOCATION=location]* [, *[NO]DATE[=date]*])**

**/NOONSITE**

This qualifier specifies the date that the magazine is to be brought back onsite and the onsite location. The location field is required when using the /ONSITE qualifier if no location has been previously specified. The LOCATION keyword cannot be negated and if specified must point to a valid location object. Specify a VMS absolute or delta time in the date field. The NODATE keyword may be used to remove the date. This has the effect of disabling the schedule for the magazine, while retaining the location. To clear the onsite date, specify /NOONSITE.

The default onsite location is that stored in the domain object.

**/OWNER\_NAME=*node::username***

**/OWNER\_NAME=*group::username***

**/NOOWNER\_NAME**

The /OWNER\_NAME qualifier specifies the owner of the object. Specify an owner as *node::username* or *group::username*. The node, group and username portions may be wild-carded.

The owner of an object has default access control of all allowances to the object, regardless of whether a specific access control entry is present or the state of the default domain access control. If you wish to remove this unconditional owner access, specify /NOOWNER\_NAME. Note that being an owner of an object overrides only the access control; the appropriate MDMS rights for a request on the object must still be honored.

By default, the owner of an object is the *group\_name::username* of the user entering the CREATE command on the object from a node in the group.

**/PLACEMENT=*keyword***

This is a protected field that should only be modified to recover on error. Use the MOVE MAGAZINE command to set up the placement under normal operation. This qualifier defines the current placement of the magazine. Select one of the following options:

JUKEBOX

OFFSITE

ONSITE

MOVING

This qualifier requires the right MDMS\_SET\_PROTECTED.



**/POSITION=position=(tower,face,level)**

This is a protected field that should only be modified to recover on error. Use the MOVE MAGAZINE command to set up the position under normal operation. The /POSITION qualifier specifies the position in the jukebox that the magazine resides.

The tower, face and level specification represent the relative number of the tower, face and level, starting from 0. So for absolute jukebox slot of zero, the corresponding position is (0,0,0). The next position in the jukebox would be (0,0,1) and so on, according to the topology defined for the jukebox.

This qualifier requires the right MDMS\_SET\_PROTECTED.

**/SLOT\_COUNT=number**

The /SLOT\_COUNT qualifier specifies the number of slots in a magazine to store volumes. The default slot count is 32 which can be used for all magazines, but you should specify the actual value for best results.

**/SPACES=(range)****/NOSPACES**

This qualifier specifies the space(s) in a location in which the magazine is stored when not in a jukebox. Spaces are alphanumeric strings of up to 8 characters. The /NOSPACES qualifier removes all spaces.

**/START\_SLOT=(number)**

This is a protected field that should only be modified to recover on error. Use the MOVE MAGAZINE command to set up the start slot under normal operation. This qualifier specifies the starting jukebox slot when the magazine is placed in a jukebox.

This qualifier requires the right MDMS\_SET\_PROTECTED.

## Examples

```
$ MDMS CREATE MAGAZINE MYMAG01 /SLOT_COUNT=7 -
```

```
/ONSITE=(LOCATION=SHELF1) /SPACE=2
```

This command creates a magazine called MYMAG01 with 7 slots, which is stored in location space 2 in location SHELF1 when not in a jukebox.

```
$ MDMS CREATE MAGAZINE MAG002 /SLOT_COUNT=32 -
```

```
/SPACES=(SA001-SA032)/ONSITE=(LOCATION=HEADQUARTERS, -
```

```
DATE=01-JAN-2001)/OFFSITE=(LOCATION=DPS, -
```

```
DATE=01-JAN-2000)
```

This command creates a magazine MAG002, with 32 slots, which is stored in onsite location HEADQUARTERS in spaces SA001-SA032, due to be moved offsite on 01-Jan-2000 to location DPS, and back onsite on 01-Jan-2001.

```
$ MDMS CREATE MAGAZINE MAG003 /SLOT_COUNT=32 -
```

```
/SPACES=SPC001 /JUKE=JUKE_1 /POSITION=(0,1,2)
```

This command creates a magazine MAG003, with 32 slots, which is stored in space SPC001. When in a jukebox, the magazine resides in jukebox JUKE\_1 in position tower 0, face 1, level 2. Note that jukebox and position should not normally be specified - rather, these are set up when moving the magazine into the jukebox with a MOVE MAGAZINE command.

---

## MDMS CREATE MEDIA\_TYPE

The MDMS CREATE MEDIA\_TYPE command creates a new media type definition in the MDMS database.

Equivalent STORAGE Command: None

### Format:

MDMS CREATE MEDIA\_TYPE media\_type [...]

### Parameters

media\_type

Specifies the name of the media type.

The maximum length of the media type name is 31 characters.

Valid characters are "\$,-,0-9,A-Z,\_,a-z".

A comma-separated list of media types may be entered.

### Description

The MDMS CREATE MEDIA\_TYPE command creates a new media type definition in the MDMS database. A media type definition consists of a required density, and optional compaction length and capacity, if applicable.

### Privileges

The request requires MDMS\_CREATE\_ALL.

### Restrictions

None

### Qualifiers

**/ACCESS\_CONTROL=(USERNAME=user\_id, ALLOW=(option[...]))**

**/NOACCESS\_CONTROL (D)**

Access control allows the system administrator to restrict access to the object to all or specific users on specific nodes or groups. Select one or more of the following options to allow the operation:

- SET - Allows setting the attributes of the object
- SHOW - Allows showing the object
- DELETE - Allows deleting the object
- READ - Allows restoring data using the object
- WRITE - Allows saving data using the object
- EXECUTE - Allows execution operations using the object
- CONTROL - Allows setting the access control on the object

By default, access controls are added to the list of access controls - only one user can be added per command. However, you can remove users using the /REMOVE qualifier, and replace the entire access control using the /REPLACE qualifier. You can remove all access controls on the object by specifying /NOACCESS\_CONTROL.

The user\_id should be in the format node::username or group::username. Wildcards are supported in the node, group and username portions.

Access control restrictions are in addition to the normal MDMS rights allowing operations on an object. If no access control is defined, then the default domain access control is applied.

**/CAPACITY=number**

The /CAPACITY qualifier specifies the capacity in megabytes of the tape. This is used by some MDMS clients to estimate end-of-tape conditions. By default, capacity is set to zero.

**/COMPACTION (D)**

**/NOCOMPACTION**

The /COMPACTION qualifier specifies that the media type should use compaction when writing to tape. This is the default. If you do not wish to use compaction, then specify /NOCOMPACTION.

**/DENSITY=density**

Specifies a density string between 1 and 31 characters in length that the media type supports. Note that the COMP keyword for compaction should be specified in the /COMPACTION attribute, not density.

**/DESCRIPTION="text"**

Defines comments about the media type. If the text contains spaces, then it must be enclosed within quotation marks. The length of the description is can range from 0 to 255 characters. Specify "" to clear the description.

**/INHERIT=media\_type**

This qualifier allows you to specify an existing media type record to inherit default attributes. The default is that MDMS supplies values you do not specify on creation. All attributes may be inherited except for media type name.

**/LENGTH=length**

The /LENGTH qualifier specifies the length of a 9-track magnetic tape, and is expressed in feet. By default, length is set to zero.

**/OWNER\_NAME=node::username**

**/OWNER\_NAME=group::username**

**/NOOWNER\_NAME**

The /OWNER\_NAME qualifier specifies the owner of the object. Specify an owner as node::username or group::username. The node, group and username portions may be wild-carded.

The owner of an object has default access control of all allowances to the object, regardless of whether a specific access control entry is present or the state of the default domain access control. If you wish to remove this unconditional owner access, specify /NOOWNER\_NAME. Note that being an owner of an object overrides only the access control; the appropriate MDMS rights for a request on the object must still be honored.

By default, the owner of an object is the group\_name::username of the user entering the CREATE command on the object from a node in the group.

## MDMS CREATE MEDIA\_TYPE

### Examples

```
$ MDMS CREATE MEDIA_TYPE TK85K_COMP /COMPACTION
```

This command creates a new media type called TK85K\_COMP with compaction enabled.

```
$ MDMS CREATE MEDIA_TYPE TAPE_800 /DENSITY=800 -
```

```
/LENGTH=2400
```

This command creates a new media type called TAPE\_800, a 9-track media type with 800 bpi density and a length of 2400 feet.

```
$ MDMS CREATE MEDIA_TYPE TAPE_1600 -
```

```
/INHERIT=TAPE_800 /DENSITY=1600
```

This command creates a new media type called TAPE\_1600, a 9-track media type with 1600 bpi density and a length of 2400 feet (inherited from media type TAPE\_800).

---

## MDMS CREATE NODE

The MDMS CREATE NODE command creates a new node definition in the MDMS node database.

Equivalent STORAGE Command: None

### Format:

MDMS CREATE NODE node\_name [...]

### Parameters

node\_name

Specifies the name of the node. Do not append colons to the node name. The node name should be the DECnet (Phase IV) node name (i.e. logical SYS\$NODE) if DECnet (Phase IV) is used on the node.

Otherwise it should be a unique name chosen by the MDMS administrator and it should match the SYSGEN parameter SCSNODE. If SCSNODE is not defined the node name has to match the host or node name portion of the DECnet-Plus (Phase V) or TCP/IP fullname. If SCSNODE name is not defined and no network name is available the server starts up with a name of "MDMS\$SERVER".

If DECnet-Plus (Phase V) and/or TCP/IP are used, the appropriate fullnames should be stored as attributes of the node. Do not use the node name to specify fullnames.

The maximum length of the node name is 31 characters. Valid characters are "\$,-,0-9,A-Z,\_,a-z".

A comma-separated list of node names may be entered.

### Description

The MDMS CREATE NODE command creates a new node definition in the MDMS database. A node record must exist for each node that can act as a client to the database server. In addition, a node record must exist for each node capable of being a database server.

When creating a new object record, default attributes are supplied by MDMS. Alternatively, they can be inherited from a specified node using the /INHERIT qualifier.

### Privileges

The request requires MDMS\_CREATE\_ALL.0

### Restrictions

The /ADD, /REMOVE and /REPLACE qualifiers are mutually exclusive. If none are specified, attributes are added to list attributes by default.

The MDMS server will refuse this command if mutually exclusive qualifiers are used in the affirmative forms.

## Qualifiers

**/ACCESS\_CONTROL=(USERNAME=user\_id, ALLOW=(option[,...]))**

**/NOACCESS\_CONTROL (D)**

Access control allows the system administrator to restrict access to the object to all or specific users on specific nodes or groups. Select one or more of the following options to allow the operation:

- SET - Allows setting the attributes of the object
- SHOW - Allows showing the object
- DELETE - Allows deleting the object
- READ - Allows restoring data using the object
- WRITE - Allows saving data using the object
- EXECUTE - Allows execution operations using the object
- CONTROL - Allows setting the access control on the object

By default, access controls are added to the list of access controls - only one user can be added per command. However, you can remove users using the /REMOVE qualifier, and replace the entire access control using the /REPLACE qualifier. You can remove all access controls on the object by specifying /NOACCESS\_CONTROL.

The user\_id should be in the format node::username or group::username. Wildcards are supported in the node, group and username portions.

Access control restrictions are in addition to the normal MDMS rights allowing operations on an object. If no access control is defined, then the default domain access control is applied.

**/ADD (D)**

The /ADD qualifier works in conjunction with certain qualifiers that accept lists and adds the specified attributes to the existing list. This is the default when specifying list items.

**/DATABASE\_SERVER (D)**

**/NODATABASE\_SERVER**

The /DATABASE\_SERVER qualifier means the node can be a database server, supporting fail-over operations. To be a database server, the node must have direct access to the MDMS Database files.

In addition, this node name should be added to the definition of the logical name MDMS\$DATABASE\_SERVERS in SYS\$STARTUP:MDMS\$SYSTARTUP.COM on all nodes in the domain.

**/DECNET\_PLUS\_FULLNAME=node\_fullname**

This qualifier allows you to specify the DECnet-Plus (Phase V) fullname for a node. The fullname may be up to 255 characters. If this node has a DECnet-Plus name defined by logical name "SYS\$NODE\_FULLNAME" then the DECNET\_PLUS\_FULLNAME has to be defined for this node and has to exactly match the DECnet-Plus (Phase V) name.

The DECNET\_PLUS\_FULLNAME has to be defined in order for this node to be fully enabled when the DECnet transport has been enabled and DECnet-Plus is running on the system. The fullname can be specified in upper or lower case.

**/DESCRIPTION="text"**

Defines comments about the node. If the text contains spaces, then it must be enclosed within quotation marks. The length of the description can range from 0 to 255 characters. Specify "" to clear the description.

**/DISABLED**

Places the node in the disabled state. This prevents the node from participating in the MDMS domain as either a server or a client. This takes effect immediately.

**/ENABLED (D)**

Places the node in the enabled state. This allows the node to participate in MDMS operations. This takes effect immediately. This is the default.

**/INHERIT=node\_name**

This qualifier allows you to specify an existing node record to inherit attributes. The default is that MDMS supplies attribute values you do not supply on creation. All attributes may be inherited except for:

Node name

DECnet\_Plus fullname

TCPIP fullname

**/LOCATION=location****/NOLOCATION**

The /LOCATION qualifier specifies the location of the node, which is used when allocating volumes and drives. If not specified, or /NOLOCATION is specified, the default onsite location from the domain record is used as the node location.

**/OPCOM=(class[,...])****/NOOPCOM**

The /OPCOM qualifier adds the specified classes used for notifying operators. All OPCOM for devices on the node are sent to all specified classes on the node. The /REMOVE or /REPLACE qualifiers can be used to remove or replace classes in the list, rather than adding them by default. Specify /NOOPCOM to disable OPCOM notification. By default, the node acquires OPCOM classes from the domain record. The following classes are valid:

CARDS	NETWORK	OPER6	OPER12
CENTRAL	OPER1	OPER7	PRINTER
CLUSTER	OPER2	OPER8	REPLY
DEVICES	OPER3	OPER9	SECURITY
DISKS	OPER4	OPER10	SOFTWARE
LICENSE	OPER5	OPER11	TAPES

**/OWNER\_NAME=node::username**

**/OWNER\_NAME=group::username**

**/NOOWNER\_NAME**

The /OWNER\_NAME qualifier specifies the owner of the object. Specify an owner as node::username or group::username. The node, group and username portions may be wild-carded.

The owner of an object has default access control of all allowances to the object, regardless of whether a specific access control entry is present or the state of the default domain access control. If you wish to remove this unconditional owner access, specify /NOOWNER\_NAME. Note that being an owner of an object overrides only the access control; the appropriate MDMS rights for a request on the object must still be honored.

By default, the owner of an object is the group\_name::username of the user entering the CREATE command on the object from a node in the group.

**/REMOVE**

The /REMOVE qualifier works in conjunction with certain qualifiers that accept lists and removes specified attributes from them.

**/REPLACE**

The /REPLACE qualifier works in conjunction with certain qualifiers that accept lists and replaces the existing attributes list with the specified list. By default, attributes are added to the existing list.

**/TCPIP\_FULLNAME=node\_fullname[:low\_port-high\_port]**

This qualifier allows you to specify the TCP/IP full name for a node. The full name may be up to 255 characters. If this node has a TCP/IP name defined by logical name "\*INET\_HOST" the TCPIP\_FULLNAME has to be defined and has to exactly match the full IP name as "<INET\_HOST>.<INET\_DOMAIN>". For INET\_DOMAIN see logical name "\*INET\_DOMAIN". The TCPIP\_FULLNAME has to be defined in order for this node to be fully enabled when the TCPIP transport has been enabled. The fullname can be specified in upper or lower case.

The low\_port and high\_port numbers specify the range of TCP/IP port numbers used by the server to listen for incoming requests. The default is 2501-2510. If this conflicts with other applications, a new range above 1023 can be specified. The range should contain at least 10 port numbers for the MDMS server to select one at a time.

Note that the MDMS GUI requires TCP/IP running on all GUI nodes, and on the MDMS server nodes to which the GUI may connect.

**/TRANSPORT=(keyword[,...])**

Specifies the network transports to be used, as a prioritized ordered list. The /REMOVE or /REPLACE qualifiers can be used to remove or replace objects in the list, rather than adding them by default.

Enter one or more of:

- DECNET - listen to incoming requests from other MDMS servers on DECnet (Phase IV) and DECnet-Plus (Phase V)
- TCPIP - listen to incoming requests from other MDMS servers on TCP/IP

Setting a new transport will automatically start the listener for this transport on the database server node. Likewise, removing a transport will take place within 10 seconds on the database server node. For client nodes, transport changes will take place the next time network connections time out (usually within 10minutes). If the change needs to take place immediately, the client node server process must be restarted.



The node name and/or the node full names have to be set accordingly for a transport to work correctly.

## Examples

```
$ MDMS CREATE NODE TABLES /LOCATION=COMPUTER_LAB_1 -  
/TRANSPORT=(DECNET,TCPIP) /OPCOM=(CENTRAL,TAPES) -  
/TCPIP_FULLNAME=TABLES.CXO.DEC.COM
```

This command creates a new node definition named "TABLES" with a location, transport protocols and OPCOM classes. The node supports DECnet (node name TABLES) and TCP/IP with a fullname of TABLES.CXO.DEC.COM.

```
$ MDMS CREATE NODE CHAIRS /INHERIT=TABLES -  
/DECNET_PLUS_FULLNAME=DEC:CHAIRS.CXO.DEC.COM -  
/TCPIP_FULLNAME=CHAIRS.CXO.DEC.COM:3000-3050
```

This command creates a new node named CHAIRS, which inherits location, OPCOM classes and transport definitions from node TABLES, with specified DECnet-Plus and TCPIP full names. The TCP/IP connections may be received over ports 3000 - 3050.

---

## MDMS CREATE POOL

The MDMS CREATE POOL command creates a new pool definition in the MDMS database.  
Equivalent STORAGE Command: None

### Format:

MDMS CREATE POOL pool\_name [...]

### Parameters

pool\_name

Specifies the name of the pool.

The maximum length of the pool name is 31 characters. Valid characters are "\$,-,0-9,A-Z,\_,a-z".

A comma-separated list of pool names may be entered.

### Description

The MDMS CREATE POOL command creates a new pool definition in the MDMS pool database. A pool definition consists of a list of authorized users, and users for whom the pool is the default pool. If a user is listed in either list, he/she is authorized for the pool.

### Privileges

The request requires MDMS\_CREATE\_ALL.

### Restrictions

The /ADD, /REMOVE and /REPLACE qualifiers are mutually exclusive. If none are specified, attributes are added to list attributes by default.

The MDMS server will refuse this command if mutually exclusive qualifiers are used in the affirmative forms.

### Qualifiers

**/ACCESS\_CONTROL=(USERNAME=user\_id, ALLOW=(option[,...]))**

**/NOACCESS\_CONTROL (D)**

Access control allows the system administrator to restrict access to the object to all or specific users on specific nodes or groups. Select one or more of the following options to allow the operation:

- SET - Allows setting the attributes of the object
- SHOW - Allows showing the object
- DELETE - Allows deleting the object
- READ - Allows restoring data using the object
- WRITE - Allows saving data using the object
- EXECUTE - Allows execution operations using the object
- CONTROL - Allows setting the access control on the object

By default, access controls are added to the list of access controls - only one user can be added per command. However, you can remove users using the /REMOVE qualifier, and replace the

entire access control using the /REPLACE qualifier. You can remove all access controls on the object by specifying /NOACCESS\_CONTROL.

The user\_id should be in the format node::username or group::username. Wildcards are supported in the node, group and username portions.

Access control restrictions are in addition to the normal MDMS rights allowing operations on an object. If no access control is defined, then the default domain access control is applied.

**/ADD (D)**

The /ADD qualifier works in conjunction with certain qualifiers that accept lists and adds the specified attributes to the existing list. This is the default when specifying list items.

**/AUTHORIZED\_USERS=(node/group\_name::username[,...])**

**/NOAUTHORIZED\_USERS**

Specifies one or more distinct users to the pool specified by node or group name and user name. Only authorized or default users can allocate volumes belonging to the pool. The /REMOVE or /REPLACE qualifiers can be used to remove or replace users in the list, rather than adding them by default.

**/DEFAULT\_USERS=(node/group\_name::username[,...])**

**/NODEFAULT\_USERS**

Specifies one or more distinct users to the pool as the users' default pool. Only authorized or default users can allocate volumes belonging to the pool. The /REMOVE or /REPLACE qualifiers can be used to remove or replace users in the list, rather than adding them by default. A particular node/group::user combination should only be defined with the /DEFAULT qualifier for one pool.

**/DESCRIPTION="text"**

Defines comments about the pool. If the text contains spaces, then it must be enclosed within quotation marks. The length of the description can range from 0 to 255 characters. Specify "" to clear the description.

**/INHERIT=pool\_name**

This qualifier allows you to specify an existing pool record to inherit attributes. The default is that MDMS supplies values you do not supply on creation. All attributes except pool name may be inherited.

**/OWNER\_NAME=node::username**

**/OWNER\_NAME=group::username**

**/NOOWNER\_NAME**

The /OWNER\_NAME qualifier specifies the owner of the object. Specify an owner as node::username or group::username. The node, group and username portions may be wild-carded.

The owner of an object has default access control of all allowances to the object, regardless of whether a specific access control entry is present or the state of the default domain access control. If you wish to remove this unconditional owner access, specify /NOOWNER\_NAME. Note that being an owner of an object overrides only the access control; the appropriate MDMS rights for a request on the object must still be honored.

By default, the owner of an object is the group\_name::username of the user entering the CREATE command on the object from a node in the group.

**/REMOVE**

The /REMOVE qualifier works in conjunction with certain qualifiers that accept lists and removes specified attributes from them.

## MDMS CREATE POOL

### **/REPLACE**

The /REPLACE qualifier works in conjunction with certain qualifiers that accept lists and replaces the existing attributes list with the specified list. By default, attributes are added to the existing list.

### **/THRESHOLD=number**

Specifies that an OPCOM message is output when the number of free volumes in the pool falls below the specified number. The default value is zero, which disables the feature. The OPCOM message is output on the database server node.

## **Examples**

```
$ MDMS CREATE POOL TEST_POOL/AUTHORIZE=COOKIE::ABS
```

This command creates a pool called TEST\_POOL with one authorized user.

```
$ MDMS CREATE POOL JIMS_POOL /DEFAULT=(OREO::JIM, -  
CRUMBS::JIM, DSORDS::JIM) /ADD
```

This command adds default users to pool JIMS\_POOL. The names before the double colons are a mixture of node names and group names.

---

## MDMS CREATE SCHEDULE

The MDMS CREATE SCHEDULE command creates a new schedule definition in the MDMS database. Schedule is a new object type.

Equivalent ABS Command: None

### Format:

MDMS CREATE SCHEDULE schedule\_name

### Parameters

schedule\_name

Specifies the name of the schedule.

The maximum length of the schedule name is 63 characters.

Valid characters are "\$,-,0-9,A-Z,\_,a-z".

A comma-separated list of schedule names may be entered.

### Description

The MDMS CREATE SCHEDULE command creates a new schedule definition in the MDMS database. You can use schedules to execute any DCL command (including MDMS commands) at regular intervals.

Schedules are used by MDMS to execute saves and restores. However, these schedules are created implicitly by MDMS and do not need to be created with this command. However, you can modify CUSTOM schedules by using the SET SCHEDULE command.

When creating a new object record, MDMS supplies default values on attributes you do not specify. Alternatively, they can be inherited from a specified schedule using the /INHERIT qualifier.

### Privileges

The request requires MDMS\_CREATE\_ALL.

### Restrictions

The /ADD, /REMOVE and /REPLACE qualifiers are mutually exclusive. If none are specified, attributes are added to list attributes by default.

The MDMS server will refuse this command if mutually exclusive qualifiers are used in the affirmative forms.

## Qualifiers

**/ACCESS\_CONTROL=(USERNAME=user\_id, ALLOW=(option[,...]))**

**/NOACCESS\_CONTROL (D)**

Access control allows the system administrator to restrict access to the object to all or specific users on specific nodes or groups. Select one or more of the following options to allow the operation:

- SET - Allows setting the attributes of the object
- SHOW - Allows showing the object
- DELETE - Allows deleting the object
- READ - Allows restoring data using the object
- WRITE - Allows saving data using the object
- EXECUTE - Allows execution operations using the object
- CONTROL - Allows setting the access control on the object

By default, access controls are added to the list of access controls - only one user can be added per command. However, you can remove users using the /REMOVE qualifier, and replace the entire access control using the /REPLACE qualifier. You can remove all access controls on the object by specifying /NOACCESS\_CONTROL.

The user\_id should be in the format node::username or group::username. Wildcards are supported in the node, group and username portions.

Access control restrictions are in addition to the normal MDMS rights allowing operations on an object. If no access control is defined, then the default domain access control is applied.

**/ADD (D)**

The /ADD qualifier works in conjunction with certain qualifiers that accept lists and adds the specified attributes to the existing list. This is the default when specifying list items.

**/AFTER\_SCHEDULE=(NAME=schedule\_name, WHEN=option)**

**/NOAFTER\_SCHEDULE**

The /AFTER\_SCHEDULE qualifiers specifies that this schedule is scheduled for execution after the AFTER\_SCHEDULE name, under the circumstances defined in WHEN. This can be used to serialize a series of operations one after the other, and conditionally execute this schedule depending on the status of the AFTER\_SCHEDULE.

The SCHEDULE should be a schedule name defined in the MDMS database. For WHEN, select one of the following:

---

ALL (D)	Unconditionally execute this schedule when the AFTER_SCHEDULE completes
ERROR	Execute this schedule if the AFTER_SCHEDULE completed with ERROR or FATAL status
FATAL	Execute this schedule if the AFTER_SCHEDULE completed with FATAL status
NONE	Do not execute this schedule (can be used as a placeholder)
SUCCESS	Execute this schedule if the AFTER_SCHEDULE completed with SUCCESS or INFORMATIONAL status

---

---

<b>WARNING</b>	Execute this schedule if the AFTER_SCHEDULE completed with WARNING, ERROR or FATAL status
----------------	---

---

If an /AFTER\_SCHEDULE name is specified, the default WHEN option is ALL.

/NOAFTER\_SCHEDULE removes the dependency to execute after another schedule.

**/COMMAND="string"**

The /COMMAND qualifier specifies the command to submit to DCL when the schedule executes. Specify a valid DCL command line in quotes. This qualifier must be entered for the schedule to do any useful work.

**/DATES=(date[,...])**

**/NODATES (D)**

The /DATES qualifier specifies on which days of the month you wish the schedule to execute. Valid values are 1-31. You can enter a date, a list of dates, a range of dates and a list of ranges (e.g. 1-7, 15-21). /NODATES indicates that all dates are valid (1-31) and other criteria are used for scheduling. You can use the /ADD, /REMOVE and /REPLACE qualifiers to manipulate the dates list.

The default is /NODATES.

**/DAYS=(day[,...])**

**/NODAYS (D)**

The /DAYS qualifier specifies on which days of the week you wish the schedule to execute. Valid values use at least the first three letters of the days of the week (English only). You can enter a day, a list of days, a range of days and a list of ranges (e.g. MON-WED, SAT-SUN). /NODAYS indicates that all day are valid (MON-SUN) and other criteria are used for scheduling. You can use the /ADD, /REMOVE and /REPLACE qualifiers to manipulate the days list.

The default is /NODAYS.

**/DESCRIPTION="text"**

Comments about the object. If the text contains spaces, then it must be enclosed within quotation marks. The length of the description can range from 0 to 255 characters. To clear the description, specify "".

**/EXCLUDE=(date[,...])**

**/NOEXCLUDE (D)**

The /EXCLUDE qualifier specifies specific dates that you wish to exclude from execution, even though the other scheduling parameters would normally schedule on that date. This can be used to exclude scheduled operations on holidays for example. Enter one or more dates in OpenVMS date format (04-JUL-2001), separated by commas. You can specify dates up to 9 years in the future. /NOEXCLUDE removes all dates from the exclude list. You can use the /ADD, /REMOVE and /REPLACE qualifiers to manipulate the exclude list.

By default, no exclude dates are defined.

**/INCLUDE=(date[,...])**

**/NOINCLUDE (D)**

The /INCLUDE qualifier specifies specific dates that you wish to include for execution, even though the other scheduling parameters would normally not schedule on that date. This can be used to include operations on special days that require special attention. Enter one or more dates in OpenVMS date format (31-DEC-2001), separated by commas. You can specify dates up to 9 years in the future. /NOINCLUDE removes all dates from the include list. You can use the /ADD, /REMOVE and /REPLACE qualifiers to manipulate the include list.

By default, no include dates are defined.

**/INHERIT=schedule\_name**

This qualifier allows you to specify an existing schedule record from which the new object record inherits attribute values. MDMS supplies default values if you specify none. All attributes may be inherited with the exception of the following:

Schedule Name

**/MONTHS=(month[,...])**

**/NOMONTHS (D)**

The /MONTHS qualifier specifies on which months of the year you wish the schedule to execute. Valid values use at least the first three letters of the months of the year (English only). You can enter a month, a list of months, a range of months and a list of ranges (e.g. JAN-MAR, JUL-SEP). /NOMONTHS indicates that all months are valid (JAN-DEC) and other criteria are used for scheduling. You can use the /ADD, /REMOVE and /REPLACE qualifiers to manipulate the months list.

The default is /NOMONTHS.

**/OWNER\_NAME=node::username**

**/OWNER\_NAME=group::username**

**/NOOWNER\_NAME**

The /OWNER\_NAME qualifier specifies the owner of the object. Specify an owner as node::username or group::username. The node, group and username portions may be wildcarded.

The owner of an object has default access control of all allowances to the object, regardless of whether a specific access control entry is present or the state of the default domain access control. If you wish to remove this unconditional owner access, specify /NOOWNER\_NAME. Note that being an owner of an object overrides only the access control; the appropriate MDMS rights for a request on the object must still be honored.

By default, the owner of an object is the group\_name::username of the user entering the CREATE command on the object from a node in the group.

**/REMOVE**

The /REMOVE qualifier works in conjunction with certain qualifiers that accept lists and removes specified attributes from them.

**/REPLACE**

The /REPLACE qualifier works in conjunction with certain qualifiers that accept lists and replaces the existing attributes list with the specified list. By default, attributes are added to the existing list.

**/TIMES=(time[,...])**

**/NOTIMES (D)**

The /TIMES qualifier specifies the times of the day you wish the schedule to execute. Enter from 1 to 100 times in OpenVMS format, separated by commas. For example (08:00, 10:00, 12:00, 14:00, 16:00). /NOTIMES clears out all times and effectively disables the schedule from execution. /TIMES=00:00 signifies midnight.

By default, no times are set.



**Examples:**

```
$ MDMS CREATE SCHEDULE SCHED1 -  
_$/DAYS=SUN -  
_$/TIMES=20:00 -  
_$/COMMAND="@USER1:[SMITH]SCHED1.COM"
```

This command creates a schedule called SCHED1, that executes every Sunday at 20:00 by issuing the command procedure @USER1:[SMITH]SCHED1.COM.

```
$ MDMS CREATE SCHEDULE SCHED2 -  
_$/MONTHS=(JAN, APR, JUL, OCT) -  
_$/DATES=1  
_$/TIMES=00:00  
_$/COMMAND="MDMS DELETE SAVE *QUARTER*"
```

This command creates a schedule called SCHED2, that executes every three months on the first of January, April, July and October at midnight, and executes an MDMS command to delete saves.

---

## MDMS CREATE VOLUME

The MDMS CREATE VOLUME command creates a new volume definition in the MDMS volume database.

Equivalent STORAGE Command: STORAGE ADD VOLUME

### Format:

MDMS CREATE VOLUME [volume\_id [...]]

### Parameters

volume\_id

Specifies the volume ID of the volume to be added. The volume ID is the external label for the volume. Alternatively, a volume range, separated by a dash, may be specified. A volume range is a numeric range for up to the last three characters of the volume ID.

Example ranges are (ABC001-ABC250), (ABC120-ABC125).

The maximum length of the volume ID is 6 characters. Valid characters are " ,! , " , % - ? , A - Z , \_ " .

There is a limit of 1000 volumes in a valid range, and ranges must be numerically increasing.

Either a volume ID or a volume range must be entered in the command, unless the /VISION qualifier is used to automatically determine volume identifiers in a jukebox.

A comma-separated list of volume IDs or ranges may be entered.

### Description

The MDMS CREATE VOLUME command creates a new volume definition in the MDMS volume database. The external label will be used to track the volume in the database. The external label must match the on-tape internal volume label when the volume is initialized.

### Privileges

The request requires MDMS\_CREATE\_ALL, MDMS\_CREATE\_VOLUME or MDMS\_CREATE\_POOL.

If the user is creating a volume in a named pool to which he is authorized, MDMS\_CREATE\_POOL is sufficient. The /POOL qualifier must be specified. Otherwise the request requires MDMS\_CREATE\_VOLUME or MDMS\_CREATE\_ALL.

Several protected qualifiers also require MDMS\_SET\_PROTECTED since these attributes are normally managed by MDMS and should not be modified by the user, unless in case of recovery from abnormal situations.

### Restrictions

The /VISION qualifier is incompatible with the volume\_id parameter.

The /ADD, /REMOVE and /REPLACE qualifiers are mutually exclusive. If none are specified, attributes are added to list attributes by default.

The MDMS server will refuse this command if mutually exclusive qualifiers are used in the affirmative forms.

## Qualifiers

**/ACCESS\_CONTROL=(USERNAME=user\_id, ALLOW=(option[,...]))**

**/NOACCESS\_CONTROL (D)**

Access control allows the system administrator to restrict access to the object to all or specific users on specific nodes or groups. Select one or more of the following options to allow the operation:

- SET - Allows setting the attributes of the object
- SHOW - Allows showing the object
- DELETE - Allows deleting the object
- READ - Allows restoring data using the object
- WRITE - Allows saving data using the object
- EXECUTE - Allows execution operations using the object
- CONTROL - Allows setting the access control on the object

By default, access controls are added to the list of access controls - only one user can be added per command. However, you can remove users using the /REMOVE qualifier, and replace the entire access control using the /REPLACE qualifier. You can remove all access controls on the object by specifying /NOACCESS\_CONTROL.

The user\_id should be in the format node::username or group::username. Wildcards are supported in the node, group and username portions.

Access control restrictions are in addition to the normal MDMS rights allowing operations on an object. If no access control is defined, then the default domain access control is applied.

**/ACCOUNT="text"**

Defines the account name of the volume. The account name may be up to 31 characters. If it contains spaces, it must be enclosed in quotation marks. Specify "" to clear the account name.

This qualifier requires the right MDMS\_SET\_PROTECTED.

**/ADD (D)**

The /ADD qualifier works in conjunction with certain qualifiers that accept lists and adds the specified attributes to the existing list. This is the default when specifying list items.

**/ALLOCATED\_DATE=date**

**/NOALLOCATED\_DATE**

Specifies the date the volume was allocated. Normally this is set by MDMS. Specify a VMS absolute date and time. Specify /NOALLOCATED\_DATE to clear the allocated date.

This qualifier requires the right MDMS\_SET\_PROTECTED.

**/AVAILABLE**

The /AVAILABLE qualifier moves a volume from the UNAVAILABLE state to the state it was previously in prior to the UNAVAILABLE state. The volume may then be moved into the TRANSITION or FREE state if the scratch date and/or transition time have expired.

**/BLOCK\_FACTOR=number**

Specifies the block factor for the volume. The default is a block factor of zero.

**/BRAND="text"**

The media manufacturer. The maximum length of the brand name is 31 characters. If it contains spaces it must be contained in quotation marks. Specify "" to clear the brand.

**/CLEANED\_DATE=date**

**/NOCLEANED\_DATE**

This qualifier specifies the date the volume was last cleaned and is entered as a VMS absolute time. Specify /NOCLEANED\_DATE to clear the cleaned date. The default cleaned date is the date/time the volume was created.

**/CREATION\_DATE=date**

**/NOCREATION\_DATE**

The date the volume is created. This attribute is set by MDMS, but may be overridden if necessary. The default creation date is the date/time the volume was created.

This qualifier requires the right MDMS\_SET\_PROTECTED.

**/DEALLOCATED\_DATE=date**

**/NODEALLOCATED\_DATE**

This qualifier specifies the actual deallocation date for the volume. Specify a VMS absolute time. This date is normally set by MDMS.

This qualifier requires the right MDMS\_SET\_PROTECTED.

**/DESCRIPTION="text"**

Comments about the volume. If the text contains spaces, then it must be enclosed within quotation marks. The length of the description can range from 0 to 255 characters. Specify "" to clear the description.

**/DRIVE=drive\_name**

**/NODRIVE**

This is a protected field that should be modified only to recover on error. Use the LOAD and UNLOAD commands to manipulate the drive name under normal operation. This qualifier specifies the drive that the volume currently resides in or last resided in. This is normally set up by MDMS. To clear the drive, specify /NODRIVE.

This qualifier requires the right MDMS\_SET\_PROTECTED.

**/FORMAT=keyword**

Specifies the format of the tape. Possible values are:

ASCII

BACKUP

EBCDIC

NONE (D)

RMUBACKUP

**/FREED\_DATE=date**

**/NOFREED\_DATE**

Specifies the date the volume was last freed (i.e. put in the FREE state). Specify an OpenVMS absolute date and time. This is normally set up by MDMS. To clear the freed date, specify /NOFREED\_DATE.

This qualifier requires the right MDMS\_SET\_PROTECTED.

**/INHERIT=volume\_id**

This qualifier allows you to specify an existing volume record to inherit default attributes. The default is that MDMS supplies values you do not supply on creation. All attributes may be inherited with the exception of the following protected fields:

---

Account	Job name
Allocate date	Jukebox name
Accessed date	Magazine name
Available state	Owner
Creation date	Placement
Deallocated date	Slot
Drive name	State
Freed date	User name
Initialized date	Next and Previous Volumes (not settable)

---

**/INITIALIZED\_DATE=date****/NOINITIALIZED\_DATE**

Specifies the date the volume was last initialized. Specify a VMS absolute date and time. This is normally set up by MDMS. To clear the initialized date, specify **/NOINITIALIZED\_DATE**.

This qualifier requires the right MDMS\_SET\_PROTECTED.

**/IO\_ERROR\_COUNT=number**

This qualifier allows you to set the number of I/O errors on the volume. The default value is zero.

**/JOB\_NAME="text"**

This qualifier allows you to specify the last job that accessed the volume. The job name can be from 0 to 31 characters. If it contains spaces, it must be enclosed in quotation marks. Specify "" to clear the job name.

This qualifier requires the right MDMS\_SET\_PROTECTED.

**/JUKEBOX=jukebox\_name****/NOJUKEBOX**

This is a protected field that should be modified only to recover on error. Use the MOVE VOLUME command to manipulate the jukebox name under normal operation. This qualifier allows you to specify that the volume is currently residing or last resided in the specified jukebox.

The maximum length of a jukebox name is 31 characters.

Valid characters are "\$,-,0-9,A-Z,\_,a-z".

This is normally set up by MDMS. To clear the jukebox name, specify **/NOJUKEBOX**.

This qualifier requires the right MDMS\_SET\_PROTECTED.

**/LAST\_ACCESS\_DATE=date****/NOLAST\_ACCESS\_DATE**

Specifies the date the volume was last loaded by MDMS. Specify a VMS absolute date and time. This is normally set up by MDMS. To clear the last access date, specify

**/NOLAST\_ACCESS\_DATE**.

This qualifier requires the right MDMS\_SET\_PROTECTED.

**/MAGAZINE=magazine\_name**

**/NOMAGAZINE**

This is a protected field that should be modified only to recover on error. Use the MOVE VOLUME command to manipulate the magazine name under normal operation. This qualifier specifies the magazine name if the volume resides in a magazine. This is normally set up by MDMS. To clear the magazine name, specify /NOMAGAZINE.

This qualifier requires the right MDMS\_SET\_PROTECTED.

**/MEDIA\_TYPES=(media\_type[,...])**

**/NOMEDIA\_TYPES**

The media type qualifier allows you to add the media type(s) that the volume can support. Multiple media types are supported prior to the volume being initialized. After initialization, a volume can only support one media type. The /REMOVE or /REPLACE qualifiers can be used to remove or replace objects in the list, rather than adding them by default. To specify the volume supports no media types, enter /NOMEDIA\_TYPES. If a volume is created with no media types, the default media type from the domain record is used.

**/MOUNT\_COUNT=number**

Specifies the number of times the volume has been loaded by MDMS. Normally set up by MDMS. The default mount count is zero.

**/OFFSITE=(LOCATION=location][,[NO]DATE[=date])**

**/NOOFFSITE**

This qualifier specifies the date that the volume is to be taken offsite and the offsite location. The location field is required when using the /OFFSITE qualifier if no location has been previously specified. The LOCATION keyword cannot be negated and if specified must point to a valid location object. Specify a VMS absolute or delta time in the date field. The NODATE keyword may be used to remove the date. This has the effect of disabling the schedule for the volume, while retaining the location. To clear the offsite date specify /NOOFFSITE. If a volume is under magazine control, the /OFFSITE qualifier is not allowed. The volume uses the values in the magazine.

The default offsite location is that stored in the domain object.

**/ONSITE=(LOCATION=location][, [NO]DATE=date])**

**/NOONSITE**

This qualifier specifies the date that the volume is to be brought back onsite and the onsite location. The location field is required when using the /ONSITE qualifier if no location has been previously specified. The LOCATION keyword cannot be negated and if specified must point to a valid location object. Specify a VMS absolute or delta time in the date field. The NODATE keyword may be used to remove the date. This has the effect of disabling the schedule for the volume, while retaining the location. To clear the onsite date specify /NOONSITE. If a volume is under magazine control, the /ONSITE qualifier is not allowed. The volume uses the values in the magazine.

