Archive Backup System for OpenVMS

Installation Guide

Software Version	Archive Backup System for OpenVMS Version 3.2
Required Operating System	OpenVMS VAX Version 6.2, 7.1, 7.2 and 7.3 OpenVMS Alpha Version 6.2, 7.1-2, 7.2-1 and 7.3
Required Software	Media and Device Management Services for OpenVMS Version 3.2
	DECnet (Phase IV) or DECnet-Plus(Phase V)
	TCP/IP Services for OpenVMS

Contents

4 Performing Postinstallation Tasks

A Examples of Authorizing NT and UNIX Clients

B File and Logical Names

C MDMS Files and Logical Names

D MDMS V3 Rights and Privileges

E   Sample Configuration of MDMS

F   Upgrading ABS From the ABS-OMT License

G   How to Delete ABS/MDMS From Your System

Preface

Intended Audience

This document is intended for storage administrators who are experienced OpenVMS system managers. This document should be used in conjunction with the *Introduction to OpenVMS System Management* manual.

Conventions

The following conventions are used in this document:

Convention	Description
{}	In format command descriptions, braces indicate required elements.
[ ]	In format command descriptions, square brackets indicate optional elements of the command syntax. You can omit these elements if you wish to use the default responses.
boldface type	Boldface type in text indicates the first instance of a term defined in the Glossary or defined in text.
italic type	Italic type emphasizes important information, indicates variables, indicates complete titles of manuals, and indicates parameters for system information.
`Starting test ...`	This type font denotes system response, user input, and examples.
Ctrl/*x*	Hold down the key labels Ctrl (Control) and the specified key simultaneously (such as Ctrl/Z).
PF1 *x*	The key sequence PF1 *x* instructs you to press and release the PF1 key, and then press and release another key (indicated here by *x*).
n	A lowercase italic *n* denotes the generic use of a number. For example, 19*nn* indicates a four digit number in which the last two digits are unknown.
x	A lowercase *x* denotes the generic use of a letter. For example, *xxx* indicates any combination of three alphabetic characters.

Related Products

The following related products may be mentioned in this document:

Product	Description
HSM	HSM refers to Hierarchical Storage Management for OpenVMS software.
MDMS	MDMS refers to Media and Device Management Services for OpenVMS software.
OpenVMS	OpenVMS refers to OpenVMS operating system.
SMF	SMF refers to Sequential Media Filesystem for OpenVMS software.
SLS	SLS refers to Storage Library System for OpenVMS software.

Associate Documents

The following documents are part of Archive Backup System for OpenVMS documentation set:

- Archive Backup System for OpenVMS Installation Guide
- Archive Backup System for OpenVMS Guide to Operations
- Archive Backup System for OpenVMS Command Reference Guide

1

Welcome To ABS

_____Note_____

If you are installing for the first time, it is essential that you review the information presented in this chapter. If you are a current ABS customer and are upgrading, you may skip to Chapter 2, _Meeting Installation Requirements_.

Not all the ABS features are available with ABS-OMT license. The following lists the features that are not available for the ABS-OMT license:

• Command line interface (CLI)
• Support for remote tape drives using RDF
• Support for OpenVMS clients
• Creation of storage policies
• Creation of environment policies
• Save request scheduling options except for the following:
– On demand (ON_DEMAND)
– Weekly full/daily incremental (DAILY_FULL_WEEKLY)
– One time only (ONE_TIME_ONLY)
• Support of STK silos and DCSC jukeboxes
• Support for the following scheduling options:
– POLYCENTER Scheduler V2.1b (DECscheduler) (DECSCHEDULER)
– External scheduler (EXT_SCHEDULER)
– External queue manager (EXT_QUEUE_MANAGER)
_____

Welcome to Archive Backup System for OpenVMS (ABS). Because you have selected ABS to help you manage your data safety needs (archive and backup operations), you have a hardware and software environment (subsequently referred to as a storage environment) that contains a set of common elements. Your storage environment also contains very specific or unique elements. The information presented in this chapter is intended to give you an "overall picture" of a typical storage environment, and to explain how ABS compliments that environment.