The default onsite location is that stored in the domain object.

**/OWNER\_NAME=node::username**

**/OWNER\_NAME=group::username**

**/NOOWNER\_NAME**

The /OWNER\_NAME qualifier specifies the owner of the object. Specify an owner as node::username or group::username. The node, group and username portions may be wild-carded.

The owner of an object has default access control of all allowances to the object, regardless of whether a specific access control entry is present or the state of the default domain access control. If you wish to remove this unconditional owner access, specify /NOOWNER\_NAME. Note that being an owner of an object overrides only the access control; the appropriate MDMS rights for a request on the object must still be honored.

By default, the owner of an object is the group\_name::username of the user entering the CREATE command on the object from a node in the group. However, once the volume is allocated, the node::user performing the allocation becomes the owner.

**/OWNER\_NAME\_UIC=uic**

**/NOOWNER\_NAME\_UIC**

This qualifier specifies the owner UIC of a volume. The owner UIC field must be a UIC in the format [USER] or [group, user]. This is normally set up by MDMS on allocate volume. To clear the owner UIC field, specify /NOOWNER\_NAME\_UIC.

This qualifier requires the right MDMS\_SET\_PROTECTED.

**/PLACEMENT=keyword**

This is a protected field that should be modified only to recover on error. Use the MOVE, LOAD or UNLOAD commands to manipulate the placement field under normal operation. This qualifier defines the current placement of the volume. This is normally managed by MDMS. Select one of the following options:

DRIVE

ONSITE

MAGAZINE

JUKEBOX

OFFSITE

MOVING

If a magazine name is specified on the /MAGAZINE qualifier, the volume placement can be in one of three states:

MAGAZINE

DRIVE

MOVING

During a MOVE, LOAD or UNLOAD, a volume's placement may be set to MOVING indicating that the volume is being moved. If a volume is in a magazine, it is set to MOVING when the volume is being loaded or unloaded to/from a drive.

This qualifier requires the right MDMS\_SET\_PROTECTED.

**/POOL=pool\_name**

**/NOPOOL (D)**

The pool in which the volume belongs. The maximum length of the pool name is 31 characters. Spaces are not allowed in the pool name. If no pool is specified, the volume is considered to be part of a scratch pool and can be allocated by any user.

**/PREINITIALIZED**

**/NOPREINITIALIZED (D)**

This qualifier specifies whether the volume has been initialized before creation. If /PREINITIALIZED is specified, the volume is placed in the FREE state rather than the UNINITIALIZED state.

**/PROTECTION=protection**

The protection code for the volume. Use the standard OpenVMS protection code format. This protection is written to volumes when initialized by MDMS. If not specified, the default protection from the domain record is used.

**/PURCHASED\_DATE=date**

**/NOPURCHASED\_DATE**

The date when the volume was purchased. The date should be specified as an OpenVMS absolute time. Specify /NOPURCHASED\_DATE to clear the purchased date. The default purchased date is the date/time that the volume was created.

**/RECLength=number**

This qualifier specifies the record length used on the volume. The default record length is zero.

**/RELEASE**

This qualifier puts the volume into the FREE state from the TRANSITION state. It is not valid to release an allocated volume.

**/REMOVE**

The /REMOVE qualifier works in conjunction with certain qualifiers that accept lists and removes specified attributes from them.

**/REPLACE**

The /REPLACE qualifier works in conjunction with certain qualifiers that accept lists and replaces the existing attributes list with the specified list. By default, attributes are added to the existing list.

**/RETAIN**

This qualifier puts the volume that is in the FREE state or TRANSITION state back into the allocated state with the former owner as the current owner.

**/SCRATCH\_DATE=date**

**/NOSCRATCH\_DATE**

The /SCRATCH\_DATE qualifier specifies the planned date to return the volume from the ALLOCATED state to either the TRANSITION state or the FREE state. Specify /NOSCRATCH\_DATE if the volume should not automatically transition from the ALLOCATED state.

**/SLOTS=(range[,...])**

**/NOSLOTS**

This qualifier specifies the jukebox or magazine slot that the volume currently resides in. Specify a number in the range of the jukebox or magazine. This is normally set up by MDMS. If the volume does not currently reside in a jukebox or magazine slot, specify /NOSLOTS.

If no volume ID is specified, or a volume range is specified, you can specify a slot range, and each volume will be placed in each slot in order. For a single volume, specify a single slot. This qualifier requires the right MDMS\_SET\_PROTECTED.



**/SPACES=(range[,...])**

**/NOSPACES**

This qualifier specifies the non-jukebox space in the specified location that the volume resides in. If the volume does not reside in a location space, specify /NOSPACES. Use a space range only when creating multiple volumes - each volume will be placed in each space in order. For a single volume, specify a single space.

**/STATE=keyword**

This is a protected field that should be modified only to recover on error. Use the ALLOCATE VOLUME or DEALLOCATE VOLUME command to manipulate the state field under normal operation. This qualifier allows you to modify the state of the volume. This is normally set up by MDMS and manual modification is not recommended. The keyword values are:

ALLOCATED

FREE

TRANSITION

UNAVAILABLE

UNINITIALIZED

This qualifier requires the right MDMS\_SET\_PROTECTED.

**/TIMES\_CLEANED=number**

This qualifier allows you to specify the number of times the volume has been cleaned. The default is zero.

**/TRANSITION\_TIME=delta\_time**

**/NOTTRANSITION\_TIME**

The /TRANSITION\_TIME qualifier specifies that the volume enters the TRANSITION state when the scratch date is reached, and is to stay in the TRANSITION state for the specified delta time. When the transition time has expired, the volume enters the FREE state. The /NOTTRANSITION\_TIME qualifier specifies that the volume enters the FREE state directly at the scratch date.

**/UNAVAILABLE**

Puts the volume in the UNAVAILABLE state. The previous state is retained for when the volume is made available again.

**/USER\_NAME=username**

**/NOUSER\_NAME**

Specifies the user for the volume. The username can be from 1-31 characters, and must reflect an authorized VMS username. To clear the username, enter /NOUSER\_NAME.

This qualifier requires the right MDMS\_SET\_PROTECTED.

**/VISION**

This qualifier specifies that volume IDs are to be read using a vision system in a suitably-equipped jukebox. No volume ID or volume range should be specified when using this qualifier. Valid only for MRD jukeboxes equipped with a VISION system.

## MDMS CREATE VOLUME

### Examples

```
$ MDMS CREATE VOLUME ABC001 /MEDIA=TK85K -  
/ONSITE=(LOCATION=SHELF2) /SPACES=1
```

This command creates a new volume definition for volume ABC001, of media type TK85K, which is stored in the location SHELF2, space 1.

```
$ MDMS CREATE VOLUME /INHERIT=DEF000 -  
/JUKEBOX=JUKE_1 /SLOTS=(0-50) /VISION
```

This command creates volume records for the newly imported volumes in slots 0-50 of the jukebox JUKE\_1, and the volume labels are generated using the vision system, and other attributes are inherited from volume DEF000.

```
$ MDMS CREATE VOLUME HS0001-HS0007 /INHERIT=TK85MG -  
/MAGAZINE=TX877A /SLOTS=(0-6)
```

This command creates seven volume records HS0001 HS0007, which are stored in magazine TX877A in slots 0-6 respectively, and other attributes are inherited from volume TK85MG.

---

## MDMS DEALLOCATE DRIVE

The MDMS DEALLOCATE DRIVE command deallocates a drive.

Equivalent STORAGE Command: None

### Format:

MDMS DEALLOCATE DRIVE drive\_name

### Parameters

drive\_name

Specifies the drive name to be deallocated. Specify a drive name or the logical name previously defined in the allocate.

The maximum length of the drive name or logical name is 31 characters. Valid characters are "\$,-,0-9,A-Z,\_,a-z".

### Description

The MDMS DEALLOCATE DRIVE command deallocates a drive. If a logical name is supplied for the drive\_name parameter, the logical name will be deassigned.

### Privileges

The request requires MDMS\_DEALLOCATE\_ALL or MDMS\_DEALLOCATE\_OWN.

### Restrictions

Can only be issued by the process that allocated the drive. If that process terminates, the drive is automatically deallocated.

### Qualifiers

None

### Examples

```
$ MDMS DEALLOCATE DRIVE MYDRIVE
```

This command deallocates the drive assigned to the MYDRIVE logical and deassigns the logical name MYDRIVE.

```
$ MDMS DEALLOCATE DRIVE $1$MUA1
```

This command deallocates drive \$1\$MUA1.

---

## MDMS DEALLOCATE VOLUME

The MDMS DEALLOCATE VOLUME command puts an allocated volume in either the transition state or the free state, depending on volume attributes, which can be overridden in the command.

Equivalent STORAGE Command: STORAGE DEALLOCATE

### Format:

MDMS DEALLOCATE VOLUME [volume\_id]

### Parameters

volume\_id

Specifies the volume ID of the volume to deallocate. This parameter is required except when the /SCHEDULE qualifier is given, in which case it must be omitted.

The maximum length of a volume ID is 6 characters. Valid characters are " ,! , % - ? , A - Z , \_ " .

### Description

The MDMS DEALLOCATE VOLUME command puts an allocated volume in either the TRANSITION state or the FREE state, depending on volume attributes, which can be overridden in the command.

In addition, this command will update the state of the volume to FREE if the deallocated date plus transition time has expired, or if you issue a DEALLOCATE VOLUME/STATE=FREE command. If you deallocate a volume set, the volume set relationships are invalidated, but can be recovered on a SET VOLUME/REATIN. The volume set relationships are deleted on the next allocation of the volume.

### Privileges

The request requires MDMS\_DEALLOCATE\_ALL or MDMS\_DEALLOCATE\_OWN.

If the user is deallocating a volume owned by him, MDMS\_DEALLOCATE\_OWN is sufficient. If the user is deallocating on behalf of another user with the /USER\_NAME qualifier, then the request requires MDMS\_DEALLOCATE\_ALL.

If the /STATE or /TRANSITION\_TIME qualifiers are entered, the user needs MDMS\_SET\_ALL if the /USER\_NAME qualifier is specified, or MDMS\_SET\_OWN or MDMS\_SET\_POOL.

### Restrictions

The /SCHEDULE and volume\_id parameters are incompatible.

The /SCHEDULE and /NOVOLSET qualifiers are incompatible.

The MDMS server will refuse this command if mutually exclusive qualifiers are used in the affirmative forms.

## Qualifiers

### **/SCHEDULE**

The /SCHEDULE qualifier selects volumes whose scratch dates or freed dates have expired, and moves them into the TRANSITION state or FREE state respectively. No volume\_id parameter is allowed when this qualifier is given.

### **/STATE=state**

Specifies the state in which to place the deallocated volume. The valid state values are FREE and TRANSITION. If the state is not specified and no transition time is specified, the deallocation state in the domain record is used.

This qualifier requires the right MDMS\_SET\_OWN or MDMS\_SET\_POOL, or MDMS\_SET\_ALL if the /USER\_NAME qualifier is used.

### **/TRANSITION\_TIME=delta\_time**

#### **/NOTTRANSITION\_TIME**

Modifies the amount of time, as a delta time, that a volume will remain in the TRANSITION state before moving to the FREE state. Use the standard OpenVMS delta time format to specify a delta time for the transition duration. If not specified, the existing transition time in the volume record is used, and if none, the default transition time in the domain record is used.

This qualifier requires the right MDMS\_SET\_OWN or MDMS\_SET\_POOL, or MDMS\_SET\_ALL if the /USER\_NAME qualifier is used.

### **/USER\_NAME=username**

Deallocate a volume which is owned by the specified user. The maximum length of the username is 31 characters.

This qualifier requires the right MDMS\_DEALLOCATE\_ALL.

### **/VOLSET (D)**

#### **/NOVOLSET**

Indicates that the entire volume set which contains the volume ID is to be deallocated. The default of /VOLSET deallocates all volumes in the volume set. If /NOVOLSET is specified, the specified volume is deallocated. This qualifier is ignored if the volume is not in a set. You cannot deallocate the first volume in a volume set using /NOVOLSET.

## MDMS DEALLOCATE VOLUME

### Examples

```
$ MDMS DEALLOCATE VOLUME VOL008 /NOVOLSET /STATE=FREE
```

This command deallocates volume VOL008 for the current user and places the volume in the FREE state.

```
$ MDMS DEALLOCATE VOLUME ACF342 /USER_NAME=SMITH
```

This command deallocates volume ACF342 which was allocated to user SMITH.

```
$ MDMS DEALLOCATE VOLUME VOL002 /VOLSET
```

This command deallocates all volumes in the set that contains volume VOL002 for the current user. The volumes are also unbound from the volume set.

```
$ MDMS DEALLOCATE VOLUME VOL3 /NOVOLSET
```

A volume set contains volumes VOL1, VOL2, VOL3, VOL4, VOL5. This command deallocates volume VOL2 for the current user and leaves two volume sets: VOL1, VOL2 and VOL4, VOL5.

```
$ MDMS DEALLOCATE VOLUME/SCHEDULE
```

Deallocates all volumes whose scratch date has expired. Also, moves volumes into the FREE state if the freed date has expired.

---

## MDMS DELETE DRIVE

The MDMS DELETE DRIVE command deletes a specified drive definition from the MDMS database. The drive must be deallocated prior to being deleted.

Equivalent STORAGE Command: None

### Format:

```
MDMS DELETE DRIVE drive_name [...]
```

### Parameters

drive\_name

Specifies the name of the drive.

The maximum length of the drive name is 31 characters.

Valid characters are "\$,-,0-9,A-Z,\_,a-z".

A comma-separated list of drive names may be entered.

### Description

The MDMS DELETE DRIVE command deletes a specified drive definition from the MDMS database.

### Privileges

The request requires MDMS\_DELETE\_ALL.

### Restrictions

None

### Qualifiers

None

### Example

```
$ MDMS DELETE DRIVE $1$MUA5
```

This command deletes the drive definition for \$1\$MUA5.

---

## MDMS DELETE GROUP

The MDMS DELETE GROUP command deletes a specified group definition from the MDMS database.

Equivalent STORAGE Command: None

### Format:

MDMS DELETE GROUP group\_name [...]

### Parameters

group\_name

Specifies the name of the group.

The maximum length of the group name is 31 characters.

Valid characters are "\$,-,0-9,A-Z,\_,a-z".

A comma-separated list of group names may be entered.

### Description

The MDMS DELETE GROUP command deletes a specified group definition from the MDMS database.

### Privileges

The request requires MDMS\_DELETE\_ALL.

### Restrictions

None

### Qualifiers

None

### Example

```
$ MDMS DELETE GROUP HOSER
```

This command deletes the group definition for HOSER.



---

## MDMS DELETE JUKEBOX

The MDMS DELETE JUKEBOX command deletes a specified jukebox definition from the MDMS database.

Before deleting a jukebox, it is highly recommended that any drives defined as being in the jukebox are also deleted (or modified), and volumes and magazines are moved out of the jukebox.

Equivalent STORAGE Command: None

### Format:

```
MDMS DELETE JUKEBOX jukebox_name [...]
```

### Parameters

jukebox\_name

Specifies the name of the jukebox.

The maximum length of the jukebox name is 31 characters.

Valid characters are "\$,-,0-9,A-Z,\_,a-z".

A comma-separated list of jukebox names may be entered.

### Description

The MDMS DELETE JUKEBOX command deletes a specified jukebox definition from the MDMS database. Before deleting a jukebox, it is highly recommended that any drives defined as being in the jukebox are also deleted (or modified), and volumes and magazines are moved out of the jukebox.

### Privileges

The request requires MDMS\_DELETE\_ALL.

### Restrictions

None

### Qualifiers

None

### Example

```
$ MDMS DELETE JUKEBOX JUKE_1
```

This command deletes the definition for jukebox JUKE\_1.

---

## MDMS DELETE LOCATION

The MDMS DELETE LOCATION command deletes a specified location definition from the MDMS database.

Equivalent STORAGE Command: None

### Format:

MDMS DELETE LOCATION location [...]

### Parameters

location

Specifies the name of the location.

The maximum length of the location is 31 characters. Valid characters are "\$,-,0-9,A-Z,\_,a-z".

A comma-separated list of locations may be entered.

### Description

The MDMS DELETE LOCATION command deletes a specified location definition from the MDMS database.

### Privileges

The request requires MDMS\_DELETE\_ALL.

### Restrictions

None

### Qualifiers

None

### Example

```
$ MDMS DELETE LOCATION ROOM_1
```

This command deletes the location definition for ROOM\_1.

---

## MDMS DELETE MAGAZINE

The MDMS DELETE MAGAZINE command deletes a specified magazine from the MDMS database.

Equivalent STORAGE Command: STORAGE REMOVE MAGAZINE

### Format:

MDMS DELETE MAGAZINE magazine\_name [...]

### Parameters

magazine\_name

Specifies the name of the magazine.

The maximum length of the magazine name is 31 characters.

Valid characters are "\$,-,0-9,A-Z,\_,a-z".

A comma-separated list of magazine names may be entered.

### Description

The MDMS DELETE MAGAZINE command deletes a specified magazine definition from the MDMS database.

### Privileges

The request requires MDMS\_DELETE\_ALL.

### Restrictions

None

### Qualifiers

None

### Example

```
$ MDMS DELETE MAGAZINE MAG001
```

This command deletes the magazine MAG001.

---

## MDMS DELETE MEDIA\_TYPE

The MDMS DELETE MEDIA\_TYPE command deletes a specified media type definition from the MDMS database.

Equivalent STORAGE Command: None

### Format:

MDMS DELETE MEDIA\_TYPE media\_type [...]

### Parameters

media\_type

Specifies the name of the media type.

The maximum length of the media type name is 31 characters.

Valid characters are "\$,-,0-9,A-Z,\_,a-z".

A comma-separated list of media types may be entered.

### Description

The MDMS DELETE MEDIA\_TYPE command deletes a specified media type definition from the MDMS database.

### Privileges

The request requires MDMS\_DELETE\_ALL.

### Restrictions

None

### Qualifiers

None

### Example

```
$ MDMS DELETE MEDIA_TYPE TK87K
```

This command deletes the definition for media type TK87K.

---

## MDMS DELETE NODE

The MDMS DELETE NODE command deletes a specified node definition from the MDMS database.

Equivalent STORAGE Command: None

### Format:

```
MDMS DELETE NODE node_name [...]
```

### Parameters

node\_name

Specifies the name of the node.

The maximum length of the node name is 31 characters. Valid characters are "\$,-,0-9,A-Z,\_,a-z".

A comma-separated list of node names may be entered.

### Description

The MDMS DELETE NODE command deletes a specified node definition from the MDMS database.

### Privileges

The request requires MDMS\_DELETE\_ALL.

### Restrictions

None

### Qualifiers

None

### Example

```
$ MDMS DELETE NODE FUDGE
```

This command deletes the node name FUDGE.

---

## MDMS DELETE POOL

The MDMS DELETE POOL command deletes a specified pool definition from the MDMS database.

Equivalent STORAGE Command: None

### Format:

MDMS DELETE POOL pool\_name [...]

### Parameters

pool\_name

Specifies the name of the volume pool.

The maximum length of the pool name is 31 characters. Valid characters are "\$,-,0-9,A-Z,\_,a-z".

A comma-separated list of pool names may be entered.

### Description

The MDMS DELETE POOL command deletes a specified pool definition from the MDMS database.

### Privileges

The request requires MDMS\_DELETE\_ALL.

### Restrictions

None

### Qualifiers

None

### Example

```
$ MDMS DELETE POOL SLSPOOL
```

This command deletes pool SLSPOOL.

---

## MDMS DELETE SCHEDULE

The MDMS DELETE SCHEDULE command deletes a specified schedule definition from the MDMS database.

Equivalent STORAGE Command: None

### Format:

MDMS DELETE SCHEDULE schedule\_name [...]

### Parameters

schedule\_name

Specifies the name of the SCHEDULE.

The maximum length of the schedule name is 63 characters.

Valid characters are "\$,-,0-9,A-Z,\_,a-z".

A comma-separated list of schedule names may be entered.

### Description

The MDMS DELETE SCHEDULE command deletes a specified schedule definition from the MDMS database.

### Privileges

The request requires MDMS\_DELETE\_ALL.

### Restrictions

None

### Qualifiers

#### **/OVERRIDE**

The /OVERRIDE qualifier allows you to delete the schedule even with outstanding references to it. This is not recommended unless you are sure that the schedule will no longer be needed.

### Example

```
$ MDMS DELETE SCHEDULE RICHS_SCHEDULE
```

This command deletes the schedule definition for RICHS\_SCHEDULE.

---

## MDMS DELETE VOLUME

The MDMS DELETE VOLUME command deletes a specified volume definition from the MDMS volume database.

Equivalent STORAGE Command: STORAGE REMOVE VOLUME

### Format:

MDMS DELETE VOLUME volume\_id [...]

### Parameters

volume\_id

Specifies the name of the volume. Alternatively, a volume range, separated by a dash, may be specified. A volume range is a numeric range for up to the last three characters of the volume ID. Example ranges are (ABC001-ABC250), (ABC120-ABC125).

The maximum length of the volume ID is 6 characters. Valid characters are " ,!,"% -?,A-Z,\_".

A comma-separated list of volume IDs or ranges may be entered.

### Description

The MDMS DELETE VOLUME command deletes a specified volume definition from the MDMS volume database. Volumes must be in the UNINITIALIZED or FREE states prior to being deleted.

### Privileges

The request requires MDMS\_DELETE\_ALL, MDMS\_DELETE\_VOLUME or MDMS\_DELETE\_POOL.

If the user is deleting a volume that is in a pool to which he is authorized, MDMS\_DELETE\_POOL is sufficient. All other volume deletes require MDMS\_DELETE\_VOLUME or MDMS\_DELETE\_ALL.

### Restrictions

None

### Qualifiers

None

### Example

```
$ MDMS DELETE VOLUME ABC001-ABC099
```

This command deletes the records for volumes ABC001 to ABC099.



---

## MDMS INITIALIZE VOLUME

The MDMS INITIALIZE VOLUME command initializes a volume or a range of volumes by writing the volume ID to tape as the tape label.

Equivalent STORAGE Command: None

### Format:

MDMS INITIALIZE VOLUME [volume\_id]

### Parameters

volume\_id

Specifies the name(s) of the volume(s) to initialize. Alternatively, a volume range, separated by a dash, may be specified. A volume range is a numeric range for up to the last three characters of the volume ID. Example ranges are (ABC001-ABC250), (ABC120-ABC125).

There is a maximum of 1000 volumes in a range.

The maximum length of the volume ID is 6 characters. Valid characters are " ,!,"%-,A-Z,\_".

This parameter is required except when /JUKEBOX and /SLOTS are specified. The /JUKEBOX and /SLOTS option is valid only for MRD-controlled jukeboxes equipped with a vision system. For all other jukeboxes, the volume\_id parameter is required.

### Description

The MDMS INITIALIZE VOLUME command initializes a volume or a range of volumes by writing the volume ID to tape as the tape label. The volume records must be created prior to the initialization. In addition, the volumes must be in the FREE or UNINITIALIZED states in order to be initialized.

If the volume(s) are contained in a jukebox, MDMS automatically loads and unloads the volumes. Otherwise, operator assistance is required.

In order to initialize volumes, the MDMS database server allocates a free drive that can handle the volumes. All volumes in a single command must support the specified media type (or the media type already in the volume record). Do not allocate a drive or load a volume for the initialization - this must be done by MDMS.

Each initialize volume command utilizes a single tape drive for all volumes. To utilize multiple drives, issue multiple initialize volume commands specifying a different volume or slot range, and use the /NOWAIT qualifier.

By default, volumes that have a different label than expected and have data written on them are not initialized. However, you can override this check and allow the initialization with the /OVERRIDE qualifier. Note that volumes with labels that are flagged as allocated or in the transition state cannot be initialized under any circumstances.

### Privileges

The request requires MDMS\_INITIALIZE\_ALL or MDMS\_INITIALIZE\_POOL.

If the user is initializing a volume that is in a pool to which he is authorized, MDMS\_INITIALIZE\_POOL is sufficient. All other initializations require MDMS\_INITIALIZE\_ALL.

MDMS\_ASSIST is required unless /NOASSIST is specified.

## Restrictions

The /JUKEBOX and /SLOTS qualifiers must be used together, and are incompatible with the volume\_id parameter.

The MDMS server will refuse this command if mutually exclusive qualifiers are used in the affirmative forms.

## Qualifiers

### **/ASSIST (D)**

#### **/NOASSIST**

The default /ASSIST qualifier is used to output an operator message if the volume requires operator intervention. You can use /ASSIST even if no assistance is needed - MDMS will perform the operation automatically if it can. If you specify /NOASSIST and assistance is needed, the operation fails.

This qualifier requires the right MDMS\_ASSIST.

### **/DRIVE=drive\_name**

This qualifier indicates the volume to be initialized is already in the specified drive and should not be loaded or unloaded. This supports an initialize in place. Only one volume may be specified with this option.

### **/JUKEBOX=jukebox\_name**

This qualifier indicates the volumes to be initialized are in the specified jukebox. A slot range should also be specified with this qualifier. Valid only for MRD jukeboxes equipped with a vision system.

### **/MEDIA\_TYPE=media\_type**

This qualifier modifies the volume record to support only this media type. The volume must already support this media type, but may support others as well. After the initialization, the other media types are removed.

### **/OVERWRITE**

#### **/NOOVERWRITE (D)**

The /OVERWRITE qualifier allows initialization if the requested volume label is different from the current label, and the volume has already been initialized and contains data. The default /NOOVERWRITE qualifier indicates that the volume should not be initialized if the label does not match and the volume contains data. Under no circumstances will the volume be initialized if the volume (according to the current label) is allocated or in the transition state.

### **/REPLY=symbol**

The name of the symbol to receive the operator's reply when operator intervention is needed. The symbol will contain the operator reply to the DCL REPLY/TO or DCL REPLY/ABORT commands.

### **/SLOTS=(range[,...])**

This qualifier specifies the slots containing volumes to initialize, when used with the /JUKEBOX qualifier. Valid only for MRD jukeboxes equipped with a vision system.

### **/USER=username**

This qualifier specifies that the volume should be initialized on behalf of this user. The user's UIC is written to the volume label. If the volume is allocated to a user, the allocated user must match the requested user. If /USER is not specified, then the username of the calling process is used by default.

This qualifier requires the right MDMS\_INITIALIZE\_ALL.

**/WAIT (D)****/NOWAIT**

The **/NOWAIT** qualifier returns an informational message indicating that the initialize is being queued. The **/WAIT** qualifier causes the MDMS INITIALIZE VOLUME command to wait until the initialization is complete.

**Examples**

```
$ MDMS INITIALIZE VOLUME WOR001 /MEDIA_TYPE=TK85K
```

This command loads, initializes and unloads volume WOR001 and modifies the volume records to set media type TK85K only.

```
$ MDMS INITIALIZE VOLUME/JUKEBOX=JUKE_1 -
```

```
/SLOTS=(0-10)/OVERWRITE
```

This command loads, initializes and unloads the volumes contained in slots 0-10 of jukebox JUKE\_1, and allows initialization even if the tapes have unexpected labels and have already been written.

```
$ MDMS INITIALIZE VOLUME ABC001-ABC050
```

This command loads, initializes and unloads all 50 volumes in the range ABC001 to ABC050

---

## MDMS INVENTORY JUKEBOX

The MDMS INVENTORY JUKEBOX command verifies the contents of a jukebox. This command is used for both MRD and DCSC controlled jukeboxes.

Equivalent STORAGE Command: STORAGE INVENTORY JUKEBOX and STORAGE INVENTORY ACS

### Format:

MDMS INVENTORY JUKEBOX jukebox\_name

### Parameters

jukebox\_name

Specifies the name of the jukebox.

The maximum length of the jukebox name is 31 characters.

Valid characters are "\$,-,0-9,A-Z,\_,a-z".

### Description

The MDMS INVENTORY JUKEBOX command verifies the contents of a jukebox. This command is used for both MRD and DCSC controlled jukeboxes.

For MRD controlled jukeboxes:

- Both the volume range and slot range options are supported, but only one per command.
- If a volume range is specified, all volumes in the range will be inventoried and their slot positions verified and updated as needed. If the volume is not detected in the jukebox, the volume's placement will be according to the /MISSING qualifier (or placed in the onsite location by default). The volume range option is only available on jukeboxes supporting a VISION system, and /VISION is specified or defaulted in the command.
- If a slot range is specified, all volumes found in the specified slots are verified and the slot position updated as needed. No other volume records are modified.
- For systems with vision systems, the inventory is performed by querying the jukebox firmware as to slot/drive/port contents. A physical inventory is not performed. Volumes detected in ports will be considered missing (not in the jukebox). It is recommended that all ports are empty during an inventory.
- For small, single-magazine jukeboxes, like the TZ877, each volume is loaded, mounted, examined and unloaded in order to read the volume label. This can be very time consuming depending on the hardware involved. Inventoring a 7-slot TZ877 jukebox takes about 20 minutes.
- Full or partial inventories are supported.

For DCSC controlled jukeboxes:

- Only the volume range option is supported.
- The inventory simply checks to see if the specified volumes are in the jukebox. This is slightly different behavior than the inventory performed on an MRD-controlled jukebox. If a volume is not in the jukebox, the volume's placement will be according to the /MISSING qualifier (or placed in the onsite location by default).

**Privileges**

The request requires MDMS\_INVENTORY\_ALL.

**Restrictions**

/SLOTS and /VOLUME\_RANGE are mutually exclusive.

/SLOTS is unsupported for DCSC jukeboxes - /VOLUME\_RANGE is required.

The MDMS server will refuse this command if mutually exclusive qualifiers are used in the affirmative forms.

**Qualifiers****/CREATE**

This qualifier specifies that any volume that is found that is not defined is created using either the /INHERIT, and/or /MEDIA\_TYPES qualifiers.

**/INHERIT=volume\_id**

When creating volumes, this qualifier specifies a volume\_id from which to inherit attributes. The same attributes can be inherited as for the CREATE VOLUME command.

**/MEDIA\_TYPE=media\_type**

When creating volumes, this qualifier specifies the media type to be used for the volume records.

**/MISSING=keyword**

This qualifier defines the action to be taken when a volume is missing in the jukebox, when it was defined to be either in a drive or slot in the jukebox. The allowable actions are:

DELETE - Delete the volume record (only valid if volume is in the FREE state)

IGNORE - Do nothing

MOVE (D) - Move the volume to its apparent location - either the onsite location or another jukebox

**/PREINITIALIZED****/NOPREINITIALIZED (D)**

When creating volumes, the volumes are placed in the FREE state if /PREINITIALIZED is specified. The default is to place them in the UNINITIALIZED state.

**/SLOTS=(range[,...])**

The slots or range of slots to inventory within the jukebox.

**/VISION (D)****/NOVISION**

Specifies whether the inventory should be performed with the Vision system on jukeboxes so equipped. A vision inventory reads the internal memory of the jukebox to perform the inventory; a physical inventory is not performed. If /NOVISION is specified, inventory is performed by loading and mounting the volumes. Applicable to MRD jukeboxes equipped with a VISIONsystem only.

**/VOLUME\_RANGE=(start\_id-end\_id)**

This qualifier specifies the volume range to inventory. A volume range is a numeric range for up to the last three characters of the volume ID. Example ranges are (ABC001-ABC250), (ABC120-ABC125).

## MDMS INVENTORY JUKEBOX

### **/WAIT (D)**

### **/NOWAIT**

The **/NOWAIT** qualifier returns an informational message indicating that the inventory is being queued. The **/WAIT** qualifier causes the MDMS INVENTORY JUKEBOX command to wait until the inventory is complete.

## **Examples**

```
$ MDMS INVENTORY JUKEBOX TESTJUKE /CREATE -
```

```
/MEDIA_TYPE=TK85K /NOWAIT
```

This command inventories the entire jukebox TESTJUKE. Any volumes not in the MDMS volume database will be created with a media type of TK85K. The command prompt is returned after the inventory is queued.

```
$ MDMS INVENTORY JUKEBOX TESTJUKE /CREATE -
```

```
/MEDIA_TYPE=TK85K /SLOTS=(0-9)
```

This command inventories the first 10 slots of jukebox TESTJUKE. Any volumes not in the MDMS volume database will be created with a media type of TK85K.

```
$ MDMS INVENTORY JUKEBOX SILO_JUKE -
```

```
/VOLUME_RANGE=(SQ0800-SQ0900) /MISSING=MOVE /CREATE
```

This command verifies that volumes SQ0800 through SQ0900 exist in the StorageTek silo. Any volumes found in the silo, but not in the MDMS volume database will be created, with default values. Any volumes found in the MDMS database but not in the silo will be marked in the onsite location of the volume

---

## MDMS LOAD DRIVE

The MDMS LOAD DRIVE command loads a volume into the specified drive.

Equivalent STORAGE Command: None

### Format:

MDMS LOAD DRIVE drive\_name

### Parameters

drive\_name

Specify a drive name, or a logical name.

The maximum length of the drive name or logical name is 31 characters. Valid characters are "\$,-,0-9,A-Z,\_,a-z".

### Description

The MDMS LOAD DRIVE command loads a volume into the specified drive. The command is used primarily for scratch loads when the volume ID is not important - only that a FREE volume is loaded into the drive. The volume may or may not already be created in the MDMS volume database. If it is not created, it may be created using the attributes using the /INHERIT or /MEDIA\_TYPE qualifiers.

### Privileges

The request requires MDMS\_LOAD\_ALL or MDMS\_LOAD\_SCRATCH.

MDMS\_ASSIST is also required unless /NOASSIST is specified.

The /CREATE qualifier requires MDMS\_CREATE\_POOL if the /POOL qualifier is specified and the user is authorized to the pool. All other uses of the /CREATE qualifier require MDMS\_CREATE\_ALL.

### Restrictions

The /NOCHECK qualifier cannot be used with /MOUNT.

/MOUNT and /NOWAIT are mutually exclusive.

The MDMS server will refuse this command if mutually exclusive qualifiers are used in the affirmative forms.

### Qualifiers

**/ASSIST (D)**

**/NOASSIST**

The default /ASSIST qualifier is used to output an operator message if the volume requires operator intervention. You can use /ASSIST even if no assistance is needed - MDMS will perform the operation automatically if it can. If you specify /NOASSIST and assistance is needed, the operation fails.

This qualifier requires the right MDMS\_ASSIST

**/CREATE**

This qualifier specifies that a volume that is not in the MDMS volume database is created in the database. You can also use the /INHERIT or /MEDIA\_TYPE qualifiers to specify the volume's attributes.

This qualifier requires the right MDMS\_CREATE\_ALL or MDMS\_CREATE\_POOL if the volume is in a pool owned by the user.

**/INHERIT=volume\_id**

If /CREATE is specified, this qualifier allows you to specify an existing volume record to inherit default attributes. The default is that MDMS supplies attribute values you do not on creation.

**/MEDIA\_TYPE=media\_type**

Specifies the media type assigned to a volume being created with the /CREATE qualifier. The media type must be supported by the drive.

**/MESSAGE="message"**

This qualifier allows the application to specify directions to the operator in the OPCOM message associated with the load request. By default MDMS puts out an appropriate message.

**/MOUNT["mount\_qualifiers"]**

Will cause MDMS to issue a DCL MOUNT command once the volume is loaded and the label verified. DCL MOUNT qualifiers can also be provided by specifying the qualifiers in a quoted string. By default, the volume is not mounted.

**/POOL=pool\_name**

Specifies the volume pool the volume is to be placed in. This qualifier must be specified if the /CREATE command is specified and the user only has MDMS\_CREATE\_POOL privilege, unless a pool is specified in a volume used with /INHERIT.

**/REPLY=symbol**

The name of the symbol to receive the operator's reply when operator intervention is needed. The symbol will contain the operator reply to the DCL REPLY/TO or DCL REPLY/ABORT commands. The maximum length of a symbol name is 31 characters.

**/WAIT (D)****/NOWAIT**

The /NOWAIT qualifier returns an informational message indicating that the load is being queued. The /WAIT qualifier causes the MDMS LOAD DRIVE command to wait until a volume is loaded.

**/WRITE (D)****/NOWRITE**

The default /WRITE qualifier specifies that the volume must be write-enabled when loaded, otherwise a warning is issued after the load. The /NOWRITE qualifier specifies that a volume can be successfully loaded with the write protect set on or off.

**Example**

```
$ MDMS LOAD DRIVE $1$MUA1: /MOUNT="/NOUNLOAD /FOREIGN"
```

This command loads a volume into drive \$1\$MUA1 and mounts the volume with "/NOUNLOAD and /FOREIGN" qualifiers. The quotes are required.



---

## MDMS LOAD VOLUME

The MDMS LOAD VOLUME command loads the specified volume into a drive.

Equivalent STORAGE Command: STORAGE LOAD

### Format:

MDMS LOAD VOLUME volume\_id

### Parameters

volume\_id

The maximum length of the volume ID is 6 characters. Valid characters are " ,! , " , % - ? , A - Z , \_ " .

This parameter is required.

### Description

The MDMS LOAD VOLUME command loads the specified volume into a drive. The volume must already be created in the MDMS database.

If a drive was allocated based on the volume ID, then that drive is chosen for the load. You can also specify a drive on this command, and the volume will be loaded into that drive. The LOAD VOLUME command does not allocate the drive to the requesting process

### Privileges

The request requires MDMS\_LOAD\_ALL, MDMS\_LOAD\_POOL or MDMS\_LOAD\_OWN.

If the user is loading a volume for which he is the owner, MDMS\_LOAD\_OWN is sufficient. If the user is loading a volume in a pool to which he is authorized, MDMS\_LOAD\_POOL is required. For all other loads, MDMS\_LOAD\_ALL is required.

MDMS\_ASSIST is also required unless /NOASSIST is specified.

The /MOVE qualifier requires MDMS\_MOVE\_OWN, MDMS\_MOVE\_POOL or MDMS\_MOVE\_ALL according to the MOVE command.

### Restrictions

The /NOCHECK qualifier cannot be used with /MOUNT.

The /MOUNT and /WAIT qualifiers are mutually exclusive.

The MDMS server will refuse this command if mutually exclusive qualifiers are used in the affirmative forms.

### Qualifiers

**/ASSIST (D)**

**/NOASSIST**

The default /ASSIST qualifier is used to output an operator message if the volume requires operator intervention. You can use /ASSIST even if no assistance is needed - MDMS will perform the operation automatically if it can. If you specify /NOASSIST and assistance is needed, the operation fails.

This qualifier requires the right MDMS\_ASSIST.

**/CHECK (D)**

**/NOCHECK**

Compares the physical ANSI label on the tape against the label for the volume ID. If the physical ANSI label does not match the label for the volume ID, operator intervention is required to resolve the conflict. The default is /CHECK.

**/DRIVE=drive\_name**

This qualifier specifies the name of the drive in which to load the volume. This qualifier is required unless a drive has been allocated for this volume (i.e. ALLOCATE DRIVE/VOLUME=volume\_id).

**/MESSAGE="message"**

This qualifier allows the application to specify directions to the operator in the OPCOM message associated with the load request.

**/MOUNT["mount\_qualifiers"]**

Will cause MDMS to issue a DCL MOUNT command once the volume is loaded and the label verified. DCL MOUNT qualifiers can also be provided by specifying the qualifiers in a quoted string. By default, the volume is not mounted.

**/MOVE (D)**

**/NOMOVE**

The default /MOVE qualifier, used with /ASSIST, allows the load request to generate a move request to move a volume from a remote location to the drive or associated jukebox. The move will generate an OPCOM move request which that must be satisfied before the load request can continue. If /NOMOVE is specified and a move is required, the load request fails with an error.

**/REPLY=symbol**

The name of the symbol to receive the operator's reply when operator intervention is needed. The symbol will contain the operator reply to the DCL REPLY/TO or DCL REPLY/ABORT commands. The maximum length of a symbol name is 31 characters. Used with /ASSIST only.

**/WAIT (D)**

**/NOWAIT**

The /NOWAIT qualifier returns an informational message indicating that the load is being queued. The /WAIT qualifier causes the MDMS LOAD VOLUME command to wait until the volume is loaded.

**/WRITE (D)**

**/NOWRITE**

The default /WRITE qualifier specifies that the volume must be write-enabled when loaded, otherwise a warning is issued after the load. The /NOWRITE qualifier specifies that a volume can be successfully loaded with the write protect set on or off.

## Examples

```
$ MDMS LOAD VOLUME ABC010 /MOUNT="/NOUNLOAD"
```

This command loads volume ABC010 into a previously- allocated drive and mounts the volume with "/NOUNLOAD" qualifier and verifies the volume label. The quotes are required.

```
$ MDMS LOAD VOLUME ABC020 /NOCHECK /DRIVE=$1$MUA1:
```

This command loads volume ABC020 into drive\$1\$MUA1: and does not check the on-tape volume label.

```
$ MDMS LOAD VOLUME ABC020 /NOWAIT
```

This command loads volume ABC020 into an allocated drive, checks the on-tape volume label, but does not wait for the load to complete before returning the command prompt.

---

## MDMS MOVE MAGAZINE

The MDMS MOVE MAGAZINE command moves a magazine from one location to another. The magazine must be created prior to the move.