1.1 What Do All Storage Environments Have In Common?

All storage environments that plan to implement ABS have the following common hardware and software:

*   OpenVMS™ VAX™ or Alpha systems

*   OpenVMS software Version 6.2, 7.1, 7.2, or 7.3 for VAX and OpenVMS software Version 6.2, 7.1-2, 7.2-1 or 7.3 for Alpha systems

*   Disk devices for online storage/transactions

*   DECnet™ Phase IV or DECnet-Plus™ for VAX or Alpha systems

*   Tape drives

• Removable media that is compatible with the tape drives for storing saved data

Figure 1–1 shows a typical storage environment.

Figure 1–1 Typical Storage Environment

CXO6010A
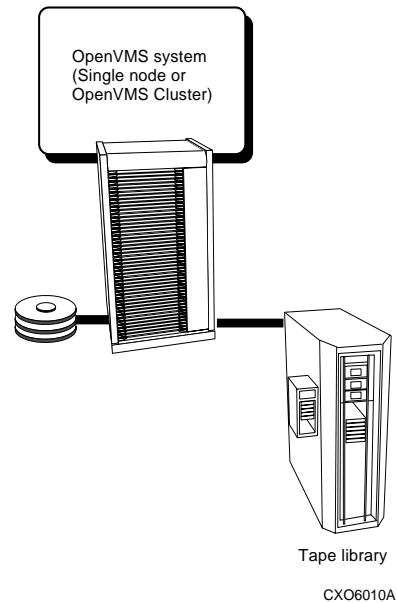
1.2 What Makes a Storage Environment Unique?

Storage environments have some or all of the following characteristics that make them unique:

• Mixed-architecture, OpenVMS Cluster (combination of OpenVMS VAX and Alpha systems). See Section 1.3 for details.

• Heterogeneous client systems (OpenVMS, NT, Tru64 UNIX)

• Types of tape drives (TLZ06, TZ877, and so forth)

• Types of robotic devices (jukeboxes or gravity-fed loaders)

• Types of tape drive connections (direct-connect SCSI or controller-connected)

• Location of tape drives (remote or local)

• Number of disks

• Number of tape drives

Figure 1–2 shows how a storage environment can be unique.

Figure 1–2 Unique Storage Environment

CXO6021A
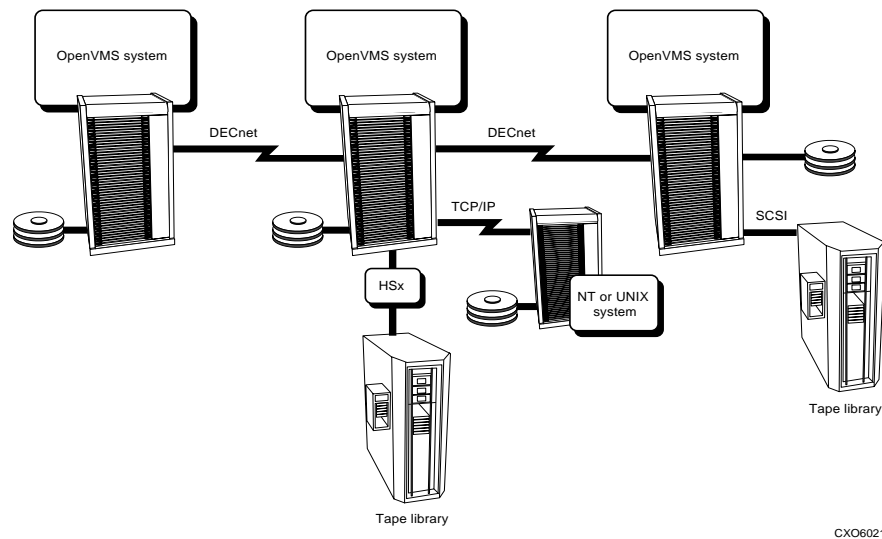
1.3 Mixed Architecture Environments

Beginning with ABS V3.0A, the installation creates new architecture specific directories under the ABS$ROOT:[SYSTEM] directory, [.VAX] and [.ALPHA].