Equivalent STORAGE Commands: STORAGE EXPORT MAGAZINE, STORAGE IMPORT MAGAZINE

### Format:

MDMS MOVE MAGAZINE magazine\_name [destination]

### Parameters

magazine\_name

Defines the name of the magazine to move. Only one magazine may be moved per command. If the /SCHEDULE qualifier is specified, the magazine\_name may be the wildcard \*, which means all scheduled magazines.

The maximum length of the magazine name is 31 characters. Valid characters are "\$,-,0-9,A-Z,\_,a-z".

destination

Specifies the name of the destination object. The destination object can be a location or a jukebox. The destination is optional when the /SCHEDULE is entered, but required for commands when /SCHEDULE is not entered. The default destination is the appropriate offsite or onsite location specified in the magazine object.

### Description

The MDMS MOVE MAGAZINE moves a magazine from one location to another. Magazines can be moved between outside locations, and from an outside location to and from a jukebox.

When /ASSIST is specified, a series of OPCOM messages may be displayed asking the operator to move magazines between locations, or in and out of the jukebox. No slots or spaces are displayed in the OPCOM message, but the operator should issue SHOW MAGAZINE/FULL for the associated magazine(s) to determine slots, spaces or positions involved in the move. These are shown in the placement field.

### Privileges

The request requires MDMS\_MOVE\_ALL.

MDMS\_ASSIST is also required unless /NOASSIST is specified.

### Restrictions

The /POSITION and /START\_SLOT qualifiers are mutually exclusive.

The MDMS server will refuse this command if mutually exclusive qualifiers are used in the affirmative forms.

## Qualifiers

### **/ASSIST (D)**

#### **/NOASSIST**

The default /ASSIST qualifier is used when the magazine has not been physically moved, and an operator needs to physically move the magazine. Use /NOASSIST if the magazine has already been physically moved, or if you plan to move it yourself.

This qualifier requires the right MDMS\_ASSIST.

#### **/OFFSITE**

When moving to a location that is not already specified in the magazine record, this qualifier specifies that the location is an offsite location, and the magazine placement is offsite, following the move. By default, an unspecified location is onsite.

#### **/POSITION=(position)**

The /POSITION qualifier specifies that the magazine is being moved into a jukebox, and specifying the jukebox position that the magazine is being moved to. The /POSITION qualifier or /START\_SLOT qualifier must be specified when moving a magazine into a multi-magazine jukebox.

In addition, the /POSITION qualifier can be used to transfer a magazine between positions inside a jukebox (if physically possible). The position parameter is in the format (tower, face, level).

The 'tower, face and level' specification, represents the relative number of the tower, face and level, starting from 0.

So for the absolute jukebox slot of zero, the corresponding position is (0,0,0). The next position in the jukebox would be (0,0,1) and so and, according to the topology defined for the jukebox. Valid for MRD-controlled jukeboxes only.

#### **/REPLY=symbol**

The name of the symbol to receive the operator's reply when operator intervention is needed. The symbol will contain the operator reply to the DCL REPLY/TO or DCL REPLY/ABORT commands. The maximum length of a symbol name is 31 characters. This qualifier is only applicable when /ASSIST is specified.

#### **/SCHEDULE[=keyword]**

The /SCHEDULE qualifier selects magazines whose offsite or onsite date has "expired" and the magazine is not in the new location. If both dates have expired, the later of the two dates is used. The optional keywords on the /SCHEDULE qualifier may be:

OFFSITE

ONSITE

If the OFFSITE keyword is used, then only those magazines scheduled to be moved offsite are selected. If the ONSITE keyword is used, then only those magazines scheduled to be moved onsite are selected. If the keyword is omitted, then the magazines scheduled to be moved onsite and offsite are selected.

#### **/START\_SLOT=number**

This qualifier specifies the starting jukebox slot when the magazine is placed in a jukebox. The default is zero. Valid for MRD-controlled jukeboxes only.

#### **/SPACES=(range[,...])**

The /SPACES qualifier specifies the space(s) in a non-jukebox location to move the magazine to. You should specify a single space if you are moving a single magazine. If you issue /SCHEDULE, each magazine will be moved to the next space in the range in order.

**/WAIT (D)**

**/NOWAIT**

The /NOWAIT qualifier returns an informational message indicating that the move is being queued. The /WAIT qualifier causes the MDMS MOVE MAGAZINE command to wait until the magazine is moved.

**Examples**

```
$ MDMS MOVE MAGAZINE MYMAG01 JUKE_1
```

This command moves magazine MYMAG01 from its current location into jukebox JUKE\_1 at start slot of zero.

```
$ MDMS MOVE MAGAZINE MYMAG01 JUKE_2 /POSITION=(2, 0, 1)
```

This command moves magazine MYMAG01 from its current location into jukebox JUKE\_2 in position (2, 0, 1) - (i.e. Tower 2, face 0, level 1).

```
$ MDMS MOVE MAGAZINE MYMAG02 ROOM_100 /SPACES=23
```

This command moves magazine MYMAG02 from its current position out of the jukebox to location ROOM\_100 in space 23.

```
$ MDMS MOVE MAGAZINE MAG002 JUKE_1 /START_SLOT=11
```

This command moves magazine MAG002 from its current location into jukebox JUKE\_1 in start slot of 11.

```
$ MDMS MOVE MAGAZINE * /SCHEDULE=OFFSITE
```

This command moves all magazines whose offsite date has expired to the magazine's offsite location.

```
$ MDMS MOVE MAGAZINE * /SCHEDULE
```

This command moves all magazines whose offsite and onsite dates have expired to the magazine's offsite and onsite locations respectively.

---

## MDMS MOVE VOLUME

The MDMS MOVE VOLUME command moves a volume or volumes(s) from one location to another. The volume(s) must be created prior to the move.

Equivalent STORAGE Commands: STORAGE BIND, STORAGE EXPORT ACS, STORAGE EXPORT VOLUME, STORAGE IMPORT ACS, STORAGE IMPORT VOLUME, STORAGE UNBIND

### Format:

MDMS MOVE VOLUME volume\_id [destination]

### Parameters

volume\_id

Defines the name of the volume to move. A volume\_id may be a single volume, a list of volumes, a volume range, separated by a dash, or a list of volume ranges. A volume range is a numeric range for up to the last three characters of the volume ID. Example ranges are (ABC001-ABC250), (ABC120-ABC125). A volume\_id may also be a wildcard \*, which means all scheduled volumes. The wildcard is only valid when the /SCHEDULE qualifier is specified. This parameter is required.

The maximum length of the volume ID is 6 characters. Valid characters are " ,!,"% -?,A-Z,\_".

destination

Specifies the name of the destination object. The destination object can be a location, magazine or a jukebox. The destination is optional when /SCHEDULE is entered, but required for commands when /SCHEDULE is not entered. For /SCHEDULE, the default destination is the appropriate offsite or onsite location specified in the volume object.

Description

The MDMS MOVE VOLUME command moves a volume or volumes from one location to another. The volume(s) must be created prior to the move. Multiple volumes can be moved, where appropriate, in a single command.

When /ASSIST is specified, a series of OPCOM messages may be displayed asking the operator to move volumes between locations, or in and out of a jukebox or magazine. No slots or spaces are displayed in the OPCOM message, but the operator should issue SHOW VOLUME/FULL for the associated volumes to determine slots or spaces involved in the move. These are shown in the placement field.

When moving volumes into a jukebox, the specification of slots is optional. If not supplied, MDMS will apply default values and move the volumes into empty slots.

This command is not used to load volumes into or out of drives.

## Privileges

The request requires MDMS\_MOVE\_ALL, MDMS\_MOVE\_POOL or MDMS\_MOVE\_OWN.

If the user is moving a volume for which he is the owner, MDMS\_MOVE\_OWN is sufficient. If the user is moving a volume in a pool to which he is authorized, MDMS\_MOVE\_POOL is required. All other moves require MDMS\_MOVE\_ALL.

MDMS\_ASSIST is also required unless /NOASSIST is specified.

The /DESCRIPTION qualifier requires MDMS\_SET\_ALL, MDMS\_SET\_POOL or MDMS\_SET\_OWN.

## Restrictions

None

## Qualifiers

### **/ASSIST (D)**

#### **/NOASSIST**

The default /ASSIST qualifier is used to output an operator message if the volume requires operator intervention. You can use /ASSIST even if no assistance is needed - MDMS will perform the operation automatically if it can. If you specify /NOASSIST and assistance is needed, the operation fails.

This qualifier requires the right MDMS\_ASSIST.

#### **/CAP=cap\_id**

If moving to a volume to/from a silo, this qualifier specifies the Cartridge Access Port identifier into which the volume is being physically injected or ejected. Required when moving a volume into or out of a silo. Valid for DCSC-controlled jukeboxes only.

#### **/DESCRIPTION="text"**

Modifies comments about the object in the volume record. If the text contains spaces, then it must be enclosed within quotation marks. The length of the description can range from 0 to 255 characters. To clear the existing description, specify "".

This qualifier requires the right MDMS\_SET\_OWN, MDMS\_SET\_POOL or MDMS\_SET\_ALL, depending on the user's ownership of the volume.

#### **/NOPHYSICAL**

This qualifier requests that no internal jukebox calls (through DCSC or MRD) are made to implement the move. When used with /NOASSIST, the effect of the MOVE VOLUME command is to simply update the database to the new location. When used with the default of /ASSIST, a single OPCOM message is displayed to move the volumes, but when the message expires, the database is automatically updated.

#### **/OFFSITE**

When moving to a location that is not already specified in the volume record, this qualifier specifies that the location is an offsite location, and the volume placement is offsite following the move. By default, an unspecified location is onsite.

#### **/REPLY=symbol**

The name of the symbol to receive the operator's reply when operator intervention is needed. The symbol will contain the operator reply to the DCL REPLY/TO or DCL REPLY/ABORT commands. The maximum length of a symbol name is 31 characters. This qualifier is only applicable when /ASSIST is specified.



**/SCHEDULE[=keyword]**

The /SCHEDULE qualifier selects volumes whose offsite or onsite date has "expired" and the volumes are not in the new location. If both dates have expired, the later of the two dates is used. The optional keywords on the schedule qualifier may be:

OFFSITE

ONSITE

If the OFFSITE keyword is used, then only those volumes scheduled to be moved offsite are selected. If the ONSITE keyword is used, then only those volumes scheduled to be moved onsite are selected. If the keyword is omitted, then volumes scheduled to be moved onsite and offsite are selected.

**/SLOTS=(range[,...])**

The /SLOTS qualifier specifies that the volume is being moved into a jukebox or magazine and specifying the slot range, or slot list, that the volumes are being moved to. If not specified, MDMS selects free slot locations for the volume(s). In addition, the /SLOTS qualifier can be used to transfer volumes between slot locations inside a jukebox or magazine. If moving a single volume, specify a single slot. If moving multiple volumes, the slots in the range are assigned in order. If any of the specified slots are full, MDMS selects alternate slots. Valid for MRD-controlled jukeboxes only.

**/SPACES=(range[,...])**

The SPACES qualifier specifies the space(s) in a non-jukebox location to move the volume(s) to.

**/WAIT (D)****/NOWAIT**

The /NOWAIT qualifier returns an informational message indicating that the move is being queued. The /WAIT qualifier causes the MDMS MOVE VOLUME command to wait until the command is complete.

## MDMS MOVE VOLUME

### Examples

```
$ MDMS MOVE VOLUME ABC001 JUKE_2 /SLOTS=20
```

This command moves volume ABC001 from its current location to jukebox JUKE\_2 at slot 20.

```
$ MDMS MOVE VOLUME AGW081 SHELF /SPACES=42
```

This command moves the volume AGW081 to location SHELF at space 42.

```
$ MDMS MOVE VOLUME FLI050-FLI056 MAG001 /SLOTS=(0-6)
```

This command moves volumes FLI050 to FLI056 into magazine MAG001 at slots (0-6). Note that this replaces the old STORAGE BIND command.

```
$ MDMS MOVE VOLUME * /SCHEDULE=OFFSITE
```

This command moves all volumes whose offsite date has expired to the volume's offsite location.

```
$ MDMS MOVE VOLUME ABC001-ABC100 ARGUS_VAULT /SCHEDULE=OFFSITE
```

This command moves all volumes in the range ABC001-ABC100 whose offsite date has expired to the volume's location ARGUS\_VAULT.

```
$ MDMS MOVE VOLUME * /SCHEDULE
```

This command moves all volumes whose offsite and onsite dates have expired to the default off-site and onsite locations respectively.

```
$ MDMS MOVE VOLUME ALS100-ALS150 WOLFCREEK /CAP=1
```

This command moves volumes ALS100-ALS150 to DCSC jukebox WOLFCREEK using CAP 1.

---

## MDMS REPORT VOLUME

The MDMS REPORT VOLUME command reports on volume objects.

Equivalent STORAGE Command: REPORT VOLUME

### Format:

MDMS REPORT VOLUME field[=value] [...]

### Parameters

#### Field

The field parameter selects attributes to display for the object.

For any string value, you can use a wildcard character. The \* character provides a wild card for any number/length of characters. The % character is a wild card character for one character.

For date and numeric values, a range may be specified by entering a two value list in parentheses. For example:

SCRATCH\_DATE=(01-Jun-2002, 01-Jul-2002)

MOUNT\_COUNT=(100,9999)

The first value is interpreted as a low value, and the second value is interpreted as a high value. All values in the range are selected for display.

At least one field must be defined for a report to be generated, except when /FORECAST or /SUMMARY is entered.

Field	Meaning	Value Format	Example Output
ACCOUNT	Owner's account	String	ENGINEERING
ALLOCATION_DATE	Last allocation	Date	18-MAY-2001
AVAILABLE_STATE	State the volume will be put in when it becomes available	Keyword	ALLOCATED
BLOCK_FACTOR	Number of records in one block	Numeric	20
BRAND	Brand name of a volume	String	Compaq
CLEANED_DATE	Last date the volume was cleaned	Date DD-MMM-YYY	18-MAY-2001
CREATION_DATE	Date volume was entered in the database	Date DD-MMM-YYY	18-MAY-2001
DEALLOCATION_DATE	Date volume was last deallocated	Date DD-MMM-YYY	18-MAY-2001
DESCRIPTION	Description	String	DRA2 BACKUP
DRIVE	Name of tape drive	String	\$1\$MUA560

MDMS REPORT VOLUME

<b>Field</b>	<b>Meaning</b>	<b>Value Format</b>	<b>Example Output</b>
ERROR_COUNT	Number of I/O errors since a volume was last cleaned	Numeric	20
FORMAT	Recording format	Keyword	BACKUP
FREED_DATE	Date volume was last freed	Date DD-MMM-YYY	18-MAY-2001
INITIALIZED_DATE	Date volume was initialized	Date DD-MMM-YYY	18-MAY-2001
JOB	Name of the job that wrote the volume	String	ABS
JUKEBOX	Name of the jukebox	String	TL812_JUKE
LAST_ACCESS_DATE	Date volume was last accessed	Date DD-MMM-YYY	18-MAY-2001
MAGAZINE	Name of magazine	String	ENG_MAG
MEDIA_TYPE	Type of media	String	TK89
MOUNT_COUNT	Number of times a volume has been loaded by MDMS	Numeric	20
NEXT_VOLUME	Next volume in a set	String	AGW200
OFFSITE_DATE	Date a volume is to be taken offsite	Date DD-MMM-YYY	18-MAY-2001
OFFSITE_LOCATION	Location where volume resides when it is offsite	String	VAULT
ONSITE_DATE	Date a volume is to be returned on site	Date DD-MMM-YYY	18-MAY-2001
ONSITE_LOCATION	Location where the volume resides when it is onsite	String	ROOM_256
OWNER_UIC	Owner's user identification code	Standard Open-VMS UIC format	[311,311][ABS]
PLACEMENT	Current placement of the volume	Keyword	JUKEBOX
PLACENAME	Current placement name of the volume	String	NODE01_JUKE
POOL	Volume's pool name	String	ENGINEERING
PREVIOUS_VOLUME	Previous volume in a volume set	String	AGW201
PURCHASED_DATE	Volume's purchase date	Date DD-MMM-YYY	18-MAY-2001
PROTECTION	Access protection code	Standard Open-VMS protection codes	"S:RWED, O:RWED, G:E, W:E"

<b>Field</b>	<b>Meaning</b>	<b>Value Format</b>	<b>Example Output</b>
ERROR_COUNT	Number of I/O errors since a volume was last cleaned	Numeric	20
FORMAT	Recording format	Keyword	BACKUP
FREED_DATE	Date volume was last freed	Date DD-MMM-YYY	18-MAY-2001
INITIALIZED_DATE	Date volume was initialized	Date DD-MMM-YYY	18-MAY-2001
JOB	Name of the job that wrote the volume	String	ABS
JUKEBOX	Name of the jukebox	String	TL812_JUKE
LAST_ACCESS_DATE	Date volume was last accessed	Date DD-MMM-YYY	18-MAY-2001
MAGAZINE	Name of magazine	String	ENG_MAG
MEDIA_TYPE	Type of media	String	TK89
MOUNT_COUNT	Number of times a volume has been loaded by MDMS	Numeric	20
NEXT_VOLUME	Next volume in a set	String	AGW200
OFFSITE_DATE	Date a volume is to be taken offsite	Date DD-MMM-YYY	18-MAY-2001
OFFSITE_LOCATION	Location where volume resides when it is offsite	String	VAULT
ONSITE_DATE	Date a volume is to be returned on site	Date DD-MMM-YYY	18-MAY-2001
ONSITE_LOCATION	Location where the volume resides when it is onsite	String	ROOM_256
OWNER_UIC	Owner's user identification code	Standard Open-VMS UIC format	[311,311][ABS]
PLACEMENT	Current placement of the volume	Keyword	JUKEBOX
PLACENAME	Current placement name of the volume	String	NODE01_JUKE
POOL	Volume's pool name	String	ENGINEERING
PREVIOUS_VOLUME	Previous volume in a volume set	String	AGW201
PURCHASED_DATE	Volume's purchase date	Date DD-MMM-YYY	18-MAY-2001
PROTECTION	Access protection code	Standard Open-VMS protection codes	"S:RWED, O:RWED, G:E, W:E"

Field	Meaning	Value Format	Example Output
SCRATCH_DATE	Date volume is to be freed	Date DD-MMM-YYY	18-MAY-2001
SLOT	Number of the jukebox slot in which the volume resides	Numeric	20
SPACE	A non-jukebox space in the specified location where the volume resides	String	A120
STATE	The state of the volume	Keyword	ALLOCATED
TIMES_CLEANED	Number of times the volume has been cleaned	Numeric	10
TRANSITION_TIME	The time a volume stays in the transition state before going to the free state	Time DDDD- HH:MM:SS	15-00:00:00
USER	Owner's user name	String	SMITH
VOLUME	Volume ID	String	AGW200

## Description

The MDMS REPORT VOLUME command generates a report on the selected volumes. The contents of the report contain fields as specified in the field parameters. Each selected field is displayed unless the /NOPRINT field qualifier is specified. The report can be sorted on the field by using the /SORT qualifier on a single field (only one sort key supported per command). In addition, the width of each field can be specified with the /WIDTH qualifier - the default width is specific for each field size.

## Privileges

The request requires MDMS\_SHOW\_ALL, MDMS\_SHOW\_POOL or MDMS\_SHOW\_OWN. If the user only has MDMS\_SHOW\_OWN, only those allocated volumes owned by the user will be included in the report. If the user has MDMS\_SHOW\_POOL, only those volumes in pool(s) for which the user is authorized will be displayed. If the user has MDMS\_SHOW\_ALL, all potential volumes matching the selection criteria will be displayed.

## Restrictions

The /USER\_NAME qualifier can only be used with the /FORECAST and /SUMMARY qualifiers. For all other types of report use the selection value of the USER field.

Fields cannot be used with the /FORECAST and /SUMMARY qualifiers.

The /FORECAST and /SUMMARY qualifiers are mutually exclusive.

Fields must be specified with the /SCHEDULE qualifier.

The MDMS server will refuse this command if mutually exclusive qualifiers are used in the affirmative forms.

## Qualifiers

### **/FORECAST**

Displays all volumes allocated to a user sorted by their scratch date. If the /USER\_NAME qualifier is not specified, only those volumes allocated to you are displayed. Do not specify any fields.

### **/OUTPUT=file\_spec**

Directs the report output to the specified file instead of SYSS\$OUTPUT, which is usually the command screen.

### **/NOPRINT**

The /NOPRINT field qualifier specifies that a field defined as a selection criteria is not included in the report.

### **SCHEDULE[=keyword]**

The /SCHEDULE qualifier selects volumes whose offsite or onsite date has "expired" and the volumes are not in the new location. If both dates have expired, the later of the two dates is used.

The optional keyword on the schedule qualifier may be:

OFFSITE

ONSITE

If The OFFSITE keyword is used, then only those volumes scheduled to be moved offsite are selected. If the ONSITE keyword is used, then only those volumes scheduled to be moved onsite are selected. If the keyword is omitted, then volumes scheduled to be moved onsite and offsite are selected

### **/SORT**

This is a field qualifier, and is used as a sort key. Only one field can be used as a sort key, and no value may be specified.

### **/SUMMARY**

Displays all volumes allocated to a user and sorted by volume ID. If the /USER\_NAME qualifier is not specified, only those volumes allocated to you are displayed. Do not specify any fields.

### **/TITLE=text**

Specifies the title of the report. If the text contains spaces, it must be enclosed in quotation marks. The maximum length of the title is 80 characters. If not supplied, a default title based on the volume ID and selection criteria is generated.

### **/USER\_NAME=username**

Selects volumes owned by this user. This qualifier can only be used with the /FORECAST and /SUMMARY qualifiers. If /USER\_NAME is not specified, only those volumes for the present user are selected.

### **/WIDTH=number**

A field qualifier, this specifies the number of characters to display for a field. If not specified, a default width is applied to each field type, and excess characters may be truncated.

## MDMS REPORT VOLUME

### Examples

```
$ MDMS REPORT VOLUME VOLUME, POOL=ABS_POOL, STATE, -  
SCRATCH_DATE
```

This command prints a report for all volumes in pool ABS\_POOL, and prints out the volume\_id, pool name, allocation state and scratch date.

```
$ MDMS REPORT VOLUME VOLUME, STATE=ALLOCATED/NOPRINT, -  
SCRATCH_DATE
```

This command prints a report for all allocated volumes, and prints out the volume id and scratch date.

```
$ MDMS REPORT VOLUME VOLUME, STATE=ALLOCATED, -  
OWNER/SORT/WIDTH=10
```

This command prints a report for all allocated volumes, and prints out the volume id, allocation state and owner (maximum of 10 characters), and the report is sorted by owner.

```
$ MDMS REPORT VOLUME /FORECAST /USER_NAME=SMITH
```

This command prints a report for all volumes allocated to user SMITH, and prints out the volume id, allocation date, scratch date, format and description fields, sorted by scratch date.



---

## MDMS SET DOMAIN

The MDMS SET DOMAIN command modifies the MDMS Domain object. The MDMS domain contains attributes affecting all nodes, devices and locations that comprise an MDMS domain.

Equivalent STORAGE Command: None

### Format:

MDMS SET DOMAIN

### Parameters

None.

### Description

The MDMS SET DOMAIN command modifies the MDMS domain. The MDMS domain contains attributes affecting all nodes, devices and locations that comprise an MDMS domain.

### Privileges

The request requires MDMS\_SET\_ALL. The qualifiers associated with setting of privilege rights also require MDMS\_SET\_RIGHTS.

### Restrictions

The /ADD, /REMOVE and /REPLACE qualifiers are mutually exclusive. If none are specified, attributes are added to list attributes by default.

The MDMS server will refuse this command if mutually exclusive qualifiers are used in the affirmative forms.

### Qualifiers

#### **/ABS\_RIGHTS**

#### **/NOABS\_RIGHTS (D)**

The /ABS\_RIGHTS qualifier enables a certain set of MDMS rights when users have an ABS right set in the UAF record. The default /NOABS\_RIGHTS qualifier does not give a users with ABS rights any additional MDMS rights.

This qualifier requires the right MDMS\_SET\_RIGHTS.

#### **/ACCESS\_CONTROL=(USERNAME=user\_id, ALLOW=(option[,...]))**

#### **/NOACCESS\_CONTROL (D)**

Access control allows the system administrator to restrict access to the object to all or specific users on specific nodes or groups. Select one or more of the following options to allow the operation:

- SET - Allows setting the attributes of the object
- SHOW - Allows showing the object
- DELETE - Allows deleting the object
- READ - Allows restoring data using the object
- WRITE - Allows saving data using the object
- EXECUTE - Allows execution operations using the object

- **CONTROL** - Allows setting the access control on the object

By default, access controls are added to the list of access controls - only one user can be added per command.

However, you can remove users using the **/REMOVE** qualifier, and replace the entire access control using the **/REPLACE** qualifier. You can remove all access controls on the object by specifying **/NOACCESS\_CONTROL**.

The `user_id` should be in the format `node::username` or `group::username`. Wildcards are supported in the node, group and username portions.

Access control restrictions are in addition to the normal MDMS rights allowing operations on an object. If no access control is defined, then the default domain access control is applied.

### **/ADD (D)**

The **/ADD** qualifier works in conjunction with certain qualifiers that accept lists and adds the specified attributes to the existing list. This is the default when specifying list items.

### **/APPLICATION\_RIGHTS[=(right[,...])**

#### **/NOAPPLICATION\_RIGHTS**

The **/APPLICATION\_RIGHTS** qualifier sets the low-level rights associated with the high-level right **MDMS\_APPLICATION**. If the qualifier is specified with no value, a default set of rights is set. The **/REMOVE** or **/REPLACE** qualifiers can be used to remove or replace rights in the list, rather than adding them by default. Use **/NOAPPLICATION\_RIGHTS** to remove all rights.

This qualifier requires the right **MDMS\_SET\_RIGHTS**.

### **/CHECK\_ACCESS**

#### **/NOCHECK\_ACCESS (D)**

The **/CHECK\_ACCESS** qualifier enables access control validation for all MDMS requests. When enabled, MDMS validates access to objects according to the operation being requested. If there are no access control entries on the object, access is either granted or rejected depending on the default access state.

The default **/NOCHECK\_ACCESS** does not perform any access control validation in MDMS regardless of the default access state or whether there are any access control entries on an object. As always, appropriate MDMS rights are also checked for each request, and setting **/NOCHECK\_ACCESS** does not disabled rights validation.

The default is **/NOCHECK\_ACCESS**.

This qualifier requires the right **MDMS\_SET\_RIGHTS**.

### **/DEALLOCATE\_STATE=state**

This attribute stores a text string stating the deallocation state of volumes. Allowable values are **FREE** and **TRANSITION**. This state is applied to volumes that are deallocated when no specific state is specified on the deallocate request and the volume record does not have a transition time defined.

### **/DEFAULT\_RIGHTS[=(right[,...])**

#### **/NODEFAULT\_RIGHTS**

The **/DEFAULT\_RIGHTS** qualifier sets the low-level rights associated with users with no MDMS rights in their UAF record. If the qualifier is specified with no value, a default set of rights (i.e. no rights) are set. The **/REMOVE** or **/REPLACE** qualifiers can be used to remove or replace rights in the list, rather than adding them by default. Use **/NODEFAULT\_RIGHTS** to remove all default rights.

This qualifier requires the right **MDMS\_SET\_RIGHTS**.

**/DESCRIPTION="text"**

Comments about the domain. If the text contains spaces, then it must be enclosed within quotation marks. The length of the description can range from 0 to 255 characters. To clear the description, specify "".

**/MAIL\_USERS=(user[,...])****/NOMAIL\_USERS**

This attribute is a list of OpenVMS system users who should receive notification when volumes are deallocated. The default is user SYSTEM.

**/MAXIMUM\_SCRATCH\_TIME=delta\_time**

The maximum scratch time is an OpenVMS delta time that specifies the greatest allocation time allowed.

**/MEDIA\_TYPE=media\_type**

This qualifier specifies the default MDMS media type, which is applied to drives and volumes if they are created without a media type definition.

**/OFFSITE\_LOCATION=location****/NOOFFSITE\_LOCATION (D)**

This attribute stores a text string identifying the name of the Location object representing your default offsite storage location. This location is applied to all volumes and magazines that do not have an explicit offsite location specified.

**/ONSITE\_LOCATION=location****/NOONSITE\_LOCATION (D)**

This attribute stores a text string identifying the name of the Location object representing your default onsite storage location. This location is applied to all volumes and magazines that do not have an explicit onsite location specified.

**/OPCOM\_CLASSES=(class[,...])**

Use this attribute to list the OpenVMS OPCOM classes to which MDMS OPCOM messages are directed. These OPCOM classes are applied to all nodes that do not have explicit OPCOM classes specified. The following classes are supported:

CARDS	NETWORK	OPER6	OPER12
CENTRAL	OPER1	OPER7	PRINTER
CLUSTER	OPER2	OPER8	REPLY
DEVICES	OPER3	OPER9	SECURITY
DISKS	OPER4	OPER10	SOFTWARE
LICENSE	OPER5	OPER11	TAPES

**/OPERATOR\_RIGHTS[=(right[,...])****/NOOPERATOR\_RIGHTS**

The /OPERATOR\_RIGHTS qualifier sets the low-level rights associated with the high-level right MDMS\_OPERATOR. If the qualifier is specified with no value, a default set of rights is set. The /REMOVE or /REPLACE qualifiers can be used to remove or replace rights in the list, rather than adding them by default. Use /NOOPERATOR\_RIGHTS to remove all rights.

This qualifier requires the right MDMS\_SET\_RIGHTS.

**/PROTECTION=protection**

Assign the default volume protection for all volumes with this attribute. Use the standard OpenVMS protection format. This protection is applied to all volumes that do not have an explicit protection specified.

**/RELAXED\_ACCESS (D)**

**/NORELAXED\_ACCESS**

The domain /[NO]RELAXED\_ACCESS qualifier determines the default user access to an object if no access control entries exist for the object. If at least one access control entry exists for the object, then the default access is not used and access is controlled by the entries.

If /RELAXED\_ACCESS is set, then all users have access to the object, if the object contains no access control entries. This option allows for a more liberal usage of access control: only use it on objects requiring it, and allow users access to all other objects. This is designed for normal environments, and those whose domain-scope is centrally organized.

If /NORELAXED\_ACCESS is set, then no users have access to the object, if the object contains no access control entries, with the exception of:

- The owner of the object, who always has full access
- Users with domain access control have the same level of access on all objects as they do the domain
- The "Last Updated By" user of the domain has full access to the domain and all objects

/NORELAXED\_ACCESS is a more conservative approach to access control: don't allow anyone access to an object unless they have specifically been granted access with an access control entry. This is designed for secure environments, or those whose domain-scope crosses organizational boundaries.

Note that access control is an object-specific method of controlling access. It is in addition to the normal MDMS rights required to perform operations. Having access control without the associated rights is not enough to operate on an object - both are required.

The default is /RELAXED\_ACCESS.

This qualifier requires the right MDMS\_SET\_RIGHTS.

**/REMOVE**

The /REMOVE qualifier works in conjunction with certain qualifiers that accept lists and removes specified attributes from them.

**/REPLACE**

The /REPLACE qualifier works in conjunction with certain qualifiers that accept lists and replaces the existing attributes list with the specified list. By default, attributes are added to the existing list.

**/REQUEST\_ID=number**

Changes the request identifier for the next MDMS request in the domain.

**/SCHEDULER\_TYPE**

When using the integrated ABS and MDMS scheduler functions, the /SCHEDULER\_TYPE qualifier defines the type of scheduler used for save and restore operations. Select from one of the following options:

- INTERNAL (D)  
MDMS internal scheduler using OpenVMS batch queues. This is the default option, and is recommended for normal backup and restore scenarios.

- **EXTERNAL**

MDMS scheduler access via command procedure. This provides more flexibility on how saves and restores are submitted to the batch queues.

- **SCHEDULER**

External scheduler access via command procedures. This is the required option when using an external scheduler product to schedule saves and restores. You should use this option if you are using the DECScheduler V2.1B that was shipped with previous versions of ABS.

The default is INTERNAL.

**/SCRATCH\_TIME=delta\_time**

Assign the default scratch date to volumes by applying the delta time specified with attribute to the allocation date of the volume. This scratch time is applied to all volumes that do not have an explicit scratch date defined.

**/SYSPRV (D)**

**/NOSYSPRV**

The default /SYSPRV qualifier enables user with OpenVMS privilege SYSPRV the low-level right MDMS\_ALL\_RIGHTS, which allows all operations. The /NOSYSPRV qualifier does not give users with SYSPRV any additional MDMS rights.

If you set the domain to /NOSYSPRV, it is highly recommended that you grant MDMS\_ALL\_RIGHTS in all accounts that require unconditional access to MDMS. This includes the system administrators' accounts and specifically the SYSTEM account.

The MDMS IVP requires unconditional access from the SYSTEM account during installation using either SYSPRV (if enabled) or MDMS\_ALL\_RIGHTS.

The default is SYSPRV.

**/TRANSITION\_TIME=delta\_time**

Defines the default transition time to be applied to volumes that do not have an explicit transition time defined.

**/USER\_RIGHTS[=(right[,...])**

**/NOUSER\_RIGHTS**

The /USER\_RIGHTS qualifier sets the low-level rights associated with the high-level right MDMS\_USER. If the qualifier is specified with no value, a default set of rights is set. The /REMOVE or /REPLACE qualifiers can be used to remove or replace rights in the list, rather than adding them by default. Use /NOUSER\_RIGHTS to remove all rights.

This qualifier requires the right MDMS\_SET\_RIGHTS.

## MDMS SET DOMAIN

### Examples

```
$ MDMS SET DOMAIN /OFFSITE_LOCATION=XCYX
```

This command sets the name of the domain offsite location to **XCYX**.

```
$ MDMS SET DOMAIN /MAIL_USERS=(NORTON,CRANDLE) /REPLACE
```

This command specifies that OpenVMS Cluster users Norton and Crandle are to be the only users notified on the mail distribution. They will be notified when volumes are deallocated.

```
$ MDMS SET DOMAIN /OPCOM_CLASSES=(OPER4) /REMOVE
```

This command prevents the further display of OPCOM messages on terminals enabled for the **OPER4** OPCOM class.

```
$ MDMS SET DOMAIN /OPERATOR_RIGHTS=(MDMS_SET_ALL, - MDMS_SET_PROTECTED)
```

This command adds the low-level rights **MDMS\_SET\_ALL** and **MDMS\_SET\_PROTECTED** to the high-level right **MDMS\_OPERATOR**.

---

## MDMS SET DRIVE

The MDMS SET DRIVE command modifies a drive definition in the MDMS database.

Equivalent STORAGE Command: None

### Format:

MDMS SET DRIVE drive\_name [...]

### Parameters

#### **drive\_name**

Specifies the name of the drive. Specify a drive name or a logical name.

The maximum length of the drive name or logical name is 31 characters.  
Valid characters are "\$,-,0-9,A-Z,\_,a-z".

A comma-separated list of drive names may be entered.

### Description

The MDMS SET DRIVE command modifies a drive definition in the MDMS database.

### Privileges

The request requires MDMS\_SET\_ALL.

### Restrictions

The /ADD, /REMOVE and /REPLACE qualifiers are mutually exclusive. If none are specified, attributes are added to list attributes by default.

The /JUKEBOX and /STACKER qualifiers are mutually exclusive.

The MDMS server will refuse this command if mutually exclusive qualifiers are used in the affirmative forms.

### Qualifiers

#### **/ACCESS=keyword**

This qualifier defines the type of access to the drive, which can be one of the following keywords:

ALL (D)- supports local node/cluster/fibre access and remote (RDF) access

LOCAL - supports local node/cluster/fibre access only

REMOTE - supports remote (RDF) access only Access to drives is restricted on allocate requests - for example, it is not possible to allocate a drive designated as local access remotely using RDF. However, with the proper rights, it is possible to issue other MDMS commands (such as LOAD) both locally and remotely.

The default is ALL.

**/ACCESS\_CONTROL=(USERNAME=user\_id, ALLOW=(option[,...]))**

**/NOACCESS\_CONTROL (D)**

Access control allows the system administrator to restrict access to the object to all or specific users on specific nodes or groups. Select one or more of the following options to allow the operation:

- SET - Allows setting the attributes of the object
- SHOW - Allows showing the object
- DELETE - Allows deleting the object
- READ - Allows restoring data using the object
- WRITE - Allows saving data using the object
- EXECUTE - Allows execution operations using the object
- CONTROL - Allows setting the access control on the object

By default, access controls are added to the list of access controls - only one user can be added per command. However, you can remove users using the /REMOVE qualifier, and replace the entire access control using the /REPLACE qualifier. You can remove all access controls on the object by specifying /NOACCESS\_CONTROL.

The user\_id should be in the format node::username or group::username. Wildcards are supported in the node, group and username portions.

Access control restrictions are in addition to the normal MDMS rights allowing operations on an object. If no access control is defined, then the default domain access control is applied.

**/ADD (D)**

The /ADD qualifier works in conjunction with certain qualifiers that accept lists and adds the specified attributes to the existing list.

This is the default when specifying list items.

**/AUTOMATIC\_REPLY (D)**

**/NOAUTOMATIC\_REPLY**

Specifies that MDMS automatically replies to all OPCOM messages that can be polled for completion on requests for this particular drive.

The default is /AUTOMATIC\_REPLY.

**/DESCRIPTION="text"**

Comments about the drive. If the text contains spaces, then it must be enclosed within quotation marks. The length of the description can range from 0 to 255 characters. To clear the description, specify "".

**/DEVICE=vms\_device\_name**

Specifies the OpenVMS device name of the drive. This qualifier is required if the OpenVMS device name is different from the drive name. Do not specify a node specification (NODE::name) in the vms\_device\_name. Instead use the /NODES or /GROUPS qualifier.

The default device name is the drive name.

**/DISABLED**

Places the drive in the disabled state. This prevents the drive from being selected and allocated for use. This takes effect immediately. However, if the drive is already in use, operations on that drive will continue until the drive is deallocated.

The default is /ENABLED.



**/DRIVE\_NUMBER=number**

This qualifier defines the drive number for robot commands if the drive is in a jukebox. This qualifier must be specified for multi-drive MRD-controlled jukeboxes.

The default is drive number 0.

**/ENABLED (D)**

Places the drive in the enabled state. This allows the drive to be selected and allocated for use. This takes effect immediately.

The default is /ENABLED.

**/GROUPS=(group\_name[,...])****/NOGROUPS**

Specifies the names of groups of nodes that share common access to this device. Usually, only one group is specified. If neither /NODES or /GROUPS are specified, the node from which the command was issued is used as the node name. Groups can also be specified with the /NODES qualifier.

**/JUKEBOX=jukebox\_name**

If the drive is in a jukebox, this qualifier specifies the jukebox name.

**/MEDIA\_TYPE=(media\_type[,...])****/NOMEDIA\_TYPE**

Specifies one or more distinct media types that the drive can support for read-write access. The /REMOVE or /REPLACE qualifiers can be used to remove or replace objects in the list, rather than adding them by default. The /NOMEDIA\_TYPES qualifier removes all media types. When a drive is created with no media types, the default media type from the domain is used.

**/NODES=(node\_name[,...])****/NONODES**

Specifies one or more distinct nodes that have direct access to the drive. The /REMOVE or /REPLACE qualifiers can be used to remove or replace objects in the list, rather than adding them by default. The /NONODE qualifier removes all nodes. If neither /NODES nor /GROUPS is specified, the node from which the command was issued is used as the node name.

**/OWNER\_NAME=node::username****/OWNER\_NAME=group::username****/NOOWNER\_NAME**

The /OWNER\_NAME qualifier specifies the owner of the object. Specify an owner as node::username or group::username. The node, group and username portions may be wild-carded.

The owner of an object has default access control of all allowances to the object, regardless of whether a specific access control entry is present or the state of the default domain access control. If you wish to remove this unconditional owner access, specify /NOOWNER\_NAME. Note that being an owner of an object overrides only the access control; the appropriate MDMS rights for a request on the object must still be honored.

By default, the owner of an object is the group\_name::username of the user entering the CREATE command on the object from a node in the group.

**/READONLY=(media\_type[,...])****/NOREADONLY**

Specifies one or more distinct media types that the drive can support for read-only access. The /REMOVE or /REPLACE qualifiers can be used to remove or replace objects in the list, rather than adding them by default. The /NOREADONLY qualifier removes all read-only media types.

**/REMOVE**

The /REMOVE qualifier works in conjunction with certain qualifiers that accept lists and removes specified attributes from them.

**/REPLACE**

The /REPLACE qualifier works in conjunction with certain qualifiers that accept lists and replaces the existing attributes list with the specified list. By default, attributes are added to the existing list.

**/SHARED**

**/NOSHARED (D)**

The /SHARED qualifier defines that the drive may be used by non-MDMS clients and that the drive is only partially managed. If the drive is set to the default /NOSHARED, the MDMS server allocates the drive at all times it is not used by an application or user. Setting the drive to /SHARED clears this allocation.

The default is /NOSHARED.

**/STACKER**

**/NOSTACKER (D)**

The /STACKER qualifier indicates that the drive is to be treated as a stacker gravity loader. The default /NOSTACKER indicates that the drive is to be treated as a standalone drive, or a robotically-controlled jukebox, as appropriate.

The default is /NOSTACKER.

**/STATE=state**

This is a protected field that should be modified only to recover on error. Use the LOAD and UNLOAD commands to manipulate the state field under normal operation. The /STATE qualifier sets the current drive state. The valid keywords are:

Empty

Full

Loading

Unloading

This qualifier requires the right MDMS\_SET\_PROTECTED.

**Example**

```
$ MDMS SET DRIVE FRED /NODES=(JOHN, PAUL)
```

This command adds nodes JOHN and PAUL for direct access to the drive with drive name or logical name of FRED.

---

## MDMS SET GROUP

The MDMS SET GROUP command modifies a group definition in the MDMS database.

Equivalent STORAGE Command: None

### Format:

MDMS SET GROUP group\_name [...]

### Parameters

#### **group\_name**

Specifies the name of the group.

The maximum length of the drive name is 31 characters.

Valid characters are "\$,-,0-9,A-Z,\_,a-z".

A comma-separated list of group names may be entered.

### Description

The MDMS SET GROUP command modifies a group definition in the MDMS database.

### Privileges

The request requires MDMS\_SET\_ALL.

### Restrictions

The /ADD, /REMOVE and /REPLACE qualifiers are mutually exclusive. If none are specified, attributes are added to list attributes by default.

The MDMS server will refuse this command if mutually exclusive qualifiers are used in the affirmative forms.

### Qualifiers

**/ACCESS\_CONTROL=(USERNAME=user\_id, ALLOW=(option[,...]))**

**/NOACCESS\_CONTROL (D)**

Access control allows the system administrator to restrict access to the object to all or specific users on specific nodes or groups. Select one or more of the following options to allow the operation:

- SET - Allows setting the attributes of the object
- SHOW - Allows showing the object
- DELETE - Allows deleting the object
- READ - Allows restoring data using the object
- WRITE - Allows saving data using the object
- EXECUTE - Allows execution operations using the object
- CONTROL - Allows setting the access control on the object

By default, access controls are added to the list of access controls - only one user can be added per command. However, you can remove users using the /REMOVE qualifier, and replace the entire access control using the /REPLACE qualifier. You can remove all access controls on the object by specifying /NOACCESS\_CONTROL.

## MDMS SET GROUP

The user\_id should be in the format node::username or group::username. Wildcards are supported in the node, group and username portions.

Access control restrictions are in addition to the normal MDMS rights allowing operations on an object. If no access control is defined, then the default domain access control is applied.

### **/ADD (D)**

The /ADD qualifier works in conjunction with certain qualifiers that accept lists and adds the specified attributes to the existing list. This is the default when specifying list items.

### **/DESCRIPTION="text"**

Comments about the group. If the text contains spaces, then it must be enclosed within quotation marks. The length of the description can range from 0 to 255 characters. To clear the description, specify "".

### **/NODES=(node\_name[,...])**

### **/NONODES**

Specifies one or more nodes that comprise the group. The /REMOVE or /REPLACE qualifiers can be used to remove or replace objects in the list, rather than adding them by default. The /NONODES qualifier removes all nodes.

### **/OWNER\_NAME=node::username**

### **/OWNER\_NAME=group::username**

### **/NOOWNER\_NAME**

The /OWNER\_NAME qualifier specifies the owner of the object. Specify an owner as node::username or group::username. The node, group and username portions may be wildcarded.

The owner of an object has default access control of all allowances to the object, regardless of whether a specific access control entry is present or the state of the default domain access control. If you wish to remove this unconditional owner access, specify /NOOWNER\_NAME. Note that being an owner of an object overrides only the access control; the appropriate MDMS rights for a request on the object must still be honored.

By default, the owner of an object is the group\_name::username of the user entering the CREATE command on the object from a node in the group.

### **/REMOVE**

The /REMOVE qualifier works in conjunction with certain qualifiers that accept lists and removes specified attributes from them.

### **/REPLACE**

The /REPLACE qualifier works in conjunction with certain qualifiers that accept lists and replaces the existing attributes list with the specified list. By default, attributes are added to the existing list.

## **Example**

```
$ MDMS SET GROUP HOSER /NODE=TOOKUS /REMOVE
```

This command removes node **TOOKUS** from the group **HOSER**

---

## MDMS SET JUKEBOX

The MDMS SET JUKEBOX command modifies a jukebox definition in the MDMS database.  
Equivalent STORAGE Command: None

### Format:

MDMS SET JUKEBOX jukebox\_name [...]

### Parameters

#### **jukebox\_name**

Specifies the name of the jukebox.

The maximum length of the jukebox name is 31 characters.  
Valid characters are "\$,-,0-9,A-Z,\_,a-z".

A comma-separated list of jukebox names may be entered.

### Description

The MDMS SET JUKEBOX command modifies a jukebox definition in the MDMS database.

### Privileges

The request requires MDMS\_SET\_ALL.

### Restrictions

The /ADD, /REMOVE and /REPLACE qualifiers are mutually exclusive. If none are specified, attributes are added to list attributes by default.

The MDMS server will refuse this command if mutually exclusive qualifiers are used in the affirmative forms.

### Qualifiers

#### **/ACCESS=keyword**

This qualifier defines the type of access to the jukebox, which can be one of the following keywords:

ALL (D) - supports local node/access and remote access

LOCAL - supports local node/cluster access only

REMOTE - supports remote access only

#### **/ACCESS\_CONTROL=(USERNAME=user\_id, ALLOW=(option[,...]))**

#### **/NOACCESS\_CONTROL (D)**

Access control allows the system administrator to restrict access to the object to all or specific users on specific nodes or groups. Select one or more of the following options to allow the operation:

- SET - Allows setting the attributes of the object
- SHOW - Allows showing the object
- DELETE - Allows deleting the object
- READ - Allows restoring data using the object

- WRITE - Allows saving data using the object
- EXECUTE - Allows execution operations using the object
- CONTROL - Allows setting the access control on the object

By default, access controls are added to the list of access controls - only one user can be added per command. However, you can remove users using the /REMOVE qualifier, and replace the entire access control using the /REPLACE qualifier. You can remove all access controls on the object by specifying /NOACCESS\_CONTROL.

The user\_id should be in the format node::username or group::username. Wildcards are supported in the node, group and username portions.

Access control restrictions are in addition to the normal MDMS rights allowing operations on an object. If no access control is defined, then the default domain access control is applied.

**/ACS=acs\_id**

The /ACS qualifier specifies the Automated Cartridge System (ACS) identifier. The default value is zero. Each MDMS jukebox maps to one Library Storage Module (LSM), and requires the specification of the library, ACS and LSM identifiers. Valid for DCSC-controlled jukeboxes only.

**/ADD (D)**

The /ADD qualifier works in conjunction with certain qualifiers that accept lists and adds the specified attributes to the existing list. This is the default when specifying list items.

**/AUTOMATIC\_REPLY (D)**

**/NOAUTOMATIC\_REPLY**

Specifies that MDMS automatically replies to all OPCOM messages that can be polled for completion on requests for this particular jukebox.

**/CAP\_SIZE=(number[,...])**

For DCSC-controlled jukeboxes equipped with Cartridge Access Points (CAPs), this attribute specifies the number of cells for each CAP. The first number is the size for CAP 0, the next for CAP 1 etc. If a size is not specified for a CAP, a default value of 40 is used. Specifying the CAP size optimizes the movement of volumes to and from the jukebox by filling the CAP to capacity for each move operation. When specifying CAP sizes, the specified numbers always replace any previous sizes in the database. Valid for DCSC-controlled jukeboxes only.

**/CONTROL=keyword**

This qualifier specifies the robot control facility used to control the jukebox's robot. The valid keywords are:

MRD (D) - The robot is controlled with the MRD facility

DCSC - The jukebox is a silo controlled by the DCSC facility - for StorageTek(R) Silos only. This option is not available when running with the ABS-OMT license: control is hard-coded to MRD.

**/DESCRIPTION="text"**

Defines comments about the object in the record. If the text contains spaces, then it must be enclosed within quotation marks. The length of the description can range from 0 to 255 characters. To clear the existing description, specify "".

**/DISABLED**

Places the jukebox in the disabled state. This prevents all drives in the jukebox from being selected and allocated for use. This takes effect immediately. However, any drives in use will continue to be used until they are deselected.

**/ENABLED (D)**

Places the drive in the enabled state. This allows drives in the jukebox to be selected and allocated for use. This takes effect immediately. This is the default.

**/GROUPS=(group\_name[,...])****/NOGROUPS**

Specifies the names of groups of nodes that share common access to this device. Normally, only one group is specified. The /REMOVE or /REPLACE qualifiers can then add them by default. The /NOGROUPS qualifier removes all groups.

**/LIBRARY=library\_id**

This qualifier specifies the library identifier for a silo. Valid values are 1,2,3,4, and the default is 1 when the jukebox is controlled by DCSC and 0 (not applicable) when controlled by MRD. Each MDMS jukebox maps to one Library Storage Module (LSM), and requires specification of the library, ACS and LSM identifiers.

**/LOCATION=location\_name****/NOLOCATION**

This qualifier specifies the location of the jukebox, which is used when moving volumes into and out of the jukebox. If not specified, or /NOLOCATION is specified, the default onsite location from the domain record is used as the jukebox location.

**/LSM=lsm\_id**

The /LSM qualifier specifies the Library Storage Module (LSM) identifier. The default value is zero. Each MDMS jukebox maps to one Library Storage Module (LSM), and requires the specification of the library, ACS and LSM identifiers. Valid for DCSC-controlled jukeboxes only.

**/NODES=(node\_name[,...])****/NONODES**

Specifies one or more nodes that can directly access the jukebox. The /REMOVE or /REPLACE qualifiers can be used to remove or replace objects in the list, rather than adding them by default. The /NONODES qualifier removes all nodes.

**/OWNER\_NAME=node::username****/OWNER\_NAME=group::username****/NOOWNER\_NAME**

The /OWNER\_NAME qualifier specifies the owner of the object. Specify an owner as node::username or group::username. The node, group and username portions may be wildcarded.

The owner of an object has default access control of all allowances to the object, regardless of whether a specific access control entry is present or the state of the default domain access control. If you wish to remove this unconditional owner access, specify /NOOWNER\_NAME. Note that being an owner of an object overrides only the access control; the appropriate MDMS rights for a request on the object must still be honored.

By default, the owner of an object is the group\_name::username of the user entering the CREATE command on the object from a node in the group.

**/REMOVE**

The /REMOVE qualifier works in conjunction with certain qualifiers that accept lists and removes specified attributes from them.

**/REPLACE**

The /REPLACE qualifier works in conjunction with certain qualifiers that accept lists and replaces the existing attributes list with the specified list. By default, attributes are added to the existing list.

**/ROBOT=robot\_name****/NOROBOT**

The /ROBOT qualifier defines the OpenVMS device name of the robot. Required for, and applicable to MRD-controlled jukeboxes only.

Do not specify a node name in the robot name.

**/SLOT\_COUNT=number**

The /SLOT\_COUNT qualifier specifies the total number of slots in the entire jukebox. For any MRD jukebox, either the slot count or topology must be specified. Valid for MRD-controlled jukeboxes only.

**/STATE=keyword**

This is a protected field that should be modified only to recover on error. Use the LOAD, UNLOAD or MOVE commands to manipulate the state field under normal operation. The /STATE qualifier specifies the usage state of the jukebox. The keyword values are:

Available - The jukebox is available for use

In\_use - The jukebox is in use

This qualifier requires the right MDMS\_SET\_PROTECTED.

**/THRESHOLD=number**

Specifies that an OPCOM message is output when the number of free volumes in the jukebox falls below the specified number. The default value is zero, which disables the feature. The OPCOM message is output on a node that can directly access the jukebox.

**/TOPOLOGY=(TOWERS=(number[,...]), FACES=(number[,...]), LEVELS=(number[,...]), SLOTS=(number[,...]))**

Specifies topology of jukebox, when a TL820-class jukebox is being used as a magazine. Valid for MRD-controlled jukeboxes only. The topology specification allows OPCOM messages to move magazines to be specified with TOWER, FACE, LEVEL rather than slot range. The specification of topology is optional. For each tower in the configuration, a corresponding entry must also be placed in FACES LEVELS and SLOTS that reflects the configuration of that tower.

The tower numbers start at zero, and additional towers must be the next number in sequence (i.e. 0,1,2 etc). Other specifications are absolute counts of the entity being specified for each tower (i.e. the total number of faces, levels and slots in each tower).

For example, for a three-tower jukebox, each tower having 8 faces, the first tower having two levels and the other two towers having three levels, and support of 11-slot bin-packs, the topology specification would be:

```
/TOPOLOGY=(TOWERS=(0,1,2), /FACES=(8,8,8), LEVELS=(2,3,3), SLOTS=(11,11,11)
```

**/USAGE=[NO]MAGAZINE**

The /USAGE=MAGAZINE qualifier specifies that the jukebox is configured for magazines, and that the movement of volumes may be performed using magazines. The /USAGE=NOMAGAZINE qualifier does not support magazine use. The default is NOMAGAZINE.

You must specify /USAGE=MAGAZINE when defining the /TOPOLOGY attribute. Note that you can use the jukebox for non-magazine moves even when the usage is magazine, but the reverse is not true. Valid for MRD-controlled jukeboxes only.



## Examples

```
$ MDMS SET JUKEBOX JUKE_1 /DESCRIPTION="Design Data Backup"
```

This command modifies the description for jukebox JUKE\_1.

```
$ MDMS SET JUKEBOX JUKE_2 /USAGE=MAGAZINE /SLOTS=(0-100)
```

This command modifies the jukebox JUKE\_2 to support magazines, and slots 0-100.

---

## MDMS SET LOCATION

The MDMS SET LOCATION command modifies a location definition in the MDMS database.  
Equivalent STORAGE Command: None

### Format:

MDMS SET LOCATION location [...]

### Parameters

#### location

Specifies the name of the location.

The maximum length of the location is 31 characters. Valid characters are "\$,-,0-9,A-Z,\_,a-z".

A comma-separated list of locations may be entered.

### Description

The MDMS SET LOCATION command modifies a location definition in the MDMS database.

### Privileges

The request requires MDMS\_SET\_ALL.

### Restrictions

None

### Qualifiers

**/ACCESS\_CONTROL=(USERNAME=user\_id, ALLOW=(option[,...]))**

**/NOACCESS\_CONTROL (D)**

Access control allows the system administrator to restrict access to the object to all or specific users on specific nodes or groups. Select one or more of the following options to allow the operation:

- SET - Allows setting the attributes of the object
- SHOW - Allows showing the object
- DELETE - Allows deleting the object
- READ - Allows restoring data using the object
- WRITE - Allows saving data using the object
- EXECUTE - Allows execution operations using the object
- CONTROL - Allows setting the access control on the object

By default, access controls are added to the list of access controls - only one user can be added per command. However, you can remove users using the /REMOVE qualifier, and replace the entire access control using the /REPLACE qualifier. You can remove all access controls on the object by specifying /NOACCESS\_CONTROL.

The user\_id should be in the format node::username or group::username. Wildcards are supported in the node, group and username portions.

Access control restrictions are in addition to the normal MDMS rights allowing operations on an object. If no access control is defined, then the default domain access control is applied.

**/DESCRIPTION="text"**

Defines comments about the object in the record. If the text contains spaces, then it must be enclosed within quotation marks. The length of the description can range from 0 to 255 characters. To clear the existing description, specify "".

**/LOCATION=location**

**/NOLOCATION (D)**

The /LOCATION qualifier allows you to specify a parent location, thus creating a location hierarchy. If there is no parent location, specify /NOLOCATION. Use parent locations to allow selection of volumes or drives in compatible locations. One location is compatible with another if it has a common parent location in the hierarchy. If you do not wish to utilize the compatible location feature, do not specify parent locations. Locations with common parents are most useful where the parents and siblings are in close proximity to one another (e.g. rooms 101 and 102, with parent location floor 1), and selection of volumes or drives from any of the locations is desired. Do not use parent locations across larger distances.

**/OWNER\_NAME=node::username**

**/OWNER\_NAME=group::username**

**/NOOWNER\_NAME**

The /OWNER\_NAME qualifier specifies the owner of the object. Specify an owner as node::username or group::username. The node, group and username portions may be wildcarded.

The owner of an object has default access control of all allowances to the object, regardless of whether a specific access control entry is present or the state of the default domain access control. If you wish to remove this unconditional owner access, specify /NOOWNER\_NAME. Note that being an owner of an object overrides only the access control; the appropriate MDMS rights for a request on the object must still be honored.

By default, the owner of an object is the group\_name::username of the user entering the CREATE command on the object from a node in the group.

**/SPACES=(range)**

**/NOSPACES**

The /SPACES qualifier defines individual spaces for volumes or magazines at the location. Spaces are alphanumeric strings of up to 8 characters. The spaces can be specified as a range - only a single range is supported. The /NOSPACES qualifier removes all spaces.

## Example

```
$ MDMS SET LOCATION SHELF_100 /SPACES=(300-500)
```

This command modifies the location called SHELF\_100 and supports spaces 300 - 500.

---

## MDMS SET MAGAZINE

The MDMS SET MAGAZINE command modifies a magazine definition in the MDMS database.

Equivalent STORAGE Command: None.

### Format:

MDMS SET MAGAZINE magazine\_name [...]

### Parameters

**magazine\_name**

Specifies the name of the magazine.

The maximum length of the magazine name is 31 characters.

Valid characters are "\$,-,0-9,A-Z,\_,a-z".

A comma-separated list of magazine names may be entered.

### Description

The MDMS SET MAGAZINE command modifies a magazine definition in the MDMS database.

### Privileges

The request requires MDMS\_SET\_ALL.

The /JUKEBOX, /PLACEMENT, /POSITION and /START\_SLOT qualifiers also require MDMS\_SET\_PROTECTED.

### Restrictions

None

### Qualifiers

**/ACCESS\_CONTROL=(USERNAME=user\_id, ALLOW=(option[,...]))**

**/NOACCESS\_CONTROL (D)**

Access control allows the system administrator to restrict access to the object to all or specific users on specific nodes or groups. Select one or more of the following options to allow the operation:

- SET - Allows setting the attributes of the object
- SHOW - Allows showing the object
- DELETE - Allows deleting the object
- READ - Allows restoring data using the object
- WRITE - Allows saving data using the object
- EXECUTE - Allows execution operations using the object
- CONTROL - Allows setting the access control on the object

By default, access controls are added to the list of access controls - only one user can be added per command. However, you can remove users using the /REMOVE qualifier, and replace the

entire access control using the /REPLACE qualifier. You can remove all access controls on the object by specifying /NOACCESS\_CONTROL.

The user\_id should be in the format node::username or group::username. Wildcards are supported in the node, group and username portions.

Access control restrictions are in addition to the normal MDMS rights allowing operations on an object. If no access control is defined, then the default domain access control is applied.

**/DESCRIPTION="text"**

Defines comments about the object in the record. If the text contains spaces, then it must be enclosed within quotation marks. The length of the description can range from 0 to 255 characters. To clear the existing description, specify "".

**/JUKEBOX=jukebox\_name**

This is a protected field that should only be modified to recover on error. Use the MOVE MAGAZINE command to set up the jukebox name under normal operations. This qualifier specifies the name of the jukebox in which the magazine resides. The maximum length of the jukebox name is 31 characters.

This qualifier requires the right MDMS\_SET\_PROTECTED.

**/OFFSITE=(LOCATION=location],[NO]DATE[=date])**

**/NOOFFSITE**

This qualifier specifies the date that the magazine is to be taken offsite and the offsite location. The location field is required when using the /OFFSITE qualifier if no location has been previously specified. The LOCATION keyword cannot be negated and if specified must point to a valid location object. Specify a VMS absolute or delta time in the date field. The NODATE keyword may be used to remove the date. This has the effect of disabling the schedule for the magazine, while retaining the location. To clear the offsite date specify /NOOFFSITE.

The default offsite location is that stored in the domain object.

**/ONSITE=(LOCATION=location],[NO]DATE[=date])**

**/NOONSITE**

This qualifier specifies the date that the magazine is to be brought back onsite and the onsite location. The location field is required when using the /ONSITE qualifier if no location has been previously specified. The LOCATION keyword cannot be negated and if specified must point to a valid location object. Specify a VMS absolute or delta time in the date field. The NODATE keyword may be used to remove the date. This has the effect of disabling the schedule for the magazine, while retaining the location. To clear the onsite date, specify /NOONSITE.

The default onsite location is that stored in the domain object.

**/OWNER\_NAME=node::username**

**/OWNER\_NAME=group::username**

**/NOOWNER\_NAME**

The /OWNER\_NAME qualifier specifies the owner of the object. Specify an owner as node::username or group::username. The node, group and username portions may be wildcarded.

The owner of an object has default access control of all allowances to the object, regardless of whether a specific access control entry is present or the state of the default domain access control. If you wish to remove this unconditional owner access, specify /NOOWNER\_NAME. Note that being an owner of an object overrides only the access control; the appropriate MDMS rights for a request on the object must still be honored.

By default, the owner of an object is the group\_name::username of the user entering the CREATE command on the object from a node in the group.

**/PLACEMENT=keyword**

This is a protected field that should only be modified to recover on error. Use the MOVE MAGAZINE command to set up the placement under normal operation. This qualifier defines the current placement of the magazine. Select one of following options:

JUKEBOX

OFFSITE

ONSITE

MOVING

This qualifier requires the right MDMS\_SET\_PROTECTED.

**/POSITION=position=(tower,face,level)**

This is a protected field that should only be modified to recover on error. Use the MOVE MAGAZINE command to set up the position under normal operation. The /POSITION qualifier specifies the position in the jukebox that the magazine resides.

The tower, face and level specification represent the relative number of the tower, face and level, starting from 0. So for absolute jukebox slot of zero, the corresponding position is (0,0,0). The next position in the jukebox would be (0,0,1) and so on, according to the topology defined for the jukebox.

This qualifier requires the right MDMS\_SET\_PROTECTED.

**/SLOT\_COUNT=number**

The /SLOT\_COUNT qualifier specifies the number of slots in a magazine to store volumes.

**/SPACES=(range)**

**/NOSPACES**

This qualifier specifies the space(s) in a location in which the magazine is stored when not in a jukebox. Spaces are alphanumeric strings of up to 8 characters. The /NOSPACES qualifier removes all spaces.

**/START\_SLOT=(number)**

This is a protected field that should only be modified to recover on error. Use the MOVE MAGAZINE command to set up the start slot under normal operation. This qualifier specifies the starting jukebox slot when the magazine is placed in a jukebox.

This qualifier requires the right MDMS\_SET\_PROTECTED.

## Example

```
$ MDMS SET MAGAZINE MYMAG01 /ONSITE=(LOCATION=SHELF_20) - /SPACES=S4
```

This command modifies the non-jukebox location of magazine MYMAG01 to "space S4 in shelf\_20".

---

## MDMS SET MEDIA\_TYPE

The MDMS SET MEDIA\_TYPE command modifies a media type definition in the MDMS database.

Equivalent STORAGE Command: None

### Format:

MDMS SET MEDIA\_TYPE media\_type [...]

### Parameters

#### **media\_type**

Specifies the name of the media type.

The maximum length of the media type name is 31 characters.

Valid characters are "\$,-,0-9,A-Z,\_,a-z".

A comma-separated list of media types may be entered.

### Description

The MDMS SET MEDIA\_TYPE command modifies a media type definition in the MDMS database. A media type definition consists of a density, compaction option and capacity, if applicable.

### Privileges

The request requires MDMS\_SET\_ALL.

### Restrictions

None

### Qualifiers

**/ACCESS\_CONTROL=(USERNAME=user\_id, ALLOW=(option[...]))**

**/NOACCESS\_CONTROL (D)**

Access control allows the system administrator to restrict access to the object to all or specific users on specific nodes or groups. Select one or more of the following options to allow the operation:

- SET - Allows setting the attributes of the object
- SHOW - Allows showing the object
- DELETE - Allows deleting the object
- READ - Allows restoring data using the object
- WRITE - Allows saving data using the object
- EXECUTE - Allows execution operations using the object
- CONTROL - Allows setting the access control on the object

By default, access controls are added to the list of access controls - only one user can be added per command. However, you can remove users using the /REMOVE qualifier, and replace the entire access control using the /REPLACE qualifier. You can remove all access controls on the object by specifying /NOACCESS\_CONTROL.

## MDMS SET MEDIA\_TYPE

The user\_id should be in the format node::username or group::username. Wildcards are supported in the node, group and username portions.

Access control restrictions are in addition to the normal MDMS rights allowing operations on an object. If no access control is defined, then the default domain access control is applied.

### **/CAPACITY=number**

The /CAPACITY qualifier specifies the capacity in megabytes of the tape. This is used by some MDMS clients to estimate end-of-tape conditions. By default, capacity is set to zero.

### **/COMPACTION (D)**

#### **/NOCOMPACTION**

The /COMPACTION qualifier specifies that the media type should use compaction when writing to tape. This is the default. If you do not wish to use compaction, then specify /NOCOMPACTION.

### **/DENSITY=density**

Specifies a density string between 1 and 31 characters in length that the media type supports.

Note that the COMP keyword for compaction should be specified in the /COMPACTION attribute, not density.

### **/DESCRIPTION="text"**

Defines comments about the media type. If the text contains spaces, then it must be enclosed within quotation marks. The length of the description is can range from 0 to 255 characters. Specify "" to clear the description.

### **/LENGTH=length**

The /LENGTH qualifier specifies the length of a 9-track magnetic tape, and is expressed in feet. By default, length is set to zero.

### **/OWNER\_NAME=node::username**

### **/OWNER\_NAME=group::username**

### **/NOOWNER\_NAME**

The /OWNER\_NAME qualifier specifies the owner of the object. Specify an owner as node::username or group::username. The node, group and username portions may be wildcarded.

The owner of an object has default access control of all allowances to the object, regardless of whether a specific access control entry is present or the state of the default domain access control. If you wish to remove this unconditional owner access, specify /NOOWNER\_NAME. Note that being an owner of an object overrides only the access control; the appropriate MDMS rights for a request on the object must still be honored.

By default, the owner of an object is the group\_name::username of the user entering the CREATE command on the object from a node in the group.

## Example

```
$ MDMS SET MEDIA_TYPE TAPE_9T /DENSITY=1600 /LENGTH=3600
```

This command modifies a media type called TAPE\_9T to use density of 1600 and length of 3600.



---

## MDMS SET NODE

The MDMS SET NODE command modifies a node definition in the MDMS database.

Equivalent STORAGE Command: None

### Format:

MDMS SET NODE node\_name [...]

### Parameters

#### **node\_name**

Specifies the name of the node. Do not append colons to the node name. The node name should be the DECnet (Phase IV) node name (i.e.SYS\$NODE) if DECnet (Phase IV) is supported on the node - otherwise it should be a unique name chosen by the MDMS administrator. If DECnet-Plus (Phase V) and/or TCP/IP are supported, the appropriate fullnames should be stored as attributes of the node. Do not use the node name to specify fullnames.

The maximum length of the node name is 31 characters. Valid characters are "\$,-,0-9,A-Z,\_,a-z".

### Description

The MDMS SET NODE command modifies a node definition in the MDMS database.

### Privileges

The request requires MDMS\_SET\_ALL.

### Restrictions

The /ADD, /REMOVE and /REPLACE qualifiers are mutually exclusive. If none are specified, attributes are added to list attributes by default.

The MDMS server will refuse this command if mutually exclusive qualifiers are used in the affirmative forms.

### Qualifiers

**/ACCESS\_CONTROL=(USERNAME=user\_id, ALLOW=(option[,...]))**

**/NOACCESS\_CONTROL (D)**

Access control allows the system administrator to restrict access to the object to all or specific users on specific nodes or groups. Select one or more of the following options to allow the operation:

- SET - Allows setting the attributes of the object
- SHOW - Allows showing the object
- DELETE - Allows deleting the object
- READ - Allows restoring data using the object
- WRITE - Allows saving data using the object
- EXECUTE - Allows execution operations using the object
- CONTROL - Allows setting the access control on the object

By default, access controls are added to the list of access controls - only one user can be added per command. However, you can remove users using the /REMOVE qualifier, and replace the

entire access control using the /REPLACE qualifier. You can remove all access controls on the object by specifying /NOACCESS\_CONTROL.

The user\_id should be in the format node::username or group::username. Wildcards are supported in the node, group and username portions.

Access control restrictions are in addition to the normal MDMS rights allowing operations on an object. If no access control is defined, then the default domain access control is applied.

#### **/ADD (D)**

The /ADD qualifier works in conjunction with certain qualifiers that accept lists and adds the specified attributes to the existing list. This is the default when specifying list items.

#### **/DATABASE\_SERVER (D)**

#### **/NODATABASE\_SERVER**

The /DATABASE\_SERVER qualifier means the node can be a database server, supporting fail-over operations. To be a database server, the node must have direct access to the MDMS database files.

#### **/DECNET\_PLUS\_FULLNAME=node\_fullname**

This qualifier allows you to specify the DECnet-Plus (Phase V) fullname for a node. The fullname may be up to 255 characters. If this node has a DECnet-Plus name defined by logical name "SYS\$NODE\_FULLNAME" then the DECNET\_PLUS\_FULLNAME has to be defined for this node and has to exactly match the DECnet-Plus (Phase V) name. The DECNET\_PLUS\_FULLNAME has to be defined in order for this node to be fully enabled when the DECnet transport has been enabled and DECnet-Plus is running on the system. The fullname can be specified in upper or lower case.

#### **/DESCRIPTION="text"**

Defines comments about the node. If the text contains spaces, then it must be enclosed within quotation marks. The length of the description can range from 0 to 255 characters. Specify "" to clear the description.

#### **/DISABLED**

Places the node in the disabled state. This prevents the node from participating in the MDMS domain as either a server or a client. This takes effect immediately.

#### **/ENABLED (D)**

Places the node in the enabled state. This allows the node to participate in MDMS operations. This takes effect immediately. This is the default.

#### **/LOCATION=location**

#### **/NOLOCATION**

The /LOCATION qualifier specifies the location of the node, which is used when allocating volumes and drives. If not specified, or /NOLOCATION is specified, the default onsite location from the domain record is used as the node location.

#### **/OPCOM=(class[,...])**

#### **/NOOPCOM**

The /OPCOM qualifier adds the specified classes used for notifying operators. All OPCOM for devices on the node are sent to all specified classes on the node. The /REMOVE or /REPLACE qualifiers can be used to remove or replace classes in the list, rather than adding them by default. Specify /NOOPCOM to disable OPCOM notification. By default, the node acquires OPCOM classes from the domain record.

The following classes are valid:

CARDS	NETWORK	OPER6	OPER12
CENTRAL	OPER1	OPER7	PRINTER
CLUSTER	OPER2	OPER8	REPLY
DEVICES	OPER3	OPER9	SECURITY
DISKS	OPER4	OPER10	SOFTWARE
LICENSE	OPER5	OPER11	TAPES

**/OWNER\_NAME=node::username**

**/OWNER\_NAME=group::username**

**/NOOWNER\_NAME**

The /OWNER\_NAME qualifier specifies the owner of the object. Specify an owner as node::username or group::username. The node, group and username portions may be wild-carded.

The owner of an object has default access control of all allowances to the object, regardless of whether a specific access control entry is present or the state of the default domain access control. If you wish to remove this unconditional owner access, specify /NOOWNER\_NAME. Note that being an owner of an object overrides only the access control; the appropriate MDMS rights for a request on the object must still be honored.

By default, the owner of an object is the group\_name::username of the user entering the CREATE command on the object from a node in the group.

**/REMOVE**

The /REMOVE qualifier works in conjunction with certain qualifiers that accept lists and removes specified attributes from them.

**/REPLACE**

The /REPLACE qualifier works in conjunction with certain qualifiers that accept lists and replaces the existing attributes list with the specified list. By default, attributes are added to the existing list.

**/TCPIP\_FULLNAME=node\_fullname[:low\_port-high\_port]**

This qualifier allows you to specify the TCP/IP full name for a node. The full name may be up to 255 characters. If this node has a TCP/IP name defined by logical name "\*INET\_HOST" the TCPIP\_FULLNAME has to be defined and has to exactly match the full IP name as "<INET\_HOST>.<INET\_DOMAIN>". For INET\_DOMAIN see logical name "\*INET\_DOMAIN". The TCPIP\_FULLNAME has to be defined in order for this node to be fully enabled when the TCPIP transport has been enabled. The fullname can be specified in upper or lower case.

The low\_port and high\_port numbers specify the range of TCP/IP port numbers used by the server to listen for incoming requests. The default is 2501-2510. If this conflicts with other applications, a new range above 1023 can be specified. The range should contain at least 10 port numbers for the MDMS server to select one at a time.

Note that the MDMS GUI requires TCP/IP running on all GUI nodes, and on the MDMS server nodes to which the GUI may connect.

**/TRANSPORT=(keyword[,...])**

Specifies the transport protocol(s) to be used when communicating with other MDMS facilities on other nodes, as a prioritized ordered list. The /REMOVE or /REPLACE qualifiers can be used to remove or replace objects in the list, rather than adding them by default. Enter one or more of:

Specifies the network transports to be used, as a prioritized ordered list. The /REMOVE or /REPLACE qualifiers can be used to remove or replace transports in the list, rather than adding them by default.

Enter one or more of:

- DECNET - listen to incoming requests from other MDMS servers on DECnet (Phase IV) and DECnet-Plus (Phase V)
- TCPIP - listen to incoming requests from other MDMS servers on TCP/IP

Setting a new transport will automatically start the listener for this transport on the database server node. Likewise, removing a transport will take place within 10 seconds on the database server node. For client nodes, transport changes will take place the next time network connections time out (usually within 10 minutes). If the change needs to take place immediately, the client node server process must be restarted.

The node name and/or the node full names have to be set accordingly for a transport to work correctly.

## Example

```
$ MDMS SET NODE COOKIE /OPCOM=(TAPES, OPER1)
```

This command modifies a node definition named "COOKIE", by adding OPCOM classes TAPES and OPER1 to the current list of OPCOM classes.

---

## MDMS SET POOL

The MDMS SET POOL command modifies a pool definition in the MDMS database.

Equivalent STORAGE Command: None

### Format:

MDMS SET POOL pool\_name [...]

### Parameters

#### **pool\_name**

Specifies the name of the pool.

The maximum length of the pool name is 31 characters. Valid characters are "\$,-,0-9,A-Z,\_,a-z".

A comma-separated list of pool names may be entered.

### Description

The MDMS SET POOL command modifies a pool definition in the MDMS database.

### Privileges

The request requires MDMS\_SET\_ALL.

### Restrictions

The /ADD, /REMOVE and /REPLACE qualifiers are mutually exclusive. If none are specified, attributes are added to list attributes by default.

The MDMS server will refuse this command if mutually exclusive qualifiers are used in the affirmative forms.

### Qualifiers

**/ACCESS\_CONTROL=(USERNAME=user\_id, ALLOW=(option[...]))**

**/NOACCESS\_CONTROL (D)**

Access control allows the system administrator to restrict access to the object to all or specific users on specific nodes or groups. Select one or more of the following options to allow the operation:

- SET - Allows setting the attributes of the object
- SHOW - Allows showing the object
- DELETE - Allows deleting the object
- READ - Allows restoring data using the object
- WRITE - Allows saving data using the object
- EXECUTE - Allows execution operations using the object
- CONTROL - Allows setting the access control on the object

By default, access controls are added to the list of access controls - only one user can be added per command. However, you can remove users using the /REMOVE qualifier, and replace the entire access control using the /REPLACE qualifier. You can remove all access controls on the object by specifying /NOACCESS\_CONTROL.

The user\_id should be in the format node::username or group::username. Wildcards are supported in the node, group and username portions.

Access control restrictions are in addition to the normal MDMS rights allowing operations on an object. If no access control is defined, then the default domain access control is applied.

**/ADD (D)**

The /ADD qualifier works in conjunction with certain qualifiers that accept lists and adds the specified attributes to the existing list.

This is the default when specifying list items.

**/AUTHORIZED\_USERS=(node/group\_name::username[,...])**

**/NOAUTHORIZED\_USERS**

Specifies one or more distinct users to the pool specified by node or group name and user name. Only authorized or default users can allocate volumes belonging to the pool. The /REMOVE or /REPLACE qualifiers can be used to remove or replace users in the list, rather than adding them by default.

**/DEFAULT\_USERS=(node/group\_name::username[,...])**

**/NODEFAULT\_USERS**

Specifies one or more distinct users to the pool as the users default pool. Only authorized or default users can allocate volumes belonging to the pool. The /REMOVE or /REPLACE qualifiers can be used to remove or replace users in the list, rather than adding them by default. A particular node /group::user combination should only be defined with the /DEFAULT qualifier for one pool.

**/DESCRIPTION="text"**

Defines comments about the pool. If the text contains spaces, then it must be enclosed within quotation marks. The length of the description can range from 0 to 255 characters. Specify "" to clear the description.

**/OWNER\_NAME=node::username**

**/OWNER\_NAME=group::username**

**/NOOWNER\_NAME**

The /OWNER\_NAME qualifier specifies the owner of the object. Specify an owner as node::username or group::username. The node, group and username portions may be wildcarded.

The owner of an object has default access control of all allowances to the object, regardless of whether a specific access control entry is present or the state of the default domain access control. If you wish to remove this unconditional owner access, specify /NOOWNER\_NAME. Note that being an owner of an object overrides only the access control; the appropriate MDMS rights for a request on the object must still be honored.

By default, the owner of an object is the group\_name::username of the user entering the CREATE command on the object from a node in the group.

**/REMOVE**

The /REMOVE qualifier works in conjunction with certain qualifiers that accept lists and removes specified attributes from them.

**/REPLACE**

The /REPLACE qualifier works in conjunction with certain qualifiers that accept lists and replaces the existing attributes list with the specified list. By default, attributes are added to the existing list.

**/THRESHOLD=number**

Specifies that an OPCOM message is output when the number of free volumes in the pool falls below the specified number. The default value is zero, which disables the feature. The OPCOM message is output on the database server node.

**Example**

```
$ MDMS SET POOL TEST_POOL/AUTHORIZE=(COOKIE::ABS, COOKIE::HSM)
```

This command adds authorized users COOKIE::ABS and COOKIE::HSM to a pool called TEST\_POOL.

---

## MDMS SET SCHEDULE

The MDMS SET SCHEDULE command modifies an existing schedule definition in the MDMS database. Schedule is a new object type.

Equivalent ABS Command: None

### Format:

MDMS SET SCHEDULE schedule\_name [...]

### Parameters

#### **schedule\_name**

Specifies the name of the schedule.

The maximum length of the schedule name is 63 characters.

Valid characters are "\$,-,0-9,A-Z,\_,a-z".

A comma-separated list of schedule names may be entered.

### Description

The MDMS SET SCHEDULE command modifies an existing schedule definition in the MDMS database. You can use schedules to execute any DCL command (including MDMS commands) at regular intervals.

Schedules are used by MDMS to execute saves and restores. You can modify only CUSTOM schedules using this command.

You can also modify any schedules that you specifically created.

### Privileges

The request requires MDMS\_SET\_ALL.

### Restrictions

The /ADD, /REMOVE and /REPLACE qualifiers are mutually exclusive. If none are specified, attributes are added to list attributes by default.

The MDMS server will refuse this command if mutually exclusive qualifiers are used in the affirmative forms.

### Qualifiers

**/ACCESS\_CONTROL=(USERNAME=user\_id, ALLOW=(option[...]))**

**/NOACCESS\_CONTROL (D)**

Access control allows the system administrator to restrict access to the object to all or specific users on specific nodes or groups. Select one or more of the following options to allow the operation:

- SET - Allows setting the attributes of the object
- SHOW - Allows showing the object
- DELETE - Allows deleting the object
- READ - Allows restoring data using the object
- WRITE - Allows saving data using the object



- EXECUTE - Allows execution operations using the object
- CONTROL - Allows setting the access control on the object

By default, access controls are added to the list of access controls - only one user can be added per command. However, you can remove users using the /REMOVE qualifier, and replace the entire access control using the /REPLACE qualifier. You can remove all access controls on the object by specifying /NOACCESS\_CONTROL.

The user\_id should be in the format node::username or group::username. Wildcards are supported in the node, group and username portions.

Access control restrictions are in addition to the normal MDMS rights allowing operations on an object. If no access control is defined, then the default domain access control is applied.

#### **/ADD (D)**

The /ADD qualifier works in conjunction with certain qualifiers that accept lists and adds the specified attributes to the existing list. This is the default when specifying list items.

#### **/AFTER\_SCHEDULE=(NAME=schedule\_name, WHEN=option)**

#### **/NOAFTER\_SCHEDULE**

The /AFTER\_SCHEDULE qualifiers specifies that this schedule is scheduled for execution after the AFTER\_SCHEDULE name, under the circumstances defined in WHEN. This can be used to serialize a series of operations one after the other, and conditionally execute this schedule depending on the status of the AFTER\_SCHEDULE.

The SCHEDULE should be a schedule name defined in the MDMS database. For WHEN, select one of the following:

ALL (D)	Unconditionally execute this schedule when the AFTER_SCHEDULE completes
ERROR	Execute this schedule if the AFTER_SCHEDULE completed with ERROR or FATAL status
FATAL	Execute this schedule if the AFTER_SCHEDULE completed with FATAL status
NONE	Do not execute this schedule (can be used as a placeholder)
SUCCESS	Execute this schedule if the AFTER_SCHEDULE completed with SUCCESS or INFORMATIONAL status
WARNING	Execute this schedule if the AFTER_SCHEDULE completed with WARNING, ERROR or FATAL status

If an /AFTER\_SCHEDULE name is specified, the default WHEN option is ALL.

/NOAFTER\_SCHEDULE removes the dependency to execute after another schedule.