ABS binary files which are architecture specific (VAX or ALPHA) are now placed into ABS$ROOT:[SYSTEM.{VAX|ALPHA}]. This allows for sharing the same ABS$ROOT in a cluster from nodes running either VAX or ALPHA. To share a common ABS$ROOT across a mixed architecture cluster, select the same location of ABS$ROOT during the installation on each node.

1.4 What Is The Purpose of a Managed Media and Device Environment?

The purpose of a managed media and device environment is to maintain a logical view of the physical elements of your storage environment to serve your nearline and offline data storage needs. A managed media and device environment defines the media and defines the drives that can use the media. It also defines the locations where media is stored, the locations of the drives that are compatible with the media, and the policy that governs the use of media.

The following list summarizes the characteristics of the managed media and device environment:

- Volumes are defined as media types in MDMS. Media type definitions are stored in the MDMS volume configuration database. All managed media are known in terms of type, location, capability, availability, and authorization (who can use that media). Before you can use media in your managed storage environment, you must add the media to the MDMS volume configuration database, and initialize the media for use. Once this is done, the media is known as a "volume". ABS recognizes these media type definitions, and depending upon which media type your storage policy uses, performs the backup operation using the appropriate media type and tape drives.

- Tape drive definitions also are stored in MDMS. Tape drives are used to serve the volumes known to MDMS. The MDMS software maintains a logical link between the volumes and the compatible tape drives, both in terms of physical and logical boundaries. Volumes and tape drives can be managed logically from locations miles away from where they are physi-

cally located. ABS depends upon MDMS to select the appropriate tape drives determined by the media type. ABS storage policies associate these logical connections.

• The MDMS software enables you to set the default criteria for moving and recycling volumes. This criteria includes rotation between onsite and offsite locations for safekeeping of the volumes, and the schedule that moves the volume through its lifecycle (retention, use, and reuse). ABS enables you to set the retention criteria for data saved on volumes, while MDMS enables you to define when to move or recycle volumes.

2

Meeting Installation Requirements

To ready your OpenVMS system for either ABS server or ABS client software installation procedure, you must perform certain tasks and requirements:

————————————————Note————————————————

OpenVMS client support is not available with an ABS-OMT license.

_____

- If you are installing ABS for the first time, Table 2–1 describes the tasks you must perform before installing ABS.

- If you are upgrading your version of ABS, refer to Table 2–2.

Table 2–1 Preinstallation Tasks For a New ABS Installation

Step	Action
1.	Decide on which system and disk to install ABS OpenVMS server software. You must make this decision before you can perform the remaining preinstallation tasks. Refer to Section 2.2 for information about ABS server software. Use the worksheet provided in Table 2–3 to record your configuration.
2.	Decide on which systems and disks to install ABS OpenVMS client software. Refer to Section 2.3 for information about ABS OpenVMS client software. Use the worksheet provided in Table 2–5 to record your configuration.
3.	Log in to the SYSTEM account and enable all privileges. Most system managers install software from the system account. See Section 2.5 for the privileges required to install ABS.
4.	Perform a system backup operation. COMPAQ recommends that you perform a backup operation on the system disk before installing any software. For details about performing a backup operation on a system disk, see *OpenVMS System Manager's Manual*.
5.	Verify the OpenVMS operating system subclasses. Subclasses are provided in the OpenVMS save set that contains support for the OpenVMS operating system. Section 2.6 lists the OpenVMS operating system subclasses required to install ABS.
6.	See Section 2.7 to verify the following requirements: • Hardware and software requirements • System parameters • Account process quotas • System processes
7.	Meet prerequisite patches requirements.