#### **/COMMAND="string"**

The /COMMAND qualifier specifies the command to submit to DCL when the schedule executes. Specify a valid DCL command line in quotes. This qualifier must be entered for the schedule to do any useful work.

**/DATES=(date[,...])****/NODATES (D)**

The /DATES qualifier specifies on which days of the month you wish the schedule to execute. Valid values are 1-31. You can enter a date, a list of dates, a range of dates and a list of ranges (e.g. 1-7, 15-21). /NODATES indicates that all dates are valid (1-31) and other criteria are used for scheduling. You can use the /ADD, /REMOVE and /REPLACE qualifiers to manipulate the dates list.

The default is /NODATES.

**/DAYS=(day[,...])****/NODAYS (D)**

The /DAYS qualifier specifies on which days of the week you wish the schedule to execute. Valid values use at least the first three letters of the days of the week (English only). You can enter a day, a list of days, a range of days and a list of ranges (e.g. MON-WED, SAT-SUN). /NODAYS indicates that all days are valid (MON-SUN) and other criteria are used for scheduling. You can use the /ADD, /REMOVE and /REPLACE qualifiers to manipulate the days list.

The default is /NODAYS.

**/DESCRIPTION="text"**

Comments about the object. If the text contains spaces, then it must be enclosed within quotation marks. The length of the description can range from 0 to 255 characters. To clear the description, specify "".

**/EXCLUDE=(date[,...])****/NOEXCLUDE (D)**

The /EXCLUDE qualifier specifies specific dates that you wish to exclude from execution, even though the other scheduling parameters would normally schedule on that date. This can be used to exclude scheduled operations on holidays for example. Enter one or more dates in OpenVMS date format (04-JUL-2001), separated by commas. You can specify dates up to 9 years in the future. /NOEXCLUDE removes all dates from the exclude list. You can use the /ADD, /REMOVE and /REPLACE qualifiers to manipulate the exclude list.

By default, no exclude dates are defined.

**/INCLUDE=(date[,...])****/NOINCLUDE (D)**

The /INCLUDE qualifier specifies specific dates that you wish to include for execution, even though the other scheduling parameters would normally not schedule on that date. This can be used to include operations on special days that require special attention. Enter one or more dates in OpenVMS date format (31-DEC-2001), separated by commas. You can specify dates up to 9 years in the future. You can use the /ADD, /REMOVE and /REPLACE qualifiers to manipulate the include list.

By default, no include dates are defined.

**/MONTHS=(month[,...])****/NOMONTHS (D)**

The /MONTHS qualifier specifies on which months of the year you wish the schedule to execute. Valid values use at least the first three letters of the months of the year (English only). You can enter a month, a list of months, a range of months and a list of ranges (e.g. JAN-MAR, JUL-SEP). /NOMONTHS indicates that all months are valid (JAN-DEC) and other criteria are used for scheduling. You can use the /ADD, /REMOVE and /REPLACE qualifiers to manipulate the months list.

The default is /NOMONTHS.

**/OWNER\_NAME=node::username**  
**/OWNER\_NAME=group::username**  
**/NOOWNER\_NAME**

The /OWNER\_NAME qualifier specifies the owner of the object. Specify an owner as node::username or group::username. The node, group and username portions may be wild-carded.

The owner of an object has default access control of all allowances to the object, regardless of whether a specific access control entry is present or the state of the default domain access control. If you wish to remove this unconditional owner access, specify /NOOWNER\_NAME. Note that being an owner of an object overrides only the access control; the appropriate MDMS rights for a request on the object must still be honored.

By default, the owner of an object is the group\_name::username of the user entering the CREATE command on the object from a node in the group.

**/REMOVE**

The /REMOVE qualifier works in conjunction with certain qualifiers that accept lists and removes specified attributes from them.

**/REPLACE**

The /REPLACE qualifier works in conjunction with certain qualifiers that accept lists and replaces the existing attributes list with the specified list. By default, attributes are added to the existing list.

**/TIMES=(time[,...])**

**/NOTIMES (D)**

The /TIMES qualifier specifies the times of the day you wish the schedule to execute. Enter from 1 to 100 times in OpenVMS format, separated by commas. For example (08:00, 10:00, 12:00, 14:00, 16:00). /NOTIMES clears out all times and effectively disables the schedule from execution. /TIMES=00:00 signifies midnight.

By default, no times are set.

## Examples:

```
$ MDMS SET SCHEDULE SCHED1 -
_ $ /DAYS=SAT -
_ $ /TIMES=21:00 -
_ $ /COMMAND="@USER2:[SMITH]SCHED1.COM"
```

This command modifies schedule SCHED1 to run every Saturday at 21:00 by issuing the command procedure @USER2:[SMITH]SCHED1.COM.

```
$ MDMS SET SCHEDULE SCHED2 -
_ $ /NOMONTHS -
_ $ /DAYS=SUN -
_ $ /DATES=(1-7) -
_ $ /TIMES=12:00 -
_ $ /COMMAND="MDMS DELETE SAVE *MONTH*"
```

This command modifies schedule SCHED2, to execute every month on the first Sunday of the month at noon, by issuing an MDMS command to delete saves.

## MDMS SET SERVER

The MDMS SET SERVER command resets server conditions. Currently it only supports resetting the MDMS logging file.

Equivalent STORAGE Command: None

### Format

MDMS SET SERVER /RESET\_LOG

### Parameters

None.

### Description

The MDMS SET SERVER command resets server conditions. Currently it only supports resetting the MDMS logging file.

### Privileges

The request requires MDMS\_SET\_ALL.

### Qualifiers

#### **/RESET\_LOG**

This qualifier closes the current log file and opens a new one. This qualifier is required.

### Example

```
$ MDMS SET SERVER /RESET_LOG
```

This command closes the current logging file and opens a new version. The new files will be opened at the location designated by the logical name MDMS\$LOGFILE\_LOCATION on the database server node.

---

## MDMS SET VOLUME

The MDMS SET VOLUME command modifies a volume definition in the MDMS volume database.

Equivalent STORAGE Command: STORAGE SET VOLUME

### Format:

MDMS SET VOLUME volume\_id [...]

### Parameters

#### **volume\_id**

Specifies the volume ID of the volume to be added. The volume ID is the external label for the volume. Alternatively, a volume range, separated by a dash, may be specified. A volume range is a numeric range for up to the last n characters of the volume ID. While specifying the volume range make sure that:

1. Same number of characters are there on both sides of the dash.
2. The data type of the corresponding position on either side of the dash must be the same.

Example ranges are (A00001-A99999), (AB0001-AB9999).

The maximum length of the volume ID is 6 characters. Valid characters are " ,!,"% -?,A-Z,\_".

A comma-separated list of volume IDs or ranges may be entered.

### Description

The MDMS SET VOLUME command modifies a volume definition in the MDMS volume database.

### Privileges

The request requires MDMS\_SET\_ALL, MDMS\_SET\_VOLUME, MDMS\_SET\_POOL or MDMS\_SET\_OWN.

If the user is modifying a volume that is allocated to him, MDMS\_SET\_OWN is sufficient. If the user is modifying a volume that belongs to a pool to which he is authorized, MDMS\_SET\_POOL is required. For any other volume, MDMS\_SET\_VOLUME or MDMS\_SET\_ALL is required.

### Restrictions

The /ADD, /REMOVE and /REPLACE qualifiers are mutually exclusive. If none are specified, attributes are added to list attributes by default.

The MDMS server will refuse this command if mutually exclusive qualifiers are used in the affirmative forms.

## Qualifiers

**/ACCESS\_CONTROL=(USERNAME=user\_id, ALLOW=(option[,...]))**

**/NOACCESS\_CONTROL (D)**

Access control allows the system administrator to restrict access to the object to all or specific users on specific nodes or groups. Select one or more of the following options to allow the operation:

- SET - Allows setting the attributes of the object
- SHOW - Allows showing the object
- DELETE - Allows deleting the object
- READ - Allows restoring data using the object
- WRITE - Allows saving data using the object
- EXECUTE - Allows execution operations using the object
- CONTROL - Allows setting the access control on the object

By default, access controls are added to the list of access controls - only one user can be added per command. However, you can remove users using the /REMOVE qualifier, and replace the entire access control using the /REPLACE qualifier. You can remove all access controls on the object by specifying /NOACCESS\_CONTROL.

The user\_id should be in the format node::username or group::username. Wildcards are supported in the node, group and username portions.

Access control restrictions are in addition to the normal MDMS rights allowing operations on an object. If no access control is defined, then the default domain access control is applied.

**/ACCOUNT="text"**

Defines the account name of the volume. The account name may be up to 31 characters. If it contains spaces, it must be enclosed in quotation marks. Specify "" to clear the account name.

This qualifier requires the right MDMS\_SET\_PROTECTED.

**/ADD (D)**

The /ADD qualifier works in conjunction with certain qualifiers that accept lists and adds the specified attributes to the existing list. This is the default when specifying list items.

**/ALLOCATED\_DATE=date**

**/NOALLOCATED\_DATE**

Specifies the date the volume was allocated. Normally this is set by MDMS. Specify a VMS absolute date and time. Specify /NOALLOCATED\_DATE to clear the allocated date.

This qualifier requires the right MDMS\_SET\_PROTECTED.

**/AVAILABLE**

The /AVAILABLE qualifier moves a volume from the UNAVAILABLE state to the state it was previously in prior to the UNAVAILABLE state. The volume may then be moved into the TRANSITION or FREE state if the scratch date and/or transition time have expired.

**/BLOCK\_FACTOR=number**

Specifies the block factor for the volume. The default is a block factor of zero.

**/BRAND="text"**

The media manufacturer. The maximum length of the brand name is 31 characters. If it contains spaces it must be contained in quotation marks. Specify "" to clear the brand.

**/CLEANED\_DATE=date****/NOCLEANED\_DATE**

This qualifier specifies the date the volume was last cleaned and is entered as a VMS absolute time. Specify /NOCLEANED\_DATE to clear the cleaned date.

**/CREATION\_DATE=date****/NOCREATION\_DATE**

The date the volume is created. This attribute is set by MDMS, but may be overridden if necessary.

This qualifier requires the right MDMS\_SET\_PROTECTED.

**/DEALLOCATED\_DATE=date****/NODEALLOCATED\_DATE**

This qualifier specifies the actual deallocation date for the volume. Specify a VMS absolute time. This date is normally set by MDMS.

This qualifier requires the right MDMS\_SET\_PROTECTED.

**/DESCRIPTION="text"**

Comments about the volume. If the text contains spaces, then it must be enclosed within quotation marks. The length of the description can range from 0 to 255 characters. Specify "" to clear the description.

**/DRIVE=drive\_name****/NODRIVE**

This is a protected field that should be modified only to recover on error. Use the LOAD and UNLOAD commands to manipulate the drive name under normal operation. This qualifier specifies the drive that the volume currently resides in or last resided in. This is normally set up by MDMS. To clear the drive, specify /NODRIVE.

This qualifier requires the right MDMS\_SET\_PROTECTED.

**/FORMAT=keyword**

Specifies the format of the tape. Possible values are:

ASCII

BACKUP

EBCDIC

NONE (D)

RMUBACKUP

**/FREED\_DATE=date****/NOFREED\_DATE**

Specifies the date the volume was last freed (i.e. put in the FREE state). Specify an OpenVMS absolute date and time. This is normally set up by MDMS. To clear the freed date, specify /NOFREED\_DATE.

This qualifier requires the right MDMS\_SET\_PROTECTED.

**/INITIALIZED\_DATE=date****/NOINITIALIZED\_DATE**

Specifies the date the volume was last initialized. Specify a VMS absolute date and time. This is normally set up by MDMS. To clear the initialized date, specify /NOINITIALIZED\_DATE.

This qualifier requires the right MDMS\_SET\_PROTECTED.

**/IO\_ERROR\_COUNT=number**

This qualifier allows you to set the number of I/O errors on the volume. The default value is zero.

**/JOB\_NAME="text"**

This qualifier allows you to specify the last job that accessed the volume. The job name can be from 0 to 31 characters. If it contains spaces, it must be enclosed in quotation marks. Specify "" to clear the job name.

This qualifier requires the right MDMS\_SET\_PROTECTED.

**/JUKEBOX=jukebox\_name**

**/NOJUKEBOX**

This is a protected field that should be modified only to recover on error. Use the MOVE VOLUME command to manipulate the jukebox name under normal operation. This qualifier allows you to specify that the volume is currently residing or last resided in the specified jukebox. The maximum length of a jukebox name is 31 characters. This is normally set up by MDMS. To clear the jukebox name, specify /NOJUKEBOX.

This qualifier requires the right MDMS\_SET\_PROTECTED.

**/LAST\_ACCESS\_DATE=date**

**/NOLAST\_ACCESS\_DATE**

Specifies the date the volume was last accessed. Specify a VMS absolute date and time. This is normally set up by MDMS. To clear the last access date, specify /NOLAST\_ACCESS\_DATE.

This qualifier requires the right MDMS\_SET\_PROTECTED.

**/MAGAZINE=magazine\_name**

**/NOMAGAZINE**

This is a protected field that should be modified only to recover on error. Use the MOVE VOLUME command to manipulate the magazine name under normal operation. This qualifier specifies the magazine name if the volume resides in a magazine. This is normally set up by MDMS. To clear the magazine name, specify /NOMAGAZINE.

This qualifier requires the right MDMS\_SET\_PROTECTED.

**/MEDIA\_TYPES=(media\_type[,...])**

**/NOMEDIA\_TYPES**

The media type qualifier allows you to add the media type(s) that the volume can support. Multiple media types are supported prior to the volume being initialized. After initialization, a volume can only support one media type. The /REMOVE or /REPLACE qualifiers can be used to remove or replace objects in the list, rather than adding them by default. To specify the volume supports no media types, enter /NOMEDIA\_TYPES. If a volume is created with no media types, the default media type from the domain record is used.

**/MOUNT\_COUNT=number**

Specifies the number of times the volume has been loaded by MDMS. Normally set up by MDMS. The default mount count is zero.



**/OFFSITE=(LOCATION=location],[NO]DATE=date)]**

**/NOOFFSITE**

This qualifier specifies the date that the volume is to be taken offsite and the offsite location. The location field is required when using the /OFFSITE qualifier if no location has been previously specified. The LOCATION keyword cannot be negated and if specified must point to a valid location object. Specify a VMS absolute or delta time in the date field. The NODATE keyword may be used to remove the date. This has the effect of disabling the schedule for the volume, while retaining the location. To clear the offsite date specify /NOOFFSITE. If a volume is under magazine control, the /OFFSITE qualifier is not allowed. The volume uses the values in the magazine. If /MAGAZINE is used on the SET VOLUME command when an existing /OFFSITE location and date are set, the values are cleared.

The default offsite location is that stored in the domain object.

**/ONSITE=(LOCATION=location],[NO]DATE=date)]**

**/NOONSITE**

This qualifier specifies the date that the volume is to be brought back onsite and the onsite location. The location field is required when using the /ONSITE qualifier if no location has been previously specified. The LOCATION keyword cannot be negated and if specified must point to a valid location object. Specify a VMS absolute or delta time in the date field. The NODATE keyword may be used to remove the date. This has the effect of disabling the schedule for the volume, while retaining the location. To clear the onsite date specify /NOONSITE. If a volume is under magazine control, the /ONSITE qualifier is not allowed. The volume uses the values in the magazine. If /MAGAZINE is used on the SET VOLUME command when an existing /ONSITE location and date are set, the values are cleared.

The default onsite location is that stored in the domain object.

**/OWNER\_NAME=node::username**

**/OWNER\_NAME=group::username**

**/NOOWNER\_NAME**

The /OWNER\_NAME qualifier specifies the owner of the object. Specify an owner as node::username or group::username. The node, group and username portions may be wildcarded.

The owner of an object has default access control of all allowances to the object, regardless of whether a specific access control entry is present or the state of the default domain access control. If you wish to remove this unconditional owner access, specify /NOOWNER\_NAME. Note that being an owner of an object overrides only the access control; the appropriate MDMS rights for a request on the object must still be honored.

By default, the owner of an object is the group\_name::username of the user entering the CREATE command on the object from a node in the group. However, once the volume is allocated, the node::user performing the allocation becomes the owner.

**/OWNER\_NAME\_UIC=uic**

**/NOOWNER\_NAME\_UIC**

This qualifier specifies the owner UIC of a volume. The owner UIC field must be a UIC in the format [USER] or [group, user]. This is normally set up by MDMS on allocate volume. To clear the owner UIC field, specify /NOOWNER\_NAME\_UIC.

This qualifier requires the right MDMS\_SET\_PROTECTED.

**/PLACEMENT=keyword**

This is a protected field that should be modified only to recover on error. Use the MOVE, LOAD or UNLOAD commands to manipulate the placement field under normal operation. This qualifier defines the current placement of the volume. This is normally managed by MDMS. Select one of the following options:

DRIVE

## MDMS SET VOLUME

ONSITE

MAGAZINE

JUKEBOX

OFFSITE

MOVING

If a magazine name is specified on the /MAGAZINE qualifier, the volume placement can be in one of three states:

MAGAZINE

DRIVE

MOVING

During a MOVE, LOAD or UNLOAD, a volume's placement may be set to MOVING indicating that the volume is being moved. If a volume is in a magazine, it is set to MOVING when the volume is being loaded or unloaded to/from a drive.

This qualifier requires the right MDMS\_SET\_PROTECTED.

**/POOL=pool\_name**

**/NOPOOL (D)**

The pool to which the volume belongs. The maximum length of the pool name is 31 characters. Spaces are not allowed in the pool name.

**/PREINITIALIZED**

**/NOPREINITIALIZED (D)**

This qualifier specifies whether the volume has been initialized before creation. If /PREINITIALIZED is specified, the volume is placed in the FREE state rather than the UNINITIALIZED state.

**/PROTECTION=protection**

The protection code for the volume. Use the standard OpenVMS protection code format.

**/PURCHASED\_DATE=date**

**/NOPURCHASED\_DATE**

The date when the volume was purchased. The date should be specified as an OpenVMS absolute time. Specify /NOPURCHASED\_DATE to clear the purchased date.

**/RECLENGTH=number**

This qualifier specifies the record length used on the volume. The default record length is zero.

**/RELEASE**

This qualifier puts the volume into the FREE state from the TRANSITION state. It is not valid to release an allocated volume. This qualifier always operates on volume sets.

**/REMOVE**

The /REMOVE qualifier works in conjunction with certain qualifiers that accept lists and removes specified attributes from them.

**/REPLACE**

The /REPLACE qualifier works in conjunction with certain qualifiers that accept lists and replaces the existing attributes list with the specified list. By default, attributes are added to the existing list.

**/RETAIN**

This qualifier puts the volume that is in the FREE state or TRANSITION state back into the allocated state with the former owner as the current owner. This qualifier always operates on volume sets.

**/SCRATCH\_DATE=date****/NOSCRATCH\_DATE**

The /SCRATCH\_DATE qualifier specifies the planned date to return the volume from the ALLOCATED state to either the TRANSITION state or the FREE state. Specify /NOSCRATCH\_DATE if the volume should not automatically transition from the ALLOCATED state.

**/SLOTS=(range[,...])****/NOSLOTS**

This qualifier specifies the jukebox or magazine slot that the volume currently resides in. Specify a number in the range of the jukebox or magazine. This is normally set up by MDMS. If the volume does not currently reside in a jukebox or magazine slot, specify /NOSLOTS.

If a volume range is specified, you can specify a slot range, and each volume will be placed in each slot in order. For a single volume, specify a single slot. This qualifier requires the right MDMS\_SET\_PROTECTED.

**/SPACES=(range[,...])****/NOSPACES**

This qualifier specifies the non-jukebox space in the specified location that the volume resides in. If the volume does not reside in a location space, specify /NOSPACES. Use a space range only when creating multiple volumes - each volume will be placed in each space in order. For a single volume, specify a single space.

**/STATE=keyword**

This is a protected field that should be modified only to recover on error. Use the ALLOCATE VOLUME or DEALLOCATE VOLUME command to manipulate the state field under normal operation. This qualifier allows you to modify the state of the volume. This is normally set up by MDMS and manual modification is not recommended. The keyword values are:

ALLOCATED

FREE

TRANSITION

UNAVAILABLE

UNINITIALIZED

This qualifier requires the right MDMS\_SET\_PROTECTED.

**/TIMES\_CLEANED=number**

This qualifier allows you to specify the number of times the volume has been cleaned. The default is zero.

**/TRANSITION\_TIME=delta\_time****/NOTTRANSITION\_TIME**

The /TRANSITION\_TIME qualifier specifies that the volume enters the TRANSITION state when the scratch date is reached, and is to stay in the TRANSITION state for the specified delta time. When the transition time has expired, the volume enters the FREE state. The /NOTTRANSITION\_TIME qualifier specifies that the volume enters the FREE state directly at the scratch date.

## MDMS SET VOLUME

### **/UNAVAILABLE**

Puts the volume in the UNAVAILABLE state. The previous state is retained for when the volume is made available again.

### **/USER\_NAME=username**

### **/NOUSER\_NAME**

Specifies the user for the volume. The username can be from 1-31 characters, and must reflect an authorized VMS username. To clear the username, enter /NOUSER\_NAME.

This qualifier requires the right MDMS\_SET\_PROTECTED.

### **/VOLSET**

### **/NOVOLSET (D)**

This qualifier specifies that all changes apply to the entire volume set. By default, attributes apply to a single volume or volume range specified. Exceptions to this are SET VOLUME/RETAIN and SET VOLUME/RELEASE, which always act on a volume set.

## Examples

```
$ MDMS SET VOLUME ABC001 /ONSITE=(LOCATION=SHELF2) -  
/SPACES=AA4
```

This command modifies the onsite location of volume ABC001.

```
$ MDMS SET VOLUME SFR024 /SLOTS=24
```

This command modifies the jukebox slot location for volume SFR024 to slot 24.

```
$ MDMS SET VOLUME HS0001-HS0007 /MAGAZINE=TX877B -  
/SLOTS=(0-6)
```

This command modifies the magazine and slot definition for the seven volume records HS0001 - HS0007.

---

## MDMS SHOW DOMAIN

The MDMS SHOW DOMAIN command displays information about the MDMS domain..  
Equivalent STORAGE Command: None

### Format:

MDMS SHOW DOMAIN

### Parameters

None.

### Description

The MDMS SHOW DOMAIN command displays information about the MDMS domain.

### Privileges

The request requires MDMS\_SHOW\_ALL.

If the /FULL qualifier is specified, MDMS\_SHOW\_RIGHTS is also required.

### Restrictions

None.

### Qualifiers

#### **/FULL**

Shows the rights assignments in the display.

This qualifier requires the right MDMS\_SHOW\_RIGHTS.

#### **/OUTPUT=file\_spec**

Specifies the output file in which to send the domain information. The default is SYSS\$OUTPUT.

#### **/SYMBOLS**

Stores selected domain information in process symbols. By default there is no output on the command. If you wish output with the symbols, use the /OUTPUT qualifier. The following symbols are created:

MDMS\_INQ\_CHECK\_ACCESS

MDMS\_INQ\_DATABASE\_DATE

MDMS\_INQ\_DEALLOCATE\_STATE

MDMS\_INQ\_DESCRIPTION

MDMS\_INQ\_LAST\_UPDATED

MDMS\_INQ\_MAIL

MDMS\_INQ\_MAX\_SCRATCH\_TIME

MDMS\_INQ\_MEDIA\_TYPE

MDMS\_INQ\_NODE\_NAME

MDMS\_INQ\_OFFSITE\_LOCATION

## MDMS SHOW DOMAIN

MDMS\_INQ\_ONSITE\_LOCATION

MDMS\_INQ\_OPCOM\_CLASS

MDMS\_INQ\_PROTECTION

MDMS\_INQ\_RELAXED\_ACCESS

MDMS\_INQ\_REQUEST\_ID

MDMS\_INQ\_SCHEDULER\_TYPE

MDMS\_INQ\_SCRATCH\_TIME

MDMS\_INQ\_TRANSITION\_TIME

Symbols are not available for access control and domain rights

### Examples

```
$ MDMS SHOW DOMAIN
```

This command displays a brief listing of the MDMS domain.

```
$ MDMS SHOW DOMAIN /OUTPUT=DOMAIN.DAT
```

This command outputs a full listing of the MDMS domain to file DOMAIN.DAT.

```
$ MDMS SHOW DOMAIN/FULL
```

This command displays a full listing of the MDMS domain including the rights attributes which are normally hidden.

---

## MDMS SHOW DRIVE

The MDMS SHOW DRIVE command displays information about specified drive(s) or all drives defined in the MDMS database.

Equivalent STORAGE Command: None

### Format:

MDMS SHOW DRIVE [drive\_name [...]]

### Parameters

#### **drive\_name**

Specifies the name(s) of the drive(s). Specify a drive name or a logical name. All drives are displayed if the drive\_name parameter is omitted. Wildcard characters are allowed in the drive\_name, where \* represents a variable-length wildcard, and % represents a single character wildcard.

The maximum length of the drive name or logical name is 31 characters. Valid characters are "\$,-,0-9,A-Z,\_,a-z".

A comma-separated list of drive names may be entered.

### Description

The MDMS SHOW DRIVE command displays information about specified drive(s) or all drives defined in the MDMS database.

### Privileges

The request requires MDMS\_SHOW\_ALL.

### Restrictions

/SYMBOLS are only defined on a show of a single drive.

### Qualifiers

#### **/BRIEF**

Display drive information in the brief 1-line format. This is the default display when a drive name is not specified.

#### **/CHECK**

#### **/NOCHECK (D)**

When the /CHECK qualifier is specified, the drive is physically accessed to determine its availability and its state.

If the state varies from what is stored in the database, the state is updated (and associated volume records are also updated) and the new values are displayed. If the check cannot be performed, the original database settings are displayed with a warning message. With the default /NOCHECK, no physical checking is performed, and the current database contents are displayed.

This qualifiers requires the right MDMS\_SET\_OWN.

#### **/FULL**

Display drive information in the full multi-line format. All attributes and status fields are displayed. This is the default display when a drive name is specified.

## MDMS SHOW DRIVE

### **/OUTPUT=file\_spec**

Specifies the output file in which to send the drive information. The default is SYS\$OUTPUT.

### **/SYMBOLS**

Stores selected drive information in process symbols. By default there is no output on the command. If you wish output with the symbols, use the /OUTPUT qualifier. The following symbols are created:

MDMS\_INQ\_ACCESS  
MDMS\_INQ\_ALLOCATED  
MDMS\_INQ\_ALLOC\_DEVICE\_NAME  
MDMS\_INQ\_ALLOC\_NODE\_NAME  
MDMS\_INQ\_ALLOC\_OWNER\_UIC  
MDMS\_INQ\_ALLOC\_PROCESS\_ID  
MDMS\_INQ\_ALLOC\_USER\_NAME  
MDMS\_INQ\_AUTOMATIC\_REPLY  
MDMS\_INQ\_AVAILABLE  
MDMS\_INQ\_DESCRIPTION  
MDMS\_INQ\_DEVICE\_NAME  
MDMS\_INQ\_DISABLED  
MDMS\_INQ\_DRIVE\_NAME  
MDMS\_INQ\_DRIVE\_NUMBER  
MDMS\_INQ\_GROUPS  
MDMS\_INQ\_JUKEBOX\_NAME  
MDMS\_INQ\_LOAD\_VOLUME\_ID  
MDMS\_INQ\_MEDIA\_TYPE  
MDMS\_INQ\_NODES  
MDMS\_INQ\_OWNER\_NAME  
MDMS\_INQ\_READONLY\_MEDIA\_TYPE  
MDMS\_INQ\_SHARED  
MDMS\_INQ\_STACKER  
MDMS\_INQ\_STATE  
MDMS\_INQ\_VOLUME\_ID

Symbols are not available for access control.



**Examples**

```
$ MDMS SHOW DRIVE $1$MUA5
```

This command displays a full listing of drive information for drive \$1\$MUA5 to SYS\$OUTPUT.

```
$ MDMS SHOW DRIVE /FULL $1$MUA5 /OUTPUT=DRIVES.DAT
```

This command displays a full listing of drive information for drive \$1\$MUA5 and puts the output into file DRIVE.DAT.

---

## MDMS SHOW GROUP

The MDMS SHOW GROUP command displays information about selected groups, or all groups defined in the MDMS database.

Equivalent STORAGE Command: None

### Format:

MDMS SHOW GROUP [group\_name [...]]

### Parameters

group\_name

Specify the group names. All groups are displayed if the group\_name parameter is omitted. Wildcard characters are allowed in the group\_name, where \* represents a variable-length wildcard, and % represents a single character wildcard.

The maximum length of the group name is 31 characters.

Valid characters are "\$,-,0-9,A-Z,\_,a-z".

A comma-separated list of group names may be entered.

### Description

The MDMS SHOW GROUP command displays information about specified groups in the MDMS database.

### Privileges

The request requires MDMS\_SHOW\_ALL.

### Restrictions

/SYMBOLS are only defined on a show of a single group.

### Qualifiers

#### **/BRIEF**

Display group information in the brief 1-line format. This is the default display when a group-name is not specified.

#### **/FULL**

Display group information in the full multi-line format. All attributes and status fields are displayed. This is the default display when a group name is specified.

#### **/OUTPUT=file\_spec**

Specifies the output file in which to send the group information. The default is SYS\$OUTPUT.

#### **/SYMBOLS**

Stores selected group information in process symbols. By default there is no output on the command. If you wish output with the symbols, use the /OUTPUT qualifier. The following symbols are created:

MDMS\_INQ\_DESCRIPTION

MDMS\_INQ\_GROUP\_NAME

MDMS\_INQ\_NODES

MDMS\_INQ\_OWNER\_NAME

Symbols are not available for access control.

## Examples

```
$ MDMS SHOW GROUP HOSER /FULL
```

This command displays a full listing group HOSER.

```
$ MDMS SHOW GROUP /OUTPUT=GROUPS.DAT
```

This command displays a brief listing of all group information and sends the output into file GROUPS.DAT

---

## MDMS SHOW JUKEBOX

The MDMS SHOW JUKEBOX command displays information about specified jukebox(es), selected jukeboxes, or all jukeboxes defined in the MDMS database.

Equivalent STORAGE Command: None

### Format:

MDMS SHOW JUKEBOX [jukebox\_name [...]]

### Parameters

jukebox\_name

Specifies the name(s) of the jukebox(es). All jukeboxes are displayed if the jukebox\_name parameter is omitted.

The maximum length of the jukebox name is 31 characters.

Valid characters are "\$,-,0-9,A-Z,\_,a-z".

Wildcard characters are allowed in the jukebox\_name, where \* represents a variable-length wildcard, and % represents a single character wildcard.

A comma-separated list of jukebox names may be entered.

### Description

The MDMS SHOW JUKEBOX command displays information about specified jukebox(es), or all jukeboxes defined in the MDMS database.

### Privileges

The request requires MDMS\_SHOW\_ALL.

### Restrictions

/SYMBOLS are only defined on a show of a single jukebox.

### Qualifiers

#### **/BRIEF**

Display jukebox information in the brief 1-line format. This is the default display when a jukebox name is not specified.

#### **/CONTENTS**

Display information about the contents of the jukebox - drives, slots, volumes and magazines.

#### **/FULL**

Display jukebox information in the full multi-line format. All attributes and status fields are displayed. This is the default display when a jukebox name is specified.

#### **/OUTPUT=file\_spec**

Specifies the output file in which to send the jukebox information. The default is SYSS\$OUTPUT.

#### **/SYMBOLS**

Stores selected jukebox information in process symbols. By default there is no output on the command. If you wish output with the symbols, use the /OUTPUT qualifier. The following symbols are created:

MDMS\_INQ\_ACCESS  
 MDMS\_INQ\_ACS\_ID  
 MDMS\_INQ\_AUTOMATIC\_REPLY  
 MDMS\_INQ\_CAP\_SIZE  
 MDMS\_INQ\_CONTROL  
 MDMS\_INQ\_DESCRIPTION  
 MDMS\_INQ\_DISABLED  
 MDMS\_INQ\_FREE\_VOLUMES  
 MDMS\_INQ\_GROUPS  
 MDMS\_INQ\_JUKEBOX\_NAME  
 MDMS\_INQ\_LIBRARY  
 MDMS\_INQ\_LOCATION\_NAME  
 MDMS\_INQ\_LSM\_ID  
 MDMS\_INQ\_NODES  
 MDMS\_INQ\_OWNER\_NAME  
 MDMS\_INQ\_ROBOT  
 MDMS\_INQ\_SLOT\_COUNT  
 MDMS\_INQ\_STATE  
 MDMS\_INQ\_THRESHOLD  
 MDMS\_INQ\_TOPOLOGY\_FACE  
 MDMS\_INQ\_TOPOLOGY\_LEVEL  
 MDMS\_INQ\_TOPOLOGY\_SLOT  
 MDMS\_INQ\_TOPOLOGY\_TOWER  
 MDMS\_INQ\_USAGE

Symbols are not available for access control.

## Examples

```
$ MDMS SHOW JUKEBOX JUKE_1
```

This command displays a full listing of jukebox information for jukebox JUKE\_1.

```
$ MDMS SHOW JUKEBOX /FULL /OUTPUT=JUKE_LIST.DAT
```

This command displays a full listing for all jukeboxes, which is stored in file JUKE\_LIST.DAT instead of SYSS\$OUTPUT.

---

## MDMS SHOW LOCATION

The MDMS SHOW LOCATION command displays information about specified location(s) or all locations defined in the MDMS database.

Equivalent STORAGE Command: None

### Format:

MDMS SHOW LOCATION [location [...]]

### Parameters

location

Specifies the name of the location. All locations are displayed if the location parameter is omitted.

The maximum length of the location is 31 characters.

Valid characters are "\$,-,0-9,A-Z,\_,a-z".

Wildcard characters are allowed in the location where \* represents a variable-length wildcard, and % represents a single character wildcard.

A comma-separated list of locations may be entered.

### Description

The MDMS SHOW LOCATION command displays information about specified locations or all locations defined in the MDMS database.

### Privileges

The request requires MDMS\_SHOW\_ALL.

### Restrictions

/SYMBOLS are only defined on a show of a single location.

### Qualifiers

#### **/BRIEF**

Display location information in the brief 1-line format. This is the default display when a location name is not specified.

#### **/FULL**

Display location information in the full multi-line format. All attributes and status fields are displayed. This is the default display when a location name is specified.

#### **/OUTPUT=file\_spec**

Specifies the output file in which to send the location information. The default is SYSS\$OUTPUT.

**/SYMBOLS**

Stores selected location information in process symbols. By default there is no output on the command. If you wish output with the symbols, use the /OUTPUT qualifier. The following symbols are created:

MDMS\_INQ\_DESCRIPTION

MDMS\_INQ\_LOCATION\_NAME

MDMS\_INQ\_OWNER\_NAME

MDMS\_INQ\_PARENT\_LOCATION

MDMS\_INQ\_SPACES

Symbols are not available for access control.

**Examples**

```
$ MDMS SHOW LOCATION
```

This command displays a brief listing of all locations.

```
$ MDMS SHOW LOCATION SHELF1, SHELF2, SHELF3, SHELF4 /FULL
```

This command displays a full listing of location information for locations SHELF1 to SHELF4.

```
$ MDMS SHOW LOCATION /FULL /OUTPUT=LOC_LIST.DAT
```

This command displays a full listing of all location information in file LOC\_LIST.DAT.

---

## MDMS SHOW MAGAZINE

The MDMS SHOW MAGAZINE command displays information about specified magazine(s), selected magazines, or all magazines defined in the MDMS database.

Equivalent STORAGE Command: STORAGE SHOW MAGAZINE

### Format:

MDMS SHOW MAGAZINE [magazine\_name [...]]

### Parameters

magazine\_name

Specifies the name(s) of the magazine(s). All magazines are displayed if the magazine\_name parameter is omitted, subject to the/SCHEDULE qualifier.

The maximum length of the magazine name is 31 characters.

Valid characters are "\$,-,0-9,A-Z,\_,a-z".

Wildcard characters are allowed in the magazine\_name, where \* represents a variable-length wildcard, and % represents a single character wildcard.

A comma-separated list of magazine names may be entered.

### Description

The MDMS SHOW MAGAZINE command displays information about specified magazine(s) or all magazines defined in the MDMS database.

### Privileges

The request requires MDMS\_SHOW\_ALL.

### Restrictions

/SYMBOLS are only defined on a show of a single magazine.

### Qualifiers

#### **/BRIEF**

Display magazine information in the brief 1-line format. This is the default display when a magazine name is not specified.

#### **/CONTENTS**

Display information about the contents of the magazine - slots and volumes.

#### **/FULL**

Display magazine information in the full multi-line format. All attributes and status fields are displayed. This is the default display when a magazine name is specified.

#### **/OUTPUT=file\_spec**

Specifies the output file in which to send the magazine information. The default is SYSS\$OUTPUT.



**/SCHEDULE[=keyword]**

The /SCHEDULE qualifier selects magazines whose offsite or onsite date has "expired" and the magazine is not in the new location. If both dates have expired, the later of the two dates is used. The optional keywords on the /SCHEDULE qualifier may be:

OFFSITE

ONSITE

If the OFFSITE keyword is used, then only those magazines scheduled to be moved offsite are selected. If the ONSITE keyword is used, then only those magazines scheduled to be moved onsite are selected. If the keyword is omitted, then the magazines scheduled to be moved onsite and offsite are selected. Do not specify a magazine name with this qualifier.

**/SYMBOLS**

Stores selected magazine information in process symbols. By default there is no output on the command. If you wish output with the symbols, use the /OUTPUT qualifier. The following symbols are created:

MDMS\_INQ\_DESCRIPTION

MDMS\_INQ\_JUKEBOX\_NAME

MDMS\_INQ\_MAGAZINE\_NAME

MDMS\_INQ\_OFFSITE\_DATE

MDMS\_INQ\_OFFSITE\_LOCATION

MDMS\_INQ\_ONSITE\_DATE

MDMS\_INQ\_ONSITE\_LOCATION

MDMS\_INQ\_OWNER\_NAME

MDMS\_INQ\_PLACEMENT

MDMS\_INQ\_PLACENAME

MDMS\_INQ\_POSITION

MDMS\_INQ\_SLOT\_COUNT

MDMS\_INQ\_SPACES

MDMS\_INQ\_START\_SLOT

Symbols are not available for access control.

**Examples**

```
$ MDMS SHOW MAGAZINE
```

This command displays a brief listing of all magazines.

```
$ MDMS SHOW MAGAZINE PDMAG* /OUTPUT=PDMAG1.DAT
```

This command outputs a full listing of magazine information for magazine matching the wildcard PDMAG\* to file PDMAG1.DAT.

```
$ MDMS SHOW MAGAZINE /SCHEDULE
```

This command outputs a brief listing of all magazines scheduled to be moved onsite or offsite.

---

## MDMS SHOW MEDIA\_TYPE

The MDMS SHOW MEDIA\_TYPE command displays information about specified media type(s) or all media types defined in the MDMS database.

Equivalent STORAGE Command: None

### Format:

MDMS SHOW MEDIA\_TYPE [media\_type [...]]

### Parameters

media\_type

Specifies the name(s) of the media\_type(s). All media types are displayed if the media\_type parameter is omitted.

The maximum length of the media\_type name is 31 characters.

Valid characters are "\$,-,0-9,A-Z,\_,a-z".

Wildcard characters are allowed in the media\_type, where \* represents a variable-length wildcard, and % represents a single character wildcard.

A comma-separated list of media types may be entered.

### Description

The MDMS SHOW MEDIA\_TYPE command displays information about specified media type(s) or all media types defined in the MDMS database.

### Privileges

The request requires MDMS\_SHOW\_ALL.

### Restrictions

/SYMBOLS are only defined on a show of a single media type.

### Qualifiers

#### **/BRIEF**

Display media type information in the brief 1-line format. This is the default display when a media\_type is not specified.

#### **/FULL**

Display media type information in the full multi-line format. All attributes and status fields are displayed. This is the default display when a media\_type is specified.

#### **/OUTPUT=file\_spec**

Specifies the output file in which to send the media type information. The default is SYSS\$OUTPUT.

#### **/SYMBOLS**

Stores selected media type information in process symbols. By default there is no output on the command. If you wish output with the symbols, use the /OUTPUT qualifier. The following symbols are created:

MDMS\_INQ\_CAPACITY

MDMS\_INQ\_COMPACTION

MDMS\_INQ\_DENSITY  
MDMS\_INQ\_DESCRIPTION  
MDMS\_INQ\_MEDIA\_TYPE  
MDMS\_INQ\_OWNER\_NAME  
MDMS\_INQ\_TAPE\_LENGTH

Symbols are not available for access control.

## Examples

```
$ MDMS SHOW MEDIA_TYPE TK88K_COMP /FULL
```

This command displays a full listing of information for media\_type TK88K\_COMP.

```
$ MDMS SHOW MEDIA_TYPE /FULL /OUTPUT=MEDIA.DAT
```

This command outputs a full listing of information for all media types to file MEDIA.DAT.

---

## MDMS SHOW NODE

The MDMS SHOW NODE command displays information about specified node(s) or all nodes defined in the MDMS database.

Equivalent STORAGE Command: None

### Format:

MDMS SHOW NODE [node\_name [...]]

### Parameters

node\_name

Specifies the name(s) of the node(s). All nodes are displayed if the node\_name parameter is omitted.

The maximum length of the node name is 31 characters.

Valid characters are "\$,-,0-9,A-Z,\_,a-z".

Wildcard characters are allowed in the node\_name, where \* represents a variable-length wildcard, and % represents a single character wildcard.