8.	Configure your hardware. Before you can use ABS software, you must first install and test your hardware configuration. It is recommended that you do this prior to installing ABS and MDMS software. These tasks are described in Section 2.8.
	Note: The Media Robot Utility (MRU) software provides you with the ability to control the operation of an automated library system or a media loader system without having to access front panel controls on the system. MRU is especially useful for installing, configuring, and testing the operation of these types of system. COMPAQ distributes MRU software with its newest libraries and loaders. If you have recently purchased MRU software and installed it, you will have the MRU software with its newest libraries and loaders. If you are using an earlier COMPAQ library or loader and would like to use MRU software, contact your COMPAQ sales representative.
9.	Register ABS licenses as described in Section 2.11.

Table 2–2 describes the preinstallation tasks to upgrade an existing ABS installation.

Table 2–2 Preinstallation Tasks For an Upgrade ABS Installation

Step	Action
1.	If you have new tape drives that you want to add to your current storage environment, follow the instructions provided in Section 2.8. It is recommended that you do this prior to upgrading ABS or MDMS software.
2.	Log in to the SYSTEM account and enable all privileges. Most system managers install software from the system account. See Section 2.5 for the privileges required to install ABS.
3.	Perform a system backup operation. COMPAQ recommends that you perform a backup operation on the system disk before installing any software. For details about performing a backup operation on a system disk, see *OpenVMS System Manager's Manual*.

4.	Shutdown ABS and MDMS.
	
	If you are upgrading ABS and it is currently running, shut down ABS by entering the following command on all nodes in the OpenVMS Cluster running ABS:
	
	    $ `@SYS$MANAGER:ABS$SHUTDOWN.COM`
	
	Shutdown MDMS by entering the following command on all nodes in the OpenVMS Cluster running MDMS:
	
	    $ `@SYS$STARTUP:MDMS$SHUTDOWN.COM`

5.	Verify the OpenVMS operating system subclasses. Subclasses are provided in the OpenVMS save set that contains support for the OpenVMS operating system. Section 2.6 lists the OpenVMS operating system subclasses required to install ABS.
6.	See Section 2.7 to verify the following requirements: • Hardware and software requirements • System parameters • Account process quotas • System processes
7.	See Section 2.12 to follow steps for converting to another scheduling option.

2.1 Recommended Installation

In the event of a disaster situation it is essential to know where to install ABS and its dependent products so that you can recover the affected system as quickly as possible. To ease the recovery process in the event of a disaster situation, review this information to understand the most efficient way to install ABS, MDMS, and any dependent layered products.

Note

The information provided is only for OpenVMS VAX or Alpha systems. If you have other types of systems to consider, see the platform specific documentation for recovery information.

Consider the following important guidelines:

- You should install ABS server software on a disk with adequate disk space for ABS.

  This disk could be the system disk, or another disk dedicated to ABS and its dependent layered products. The advantage of placing ABS and its dependent layered products on the system disk is that a less complicated restore process is enabled. However, the system disk then requires more space for ABS catalogs, MDMS database, and so forth.

- This disk should be dedicated to ABS and its dependent layered products, not shared with other interactive processes that could impede the performance of ABS.

- The system that contains this disk must have access to a tape drive compatible with the media that will contain ABS save sets for this disk.

Figure 2–1 illustrates the recommended installation.

Figure 2–1 Recommended Installation
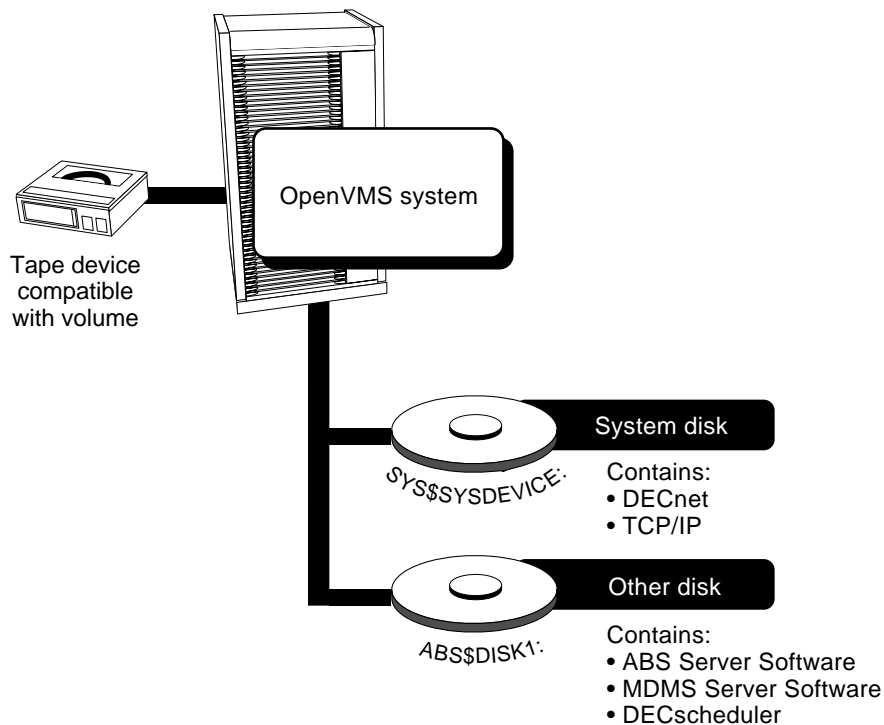
CXO6008B

Before you begin the installation procedure, use the worksheets provided in Table 2–4 to identify which OpenVMS system and disk will contain ABS server software.

2.2 Deciding Where to Install ABS/MDMS Server Software

ABS and MDMS are designed upon a client server technology. Before installing ABS/MDMS software, decide which OpenVMS node or which nodes in an OpenVMS Cluster will run an ABS/MDMS servers.

ABS server software is installed on an OpenVMS node or OpenVMS Cluster system and provides the policy database for itself and for any ABS client nodes connected to it. ABS server software makes the policy database available to all ABS clients. The MDMS software will be installed on the same OpenVMS node or OpenVMS Cluster system. The MDMS software provides media management services for itself and any MDMS client nodes connected to it. MDMS Software provides MDMS volume and magazine databases.

The volume database contains definitions of all removable media known to MDMS software, and the associated tape drive definitions, such as if the tape drive is local or remote to MDMS. It also contains information about volume locations and volume pool authorization. MDMS software updates all MDMS databases when any transactions are performed against the volumes.

Requirement:
You must install ABS/MDMS server software on at least one OpenVMS node or OpenVMS Cluster system.

Although the installation procedure does not have separate kits for the client and server software, ABS server is determined by the placement of ABS policy engine. During the installation procedure, you will be prompted to supply an OpenVMS node name (or node names in an OpenVMS

Cluster) where you want the ABS server to reside. This node (or nodes in a cluster) becomes the ABS server nodes. You cannot specify a cluster alias name in the node name list. This node also becomes the MDMS server node.

If your system configuration changes at some point after running the installation procedure, you can change the server node name(s) by modifying the ABS$SYS-TEM:ABS$POLICY_CONFIG.DAT file and by modifying the node and server information in the MDMS database.

2.3 Deciding Where To Install ABS Client Software

OpenVMS Clients

Install ABS/MDMS OpenVMS client software on any node that can communicate with ABS/MDMS server and for which you want to create ABS save requests. ABS OpenVMS client node can communicate with an ABS server node using the DECnet software. When an ABS client node needs to perform an ABS operation, ABS client node communicates with ABS server, retrieves policy information, updates ABS policy database, and then relinquishes its communications with ABS server when the operation has completed.

MDMS OpenVMS Client nodes can communicate with an MDMS server using DECnet or TCPIP.

_____Note_____

OpenVMS client support is not available with an ABS-OMT license.
_____

NT Clients

For NT clients you must install ABS NT client software provided in ABS software kit. You must also authorize access for NT clients on ABS server node. Instructions for these tasks are described in Chapter 3.