A comma-separated list of node names may be entered.

### Description

The MDMS SHOW NODE command displays information about specified node(s) or all nodes defined in the MDMS database.

### Privileges

The request requires MDMS\_SHOW\_ALL.

### Restrictions

/SYMBOLS are only defined on a show of a single node.

### Qualifiers

#### **/BRIEF**

Display node information in the brief 1-line format. This is the default display when a node name is not specified.

#### **/FULL**

Display node information in the full multi-line format. All attributes and status fields are displayed. This is the default display when a node name is specified.

#### **/OUTPUT=file\_spec**

Specifies the output file in which to send the node information. The default is SYSS\$OUTPUT.

#### **/SYMBOLS**

Stores selected node information in process symbols. By default there is no output on the command. If you wish output with the symbols, use the /OUTPUT qualifier. The following symbols are created:

MDMS\_INQ\_DATABASE\_SERVER

MDMS\_INQ\_DECNET\_FULLNAME

MDMS\_INQ\_DESCRIPTION  
MDMS\_INQ\_DISABLED  
MDMS\_INQ\_LOCATION\_NAME  
MDMS\_INQ\_NODE\_NAME  
MDMS\_INQ\_OPCOM\_CLASS  
MDMS\_INQ\_OWNER\_NAME  
MDMS\_INQ\_TCPIP\_FULLNAME  
MDMS\_INQ\_TRANSPORT

Symbols are not available for access control.

## Examples

```
$ MDMS SHOW NODE /FULL /OUTPUT=NODES.DAT
```

This command outputs a full listing of node information for all nodes to file NODES.DAT.

```
$ MDMS SHOW NODE DSORDS
```

This command displays a full listing of node information for node DSORDS.

```
$ MDMS SHOW NODE SIL*/FULL
```

This command displays a full listing of nodes matching the wildcard SIL\*.

---

## MDMS SHOW POOL

The MDMS SHOW POOL command displays information about specified pool(s) or all pools defined in the MDMS database.

Equivalent STORAGE Command: None

### Format:

MDMS SHOW POOL [pool\_name [...]]

### Parameters

pool\_name

Specifies the name of the pool. All pools are displayed if the pool\_name parameter is omitted.

The maximum length of the pool name is 31 characters. Valid characters are "\$,-,0-9,A-Z,\_,a-z".

Wildcard characters are allowed in the pool\_name, where \* represents a variable-length wildcard, and % represents a single character wildcard.

A comma-separated list of pool names may be entered.

### Description

The MDMS SHOW POOL command displays information about specified pool(s) or all pools defined in the MDMS database.

### Privileges

The request requires MDMS\_SHOW\_ALL.

### Restrictions

/SYMBOLS are only defined on a show of a single pool.

### Qualifiers

#### **/BRIEF**

Display pool information in the brief 1-line format. This is the default display when a pool name is not specified.

#### **/FULL**

Display pool information in the full multi-line format. All attributes and status fields are displayed. This is the default display when a pool name is specified.

#### **/OUTPUT=file\_spec**

Specifies the output file in which to send the pool information. The default is SYSS\$OUTPUT.

#### **/SYMBOLS**

Stores selected pool information in process symbols. By default there is no output on the command. If you wish output with the symbols, use the /OUTPUT qualifier. The following symbols are created:

MDMS\_INQ\_AUTHORIZED\_USERS

MDMS\_INQ\_DEFAULT\_USERS

MDMS\_INQ\_DESCRIPTION

MDMS\_INQ\_FREE\_VOLUMES

MDMS\_INQ\_OWNER\_NAME

MDMS\_INQ\_POOL\_NAME

MDMS\_INQ\_THRESHOLD

Symbols are not available for access control.

## Examples

```
$ MDMS SHOW POOL TEST_POOL /FULL
```

This command displays a full listing of pool information for pool TEST\_POOL.

---

## MDMS SHOW REQUEST

The MDMS SHOW REQUEST command displays information about all requests currently active on the database server node which includes virtually all requests in the domain.

Equivalent STORAGE Command: None

### Format:

MDMS SHOW REQUEST [request\_id]

### Parameters

request\_id

Specifies an identifier to show a specific request. You can obtain a list of request IDs by issuing a SHOW REQUEST command without a request ID.

### Description

The MDMS SHOW REQUESTS command displays information about requests currently active in the domain. Certain recently-completed requests may also be shown. If no request\_id is specified, all requests are shown subject to privilege. In addition, requests from a specific user can be shown.

### Privileges

The request requires MDMS\_SHOW\_ALL, MDMS\_SHOW\_POOL, or MDMS\_SHOW\_OWN.

If the user has only MDMS\_SHOW\_OWN or MDMS\_SHOW\_POOL, only requests issued by the user are shown. To see requests of other users MDMS\_SHOW\_ALL is required.

### Restrictions

None.

### Qualifiers

#### **/BRIEF**

Displays requests information in the brief 1-line format. This is the default display if no request\_id is specified.

#### **/DEBUG**

Displays addresses of certain objects for debugging purposes.

#### **/FULL**

Displays request information in the full multi-line format. This is the default display if a request\_id is specified.

#### **/OUTPUT=file\_spec**

Specifies the output file in which to send the requests information. The default is SYSS\$OUTPUT.

#### **/USER\_NAME=user\_name**

Restricts the request list to those issued by the specified user.



## Examples

```
$ MDMS SHOW REQUESTS
```

This command displays a brief listing of all requests active in the domain.

```
$ MDMS SHOW REQUEST/FULL
```

This command displays a full listing of all requests active in the domain.

```
$ MDMS SHOW REQUEST 45
```

This command displays a full listing of request identifier 45

```
$ MDMS SHOW REQUEST/USER_NAME=FROEHLIN
```

This command displays a brief listing of all requests issued by user FROEHLIN.

---

## MDMS SHOW SCHEDULE

The MDMS SHOW SCHEDULE command displays information about selected schedules, or all schedules defined in the MDMS database.

Equivalent ABS Command: SHOW STORAGE\_CLASS

### Format:

MDMS SHOW SCHEDULE [schedule\_name [...]]

### Parameters

schedule\_name

Specify the schedule names. All schedules are displayed if the schedule\_name parameter is omitted. Wildcard characters are allowed in the schedule\_name, where \* represents a variable-length wildcard, and % represents a single character wildcard.

The maximum length of the schedule name is 63 characters.

Valid characters are "\$,-,0-9,A-Z,\_,a-z".

A comma-separated list of schedule names may be entered.

### Description

The MDMS SHOW SCHEDULE command displays information about specified schedules in the MDMS database.

### Privileges

The request requires MDMS\_SHOW\_ALL.

### Restrictions

/SYMBOLS are only defined on a show of a single schedule.

### Qualifiers

#### **/BRIEF**

Display schedule information in the brief 1-line format. This is the default display when a schedule name is not specified.

#### **/FULL**

Display schedule information in the full multi-line format. All attributes and status fields are displayed. This is the default display when a schedule name is specified.

#### **/OUTPUT=file\_spec**

Specifies the output file in which to send the schedule information. The default is SYSS\$OUTPUT.

#### **/SYMBOLS**

Stores selected schedule information in process symbols. By default there is no output on the command. If you wish output with the symbols, use the /OUTPUT qualifier. The following symbols are created:

MDMS\_INQ\_AFTER\_SCHEDULE\_NAME

MDMS\_INQ\_AFTER\_SCHEDULE\_OPTION

MDMS\_INQ\_COMMAND

MDMS\_INQ\_DATES  
MDMS\_INQ\_DAYS  
MDMS\_INQ\_DESCRIPTION  
MDMS\_INQ\_EXCLUDE  
MDMS\_INQ\_INCLUDE  
MDMS\_INQ\_LAST\_START\_DATE  
MDMS\_INQ\_MONTHS  
MDMS\_INQ\_NEXT\_START\_DATE  
MDMS\_INQ\_OWNER\_NAME  
MDMS\_INQ\_SCHEDULE\_NAME  
MDMS\_INQ\_TIMES

Symbols are not available for access control.

## Examples

```
$ MDMS SHOW SCHEDULE RICHS_SCHEDULE /FULL
```

This command displays a full listing schedule RICHS\_SCHEDULE.

```
$ MDMS SHOW SCHEDULE /OUTPUT=SCHEDULES.DAT
```

This command displays a brief listing of all schedule information and sends the output into file SCHEDULES.DAT

---

## MDMS SHOW SERVER

The MDMS SHOW SERVER command displays information about the local configuration of the MDMS server on the specified node(s).

Equivalent STORAGE Command: None

### Format:

MDMS SHOW SERVER

### Parameters

None.

### Description

The MDMS SHOW SERVER command displays information about the local configuration of the MDMS server on the specified node(s). The information is derived from the specified nodes local startup configuration files rather than the MDMS database.

### Privileges

The request requires MDMS\_SHOW\_ALL, MDMS\_SHOW\_POOL or MDMS\_SHOW\_OWN.

### Qualifiers

**/NODES=(node\_name[,...])**

Displays information about the server on the specified node(s). The node name is either a DECnet (Phase IV) node name, a DECnet-Plus (Phase V) node name, or a TCP/IP node name. A TCP/IP node name can include a port number range. If no port number range is specified the default is 2501-2510.

**/OUTPUT=file\_spec**

Specifies the output file in which to send the version information. The default is SYS\$OUTPUT.

**/SYMBOLS**

Stores selected server information in process symbols. By default there is no output on the command. If you wish output with the symbols, use the /OUTPUT qualifier. The following symbols are created:

MDMS\_INQ\_DATABASE\_ACCESS  
MDMS\_INQ\_DATABASE\_LOCATION  
MDMS\_INQ\_DECNET\_FULLNAME  
MDMS\_INQ\_LOGFILE  
MDMS\_INQ\_SERVER\_NODE\_NAME  
MDMS\_INQ\_SERVER\_SEARCH\_LIST  
MDMS\_INQ\_SERVER\_VERSION  
MDMS\_INQ\_TCPIP\_FULLNAME  
MDMS\_INQ\_TRANSPORTS

## Examples

```
$ MDMS SHOW SERVER
```

This command displays MDMS version information on the current node.

```
$ MDMS SHOW SERVER /NODES=(COOKIE, DSORDS)
```

This command displays local MDMS server information for nodes COOKIE and DSORDS.

```
$ MDMS SHOW SERVER -
```

```
/NODES=(SENILE,CORP:.DOM.PARANOID, -
```

```
DISORDERS.DOM.CORP:3001-3010)
```

This command displays information about servers on DECnet node SENILE, DECnet-Plus node PARANOID and TCP/IP node DISORDERS.

---

## MDMS SHOW VERSION

The MDMS SHOW VERSION command displays information about the version of various MDMS components on the specified node(s).

Equivalent STORAGE Command: None

### Format:

MDMS SHOW VERSION

### Parameters

None.

### Description

The MDMS SHOW VERSION command displays information about the version of various MDMS components on the specified node(s).

### Privileges

The request requires MDMS\_SHOW\_ALL, MDMS\_SHOW\_POOL or MDMS\_SHOW\_OWN.

### Restrictions

None.

### Qualifiers

**/NODES=(node\_name[,...])**

Displays information about this server. The node name is either a DECnet (Phase IV) node name, a DECnet-Plus (Phase V) node name, or a TCP/IP node name. A TCP/IP node name can include a port number range. If no port number range is specified the default is 2501-2510.

**/OUTPUT=file\_spec**

Specifies the output file in which to send the version information. The default is SYSS\$OUTPUT.

### Examples

```
$ MDMS SHOW VERSION
```

This command displays MDMS version information on the current node.

```
$ MDMS SHOW VERSION /NODE=(COOKIE::, DSORDS)
```

This command displays MDMS version information for nodes COOKIE and DSORDS.

```
$ MDMS SHOW VERSION -
```

```
/NODES=(SENILE,CORP:.DOM.PARANOID, -
```

```
DISORDERS.COM.CORP:3001-3010)
```

This command displays the version of the servers on DECnet node SENILE, DECnet-Plus node PARANOID and TCP/IP node DISORDERS.

---

## MDMS SHOW VOLUME

The MDMS SHOW VOLUME command displays information about specified volume(s), selected volumes, or all volumes defined in the MDMS database.

Equivalent STORAGE Command: STORAGE SHOW VOLUME

### Format:

```
MDMS SHOW VOLUME [volume_id[,...]]
```

### Parameters

volume\_id

Specifies the identifier(s) of the volume(s). Alternatively, a volume range, separated by a dash, may be specified. A volume range is a numeric range for up to the last three characters of the volume ID. Example ranges are (ABC001-ABC250), (ABC120-ABC125).

The maximum length of the volume ID is 6 characters. Valid characters are " ,!,",%-,A-Z,\_".

Wildcard characters are allowed in the volume\_id, where \* represents a variable-length wildcard, and % represents a single character wildcard.

All volumes are displayed if the volume\_id parameter is omitted, subject to qualifier selection.

A comma-separated list of volume IDs or ranges may be entered.

### Description

The MDMS SHOW VOLUME command displays information about specified volume(s), selected volumes, or all volumes defined in the MDMS database.

If the volume\_id parameter is supplied, only information on the specified volume(s) is displayed, which are subject to further selection based on the qualifiers.

If no volume\_id parameter is specified, the qualifiers are used for selection; if no qualifiers are specified, all volumes are displayed.

### Privileges

The request requires MDMS\_SHOW\_ALL, MDMS\_SHOW\_VOLUME, MDMS\_SHOW\_POOL or MDMS\_SHOW\_OWN.

If the user has only MDMS\_SHOW\_OWN, only volumes allocated and owned by the user are displayed. If the user has MDMS\_SHOW\_POOL, then volumes in pools authorized to the user are displayed. To display any other volumes requires MDMS\_SHOW\_VOLUME or MDMS\_SHOW\_ALL.

### Restrictions

/SYMBOLS are only defined on a show of a single volume.

/ALLOCATED, /NOALLOCATED and /SYMBOLS are mutually exclusive.

The MDMS server will refuse this command if mutually exclusive qualifiers are used in the affirmative forms.

## Qualifiers

### **/ABS\_VOLSET**

In a wildcard specification, this qualifier displays the ABS volset records, those beginning with "&+", in addition to normal volumes records. By default, these pseudo volume records are not displayed. When used with a volume name, range or list, the associated ABS volset records are displayed instead of the normal volume records.

### **/ALLOCATED**

### **/NOALLOCATED**

Displays only volumes that are currently either allocated or not allocated respectively.

### **/BRIEF**

Display volume information in the brief 1-line format. This is the default if no volume\_id parameter is specified.

### **/FULL**

Display volume information in the full multi-line format. All attributes and status fields are displayed. This is the default when a volume\_id parameter is specified.

### **/OUTPUT=file\_spec**

Specifies the output file in which to send the volume information. The default is SYS\$OUTPUT.

### **/SCHEDULE[=keyword]**

The /SCHEDULE qualifier selects volumes whose offsite or onsite date has "expired" and the volumes are not in the new location. If both dates have expired, the later of the two dates is used. The optional keyword on the schedule qualifier may be:

OFFSITE

ONSITE

If the OFFSITE keyword is used, then only those volumes schedule to be moved offsite are selected. If the ONSITE keyword is used, then only those volumes scheduled to be moved onsite are selected. If the keyword is omitted, the volumes scheduled to be moved onsite and offsite are selected. Do not specify a volume\_id with this qualifier.

### **/SYMBOLS**

Stores selected volume information in process symbols. By default there is no output on the command. If you wish output with the symbols, use the /OUTPUT qualifier. The following symbols are created:

MDMS\_INQ\_ACCOUNT

MDMS\_INQ\_ALLOCATED\_DATE

MDMS\_INQ\_BLOCK\_FACTOR

MDMS\_INQ\_BRAND

MDMS\_INQ\_CLEANED\_DATE

MDMS\_INQ\_CREATION\_DATE

MDMS\_INQ\_DEALLOCATION\_DATE

MDMS\_INQ\_DESCRIPTION

MDMS\_INQ\_DRIVE

MDMS\_INQ\_FORMAT

MDMS\_INQ\_FREED\_DATE

MDMS\_INQ\_INITIALIZED\_DATE



MDMS\_INQ\_IO\_ERROR\_COUNT  
 MDMS\_INQ\_JOB\_NAME  
 MDMS\_INQ\_JUKEBOX\_NAME  
 MDMS\_INQ\_LAST\_ACCESS\_DATE  
 MDMS\_INQ\_MAGAZINE\_NAME  
 MDMS\_INQ\_MEDIA\_TYPE  
 MDMS\_INQ\_MOUNT\_COUNT  
 MDMS\_INQ\_NEXT\_VOLUME  
 MDMS\_INQ\_OFFSITE\_DATE  
 MDMS\_INQ\_OFFSITE\_LOCATION  
 MDMS\_INQ\_ONSITE\_DATE  
 MDMS\_INQ\_ONSITE\_LOCATION  
 MDMS\_INQ\_OWNER\_UIC  
 MDMS\_INQ\_PLACEMENT  
 MDMS\_INQ\_PLACENAME  
 MDMS\_INQ\_POOL\_NAME  
 MDMS\_INQ\_PREVIOUS\_VOLUME  
 MDMS\_INQ\_PROTECTION  
 MDMS\_INQ\_PURCHASED\_DATE  
 MDMS\_INQ\_RECORD\_LENGTH  
 MDMS\_INQ\_SCRATCH\_DATE  
 MDMS\_INQ\_SLOTS  
 MDMS\_INQ\_SPACES  
 MDMS\_INQ\_STATE  
 MDMS\_INQ\_TIMES\_CLEAVED  
 MDMS\_INQ\_TRANSITION\_TIME  
 MDMS\_INQ\_USER\_NAME  
 MDMS\_INQ\_VOLUME\_ID

Symbols are not available for access control.

**/USER\_NAME=username**

Selects volumes owned by this user.

**/VOLSET**

**/NOVOLSET(D)**

This qualifier specifies that the show applies to all volumes of the volume set. By default, the show command only applies to specified volume(s).

## MDMS SHOW VOLUME

### Examples

```
$ MDMS SHOW VOLUME FRM001 /SYMBOLS
```

This command stores selected information in DCL symbols for volume FRM001. No output is displayed by default.

```
$ MDMS SHOW VOLUME /ALLOCATED /USER_NAME=HSM$SERVER /FULL
```

This command displays a full listing of volume information for all volumes allocated to user HSM\$SERVER.

```
$ MDMS SHOW VOLUME /NOALLOCATED
```

This command displays a brief listing of volume information for the all volumes that are not allocated (i.e. those that are in the UNINITIALIZED, FREE or TRANSITION states).

```
$ MDMS SHOW VOLUME BDJ*/FULL
```

This command displays a full listing of the volumes matching the wildcard BDJ\*.

---

## Start\_Session

Equivalent STORAGE Command: None.

### Format

MDMS

### Parameter

None.

### Description

The MDMS starts an MDMS session. The default session is a DCL session, from which you may enter multiple MDMS commands without the MDMS verb. Your prompt is MDMS>. If you enter the /INTERFACE=GUI qualifier, you will instead initiate a GUI session on the system.

### Qualifier

#### **/INTERFACE=GUI**

This qualifier is required to start the Graphical User Interface on OpenVMS.

Before initiating the MDMS/INTERFACE=GUI command, the following command should be issued:

```
$ SET DISPLAY/CREATE/NODE=monitor_node_name-/TRANSPORT=transport
```

monitor\_node\_name:

Is the name of the node on which the monitor resides. Depending on the transport you select, you may need to enter the TCP/IP fullname, or the DECnet-Plus fullname, rather than the DECnet Phase IV node name.

transport:

Is one of the following:

- DECnet - if you are using DECnet between the OpenVMS node running the GUI and the node whose monitor you are displaying the GUI.
- TCPIP - if you are using TCP/IP between the OpenVMS node running the GUI and the node whose monitor you are displaying the GUI.
- LOCAL - if the monitor is on the same node as the one running the GUI.

Note that native GUIs are also provided that execute directly on Microsoft Windows systems. To activate these GUIs, do the following:

1. Double-click on the drive in which MDMSView is installed. The default is the C: drive.
2. Double-click on the folder MDMSView to open it.
3. Double-click on the batch file to start the GUI.

After activating the GUI, you may log into any OpenVMS system in the MDMS domain to manipulate MDMS. Both the Windows system and the OpenVMS system must support the TCP/IP protocol for the GUI to access MDMS.

While using the GUI (from any system), you cannot set the SYSPRV privilege to grant all rights from the GUI. As such, the account you are logging into should have SYSPRV defined as an

authorized privilege to support the SYSPRV privilege. If this is not desired, then the appropriate MDMS rights need to be defined in the account.

## Examples

1. \$ MDMS

MDMS>

This command initiates an MDMS DCL session.

2. \$ SET DISPLAY/NODE=NOD001.XYZ.COM/TRANSPORT=TCPIP  
\$ MDMS/INTERFACE=GUI

This command initiates an MDMS GUI session on the current system (running OpenVMS V7.2-1 or later), on the monitor connected to node NOD001 (which can be running an OpenVMS version supporting X-windows, a UNIX operating system or a Microsoft Windows operating system) running the TCP/IP protocol.

---

## MDMS UNBIND VOLUME

The MDMS UNBIND VOLUME command unbinds a volume from a volume set.

Equivalent STORAGE Command: STORAGE SPLIT

### Format:

MDMS UNBIND VOLUME volume\_id

### Parameters

volume\_id

Specifies the volume ID of the volume to unbind.

The maximum length of a volume ID is 6 characters. Valid characters are " ,! ,% -? ,A-Z ,\_ ".

### Description

The MDMS UNBIND VOLUME command unbinds a volume from a volume set. By default, when a volume is unbound from a volume set, all volumes in the volume set are unbound. To cause the volume set to be split into two volume sets, use the /NOVOLSET qualifier. Be aware that unbinding a volume from a volume set could cause data loss if the volume set contains valid data on every volume. Unbound volumes will remain in the ALLOCATED state.

### Privileges

The request requires MDMS\_UNBIND\_ALL or MDMS\_UNBIND\_OWN.

If the user has only MDMS\_UNBIND\_OWN, he can unbind only those volumes allocated to him. Unbinding any other volume requires MDMS\_UNBIND\_ALL.

### Restrictions

None

### Qualifiers

**/USER\_NAME=username**

The user that owns the volume set. This qualifier is used to unbind a volume from a volume set on behalf of the user. The maximum length of the username is 31 characters.

This qualifier requires the right MDMS\_UNBIND\_ALL.

**/VOLSET (D)**

**/NOVOLSET**

By default, the entire volume set containing the volume will be split into single volumes. Use the /NOVOLSET qualifier to split the volume set into two volume sets, with the second set beginning with the specified volume.

Note that it is not possible to unbind the first volume of a volume set using the /NOVOLSET qualifier.

## MDMS UNBIND VOLUME

### Examples

```
$ MDMS UNBIND VOLUME VOL002
```

Volume set contains VOL001, VOL002, VOL003, VOL004. This command unbinds all the volumes in the volume set, leaving VOL001, VOL002, VOL003 and VOL004 all as single volumes. The volumes remain allocated.

```
$ MDMS UNBIND VOLUME VOL003 /USER_NAME=SMITH /NOVOLSET
```

Volume set contains VOL001, VOL002, VOL003, VOL004 owned by user SMITH. This command unbinds the volume set starting at VOL003 from the volume set. The remaining volume sets contain volumes VOL001 and VOL002 and the second set contains VOL003 and VOL004.

---

## MDMS UNLOAD DRIVE

The MDMS UNLOAD DRIVE command unloads the volume contained in the specified drive.  
Equivalent STORAGE Command: STORAGE UNLOAD DRIVE

### Format:

MDMS UNLOAD DRIVE drive\_name

### Parameters

drive\_name

Specifies the name of the drive to unload. Specify a drive name or a logical name.

The maximum length of the drive name or logical name is 31 characters.

Valid characters are "\$,-,0-9,A-Z,\_,a-z".

### Description

The MDMS UNLOAD DRIVE command unloads the volume currently contained in the specified drive.

### Privileges

The request requires MDMS\_UNLOAD\_ALL.

MDMS\_ASSIST is also required unless /NOASSIST is specified.

### Restriction

None

### Qualifiers

**/ASSIST (D)**

**/NOASSIST**

The default /ASSIST qualifier requests operator assistance to prompt the operator to unload the volume from the drive. If /NOASSIST is specified, the operator is not notified.

This qualifier requires the right MDMS\_ASSIST.

**/REPLY=symbol**

The name of the symbol to receive the operator's reply when operator intervention is needed. The symbol will contain the operator reply to the DCL REPLY/TO or DCL REPLY/ABORT commands. The maximum length of a symbol name is 31 characters. This qualifier is only applicable when /ASSIST is specified.

**/WAIT (D)**

**/NOWAIT**

The /NOWAIT qualifier returns an informational message indicating that the unload is being queued. The /WAIT qualifier causes the MDMS UNLOAD DRIVE command to wait until a drive is unloaded.

## MDMS UNLOAD DRIVE

### **Example**

```
$ MDMS UNLOAD DRIVE $1$MUA151:
```

This command unloads the volume in drive \$1\$MUA151:. There is no timeout value, and no operator assistance is requested.



---

## MDMS UNLOAD VOLUME

The MDMS UNLOAD VOLUME command unloads the specified volume from a drive.  
Equivalent STORAGE Command: STORAGE UNLOAD VOLUME

### Format:

MDMS UNLOAD VOLUME volume\_id

### Parameters

volume\_id

Specifies the volume ID of the volume to unload.

The maximum length of the volume ID is 6 characters. Valid characters are ",!","%-?,A-Z,\_".

If there is a discrepancy between the "Load Volume" and "Volume" fields in a SHOW DRIVE display, try the "Load Volume" field first for unload. If this does not work, try the "Volume" field next.

Finally, try UNLOAD DRIVE.

### Description

The MDMS UNLOAD VOLUME command will unload a volume from a drive. If the volume resides in a jukebox, it will be returned to its jukebox slot.

### Privileges

The request requires MDMS\_UNLOAD\_ALL, MDMS\_UNLOAD\_POOL or MDMS\_UNLOAD\_OWN.

If the user only has MDMS\_UNLOAD\_OWN, only those volumes allocated to the user can be unloaded. With MDMS\_UNLOAD\_POOL, the user can unload a volume in a pool to which he is authorized. To unload any other volume requires MDMS\_UNLOAD\_ALL.

MDMS\_ASSIST is also required unless /NOASSIST is specified.

### Restrictions

The volume cannot be unloaded if it is currently mounted.

### Qualifiers

**/ASSIST (D)**

**/NOASSIST**

The default /ASSIST qualifier requests operator assistance to prompt the operator to unload the volume from the drive. If /NOASSIST is specified, the operator is not notified.

This qualifier requires the right MDMS\_ASSIST.

**/REPLY=symbol**

The name of the symbol to receive the operator's reply when operator intervention is needed. The symbol will contain the operator reply to the DCL REPLY/TO or DCL REPLY/ABORT commands. The maximum length of a symbol name is 31 characters. This qualifier is only applicable when /ASSIST is specified.

## MDMS UNLOAD VOLUME

### **/WAIT (D)**

### **/NOWAIT**

The **/NOWAIT** qualifier returns an informational message indicating that the unload is being queued. The **/WAIT** qualifier causes the MDMS UNLOAD VOLUME command to wait until a volume is unloaded.

### **Example**

```
$ MDMS UNLOAD VOLUME VOL003 /NOWAIT
```

This command unloads volume VOL003 from the drive on which it is loaded and the command does not wait until the unload is complete.

# A

---

## MDMS Files and Logical Names

The MDMS installation procedure installs files and defines logical names on your system. This appendix lists the names of the files installed and logical names that are added to the system logical name table. Section A.1 lists names of the files installed and Section A.2 lists the logical names that are added to the system logical names table.

---

### Note

**They are automatically entered into these logical name tables whenever the system reboots or whenever the software is invoked. The LNM\$JOB table also contains logical names that are defined when a user logs in to a system running MDMS software.**

---

### A.1 MDMS File Names

Table A–1 contains the names of all MDMS files created on the system after MDMS V4.0A is successfully installed.

---

### Note

**Some files may not be installed depending on the options you select.**

---

**Table A–1 MDMS Installed Files**

---

File Name	File Name
	<b>SY\$HELP</b>
MDMS.HLP	MDMSA040.RELEASE_NOTES
MDMSA040.RELEASE_NOTES.PS	
	<b>SY\$LIBRARY</b>
MDM\$SHR.EXE	
	<b>SY\$MESSAGE</b>
MDM\$MSG.EXE	RDF\$MSG.EXE
	<b>SY\$MANAGER</b>
MDM\$SYSTARTUP.COM	MDM\$SYSTARTUP:TEMPLATE
	<b>SY\$SHARE</b>
MDM\$SHR.EXE	MRD\$RTL.EXE

---

# MDMS Files and Logical Names

## A.1 MDMS File Names

**Table A-1 MDMS Installed Files**

File Name	File Name
<b>SY\$STARTUP</b>	
MDM\$SHUTDOWN.COM MDM\$UNINSTALL.COM	MDM\$STARTUP.COM
<b>SY\$SYSTEM</b>	
MDM\$SERVER.EXE, MDM\$CONVERT_V3_TO_V2.EXE, MDM\$CONVERT_V2_TO_V3.EXE	MDM\$DCL.EXE MDM\$SERVER.EXE
<b>SY\$TEST</b>	
MDM\$IVP.COM	
<b>MDM\$ROOT:[DATABASE]</b>	
MDM\$DOMAIN_DB.DAT MDM\$GROUP_DB.DAT MDM\$LOCATION_DB.DAT MDM\$MEDIA_DB.DAT MDM\$POOL_DB.DAT	MDM\$DRIVE_DB.DAT MDM\$JUKEBOX_DB.DAT MDM\$MAGAZINE_DB.DAT MDM\$NODE_DB.DAT MDM\$VOLUME_DB.DAT
<b>MDM\$ROOT:[GUL.VMS]</b>	
ALPDCL01_071.A-DCX_AXPEXE ALPACRT08_071.A-DCX_AXPEXE ALPSYB02_071.A-DCX_AXPEXE DEC-AXPVMS-JAVA-V0101-6- 1.PCSI_DCX_AXPEXE MDM.INI MDM_GUI_HELP.HTML	ALPBASE02_071.A-DCX_AXPEXE ALPSYSA02_071.A-DCX_AXPEXE ALPTHREADS_03071.A-DCX_AXPEXE DEC-AXPVMS-VMS712_PTHREADS-V0100-- 4.PCSI-DCX_AXPEXE MDM.ZIP SYMANTEC.ZIP
<b>MDM\$ROOT:[GUL.VMS.GRAPHICS]</b>	
CONFWIZ4.GIF DOMAIN2.GIF GROUP.GIF LOCATION.GIF MEDIATYPE.GIF POOL.GIF SERVJUKE.GIF VOLROT.GIF	CONFWIZ5.GIF DRIVE.GIF JUKEBOX.GIF MAGAZINE.GIF NODE.GIF REQUESTS.GIF SPLASH.GIF VOLUME.GIF
<b>MDM\$ROOT:[GUL.ALPHA_NT]</b>	
JRE116ALPHANT.EXE	SETUP_ALPHA_NT.EXE
<b>MDM\$ROOT:[GUL.WINTEL]</b>	
JRE117WINTEL.EXE	SETUP_INTEL.EXE
<b>MDM\$ROOT:[PATCHES.ALPHA]</b>	
ALPY2K02_062.A	ALPY2K02_062.CVRLET_TXT

**MDMS Files and Logical Names**  
**A.1 MDMS File Names**

**Table A-1 MDMS Installed Files**

File Name	File Name
<b>MDMS\$ROOT:[PATCHES.VAX]</b>	
VAXLIBR06_070.A	VAXLIBR06_070.B
VAXLIBR06_070.C	VAXLIBR06_070.CVRLET.TXT
VAXLIBR06_070.D	VAXLIBR06_070.E
VAXLIBR06_070.F	VAXLIBR06_070.G
<b>MDMS\$ROOT:[SYSTEM]</b>	
MDMS\$ALL_OTHER_DB.FDL	MDMS\$CONVERT_V3_TO_V2.COM
MDMS\$CONVERT_V2_TO_V3.COM	MDMS\$COPY_DB_FILES.COM
MDMS\$CREATE_DB_FILES.COM	MDMS\$REPLACE_SLS_LOADER.COM
MDMS\$START_GUI.COM	MDMS\$VOLUME_DB.FDL
<b>MDMS\$ROOT:[TTI_RDEV.ALPHA]</b>	
CONFIG_EXAMPLE.DAT	RDALLOCATE.COM
RDCDRIVER_AXP.OPT	RDCDRIVER_AXP61.OPT
RDCDRIVER_AXP70.OPT	RDCLIENT_SHUTDOWN.COM
RDCLIENT_STARTUP.COM	RDCTL_EXE.OPT
RDEV_AXP70.OLB	RDEV_AXP61.OLB
RDDEALLOCATE.COM	RDEV_BUILD.COM
RDEV_CHECK_STATE.COM	RDEV_CLIENT.COM
RDEV_CONFIGURE.COM	RDEV_COPYRIGHT.COM
RDEV_GATHER.COM	RDEV_LOGICALS.COM
RDEV_RMT_SHUTDOWN.COM	RDEV_RMT_STARTUP.COM
RDEV_SERVER.COM	RDEV_UCXSTUB_AXP61.OLB
RDEV_UCXSTUB_AXP70.OLB	PDFREE.COM
RDF_UCX_RSHD_STARTUP.COM	RDLOG.COM
RDRMT_STARTUP.COM	RDSERVER_SHUTDOWN.COM
RDSERVER_STARTUP.COM	RDSHOW.COM
RLINK_AXP.OPT	RMTSRV_AXP.OPT
SHRLINK_AXP.OPT	
<b>MDMS\$ROOT:[TTI_RDEV.VAX]</b>	
CONFIG_EXAMPLE.DAT	DRVRDEFS_V62.STB
DRVRDEFS_V62.STB	DRVRDEFS_V62.STB
DRVRDEFS_V62.STB	DRVRDEFS_V62.STB
DRVRDEFS_V62.STB	DRVRDEFS_V62.STB
RDALLOCATE.COM	RDCDRIVER_V62.EXE
RDCDRIVER_V62.STB	RDCDRIVER_V62.STB
RDCDRIVER_V62.STB	RDCDRIVER_V62.STB
RDCDRIVER_V62.STB	RDCDRIVER_V62.STB
RDCDRIVER_V62.STB	RDCDRIVER_V62.STB
RDCDRIVER_V62.STB	RDCDRIVER_V62.STB
RDCDRIVER_V62.STB	RDCDRIVER_VAX.OPT
RDCLIENT_SHUTDOWN.COM	RDCLIENT_STARTUP.COM
RDCLIENT_V62.EXE	RDCONTROL_V62.EXE
RDCTL_EXE.OPT	RDDEALLOCATE.COM
RDEV_BUILD.COM	RDEV_CHECK_STATE.COM
RDEV_CLIENT.COM	RDEV_CONFIGURE.COM
RDEV_CONTROL_SHR_V62.EXE	RDEV_COPYRIGHT.COM
RDEV_GATHER.COM	RDEV_LOGICALS.COM
RDEV_RMT_SHUTDOWN.COM	RDEV_RMT_STARTUP.COM
RDEV_SERVER.COM	RDEV_UCXSTUB_VAX.OLB

## MDMS Files and Logical Names

### A.2 MDMS Logical Names

**Table A–1 MDMS Installed Files**

File Name	File Name
RDEV_VAX.OLB	SHRLINK_VAX.OPT
RDF_UCX_RSHD_STARTUP.COM	RDFREE.COM
RDLOG.COM	RDLOG.COM
RDRMT_STARTUP.COM	RDSERVER_SHUTDOWN.COM
RDSERVER_STARTUP.COM	RDSERVER_V62.EXE
RDSHOW.COM	RLINK_VAX.OPT
RMTSRV_V62.EXE	RMTSRV_VAX.OPT

## A.2 MDMS Logical Names

When the MDMS installation procedure is complete, logical names are entered into the system logical name table and stored in the startup file, SYS\$STARTUP:MDMS\$SYSTARTUP.COM. They are automatically entered into the system logical name table whenever the system reboots or whenever MDMS is started with this command:

```
SYS$STARTUP:MDMS$STARTUP.COM.
```

Table A–2 describes the logical names in the system table

**Table A–2 MDMS Logical Names**

Logical Name	Definition and Description
MDMS\$DATABASE_LOCATION MDMS\$DB	These logical names point to the location of the MDMS database files.
MDMS\$DATABASE_SERVERS	<p>This logical name is a comma separated list of full node names of potential database servers. When a server starts up, it uses this logical to see if it may be a database server. If the server finds its node name in the list, it tries to become the database server. If the server does not find itself in the list, it then knows that it is not a database server but it then tries to communicate with the node in the list to find the database server. The name of the node defines how the two server communicate with each other.</p> <p>This list of names must be DECnet, DECnet-Plus, or TCP/IP node names.</p> <p>They can be a mix of different protocols or the same. For example the node list could look like this:</p> <p>NODE1,NODE2.SITE.INC.COM, - INC.:SITE.NODE3</p> <p>The above example shows that to communicate with:</p> <ul style="list-style-type: none"> <li>• NODE1 - use DECnet</li> <li>• NODE2 - use DECnet-Plus</li> <li>• NODE3 - use TCP/IP</li> </ul>
MDMS\$LOGFILE_LOCATION MDMS\$LOG	These logical names point to the location of the MDMS log files.

**Table A–2 MDMS Logical Names**

---

<b>Logical Name</b>	<b>Definition and Description</b>
MDMS\$LOGFILTER	Bitmask that defines the level of logging to be performed by the MDMS\$SERVER process. If not defined, the default value is 15, which turns on completion audit logging of requests that change the database, and event and serious error logging. See MDMS\$SYSTARTUP.COM for bitmask definitions.
MDMS\$MAILBOX	This logical name is the name of the mailbox used by the MDMS\$SERVER process.
MDMS\$ROOT	This logical name points to the device and directory of the root for the MDMS files.
MDMS\$SBT_TRACE_LEVEL	This logical name is used by the System Backup to Tape module. It controls what is written to the Oracle trace file for SBT.
MDMS\$STARTUP_QUEUE	This logical name should point to the name of the batch queue to be used for MDMS\$STARTUP.COM.
MDMS\$SUPPORT_PRE_V3	<p>This logical name enables or disables support of SLS/MDMS V2.9x remote servers.</p> <p>When this logical is TRUE, an SLS/MDMS V2.9x client can communicate with an MDMS V3.0 server. If you do not have any SLS/MDMS V2.9x clients, define this logical as FALSE.</p>

---





---

## MDMS Messages

### **ABORT**

request aborted by operator

**Explanation:** The request issued an OPCOM message that has been aborted by an operator. This message can also occur if no terminals are enabled for the relevant OPCOM classes on the node.

**User Action:** Either nothing or enable an OPCOM terminal, contact the operator and retry.

### **ACCCTRLONLY**

updated access control only

**Explanation:** You entered a SET command and you only had CONTROL access to the object, so only the access control information (if any) was updated.

**User Action:** If this is what was intended no action is needed. If you wish to update other fields in the object, you require SET access control. See your administrator.

### **ACCVIO**

access violation

**Explanation:**

The MDMS software caused an access violation. This is an internal error.

**User Action:**

Provide copies of the MDMS command issued, the database files and the server's logfile for further analysis.

### **ALLOCDRIVE**

drive !AD allocated

**Explanation:**

The named drive was successfully allocated.

**User Action:**

None.

### **ALLOCDRIVEDEV**

drive !AD allocated as device !AD

**Explanation:**

The named drive was successfully allocated, and the drive may be accessed with DCL commands using the device name shown.

**User Action:**

None.

## MDMS Messages

### **ALLOCVOLUME**

volume !AD allocated

#### **Explanation:**

The named volume was successfully allocated.

#### **User Action:**

None.

### **ALTSUCCESS**

alternative success

#### **Explanation:**

The request was successful, but extended status contains information.

#### **User Action:**

Examine the extended status, and retry command as needed.

### **APIBUGCHECK**

internal inconsistency in API

#### **Explanation:**

The MDMS API (MDMS\$\$SHR.EXE) detected an inconsistency. This is an internal error.

#### **User Action:**

Provide copies of the MDMS command issued, the database files and the server's logfile for further analysis.

### **APIUNEXP**

unexpected error in API !AZ line !UL

#### **Explanation:**

The shareable image MDMS\$\$SHR detected an internal inconsistency.

#### **User Action:**

Provide copies of the MDMS command issued, the database files and the server's logfile for further analysis.

### **ATTRINMAG**

onsite/offsite attributes invalid for magazine-based volumes

#### **Explanation:**

You attempted to specify offsite or onsite dates or locations for a volume whose placement is in a magazine. These attributes are controlled by the magazine and are not valid for individual volumes.

#### **User Action:**

Specify the dates and locations in the magazine object, or do not use magazines for volumes if you want the individual offsite/onsite dates to be different for each volume.

### **BINDVOLUME**

volume !AD bound to set !AD

#### **Explanation:**

The specified volume (or volume set) was successfully bound to the end of the named volume set.

**User Action:**

None.

**BUGCHECK**  
internal inconsistency

**Explanation:**

The server software detected an inconsistency. This is an internal error.

**User Action:**

Provide copies of the MDMS command issued, the database files and the server's logfile for further analysis. Restart the server.

**CANCELLED**  
request cancelled by user

**Explanation:**

The request was cancelled by a user issuing a cancel request command.

**User Action:**

None, or retry command.

**CLEANVOL**  
cleaning volume loaded

**Explanation:**

During a load of a volume, a cleaning volume was loaded.

**User Action:**

During an inventory this message can be ignored. During a load of a requested volume or a scratch load on a drive, or an initialize command, a cleaning volumes was loaded. Check location of the cleaning volume, update database as needed, and re-issue command using a non-cleaning volume.

**CONFLITEMS**  
conflicting item codes specified

**Explanation:**

The command cannot be completed because there are conflicting item codes in the command. This is an internal error.

**User Action:**

Provide copies of the MDMS command issued, the database files and the server's logfile for further analysis.

**CREATVOLUME**  
volume !AD created

**Explanation:**

The named volume was successfully created.

**User Action:**

None.

**DBLOCACC**  
local access to database

## MDMS Messages

**Explanation:**

This node has the database files open locally.

**User Action:**

None.

**DBRECERR**

error !AZ !AZ record for !AZ:

**Explanation:**

The search for a database server received an error from a remote server.

**User Action:**

Check the logfile on the remote server for more information. Check the logical name MDMS\$DATABASE\_SERVERS for correct entries of database server node.

**DBREMACC**

access to remote database server on node !AZ

**Explanation:**

This node has access to a remote database server.

**User Action:**

None.

**DBREP**

Database server on node !AZ reports:

**Explanation:**

The remote database server has reported an error condition. The next line contains additional information.

**User Action:**

Depends on the additional information.

**DCLARGLSOVR**

DCL extended status format, argument list overflow

**Explanation:**

During formatting of the extended status, the number of arguments exceeded the allowable limit. This is an internal error.

**User Action:**

Provide copies of the MDMS command issued, the database files and the server's logfile for further analysis.

**DCLBUGCHECK**

internal inconsistency in DCL

**Explanation:**

The MDMS command line software (MDMS\$DCL.EXE) detected an inconsistency. This is an internal error.

**User Action:**

Provide copies of the MDMS command issued, the database files and the server's logfile for further analysis.

### **DCSCERROR**

error accessing jukebox with DCSC

#### **Explanation:**

MDMS encountered an error when performing a jukebox operation. An accompanying message gives more detail.

#### **User Action:**

Examine the accompanying message and perform corrective actions to the hardware, the volume or the database, and optionally retry the operation.

### **DCSCMSG**

!AZ

#### **Explanation:**

This is a more detailed DCSC error message which accompanies DCSCERROR.

#### **User Action:**

Check the DCSC error message file.

### **DECNETLISEXIT**

DECnet listener exited

#### **Explanation:**

The DECnet listener has exited due to an internal error condition or because the user has disabled the DECNET transport for this node. The DECnet listener is the server's routine to receive requests via DECnet (Phase IV and Phase V).

#### **User Action:**

The DECnet listener should be automatically restarted unless the DECNET transport has been disabled for this node. Provide copies of the MDMS command issued, the database files and the server's logfile for further analysis if the transport has not been disabled by the user.

### **DECNETLISRUN**

listening on DECnet node !AZ object !AZ

#### **Explanation:**

The server has successfully started a DECnet listener. Requests can now be sent to the server via DECnet.

#### **User Action:**

None.

### **DEVNAMICM**

device name item code missing

#### **Explanation:**

During the allocation of a drive, the drive name was not returned by the server. This is an internal error.

#### **User Action:**

Provide copies of the MDMS command issued, the database files and the server's logfile for further analysis.

### **DRIVEEXISTS**

specified drive already exists

## MDMS Messages

**Explanation:**

The specified drive already exists and cannot be created.

**User Action:**

Use a set command to modify the drive, or create a new drive with a different name.

**DRVACCERR**

error accessing drive

**Explanation:**

MDMS could not access the drive.

**User Action:**

Verify the VMS device name, node names and/or group names specified in the drive record. Fix if necessary. Verify MDMS is running on a remote node. Check status of the drive, correct and retry.

**DRVALRALLOC**

drive is already allocated

**Explanation:**

An attempt was made to allocate a drive that was already allocated.

**User Action:**

Wait for the drive to become deallocated, or if the drive is allocated to you, use it.

**DRVEMPTY**

drive is empty or volume in drive is unloaded

**Explanation:** The specified drive is empty, or the volume in the drive is unloaded, spun-down and inaccessible.

**User Action:**

Check status of drive, correct and retry.

**DRVINITERR**

error initializing drive on platform

**Explanation:**

MDMS could not initialize a volume in a drive.

**User Action:**

There was a system error initializing the volume. Check the log file.

**DRVINUSE**

drive is currently in use

**Explanation:**

The specified drive is already in use.

**User Action:**

Wait for the drive to free up and re-enter command, or try to use another drive.

**DRVLOADED**

drive is already loaded

**Explanation:**

A drive unload appeared to succeed, but the specified volume was still detected in the drive.

**User Action:**

Check the drive and check for duplicate volume labels, or if the volume was reloaded.

**DRVLOADING**

drive is currently being loaded or unloaded

**Explanation:**

The operation cannot be performed because the drive is being loaded or unloaded.

**User Action:**

Wait for the drive to become available, or use another drive. If the drive is stuck in the loading or unloading state, check for an outstanding request on the drive and cancel it. If all else fails, manually adjust the drive state.

**DRVNOTALLOC**

drive is not allocated

**Explanation:**

The specified drive could not be allocated.

**User Action:**

Check again if the drive is allocated. If it is, wait until it is deallocated. Otherwise there was some other reason the drive could not be allocated. Check the log file.

**DRVNOTALLUSER**

drive is not allocated to user

**Explanation:**

You cannot perform the operation on the drive because the drive is not allocated to you.

**User Action:**

Either defer the operation or (in some cases) you may be able to perform the operation specifying a user name.

**DRVNOTAVAIL**

drive is not available on system

**Explanation:**

The specified drive was found on the system, but is not available for use.

**User Action:**

Check the status of the drive and correct.

**DRVNOTDEALLOC**

drive was not deallocated

**Explanation:**

MDMS could not deallocate a drive.

## MDMS Messages

### **User Action:**

Either the drive was not allocated or there was a system error deallocating the drive. Check the log file.

### **DRVNOTFOUND**

drive not found on system

### **Explanation:**

The specified drive cannot be found on the system.

### **User Action:**

Check that the OpenVMS device name, node names and/or group names are correct for the drive. Verify MDMS is running on a remote node. Re-enter command when corrected.

### **DRVNOTSPEC**

drive not specified or allocated to volume

### **Explanation:**

When loading a volume a drive was not specified, and no drive has been allocated to the volume.

### **User Action:**

Retry the operation and specify a drive name.

### **DRVREMOTE**

drive is remote

### **Explanation:**

The specified drive is remote on a node where it is defined to be local.

### **User Action:**

Check that the OpenVMS device name, node names and/or group names are correct for the drive. Verify MDMS is running on a remote node. Re-enter command when corrected.

### **DRVSINUSE**

all drives are currently in use

### **Explanation:**

All of the drives matching the selection criteria are currently in use.

### **User Action:**

Wait for a drive to free up and re-enter command.

### **DRVUNDEFINED**

referenced drive !AZ undefined

### **Explanation:**

When creating or modifying a valid object, the object's record contains a reference to a drive name that does not exist.

### **User Action:**

Check spelling of the drive name and retry, or create the drive object in the database.

### **ERROR**

error



**Explanation:**

A general internal MDMS error occurred.

**User Action:**

Provide copies of the MDMS command issued, the database files and the server's logfile for further analysis.

**EXECOMFAIL**

execute command failed, see log file for more explanation

**Explanation:**

While trying to execute a command during scheduled activities, a system service called failed.

**User Action:**

Check the log file for the failure code from the system server call.

**EXIT**

MDMS server exiting with fatal error, restarting

**Explanation:**

The MDMS server has encountered a fatal error and is exiting. The server will be restarted.

**User Action:**

Report incident to Compaq.

**EXSCHED**

internal schedules are inoperable; external scheduler in use

**Explanation:**

You have created or modified an MDMS schedule object. This is allowed, but since the domain scheduler type is set up to an external scheduler product, this schedule object will never be executed.

**User Action:**

If you are not planning to change the scheduler type to INTERNAL or EXTERNAL, you should modify the associated save or restore request to use a standard frequency or an explicit frequency.

**EXTRAVOL**

extra volume(s) processed

**Explanation:**

One or more volumes unknown to MDMS have been processed by this command.

**User Action:**

See next message line(s) for more details. Use MDMS or jukebox utility programs (MRU or CARTRIDGE) to correct the problem.

**FAILALLOCDRV**

failed to allocate drive

**Explanation:**

Failed to allocate drive.

**User Action:**

The previous message is the error that caused the failure.

## MDMS Messages

### **FAILCONSVR**

failed connection to server

#### **Explanation:**

The connection to an MDMS server either failed or could not be established. See additional message lines and/or check the server's logfile.

#### **User Action:**

Depends on additional information.

### **FAILCONSVRD**

failed connection to server via DECnet

#### **Explanation:**

The DECnet connection to an MDMS server either failed or could not be established. See additional message lines and/or check the server's logfile.

#### **User Action:**

Depends on additional information.

### **FAILCONSVRT**

failed connection to server via TCP/IP

#### **Explanation:**

The TCP/IP connection to an MDMS server either failed or could not be established. See additional message lines and/or check the server's logfile.

#### **User Action:**

Depends on additional information.

### **FAILCREATE**

failed to create !AZ

#### **Explanation:**

The reported file or object could not be created. The next line contains additional information.

#### **User Action:**

Depends on the additional information.

### **FAILDEALLOCDRV**

failed to deallocate drive

#### **Explanation:**

Failed to deallocate drive.

#### **User Action:**

The previous message is the error that caused the failure.

### **FAILDELETE**

failed to delete !AZ

#### **Explanation:**

The reported file or object could not be deleted. The next line contains additional information.

#### **User Action:**

Depends on the additional information.

### **FAILEDMNTVOL**

failed to mount volume

#### **Explanation:**

MDMS was unable to mount the volume.

#### **User Action:**

The error above this contains the error that cause the volume not to be mounted.

### **FAILICRES**

failed item code restrictions

#### **Explanation:**

The command cannot be completed because there are conflicting item codes in the command. This is an internal error.

#### **User Action:**

Provide copies of the MDMS command issued, the database files and the server's logfile for further analysis.

### **FAILINIEXTSTAT**

failed to initialize extended status buffer

#### **Explanation:**

The API could not initialize the extended status buffer. This is an internal error.

#### **User Action:**

Provide copies of the MDMS command issued, the database files and the server's logfile for further analysis.

### **FAILLOOKUP**

failed to lookup !AZ

#### **Explanation:**

The reported file or object could not be looked up. The next line contains additional information.

#### **User Action:**

Depends on the additional information.

### **FAILURE**

fatal error

#### **Explanation:**

The MDMS server encountered a fatal error during the processing of a request.

#### **User Action:**

Provide copies of the MDMS command issued, the database files and the server's logfile for further analysis.

### **FILOPNERR**

file !AZ could not be opened

#### **Explanation:**

An MDMS database file could not be opened.

## MDMS Messages

### **User Action:**

Check the server's logfile for more information.

### **FIRSTVOLUME**

specified volume is first in set

### **Explanation:**

The specified volume is the first volume in a volume set.

### **User Action:**

You cannot deallocate or unbind the first volume in a volume set. However, you can unbind the second volume and then deallocate the first, or unbind and deallocate the entire volume set.

### **FUNCFAILED**

Function !AZ failed with:

### **Explanation:**

An internal call to a system function has failed. The following lines identify the function called and the failure status.

### **User Action:**

Depends on information following this message.

### **GRPUNDEFINED**

referenced group(s) !AZ undefined

### **Explanation:**

When creating or modifying a valid object, the object's record contains a reference to a group name that does not exist. One or more of the specified groups may be undefined.

### **User Action:**

Check spelling of the group names and retry, or create the group objects in the database.

### **ILLEGALOP**

illegal move operation

### **Explanation:**

You attempted to move a volume within a DCSC jukebox, and this is not supported.

### **User Action:**

None.

### **INCOMFREQ**

incompatible frequency for !AZ !AZ

### **Explanation:**

After changing the domain scheduler type, MDMS has determined that this save or restore request has a frequency that is incompatible with the new scheduler type. The frequencies that are not valid for the given scheduler types are:

- INTERNAL and EXTERNAL: Explicit
- DECSCHEDULER and SCHEDULER: Custom

### **User Action:**

Modify the frequency to a valid one for this scheduler type.

**INCOMPATMED**

volume's media type incompatible with the drive

**Explanation:**

The media type for the volume is incompatible with the media type(s) for the drive on a load operation.

**User Action:**

Verify that the volume can be physically loaded and used in the specified drive. If not, select another drive. If so, then add the volume's media type to the drive or otherwise aligned the media types of the volume and the drive.

**INCOMPATOPT**

incompatible options specified

**Explanation:**

You entered a command with incompatible options.

**User Action:**

Examine the command documentation and re-enter with allowed combinations of options.

**INCOMPATVOL**

volume is incompatible with volumes in set

**Explanation:**

You cannot bind the volume to the volume set because some of the volume's attributes are incompatible with the volumes in the volume set.

**User Action:**

Check that the new volume's media type, onsite location and offsite location are compatible with those in the volume set. Adjust attributes and retry, or use another volume with compatible attributes.

**INSCMDPRIV**

insufficient privilege to execute request

**Explanation:**

You do not have sufficient privileges to enter the request.

**User Action:**

Contact your system administrator and request additional privileges, or give yourself privs and retry.

**INSOPTPRIV**

insufficient privilege for request option

**Explanation:**

You do not have sufficient privileges to enter a privileged option of this request.

**User Action:**

Contact your system administrator and request additional privileges, or give yourself privs and retry. Alternatively, retry without using the privileged option.

**INSSHOWPRIV**

some volumes not shown due to insufficient privilege

## MDMS Messages

### **Explanation:**

Not all volumes were shown because of restricted privilege.

### **User Action:**

None if you just want to see volumes you own. You need MDMS\_SHOW\_ALL privilege to see all volumes.

### **INSSVRPRV**

insufficient server privileges

### **Explanation:**

The MDMS server is running with insufficient privileges to perform system functions.

### **User Action:**

Refer to the Installation Guide to determine the required privileges. Contact your system administrator to add these privileges in the MDMS\$SERVER account.

### **INTBUFOVR**

internal buffer overflow

### **Explanation:**

The MDMS software detected an internal buffer overflow. This an internal error.

### **User Action:**

Provide copies of the MDMS command issued, the database files and the server's logfile for further analysis. Restart the server.

### **INTINVMSG**

internal invalid message

### **Explanation:**

An invalid message was received by a server. This could be due to a network problem or, a remote non-MDMS process sending messages in error or, an internal error.

### **User Action:**

If the problem persists and no non-MDMS process can be identified then provide copies of the MDMS command issued, the database files and the server's logfile for further analysis.

### **INTSCHEDULE**

cannot modify or delete internal schedule

### **Explanation:**

You attempted to modify or delete a schedule object that was internally generated for a save or restore request. This is not allowed.

### **User Action:**

Modify or delete the associated save or restore request instead, and the schedule will be updated accordingly.

### **INVABSTIME**

invalid absolute time

### **Explanation:**

The item list contained an invalid absolute date and time. Time cannot be earlier than 1-Jan-1970 00:00:00 and cannot be greater than 7-Feb-2106 06:28:15

**User Action:**

Check that the time is between these two times.

**INVALIDRANGE**

invalid volume ID or invalid range specified

**Explanation:**

The specified volume ID, volume range, slot range or space range is invalid.

**User Action:**

A volume ID may contain up to 6 characters. A volume range may contain up to 1000 volume IDs where the first 3 characters must be alphabetic and the last 3 may be alphanumeric. Only the numeric portions may vary in the range. Examples are ABC000-ABC999, or ABCD01-ABCD99. A slot range can contain up to 1000 slots and must be numeric. Also, all slots in the range must be less than the slot count for the jukebox or magazine. Example: 0-255 for a slot count of 256. A space range can contain up to 1000 spaces where the first and last spaces must have the same number of characters. Spaces must be within the range defined for the location. Examples: 000-999, or Space A1-Space C9

**INVCONSOLVAL**

invalid value for consolidation savesets or volumes

**Explanation:**

You specified an invalid value for consolidation savesets or volumes.

**User Action:**

Use a value in the range 0 to maximum integer.

**INVDBSVRLIS**

invalid database server search list

**Explanation:**

The logical name MDMS\$DATABASE\_SERVERS contains invalid network node names or is not defined.

**User Action:**

Correct the node name(s) in the logical name MDMS\$DATABASE\_SERVERS in file MDMS\$SYSTARTUP.COM. Redefine the logical name in the current system. Then start the server.

**INVDELSTATE**

object is in invalid state for delete

**Explanation:**

The specified object cannot be deleted because its state indicates it is being used.

**User Action:**

Defer deletion until the object is no longer being used, or otherwise change its state and retry.

**INVDELTATIME**

invalid delta time

**Explanation:**

The item list contained an invalid delta time.

## MDMS Messages

### **User Action:**

Check that the item list has a correct delta time.

### **INVDFULLNAM**

invalid DECnet fullname

### **Explanation:**

A node full name for a DECnet-Plus (Phase V) node specification has an invalid syntax.

### **User Action:**

Correct the node name and retry.

### **INVDRVCOUNT**

invalid value for drive count, use 1-32

### **Explanation:**

You specified an invalid value for drive count.

### **User Action:**

Use a value in the range 1-32.

### **INVEXTSTS**

invalid extended status item desc/buffer

### **Explanation:**

The error cannot be reported in the extended status item descriptor. This error can be caused by one of the following:

- Not being able to read any one of the item descriptors in the item list
- Not being able to write to the buffer in the extended status item descriptor
- Not being able to write to the return length in the extended status item descriptor
- Not being able to initialize the extended status buffer

### **User Action:**

Check for any of the errors stated above in your program and fix the error.

### **INVFREQUENCY**

invalid frequency for domain scheduler type

### **Explanation:**

You specified an invalid save or restore frequency the scheduler type specified in the domain. Invalid combinations include: CUSTOM, with NONE, DECSCHEDULER, SCHEDULER or LOCAL EXPLICIT, with NONE, INTERNAL, EXTERNAL, or SINGLE

### **User Action:**

Specify a valid frequency for the scheduler type specified in the domain.

### **INVINITOPT**

invalid initialize options specified

### **Explanation:**

You attempted initialize volumes in a jukebox by specifying a slot range and the jukebox is not a vision-equipped, MRD-controlled jukebox.



### **User Action:**

Specify a volume range instead of a slot range to initialize volumes in a DCSC jukebox or an MRD jukebox without a vision system.

### **INVITCODE**

invalid item code for this function

### **Explanation:**

The item list had an invalid item code. The problem could be one of the following:

- Item codes do not meet the restrictions for that function.
- An item code cannot be used in this function.

### **User Action:**

Refer to the API specification to find out which item codes are restricted for each function and which item codes are allowed for each function.

### **INVITDESC**

invalid item descriptor, index !@UL

### **Explanation:**

The item descriptor is in error. The previous message gives the error. Included is the index of the item descriptor in the item list.

### **User Action:**

Refer to the index number and the previous message to indicate the error and which item descriptor is in error.

### **INVITLLENGTH**

invalid item list buffer length

### **Explanation:**

The item list buffer length is zero. The item list buffer length cannot be zero for any item code.

### **User Action:**

Refer to the API specification to find an item code that would be used in place of an item code that has a zero buffer length.

### **INVMEDIATYPE**

media type is invalid or not supported by volume

### **Explanation:**

The specified volume supports multiple media types where a single media type is required, or the volume does not support the specified media type.

### **User Action:**

Re-enter the command specifying a single media type that is already supported by the volume.

### **INVMSG**

invalid message via !AZ

### **Explanation:**

An invalid message was received MDMS software. This could be due to a network problem or, a non-MDMS process sending messages in error or, an internal error.

## MDMS Messages

### **User Action:**

If the problem persists and no non-MDMS process can be identified then provide copies of the MDMS command issued, the database files and the server's logfile for further analysis.

### **INVNODNAM**

invalid node name specification

### **Explanation:**

A node name for a DECnet (Phase IV) node specification has an invalid syntax.

### **User Action:**

Correct the node name and retry.

### **INVPORTS**

invalid port number specification

### **Explanation:**

The MDMS server did not start up because the logical name MDMS\$TCPIP\_SND\_PORTS in file MDMS\$SYSTARTUP.COM specifies an illegal port number range. A legal port number range is of the form "low\_port\_number-high\_port\_number".

### **User Action:**

Correct the port number range for the logical name MDMS\$TCPIP\_SND\_PORTS in file MDMS\$SYSTARTUP.COM. Then start the server.

### **INVPOSITION**

invalid jukebox position

### **Explanation:**

The position specified is invalid.

### **User Action:**

Position is only valid for jukeboxes with a topology defined. Check that the position is within the topology ranges, correct and retry. Example: /POSITION=(1,2,1)

### **INVRETDAYS**

invalid retention days specified

### **Explanation:**

You entered an invalid value for the retention days. Valid values are 0 to 9999 days. If you wish for no expiration of volumes, specify /NOEXPIRATION\_DATE.

### **User Action:**

Enter a value between 0 and 9999.

### **INVRETRY**

invalid value for retry count or interval

### **Explanation:**

You specified an invalid value for either or both the retry count or interval. In addition, it is invalid to specify an interval with a retry limit of zero or nolimit.

**User Action:**

Use values within the following ranges:

- RETRY\_LIMIT: 0 - 10000 or NOLIMIT
- INTERVAL: 00:01:00 - 01:00:00 (1 - 60 mins)

**INVRETRYINTERVAL**

invalid value for retry interval

**Explanation:**

You specified an invalid value for retry interval. In addition, it is invalid to specify an interval with a retry limit of zero.

**User Action:**

Use a value within the following range only if retry limit is non-zero: 00:01:00 - 01:00:00 (1 - 60 mins)

**INVRETRYLIMIT**

invalid value for retry limit

**Explanation:**

You specified an invalid value for retry limit.

**User Action:**

Use a value in the range 0 to maximum integer or use /NORETRY\_LIMIT

**INVSCHEDENUM**

invalid scheduling translation defined

**Explanation:**

An invalid parameter translation was entered for a scheduling option.

**User Action:**

Report the incident to Compaq.

**INVSCHEDOPT**

invalid schedule options entered

**Explanation:**

You entered invalid schedule date/time options for a schedule object. The following values are allowed:

- DATES: List of values or ranges, values 1 - 31
- DAYS: List of values or ranges, values MON - SUN
- MONTHS: List of values or ranges, values JAN - DEC
- TIMES: List of values, 00:00 - 23:59
- INCLUDE: List of dates 01-Jan-yyyy - 31-Dec-yyyy
- EXCLUDE: List of dates 01-Jan-yyyy - 31-Dec-yyyy

The yyyy for INCLUDE and EXCLUDE must be between the current year and up to 9 years into the future (e.g. 2001-2010). If omitted, the current year is used.

**User Action:**

Re-enter the command with valid values.

## MDMS Messages

### **INVSCHEDPARAM**

invalid scheduling parameter defined

#### **Explanation:**

An invalid parameter was entered for a scheduling option.

#### **User Action:**

Report the incident to Compaq.

### **INVSELECT**

invalid selection criteria

#### **Explanation:**

The selection criteria specified on an allocate command are invalid.

#### **User Action:**

Check the command with the documentation and re-enter with a valid combination of selection criteria.

### **INVSLOT**

invalid slot or slot range specified

#### **Explanation:**

The slot or slot range specified when moving volumes into a magazine or jukebox was invalid, or the specified slots were already occupied.

#### **User Action:**

Specify valid empty slots and re-enter.

### **INVSLOTRANGE**

invalid slot range

#### **Explanation:**

The slot range was invalid. It must be of the form: 1-100 1,100-200,300-400 The only characters allowed are comma, dash, and numbers (0-9).

#### **User Action:**

Check that you are using the correct form.

### **INVSPACE**

invalid space or space range specified

#### **Explanation:**

The space or space range specified when moving volumes into a location was invalid.

#### **User Action:**

Specify valid spaces already defined for the location, or specify a space range for the location

### **INVSRCDEST**

invalid source or destination for move

#### **Explanation:**

Either the source or destination of a move operation was invalid (does not exist).

**User Action:**

If the destination is invalid, enter a correct destination and retry. If a source is invalid, either create the source or correct the current placement of the affected volumes or magazines.

**INVSTATE**

volume !AZ is in an invalid state for initialization

**Explanation:**

The volume loaded in the drive for initialization was either allocated or in the transition state and cannot be initialized.

**User Action:**

Either the wrong volume was loaded, or the requested volume was in an invalid state. If the wrong volume was loaded, perform an inventory on the jukebox and retry. If the volume is allocated or in transition, you should not try to initialize the volume.

**INVTFULLNAM**

invalid TCP/IP fullname

**Explanation:**

A node full name for a TCP/IP node specification has an invalid syntax.

**User Action:**

Correct the node name and retry.

**INVTPOLOGY**

invalid jukebox topology

**Explanation:**

The specified topology for a jukebox is invalid.

**User Action:**

Check topology definition; the towers must be sequentially increasing from 0; there must be a face, level and slot definition for each tower. Example: /TOPOLOGY=(TOWER=(0,1,2), FACES=(8,8,8), - LEVELS=(2,3,2), SLOTS=(13,13,13))

**INVVOLPLACE**

invalid volume placement for operation

**Explanation:**

The volume has an invalid placement for a load operation.

**User Action:**

Re-enter the command and use the move option.

**INVVOLSTATE**

volume in invalid state for operation

**Explanation:**

The operation cannot be performed on the volume because of the volume state does not allow it.

**User Action:**

Defer the operation until the volume changes state. If the volume is stuck in a transient state (e.g. moving), check for an outstanding request and cancel it. If all else fails, manually change the state.

## MDMS Messages

### **JUKEBOXEXISTS**

specified jukebox already exists

#### **Explanation:**

The specified jukebox already exists and cannot be created.

#### **User Action:**

Use a set command to modify the jukebox, or create a new jukebox with a different name.

### **JUKENOTINIT**

jukebox could not be initialized

#### **Explanation:**

An operation on a jukebox failed because the jukebox could not be initialized.

#### **User Action:**

Check the control, robot name, node name and group name of the jukebox, and correct as needed. Check access path to jukebox (HSJ etc), correct as needed. Verify MDMS is running on a remote node. Then retry operation.

### **JUKETIMEOUT**

timeout waiting for jukebox to become available

#### **Explanation:**

MDMS timed out waiting for a jukebox to become available. The timeout value is 10 minutes.

#### **User Action:**

If the jukebox is in heavy use, try again later. Otherwise, check requests for a hung request - cancel it. Set the jukebox state to available if all else fails.

### **JUKEUNAVAIL**

jukebox is currently unavailable

#### **Explanation:**

The jukebox is disabled.

#### **User Action:**

Re-enable the jukebox.

### **JUKUNDEFINED**

referenced jukebox !AZ undefined

#### **Explanation:**

When creating or modifying a valid object, the object's record contains a reference to a jukebox name that does not exist.

#### **User Action:**

Check spelling of the jukebox name and retry, or create the jukebox object in the database.

### **LOCATIONEXISTS**

specified location already exists

#### **Explanation:**

The specified location already exists and cannot be created.

**User Action:**

Use a set command to modify the location, or create a new location with a different name.

**LOCUNDEFINED**

referenced location !AZ undefined

**Explanation:**

When creating or modifying a valid object, the object's record contains a reference to a location name that does not exist.

**User Action:**

Check spelling of the location name and retry, or create the location object in the database.

**LOGRESET**

Log file !AZ by !AZ on node !AZ

**Explanation:**

The server logfile has been closed and a new version has been created by a user.

**User Action:**

None.

**MAGAZINEEXISTS**

specified magazine already exists

**Explanation:**

The specified magazine already exists and cannot be created.

**User Action:**

Use a set command to modify the magazine, or create a new magazine with a different name.

**MAGUNDEFINED**

referenced magazine !AZ undefined

**Explanation:**

When creating or modifying a valid object, the object's record contains a reference to a magazine name that does not exist.

**User Action:**

Check spelling of the magazine name and retry, or create the magazine object in the database.

**MBLISEXIT**

mailbox listener exited

**Explanation:**

The mailbox listener has exited due to an internal error condition. The mailbox listener is the server's routine to receive local user requests through mailbox MDMS\$MAILBOX.

**User Action:**

The mailbox listener should be automatically restarted. Provide copies of the MDMS command issued, the database files and the server's logfile for further analysis.

**MBLISRUN**

listening on mailbox !AZ logical !AZ

## MDMS Messages

### **Explanation:**

The server has successfully started the mailbox listener. MDMS commands can now be entered on this node.

### **User Action:**

None.

### **MEDIATYPEEXISTS**

specified media type already exists

### **Explanation:**

The specified media type already exists and cannot be created.

### **User Action:**

Use a set command to modify the media type, or create a new media type with a different name.

### **MEDUNDEFINED**

referenced media type(s) !AZ undefined

### **Explanation:**

When creating or modifying a valid object, the object's record contains a reference to a media type that does not exist. One or more of the specified media types may be undefined.

### **User Action:**

Check spelling of the media types and retry, or create the media type objects in the database.

### **MOVEINCOMPL**

move is incomplete

### **Explanation:**

When moving volumes into and out of a jukebox, some of the volumes were not moved.

### **User Action:**

Check that there are enough empty slots in the jukebox when moving in and retry. On a move out, examine the cause of the failure and retry.

### **MRDERROR**

error accessing jukebox with MRD

### **Explanation:**

MDMS encountered an error when performing a jukebox operation. An accompanying message gives more detail.

### **User Action:**

Examine the accompanying message and perform corrective actions to the hardware, the volume or the database, and optionally retry the operation.

### **MRDMSG**

!AZ

### **Explanation:**

This is a more detailed MRD error message which accompanies MRDERROR.

### **User Action:**

Check the MRU error message file.



### **NOACCESS**

no user access to object for operation

#### **Explanation:**

You attempted to perform an operation on an object for which you have no access.

#### **User Action:**

You need an authorized user to add you to the access control list, otherwise you cannot perform the requested operation.

### **NOBINDSELF**

volume is already in volume set

#### **Explanation:**

You cannot bind this volume into this volume set because it already a member of the volume set.

#### **User Action:**

Use another volume.

### **NOCHANGES**

no attributes were changed in the database

#### **Explanation:**

Your set command did not change any attributes in the database because the attributes you entered were already set to those values.

#### **User Action:**

Double-check your command, and re-enter if necessary. Otherwise the database is already set to what you entered.

### **NOCHANGESOBJ**

no attributes were changed for !AZ !AZ

#### **Explanation:**

Your set command did not change any attributes in the database because the attributes you entered were already set to those values. The message indicates which object was not changed.

#### **User Action:**

Double-check your command, and re-enter if necessary. Otherwise the database is already set to what you entered.

### **NOCHECK**

drive not accessible, check not performed

#### **Explanation:**

The specified drive could not be physically accessed and the label check was not performed. The displayed attributes are taken from the database.

#### **User Action:**

Verify the VMS device name, node name or group name in the drive object. Check availability on system. Verify MDMS is running on a remote node. Determine the reason the drive was not accessible, fix it and retry.

### **NODBACC**

no access to database server

## MDMS Messages

**Explanation:**

This server has no access to a database server.

**User Action:**

Verify the setting of logical name MDMS\$DATABASE\_SERVERS. Check each node listed using MDMS SHOW SERVER/NODE=... for connectivity and database access status. Check the servers logfiles for more information.

**NODCSC**

DCSC not running

**Explanation:**

DCSC has not been started.

**User Action:**

Execute command procedure SYS\$STARTUP:DCSC\$STARTUP.COM and retry command.

**NODEDISABLED**

node disabled

**Explanation:**

The server failed to start up because it is disabled in the database.

**User Action:**

If necessary correct the setting and start the server again.

**NODEEXISTS**

specified node already exists

**Explanation:**

The specified node already exists and cannot be created.

**User Action:**

Use a set command to modify the node, or create a new node with a different name.

**NODENOPRIV**

node is not privileged to access database server

**Explanation:**

A remote server access failed because the user making the DECnet connection is not MDMS\$SERVER or the remote port number is not less than 1024.

**User Action:**

Verify with DCL command SHOW PROCESS that the remote MDMS server is running under a username of MDMS\$SERVER and/or, verify that logical name MDMS\$TCPIP\_SND\_PORTS on the remote server node specifies a port number range between 0-1023.

**NODENOTENA**

node not in database or not fully enabled

**Explanation:**

The server was not allowed to start up because there is no such node object in the database or its node object in the database does not specify all network full names correctly.

**User Action:**

For a node running DECnet (Phase IV) the node name has to match logical name SYS\$NODE on that node. For a node running DECnet-Plus (Phase V) the node's DECNET\_PLUS\_FULLNAME has to match the logical name SYS\$NODE\_FULLNAME on that node. For a node running TCP/IP the node's TCPIP\_FULLNAME has to match the full name combined from logical names \*INET\_HOST and \*INET\_DOMAIN.

**NODENOTINDB**

no node object with !AZ name !AZ in database

**Explanation:**

The current server could not find a node object in the database with a matching DECnet (Phase IV) or DECnet-Plus (Phase V) or TCP/IP node full name.

**User Action:**

Use SHOW SERVER/NODES=(...) to see the exact naming of the server's network names. Correct the entry in the database and restart the server.

**NODRIVES**

no drives match selection criteria

**Explanation:**

When allocating a drive, none of the drives match the specified selection criteria.

**User Action:**

Check spelling and re-enter command with valid selection criteria.

**NODRVACC**

access to drive disallowed

**Explanation:**

You attempted to allocate, load or unload a drive from a node that is not allowed to access it.

**User Action:**

The access field in the drive object allows local, remote or all access, and your attempted access did not conform to the attribute. Use another drive.

**NODRVSAVAIL**

no drives are currently available

**Explanation:**

All of the drives matching the selection criteria are currently in use or otherwise unavailable.

**User Action:**

Check to see if any of the drives are disabled or inaccessible. Re-enter command when corrected.

**NODRVSGRP**

no drives in the specified group were found

**Explanation:**

When allocating a drive, no drives on nodes in the specified group were found.

**User Action:**

Check group name and retry command.

## MDMS Messages

### **NODRVSJUKE**

no drives in the specified jukebox were found

#### **Explanation:**

When allocating a drive, no drives in the specified jukebox were found.

#### **User Action:**

Check jukebox name and retry command.

### **NODRVSLOC**

no drives in the specified location were found

#### **Explanation:**

When allocating a drives, no drives in the specified location were found.

#### **User Action:**

Check location name and retry command.

### **NODRVSMED**

no drives with the specified media type were found

#### **Explanation:**

When allocating a drive, no drives with the specified media type were found.

#### **User Action:**

Check media type and retry command, or specify the media type for more drives.

### **NODRVSNOD**

no drives on the specified node were found

#### **Explanation:**

When allocating a drive, no drives on the specified node were found.

#### **User Action:**

Check the node name and retry command.

### **NODRVSVOL**

no drives that can support the specified volume were found

#### **Explanation:**

When allocating a drive, no drives that could support the specified volume were found.

#### **User Action:**

Check the volume ID and retry command, or check and adjust volume attributes to match a valid drive.

### **NODUNDEFINED**

referenced node(s) !AZ undefined

#### **Explanation:**

When creating or modifying a valid object, the object's record contains a reference to a node name that does not exist. One or more of the specified nodes may be undefined.

#### **User Action:**

Check spelling of the node names and retry, or create the node objects in the database.

### **NOFIELDS**

no fields specified for report

#### **Explanation:**

A REPORT VOLUME command was entered with no fields to select or display.

#### **User Action:**

Enter at least one field for the report.

### **NOINCLUDE**

selection attributes not set with no include data

#### **Explanation:**

You specified one or more of the following attributes which are not valid unless an include specification is present: DATA\_TYPE, INCREMENTAL, NODES, GROUPS The save or restore object was updated, but selection attributes were not set.

#### **User Action:**

These attributes are applicable only when an INCLUDE statement is present. Re-enter the command with an INCLUDE qualifier.

### **NOINCLUDES**

no include specification for selection

#### **Explanation:**

A save or restore object had some selection attributes specified, but no include file specification. The following attributes require an include specification:

- Data type
- Incremental
- Groups
- Nodes

#### **User Action:**

Re-enter the command with an include specification.

### **NOINTSCHED**

internal scheduling not enabled

#### **Explanation:**

You attempted to create a schedule object but the domain's scheduler option is set to an external scheduler. The MDMS schedule object is valid only with scheduler options INTERNAL, EXTERNAL and SINGLE\_SCHEDULER.

#### **User Action:**

Schedule your request using the specified external scheduler product and interface.

### **NOJUKEACC**

access to jukebox disallowed

#### **Explanation:**

You attempted to use a jukebox from a node that is not allowed to access it.

## MDMS Messages

### **User Action:**

The access field in the jukebox object allows local, remote or all access, and your attempted access did not conform to the attribute. Use another jukebox.

### **NOJUKESPEC**

jukebox required on vision option

### **Explanation:**

The jukebox option is missing on a create volume request with the vision option.

### **User Action:**

Re-enter the request and specify a jukebox name and slot range.

### **NOLICENSE**

your current license does not support this operation

### **Explanation:**

The requested operation is not licensed. If you are licensed for ABS\_OMT only, you have attempted to perform an operation that requires a full ABS license.

### **User Action:**

Use an alternative mechanism to perform the operation. If this is not possible, you cannot perform the operation with your current license. You may purchase an upgrade ABS license to enable full ABS functionality. Contact Compaq for details.

### **NOMAGAZINES**

no magazines match selection criteria

### **Explanation:**

On a move magazine request using the schedule option, no magazines were scheduled to be moved.

### **User Action:**

None.

### **NOMAGSMOVED**

no magazines were moved

### **Explanation:**

No magazines were moved for a move magazine operation. An accompanying message gives a reason.

### **User Action:**

Check the accompanying message, correct and retry.

### **NOMEDIATYPE**

no media type specified when required

### **Explanation:**

An allocation for a volume based on node, group or location also requires the media type to be specified.

### **User Action:**

Re-enter the command with a media type specification.

### **NOMEMORY**

not enough memory

#### **Explanation:**

The MDMS server failed to allocate enough virtual memory for an operation. This is an internal error.

#### **User Action:**

Provide copies of the MDMS command issued, the database files and the server's logfile for further analysis. Restart the server.

### **NOOBJECTS**

no such objects currently exist

#### **Explanation:**

On a show command, there are no such objects currently defined.

#### **User Action:**

None.

### **NOPARAM**

required parameter missing

#### **Explanation:**

A required input parameter to a request or an API function was missing.

#### **User Action:**

Re-enter the command with the missing parameter, or refer to the API specification for required parameters for each function.

### **NOPOOLSPEC**

no free volumes with no pool or your default pool were found

#### **Explanation:**

When allocating a volume, no free volumes that do not have a pool defined or that are in your default pool were found.

#### **User Action:**

Add a pool specification to the command, or define more free volumes with no pool or your default pool.

### **NORANGESUPP**

slot or space ranges not supported with volset option

#### **Explanation:**

On a set volume, you entered the volset option and specified either a slot range or space range.

#### **User Action:**

If you want to assign slots or spaces to volumes directly, do not use the volset option.

### **NORECVPORTS**

no available receive port numbers for incoming connections

#### **Explanation:**

The MDMS could not start the TCP/IP listener because none of the receive ports specified with this node's TCPIP\_FULLNAME are currently available.

## MDMS Messages

### **User Action:**

Use a suitable network utility to find a free range of TCP/IP ports which can be used by the MDMS server. Use the MDMS SET NODE command to specify the new range with the /TCPIP\_FULLNAME then restart the server.

### **NOREMCONNECT**

unable to connect to remote node

### **Explanation:**

The server could not establish a connection to a remote node. See the server's logfile for more information.

### **User Action:**

Depends on information in the logfile.

### **NOREQUESTS**

no such requests currently exist

### **Explanation:**

No requests exist on the system.

### **User Action:**

None.

### **NORESEFN**

not enough event flags

### **Explanation:**

The server ran out of event flags. This is an internal error.

### **User Action:**

Provide copies of the MDMS command issued, the database files and the server's logfile for further analysis. Restart the server.

### **NORIGHTS**

no rights are shown

### **Explanation:**

When showing a domain, the rights are not shown because you don't have privilege to see the rights.

### **User Action:**

Nothing. To see rights you need MDMS\_SHOW\_RIGHTS.

### **NOSCHEDULE**

schedule object invalid for scheduler type or frequency

### **Explanation:**

You specified a schedule object for a non-custom frequency or for an external scheduler option. A schedule object can only be specified for frequency CUSTOM with domain scheduler type of INTERNAL, EXTERNAL or SINGLE.

### **User Action:**

Do not specify a schedule name.



### **NOSCRATCH**

scratch loads not supported for jukebox drives

#### **Explanation:**

You attempted a load drive command for a jukebox drive.

#### **User Action:**

Scratch loads are not supported for jukebox drives. You must use the load volume command to load volumes in jukebox drives.

### **NOSENDPORTS**

no available send port numbers for outgoing connection

#### **Explanation:**

The server could not make an outgoing TCP/IP connection because none of the send ports specified for the range in logical name MDMS\$TCPIP\_SND\_PORTS are currently available.

#### **User Action:**

Use a suitable network utility to find a free range of TCP/IP ports which can be used by the MDMS server. Change the logical name MDMS\$TCPIP\_SND\_PORTS in file MDMS\$SYS-TARTUP.COM. Then restart the server.

### **NOSLOT**

not enough slots defined for operation

#### **Explanation:**

The command cannot be completed because there are not enough slots specified in the command, or because there are not enough empty slots in the jukebox.

#### **User Action:**

If the jukebox is full, move some other volumes out of the jukebox and retry. If there are not enough slots specified in the command, re-enter with a larger slot range.

### **NOSTATUS**

no status defined

#### **Explanation:**

An uninitialized status has been reported. This an internal error.

#### **User Action:**

Provide copies of the MDMS command issued, the database files and the server's logfile for further analysis.

### **NOSUCHDEST**

specified destination does not exist

#### **Explanation:**

In a move command, the specified destination does not exist.

#### **User Action:**

Check spelling or create the destination as needed.

### **NOSUCHDRIVE**

specified drive does not exist

## MDMS Messages

**Explanation:**

The specified drive does not exist.

**User Action:**

Check spelling or create drive as needed.

**NOSUCHGROUP**

specified group does not exist

**Explanation:**

The specified group does not exist.

**User Action:**

Check spelling or create group as needed.

**NOSUCHINHERIT**

specified inherited object does not exist

**Explanation:**

On a create of an object, the object specified for inherit does not exist.

**User Action:**

Check spelling or create the inherited object as needed.

**NOSUCHJUKEBOX**

specified jukebox does not exist

**Explanation:**

The specified jukebox does not exist.

**User Action:**

Check spelling or create jukebox as needed.

**NOSUCHLOCATION**

specified location does not exist

**Explanation:**

The specified location does not exist.

**User Action:**

Check spelling or create location as needed.

**NOSUCHMAGAZINE**

specified magazine does not exist

**Explanation:**

The specified magazine does not exist.

**User Action:**

Check spelling or create magazine as needed.

**NOSUCHMEDIATYPE**

specified media type does not exist

**Explanation:**

The specified media type does not exist.

**User Action:**

Check spelling or create media type as needed.

**NOSUCHNODE**

specified node does not exist

**Explanation:**

The specified node does not exist.

**User Action:**

Check spelling or create node as needed.

**NOSUCHOBJECT**

specified object does not exist

**Explanation:**

The specified object does not exist.

**User Action:**

Check spelling or create the object as needed.

**NOSUCHPOOL**

specified pool does not exist

**Explanation:**

The specified pool does not exist.

**User Action:**

Check spelling or create pool as needed.

**NOSUCHREQUESTID**

specified request does not exist

**Explanation:**

The specified request does not exist on the system.

**User Action:**

Check the request id again, and re-enter if incorrect.

**NOSUCHUSER**

no such user on system

**Explanation:**

The username specified in the command does not exist.

**User Action:**

Check spelling of the username and re-enter.

**NOSUCHVOLUME**

specified volume(s) do not exist

**Explanation:**

The specified volume or volumes do not exist.

**User Action:**

Check spelling or create volume(s) as needed.

## MDMS Messages

### **NOSVRACCOUNT**

username !AZ does not exist

#### **Explanation:**

The server cannot startup because the username MDMS\$SERVER is not defined in file SYSUAF.DAT.

#### **User Action:**

Enter the username of MDMS\$SERVER (see Installation manual for account details) and then start the server.

### **NOSVRMB**

no server mailbox or server not running

#### **Explanation:**

The MDMS server is not running on this node or the server is not servicing the mailbox via logical name MDMS\$MAILBOX.

#### **User Action:**

Use the MDMS\$STARTUP procedure with parameter RESTART to restart the server. If the problem persists, check the server's logfile and file SYS\$MANAGER:MDMS\$SERVER.LOG for more information.

### **NOSYMBOLS**

symbols not supported for multiple volumes

#### **Explanation:**

A SHOW VOLUME/SYMBOLS command was entered for multiple volumes. The /SYMBOLS qualifier is only supported for a single volume.

#### **User Action:**

Re-enter command with a single volume ID, or don't use the /SYMBOLS qualifier.

### **NOTALLOCUSER**

volume is not allocated to user

#### **Explanation:**

You cannot perform the operation on the volume because the volume is not allocated to you.

#### **User Action:**

Either use another volume, or (in some cases) you may be able to perform the operation specifying a user name.

### **NOTSCHEDULED**

specified save or restore is not scheduled for execution

#### **Explanation:**

The save or restore request did not contain enough information to schedule the request for execution. The request requires the definition of an archive, an environment and a start time.

#### **User Action:**

If you wish this request to be scheduled, enter a SET SAVE or SET RESTORE and enter the required information.

### **NOUNALLOCDRV**

no unallocated drives found for operation

**Explanation:**

On an initialize volume request, MDMS could not locate an unallocated drive for the operation.

**User Action:**

If you had allocated a drive for the operation, deallocate it and retry. If all drives are currently in use, retry the operation later.

**NOVOLSJUKE**

no free volumes in the specified jukebox were found

**Explanation:**

When allocating a volume, no free volumes in the specified jukebox were found.

**User Action:**

Check jukebox name and retry command, or move some free volumes into the jukebox.

**NOVOLSLOC**

no free volumes in the specified location were found

**Explanation:**

When allocating a volume, no free volumes in the specified location were found.

**User Action:**

Check location name and retry command, or move some free volumes into the location.

**NOVOLSMED**

no free volumes with the specified media type were found

**Explanation:**

When allocating a volume, no free volumes with the specified media type were found.

**User Action:**

Check media type and retry command, or specify the media type for more free volumes.

**NOVOLSMOVED**

no volumes were moved

**Explanation:**

No volumes were moved for a move volume operation. An accompanying message gives a reason.

**User Action:**

Check the accompanying message, correct and retry.

**NOVOLSPPOOL**

no free volumes in the specified pool were found

**Explanation:**

When allocating a volume, no free volumes in the specified pool were found.

**User Action:**

Check pool name and retry command, or specify the pool for more free volumes (add them to the pool).

**NOVOLSPROC**

no volumes were processed

## MDMS Messages

### **Explanation:**

In a create, set or delete volume command, no volumes were processed.

### **User Action:**

Check the volume identifiers and re-enter command.

### **NOVOLSVOL**

no free volumes matching the specified volume were found

### **Explanation:**

When allocating a volume, no free volumes matching the specified volume were found.

### **User Action:**

Check the volume ID and retry command, or add more free volumes with matching criteria.

### **NOVOLUMES**

no volumes match selection criteria

### **Explanation:**

When allocating a volume, no volumes match the specified selection criteria.

### **User Action:**

Check the selection criteria. Specifically check the relevant volume pool. If free volumes are in a volume pool, the pool name must be specified in the allocation request, or you must be a default user defined in the pool. You can re-enter the command specifying the volume pool as long as you are an authorized user. Also check that newly-created volumes are in the FREE state rather than the UNINITIALIZED state.

### **OBJECTEXISTS**

specified object already exists

### **Explanation:**

The specified object already exists and cannot be created.

### **User Action:**

Use a set command to modify the object, or create a new object with a different name.

### **OBJNOTEXIST**

referenced object !AZ does not exist

### **Explanation:**

When attempting to allocate a drive or volume, you specified a selection object that does not exist.

### **User Action:**

Check spelling of selection criteria objects and retry, or create the object in the database.

### **OBJREFZERO**

dereferenced object with zero count

### **Explanation:**

The MDMS server software detected an internal inconsistency. This is an internal error.

### **User Action:**

Provide copies of the MDMS command issued, the database files and the server's logfile for further analysis.

**PARTIALSUCCESS**

some volumes in range were not processed

**Explanation:**

On a command using a volume range, some of the volumes in the range were not processed.

**User Action:**

Verify the state of all objects in the range, and issue corrective commands if necessary.

**POLUNDEFINED**

referenced pool !AZ undefined

**Explanation:**

When creating or modifying a valid object, the object's record contains a reference to a pool name that does not exist.

**User Action:**

Check spelling of the pool name and retry, or create the pool object in the database.

**POOLEXISTS**

specified pool already exists

**Explanation:**

The specified pool already exists and cannot be created.

**User Action:**

Use a set command to modify the pool, or create a new pool with a different name.

**QUEUED**

operation is queued for processing

**Explanation:**

The asynchronous request you entered has been queued for processing.

**User Action:**

You can check on the state of the request by issuing a show requests command.

**RDFERROR**

error allocating or deallocating RDF device

**Explanation:**

During an allocation or deallocation of a drive using RDF, the RDF software returned an error.

**User Action:**

The error following this error is the RDF error return.

**REQUESTID**

request ID is !@UL

**Explanation:**

The number is the request ID for the command just queued.

**User Action:**

None

**RESUNDEFINED**

referenced restore(s) !AZ undefined

## MDMS Messages

**Explanation:**

When creating or modifying a valid object, the object's record contains a reference to a restore name that does not exist. One or more of the specified restores may be undefined.

**User Action:**

Check spelling of the restore names and retry, or create the restore objects in the database.

**SCHEDCREATEERR**

failed to create a scheduling job

**Explanation:**

MDMS failed to create a scheduling job.

**User Action:**

Report the incident to Compaq.

**SCHEDELETEERR**

failed to delete a scheduling job

**Explanation:**

MDMS failed to delete a scheduling job.

**User Action:**

Report the incident to Compaq.

**SCHEDDISCONNECT**

scheduler disconnected from mailbox

**Explanation:**

The scheduler was disconnected from a mailbox

**User Action:**

Report the incident to Compaq.

**SCHEDDUPJOB**

duplicate scheduler job found

**Explanation:**

MDMS found a duplicate scheduling job

**User Action:**

Report the incident to Compaq.

**SCHEDEXTSTATUS**

external schedule job exited with bad status

**Explanation:**

An external schedule job exited with bad status

**User Action:**

Report the incident to Compaq.

**SCHELOOPERR**

schedule thread terminating with fatal error, restarting



**Explanation:**

The MDMS internal schedule thread encountered an error and terminated. The thread is restarted.

**User Action:**

Report the problem to Compaq.

**SCHEDMODIFYERR**

failed to modify a scheduling job

**Explanation:**

MDMS failed to modify a scheduling job.

**User Action:**

Report the incident to Compaq.

**SCHEDNOJOBCOMPLETE**

no job complete time was returned from a scheduled job

**Explanation:**

No job complete time was returned from a scheduled job.

**User Action:**

Report the incident to Compaq.

**SCHEDNOJOBEXISTS**

no job exists was returned from a scheduled job

**Explanation:**

No job exists was returned from a scheduled job.

**User Action:**

Report the incident to Compaq.

**SCHEDNOJOBNUM**

no job number was returned from a scheduled job

**Explanation:**

No job number was returned from a scheduled job.

**User Action:**

Report the incident to Compaq.

**SCHEDNOJOBSTART**

no job start time was returned from a scheduled job

**Explanation:**

No job start time was returned from a scheduled job.

**User Action:**

Report the incident to Compaq.

**SCHEDNOJOBSTATUS**

no job status was returned from a scheduled job

## MDMS Messages

**Explanation:**

No job status was returned from a scheduled job.

**User Action:**

Report the incident to Compaq.

**SCHEDNOSUCHJOB**

failed to find a scheduling job

**Explanation:**

MDMS failed to find a scheduling job.

**User Action:**

Report the incident to Compaq.

**SCHEDSHOWERR**

failed to show a scheduling job

**Explanation:**

MDMS failed to show a scheduling job.

**User Action:**

Report the incident to Compaq.

**SCHEDSYSTEMERR**

failed to access the internal scheduler queue

**Explanation:**

An MDMS call to a system service failed in the scheduler functions.

**User Action:**

Report the incident to Compaq.

**SCHEDULECONFL**

schedule qualifier and novolume qualifier are incompatible

**Explanation:**

The /SCHEDULE and /NOVOLUME qualifiers are incompatible for this command.

**User Action:**

Use the /SCHEDULE and /VOLSET qualifiers for this command.

**SCHEDVOLCONFL**

schedule qualifier and volume parameter are incompatible

**Explanation:**

The /SCHEDULE and the volume parameter are incompatible for this command.

**User Action:**

Use the /SCHEDULE qualifier and leave the volume parameter blank for this command.

**SCHEDULECONFL**

schedule qualifier and novolume qualifier are incompatible

**Explanation:**

The /SCHEDULE and /NOVOLUME qualifiers are incompatible for this command.

**User Action:**

Use the /SCHEDULE and /VOLSET qualifiers for this command.

**SCHEDVOLCONFL**

schedule qualifier and volume parameter are incompatible

**Explanation:**

The /SCHEDULE and the volume parameter are incompatible for this command.

**User Action:**

Use the /SCHEDULE qualifier and leave the volume parameter blank for this command.

**SCHUNDEFINED**

referenced schedule(s) !AZ undefined

**Explanation:**

When creating or modifying a valid object, the object's record contains a reference to a schedule name that does not exist. One or more of the specified schedules may be undefined.

**User Action:**

Check spelling of the schedule names and retry, or create the schedule objects in the database.

**SETLOCALEFAIL**

an error occurred when accessing locale information

**Explanation:**

When executing the SETLOCALE function an error occurred.

**User Action:**

A user should not see this error.

**SETPROTECTED**

protected field(s) set, verify consistency

**Explanation:**

You have directly set a protected field with this command. Normally these fields are maintained by MDMS. This has the potential to make the database inconsistent and cause other operations to fail.

**User Action:**

Do a SHOW /FULL on the object(s) you have just modified and verify that your modifications leave the object(s) in a consistent state.

**SLSDBINUSE**

SLS\$DB network object in use

**Explanation:**

The MDMS server could not be started because it could not declare the network task SLS\$DB. The network task SLS\$DB is already in use.

**User Action:**

Check the server's logfile for more information. Check the logical MDMS\$SUPPORT\_PRE\_V3 in the system table. If this is TRUE and the SLS\$TAPMGRDB process is running the server cannot be started. Shut down the SLS\$TAPMGRDB process by shutting down SLS. Restart MDMSV3.0 server and then restart SLS.

## MDMS Messages

### **SNDMAILFAIL**

send mail failed, see log file for more explanation

#### **Explanation:**

While sending mail during the scheduled activities, a call to the mail utility failed.

#### **User Action:**

Check the log file for the failure code from the mail utility.

### **SOMESUCCESS**

some objects in list were not processed

#### **Explanation:**

The request was partially successful, but some of the objects were not processed as shown in the extended status.

#### **User Action:**

Examine the extended status, and retry command as needed.

### **SPAWNCMDBUFOVR**

spawn command buffer overflow

#### **Explanation:**

During the mount of a volume, the spawned mount command was too long for the buffer. This is an internal error.

#### **User Action:**

Provide copies of the MDMS command issued, the database files and the server's logfile for further analysis.

### **SVRBUGCHECK**

internal inconsistency in SERVER

#### **Explanation:**

The MDMS server software (MDMS\$SERVER.EXE) detected an inconsistency. This is an internal error.

#### **User Action:**

Provide copies of the MDMS command issued, the database files and the server's logfile for further analysis. Restart the server.

### **SVRDISCON**

server disconnected

#### **Explanation:**

The server disconnected from the request because of a server problem or a network problem.

#### **User Action:**

Check the server's logfile and file SYSS\$MANAGER:MDMS\$SERVER.LOG for more information. Provide copies of the MDMS command issued, the database files and the server's logfile for further analysis.

### **SVREXIT**

server exited

**Explanation:**

Server exited. Check the server logfile for more information.

**User Action:**

Depends on information in the logfile.

**SVRLOGERR**

server logged error

**Explanation:**

The server failed to execute the request. Additional information is in the server's logfile.

**User Action:**

Depends on information in the logfile.

**SVRRUN**

server already running

**Explanation:**

The MDMS server is already running.

**User Action:**

Use the MDMS\$\$SHUTDOWN procedure with parameter RESTART to restart the server.

**SVRSTART**

Server !AZ!UL.!UL-!UL started

**Explanation:**

The server has started up identifying its version and build number.

**User Action:**

None.

**SVRSTARTSTRING**

Server !AZ started

**Explanation:**

The server has started up identifying its version and build number.

**User Action:**

None.

**SVRTERM**

Server terminated abnormally

**Explanation:**

The MDMS server was shut down. This could be caused by a normal user shutdown or it could be caused by an internal error.

**User Action:**

Check the server's logfile for more information. If the logfile indicates an error has caused the server to shut down then provide copies of the MDMS command issued, the database files and the server's logfile for further analysis.

## MDMS Messages

### SVRUNEXP

unexpected error in SERVER !AZ line !UL

#### **Explanation:**

The server software detected an internal inconsistency.

#### **User Action:**

Provide copies of the MDMS command issued, the database files and the server's logfile for further analysis.

### TCPIPLISEXIT

TCP/IP listener exited

#### **Explanation:**

The TCP/IP listener has exited due to an internal error condition or because the user has disabled the TCPIP transport for this node. The TCP/IP listener is the server's routine to receive requests via TCP/IP.

#### **User Action:**

The TCP/IP listener should be automatically restarted unless the TCPIP transport has been disabled for this node. Provide copies of the MDMS command issued, the database files and the server's logfile for further analysis if the transport has not been disabled by the user.

### TCPIPLISRUN

listening on TCP/IP node !AZ port !AZ

#### **Explanation:**

The server has successfully started a TCP/IP listener. Requests can now be sent to the server via TCP/IP.

#### **User Action:**

None.

### TOOLARGE

entry is too large

#### **Explanation:**

Either entries cannot be added to a list of an MDMS object or existing entries cannot be renamed because the maximum list size would be exceeded.

#### **User Action:**

Remove other elements from list and try again.

### TOOMANY

too many objects generated

#### **Explanation:**

You attempted to perform an operation that generated too many objects.

#### **User Action:**

There is a limit of 1000 objects that may be specified in any volume range, slot range or space range. Re-enter command with a valid range.

### TOOMANYSORTS

too many sort qualifiers, use only one

**Explanation:**

When specify more than one field to sort on.

**User Action:**

Specify only one field to sort on.

**UNDEFINEDREFS**

success, but object references undefined objects

**Explanation:**

The command was successful, but the object being created or modified has references to undefined objects. Subsequent messages indicate which objects are undefined.

**User Action:**

This allows objects to be created in any order, but some operations may not succeed until the objects are defined. Verify/correct the spelling of the undefined objects or create the objects if needed.

**UNKVOLENT**

unknown volume !AZ entered in jukebox !AZ

**Explanation:**

A volume unknown to MDMS has been entered into a jukebox.

**User Action:**

Use the INVENTORY command to make the volume known to MDMS or use a jukebox utility program (CARTRIDGE or MRU) to eject the volume from the jukebox.

**UNSUPPORTED**

unsupported function

**Explanation:**

You attempted to perform an unsupported function.

**User Action:**

None.

**UNSUPPORTED1**

unsupported function !AZ

**Explanation:**

You attempted to perform an unsupported function.

**User Action:**

None.

**UNSUPRECVER**

unsupported version for record !AZ in database !AZ

**Explanation:**

The server has detected unsupported records in a database file. These records will be ignored.

**User Action:**

Consult the documentation about possible conversion procedures provided for this version of MDMS.

## MDMS Messages

### **USERNOTAUTH**

user is not authorized for volume pool

#### **Explanation:**

When allocating a volume, you specified a pool for which you are not authorized.

#### **User Action:**

Specify a pool for which you are authorized, or add your name to the list of authorized users for the pool. Make sure the authorized user includes the node name or group name in the pool object.

### **VISIONCONFL**

vision option and volume parameter are incompatible

#### **Explanation:**

You attempted to create volumes with the vision option and the volume parameter. This is not supported.

#### **User Action:**

The vision option is used to create volumes with the volume identifiers read by the vision system on a jukebox. Re-enter the command with either the vision option (specifying jukebox and slot range), or with volume identifier(s), but not both.

### **VOLALRALLOC**

specified volume is already allocated

#### **Explanation:**

You attempted to allocate a volume that is already allocated.

#### **User Action:**

Use another volume.

### **VOLALRINIT**

volume is already initialized and contains data

#### **Explanation:**

When initializing a volume, MDMS detected that the volume is already initialized and contains data.

#### **User Action:**

If you are sure you still want to initialize the volume, re-enter the command with the overwrite option.

### **VOLIDICM**

volume ID code missing

#### **Explanation:**

The volume ID is missing in a request.

#### **User Action:**

Provide volume ID and retry request.

### **VOLINDRV**

volume is currently in a drive



**Explanation:**

When allocating a volume, the volume is either moving or in a drive, and nopreferred was specified.

**User Action:**

Wait for the volume to be moved or unloaded, or use the preferred option.

**VOLINJUKE**

volume is in a jukebox

**Explanation:**

You attempted load a volume that is currently in a jukebox into a drive that is not in the jukebox.

**User Action:**

Load the volume into a drive within the current jukebox, or check the jukebox name for the drive.

**VOLINSET**

volume is already bound to a volume set

**Explanation:**

You cannot bind this volume because it is already in a volume set and is not the first volume in the set.

**User Action:**

Use another volume, or specify the first volume in the volume set.

**VOLLOST**

volume location is unknown

**Explanation:**

The volume's location is unknown.

**User Action:**

Check if the volume's placement is in a magazine, and if so if the magazine is defined. If not, create the magazine. Also check the magazine's placement.

**VOLMOVE**

volume cannot be loaded but can be moved to jukebox or drive

**Explanation:**

The volume is not currently in a placement where it can be loaded, but can be moved there.

**User Action:**

Move the volume to the drive, or use the automatic move option on the load and retry.

**VOLMOVING**

volume is currently being moved

**Explanation:**

In a move, load or unload command, the specified volume is already being moved.

**User Action:**

Wait for volume to come to a stable placement and retry. If the volume is stuck in the moving placement, check for an outstanding request and cancel it. If all else fails, manually change volume state.

## MDMS Messages

### **VOLNOTALLOC**

specified volume is not allocated

#### **Explanation:**

You attempted to bind or deallocate a volume that is not allocated.

#### **User Action:**

None for deallocate. For bind, allocate the volume and then bind it to the set, or use another volume.

### **VOLNOTBOUND**

volume is not bound to a volume set

#### **Explanation:**

You attempted to unbind a volume that is not in a volume set.

#### **User Action:**

None.

### **VOLNOTINACS**

one or more volumes are not in this ACS

#### **Explanation:**

One or more volumes for the command are not in this ACS.

#### **User Action:**

Verify that all volumes are in the same ACS and that the ACS id is correct.

### **VOLNOTINJUKE**

volume is not in a jukebox

#### **Explanation:**

When loading a volume into a drive, the volume is not in a jukebox.

#### **User Action:**

Use the move option and retry the load. This will issue OPCOM messages to move the volume into the jukebox.

### **VOLNOTINPOOL**

loaded volume is not in the specified pool

#### **Explanation:**

During a scratch load of a volume in a drive, the volume loaded was not in the requested pool.

#### **User Action:**

Load another volume that is in the requested pool. A recommended volume is printed in the OPCOM message. Note that if no pool was specified, the volume must have no pool defined.

### **VOLNOTLOADED**

the volume is not loaded in a drive

#### **Explanation:**

On an unload request, the volume is not recorded as loaded in a drive.

#### **User Action:**

If the volume is not in a drive, none. If it is, issue an unload drive command to unload it.

**VOLONOTHDRV**

volume is currently in another drive

**Explanation:**

When loading a volume, the volume was found in another drive.

**User Action:**

Wait for the volume to be unloaded, or unload the volume and retry.

**VOLSALLOC**

!AZ volumes were successfully allocated

**Explanation:**

When attempting to allocate multiple volumes using the quantity option, some but not all of the requested quantity of volumes were allocated.

**User Action:**

See accompanying message as to why not all volumes were allocated.

**VOLSDRIVES**

one or more of the volumes are in drives or are moving

**Explanation:**

One or more of the volumes in the move request are in drives and cannot be moved. A show volume /brief will identify which volumes are in drives.

**User Action:**

Unload the volume(s) in drives and retry, or retry without specifying the volumes in drives.

**VOLUMEEXISTS**

specified volume(s) already exist

**Explanation:**

The specified volume or volumes already exist and cannot be created.

**User Action:**

Use a set command to modify the volume(s), or create new volume(s) with different names.

**VOLUNDEFINED**

referenced volume !AZ undefined

**Explanation:**

When creating or modifying a valid object, the object's record contains a reference to a volume ID that does not exist.

**User Action:**

Check spelling of the volume ID and retry, or create the volume object in the database.

**VOLWRTLCK**

volume loaded with hardware write-lock

**Explanation:**

The requested volume was loaded in a drive, but is hardware write-locked when write access was requested.

## MDMS Messages

### **User Action:**

If you need to write to the volume, unload it, physically enable it for write, and re-load it.

### **WRONGLABEL**

initializing volume !AZ as !AZ is disallowed

### **Explanation:**

The label of the volume loaded in the drive for initialization does not match the requested volume label and there is data on the volume. Or initializing the volume with the requested label causes duplicate volumes in the same jukebox or location.

### **User Action:**

If you wish to overwrite the volume label, re-issue the command with the overwrite qualifier. If there are duplicate volumes in the same location or jukebox you need to move the other volume from the jukebox or location before retrying.

### **WRONGVOLUME**

wrong volume label or unlabelled volume was loaded

### **Explanation:**

On a load volume command, MDMS loaded a volume with the wrong volume label or a blank volume label into the drive.

### **User Action:**

Check the volume, and optionally perform an initialization of the volume and retry. If this message is displayed in an OPCOM message, you will need another free drive to perform the initialization. The volume has been unloaded.

---

## MDMS Rights and Privileges

This Appendix has explanation for MDMS user rights and privileges.

Every MDMS user/potential user will be assigned zero or more rights in their SYSUAF file. These rights will be examined on a per-command basis to determine whether a user has sufficient privilege to issue a command. The command is accepted for processing only if the user has sufficient privilege. In case the user has no rights the **entire command** is rejected. Each right has a name in the following format:

```
MDMS_rightname.
```

Rights are looked-up on the *client OpenVMS node* that receives the request, as such each user must have an account on the *client node*.

- in the case of DCL commands and applications, this would be the node at which the request is issued.
- from the GUI, it is the node whose MDMS\$SERVER process receives the request. The rights are translated into a bitmap and passed to the database server for validation.

### C.1 MDMS Rights - Types

MDMS has the following rights:

- High-level rights
- Low level rights
- ABS rights

#### C.1.1 High Level Rights

These rights are designed for a specific kind of user, to support a typical MDMS installation, and make the assignments of rights to users easy. The **three** high-level MDMS rights, the default right, administrator right and the additional right are described in Table C-1.

## MDMS Rights and Privileges

### C.1 MDMS Rights - Types

Table C–1 High Level Rights

High level right	Allows Privileges for...
MDMS_USER	A non-privileged MDMS user who wants to use MDMS to manage tape volumes for BACKUP, ABS or HSM purposes
MDMS_APPLICATION	Main applications that MDMS supports - ABS and HSM server processes
MDMS_OPERATOR	The user responsible for day-to-day operations in the MDMS environment.
Default Right A <i>hidden</i> high-level right	The low level rights contained in it, for users with no MDMS rights. They are additional to any specific rights a user may have been granted. It is the <i>default right</i> . By default, there are no low-level rights assigned to the <i>default right</i> . If rights are assigned to the default right, they apply to all users in the system, since every user is effectively granted the default right. The default right can be disabled with the MDMS_NO_DEFAULT identifier in a user's UAF file.
MDMS_ALL_RIGHTS Administrator Right	A system administrator to perform any operation. MDMS_ALL_RIGHTS can be enabled with the OpenVMS SYSPRV privilege.
Additional Right	All operations.
MDMS_USER	A non-privileged MDMS user who wants to use MDMS to manage tape volumes for BACKUP, ABS or HSM purposes
MDMS_APPLICATION	Main applications MDMS supports - ABS and HSM server processes
MDMS_OPERATOR	User responsible for day-to-day operations in the MDMS environment.
Default Right A <i>hidden</i> high-level right	The low level rights contained in it, for users with no MDMS rights. They are additional to any specific rights a user may have been granted. It is the <i>default right</i> . By default, there are no low-level rights assigned to the <i>default right</i> . If rights are assigned to the default right, they apply to all users in the system, since every user is effectively granted the default right. The default right can be disabled with the MDMS_NO_DEFAULT identifier in a user's UAF file.
MDMS_ALL_RIGHTS Administrator Right	A system administrator to perform any operation. MDMS_ALL_RIGHTS can be enabled with the OpenVMS SYSPRV privilege.
Additional Right	All operations.

You can disable the mapping of SYSPRV to MDMS\_ALL\_RIGHTS using a SET DOMAIN command

#### C.1.2 Low-level rights

Each command or command option will be tagged with one or more low-level rights that are needed to perform the operation. Where more than one right is specified, the command indicates the appropriate combination of rights needed. The MDMS administrator can assign a set of *low-level rights* to each high-level right. The administrator can then simply assign the high-level right to the user.

MDMS translates the high-level right to respective low-level rights while processing a command. For additional flexibility, the user can be assigned a combination of high-level and low-level rights. The result will be a sum of all rights defined.

## MDMS Rights and Privileges

### C.1 MDMS Rights - Types

The default set of mapping of high-level to low-level rights will be assigned at installation (by default) and stored in the domain record. However, the MDMS administrator can change these assignments by using the **SET DOMAIN** command.

---

#### Note

---

**By default a user has no rights and cannot use MDMS. The system administrator can change the 'rightless' user's rights using a SET DOMAIN command. These rights can again be disabled on a per-user basis as needed.**

---

The low-level rights are designed to be applied to operations. A given command, with a given set of qualifiers or options, requires the sum of the rights needed for the command **and all supplied options**. In many cases some options require more privilege than the command, and that higher privilege will be applied to the entire command if those options are specified.

The following are usable low level rights:

**Table C-2 Low Level Rights**

---

<b>Low Level Right Name</b>	<b>Allows Privilege to:</b>
MDMS_ALL_RIGHTS	Enable all operations (This right is for the system administrator.)
MDMS_ALLOCATE_ALL	Allocate volumes or drives for any user
MDMS_ALLOCATE_OWN	Allocate a drive and become "owner"
MDMS_ALLOCATE_POOL	Allocate a volume from an authorized pool
MDMS_ASSIST	Request operator assistance on calls
MDMS_BIND_ALL	Bind any volumes together in a volume set
MDMS_BIND_OWN	Bind owned volumes together in a volume set
MDMS_CANCEL_ALL	Cancel any request
MDMS_CANCEL_OWN	Cancel one's own requests
MDMS_CANCEL_POOL	Cancel a request of a member of the same pool
MDMS_CREATE_ALL	Create any database object
MDMS_CREATE_POOL	Create volumes in a pool authorized to user
MDMS_DEALLOCATE_ALL	Deallocate volumes for any user
MDMS_DEALLOCATE_OWN	Deallocate an owned volume or drive
MDMS_DELETE_ALL	Delete any database object
MDMS_DELETE_POOL	Delete volumes in pool authorized to user
MDMS_INITIALIZE_ALL	Initialize any volume
MDMS_INITIALIZE_POOL	Initialize a volume in pool authorized to user
MDMS_INVENTORY_ALL	Perform inventory on any jukebox
MDMS_LOAD_ALL	Load any volumes including scratch volumes
MDMS_LOAD_OWN	Load owned volumes into drives

---

## MDMS Rights and Privileges

### C.2 Default High-Level to Low-Level Mapping

**Table C–2 Low Level Rights**

<b>Low Level Right Name</b>	<b>Allows Privilege to:</b>
MDMS_LOAD_POOL	Load volumes in pool authorized to user
MDMS_LOAD_SCRATCH	Load scratch volumes
MDMS_MOVE_ALL	Move any volume
MDMS_MOVE_OWN	Move owned volumes
MDMS_MOVE_POOL	Move volumes in pool authorized to user
MDMS_SET_ALL	SET (modify) any database object
MDMS_SET_PROTECTED	SET internal MDMS attributes in an object
MDMS_SET_OWN	SET (modify) volumes allocated to user
MDMS_SET_POOL	SET (modify) volumes in pool authorized to user
MDMS_SET_RIGHTS	SET (modify) rights in the domain
MDMS_SHOW_ALL	SHOW or REPORT any database object
MDMS_SHOW_OWN	SHOW or REPORT volumes allocated to user
MDMS_SHOW_POOL	SHOW or REPORT volumes in pool authorized to user
MDMS_SHOW_RIGHTS	Show rights with a SHOW DOMAIN/FULL
MDMS_UNBIND_ALL	Unbind any volumes
MDMS_UNBIND_OWN	Unbind owned objects from a volume set
MDMS_UNLOAD_ALL	Unload any volumes or drives
MDMS_UNLOAD_OWN	Unload volumes allocated to user from a drive
MDMS_UNLOAD_POOL	Unload volumes in pool authorized to user

## C.2 Default High-Level to Low-Level Mapping

This section defines the default high to low-level mapping for each high-level right.

### C.2.1 MDMS\_USER:

**Table C–3 MDMS\_USER Rights**

<b>MDMS User...</b>	<b>Allows privilege to...</b>
MDMS_ALLOCATE_OWN	Allocate a drive and become “owner”
MDMS_ALLOCATE_POOL	Allocate a volume from a pool authorized to user
MDMS_ASSIST	Request operator assistance on calls
MDMS_BIND_OWN	Bind owned volumes together in a volume set
MDMS_CANCEL_OWN	Cancel one’s own requests
MDMS_DEALLOCATE_OWN	Deallocate an owned volume or drive



## MDMS Rights and Privileges

### C.2 Default High-Level to Low-Level Mapping

MDMS User...	Allows privilege to...
MDMS_LOAD_OWN	Load owned volumes into drives
MDMS_SHOW_OWN	SHOW or REPORT volumes allocated to user
MDMS_SHOW_POOL	SHOW or REPORT volumes in pool authorized to user
MDMS_UNBIND_OWN	Unbind owned objects from a volume set
MDMS_UNLOAD_OWN	Unload volumes allocated to user from a drive

### C.2.2 MDMS\_OPERATOR Rights:

**Table C-4 Operator Rights**

MDMS Operator...	Allows privilege to...
MDMS_ALLOCATE_ALL	Allocate volumes or drives for any user
MDMS_ASSIST	Request operator assistance on calls
MDMS_BIND_ALL	Bind any volumes together in a volume set
MDMS_CANCEL_ALL	Cancel any request
MDMS_CREATE_POOL	Create volumes in a pool authorized to user
MDMS_DEALLOCATE_ALL	Deallocate volumes for any user
MDMS_DELETE_POOL	Delete volumes in pool authorized to user
MDMS_INITIALIZE_ALL	Initialize any volume
MDMS_INVENTORY_ALL	Perform inventory on any jukebox
MDMS_LOAD_ALL	Load any volumes including scratch volumes
MDMS_MOVE_ALL	Move any volume
MDMS_SET_OWN	SET (modify) volumes allocated to user
MDMS_SET_POOL	SET (modify) volumes in pool authorized to user
MDMS_SHOW_ALL	SHOW or REPORT any database object
MDMS_SHOW_RIGHTS	Show rights with SHOW DOMAIN/FULL
MDMS_UNBIND_ALL	Unbind any volumes
MDMS_UNLOAD_ALL	Unload any volumes or drives

## C.2 Default High-Level to Low-Level Mapping

### C.2.2.1 Domain Commands for Mapping Privileges

```
SET DOMAIN
/[NO]ABS_RIGHTS
/ADD
/[NO]APPLICATION_RIGHTS[=(right[,...])]
/[NO]DEFAULT_RIGHTS[=(right[,...])]
/[NO]OPERATOR_RIGHTS[=(right[,...])]
/REMOVE
/[NO]SYSPRV
/[NO]USER_RIGHTS[=(right[,...])]
```

#### Example C-1

```
SET DOMAIN /OPERATOR_RIGHTS=MDMS_SET_PROTECTED /ADD
```

This command adds the MDMS\_SET\_PROTECTED right to the operator rights list.

---

# Index

---

## A

ANALYZE command 1-27

---

## B

BACKUP command 1-7

---

## C

Cache 1-53

Cache

the entire disk 1-54

CHECKPOINT command 1-30

Commands

DCL 1-97

COPY command 1-17

---

## D

DCL Commands - Summary 1-2

default low water mark 1-64

DIRECTORY command 1-6

---

## E

EXIT command 1-36

---

## H

HELP command 1-37

HSM\$BACKUP command 1-7

---

## L

LOCATE command 1-38

low water mark default 1-64

---

## M

Magneto-optical devices 1-53

MDMS ALLOCATE DRIVE 2-2

MDMS ALLOCATE VOLUME 2-6

MDMS BIND VOLUME 2-11

MDMS CANCEL REQUEST 2-13

MDMS CREATE DRIVE 2-15

MDMS CREATE GROUP 2-20

MDMS CREATE JUKEBOX 2-23

MDMS CREATE LOCATION 2-29

MDMS CREATE MAGAZINE 2-32

MDMS CREATE MEDIA\_TYPE 2-36

MDMS CREATE NODE 2-39

MDMS CREATE POOL 2-44

MDMS CREATE SCHEDULE 2-47

MDMS CREATE VOLUME 2-52

MDMS DEALLOCATE DRIVE 2-61

MDMS DEALLOCATE VOLUME 2-62

MDMS DELETE DRIVE 2-65

MDMS DELETE GROUP 2-66

MDMS DELETE JUKEBOX 2-67

MDMS DELETE LOCATION 2-68

MDMS DELETE MAGAZINE 2-69

MDMS DELETE MEDIA\_TYPE 2-70

MDMS DELETE NODE 2-71

MDMS DELETE POOL 2-72

MDMS DELETE SCHEDULE 2-73

MDMS DELETE VOLUME 2-74

MDMS INITIALIZE VOLUME 2-75

MDMS INVENTORY JUKEBOX 2-78

MDMS LOAD DRIVE 2-81

MDMS LOAD VOLUME 2-83

MDMS MOVE MAGAZINE 2-86

MDMS MOVE VOLUME 2-89

MDMS REPORT VOLUME 2-93

MDMS SET DOMAIN 2-99

MDMS SET DRIVE 2-105

MDMS SET GROUP 2-109

MDMS SET JUKEBOX 2-111

MDMS SET LOCATION 2-116

MDMS SET MAGAZINE 2-118

MDMS SET MEDIA\_TYPE 2-121

MDMS SET NODE 2-123

MDMS SET POOL 2-127

MDMS SET SCHEDULE 2-130

MDMS SET SERVER 2-134

MDMS SET VOLUME 2-135

MDMS SHOW DOMAIN 2-143

MDMS SHOW DRIVE 2-145

MDMS SHOW GROUP 2-148

MDMS SHOW JUKEBOX 2-150

MDMS SHOW LOCATION 2-152

MDMS SHOW MAGAZINE 2-154

MDMS SHOW NODE 2-158

MDMS SHOW POOL 2-160

MDMS SHOW REQUEST 2-162

MDMS SHOW SCHEDULE 2-164

MDMS SHOW SELECTION 2-166

MDMS SHOW SERVER 2-166

MDMS SHOW VERSION 2-168

MDMS SHOW VOLUME 2-169

MDMS UNBIND VOLUME 2-175  
MDMS UNLOAD DRIVE 2-177  
MDMS UNLOAD VOLUME 2-179

## **P**

---

PRESHELVE command 1-9  
Privileges  
for SMU Command 1-26  
Privileges  
for SMU Startup 1-98

## **R**

---

RANK command 1-44  
REPACK command 1-46

## **S**

---

SET ARCHIVE command 1-50  
SET CACHE command 1-53  
SET DEVICE command 1-56  
SET FACILITY command 1-60  
SET FILE command 1-12  
SET POLICY command 1-62  
SET PROCESS command 1-13  
SET SCHEDULE command 1-68  
SET SHELF command 1-71  
SET VOLUME command 1-75  
SHELVE command 1-14  
SHOW ARCHIVE command 1-30  
SHOW CACHE command 1-81  
SHOW DEVICE command 1-82  
SHOW FACILITY command 1-84  
SHOW POLICY command 1-85  
SHOW REQUESTS command 1-87  
SHOW SCHEDULE command 1-89  
SHOW SHELF command 1-91  
SHOW VERSION command 1-93  
SHOW VOLUME command 1-94  
SHUTDOWN command 1-96  
SMU CHECKPOINT command 1-30  
SMU commands 1-26  
SMU COPY command 1-32  
SMU EXIT command 1-36  
SMU HELP command 1-37  
SMU LOCATE command 1-38  
SMU RANK command 1-44  
SMU REPACK command 1-46  
SMU SET ARCHIVE command 1-50  
SMU SET CACHE command 1-53  
SMU SET DEVICE command 1-56

SMU SET FACILITY command 1-60  
SMU SET POLICY command 1-62  
SMU SET SCHEDULE command 1-68  
SMU SET SHELF command 1-71  
SMU SET VOLUME command 1-75  
SMU SHOW ARCHIVE command 1-79  
SMU SHOW CACHE command 1-81  
SMU SHOW DEVICE command 1-82  
SMU SHOW FACILITY command 1-84  
SMU SHOW POLICY command 1-85  
SMU SHOW REQUESTS command 1-87  
SMU SHOW SCHEDULE command 1-89  
SMU SHOW SHELF command 1-91  
SMU SHOW VERSION command 1-93  
SMU SHOW VOLUME command 1-94  
SMU SHUTDOWN command 1-96  
SMU SPAWN command 1-97  
SMU STARTUP command 1-98  
SPAWN command 1-97  
STARTUP command 1-98

## **U**

---

UNPRESHELVE command 1-18  
UNSHELVE command 1-21