Tru64 UNIX Clients

UNIX refers to Tru64 UNIX. UNIX clients do not require a separate client installation. However, you must transfer the GNU tar files provided by ABS software to the UNIX client system, and then you must build the executable files on the UNIX client system. Also, you must authorize access for UNIX clients on ABS server node. Instructions for these tasks are described in Chapter 3.

Both NT and UNIX clients have their backup operations occur on ABS server node, and ABS server communicates with the NT or UNIX client system for data transfer and control.

_____Note_____

Given you have the appropriate amount of licenses and adequate resources available, any number of ABS client nodes may be connected to a single ABS server node.
_____

Figure 2–2 Typical Client-Server Configuration
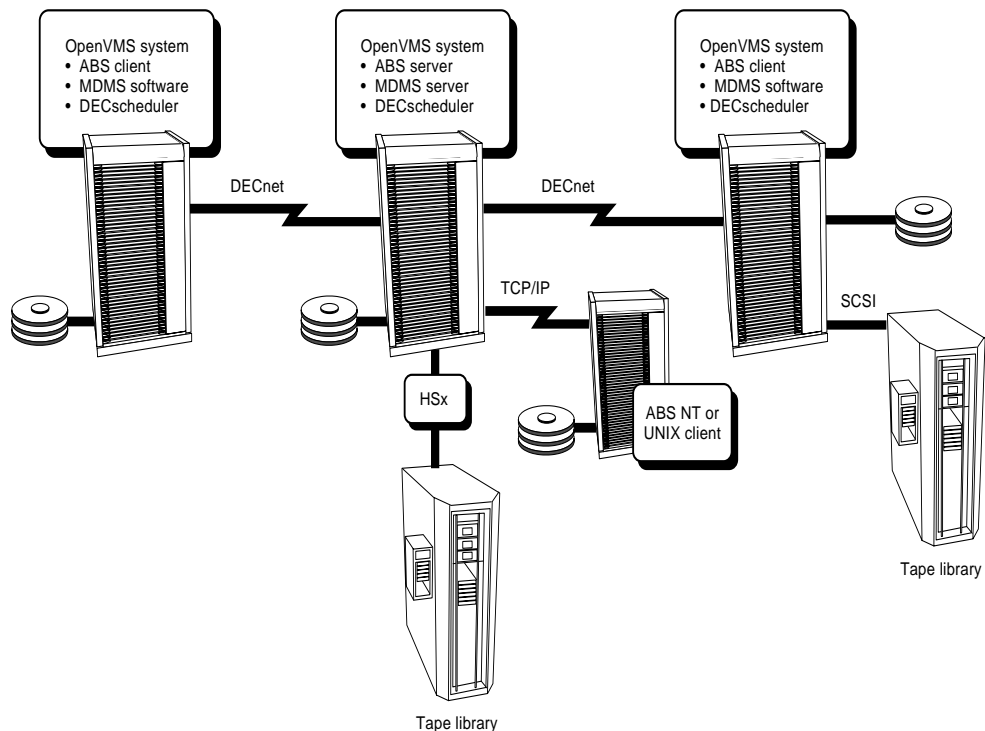
CXO6022B

2.4 Using the Installation Worksheets

To help you with the installation procedure the worksheet in Table 2–4 provides a work area to help you previously identify where you are installing ABS OpenVMS server, client software, and MDMS software.

_____Note_____

OpenVMS client support is not available with an ABS-OMT license.

Table 2–3 provides an example of how to use the worksheets provided in Table 2–4.

Table 2–3 Example Installation Worksheet

Server/Client	Node Name or OpenVMS Cluster Name	Disk Name	Support a Remote Tape Device?	Support a Remote Backup Operation?
ABS/MDMS Server Software	NODESV	DISK$USER1:	N/A	N/A
ABS OpenVMS Client and MDMS	NODECA	DISK$USER2:	Yes	No

ABS OpenVMS Client and MDMS	NODECB	DISK$USER3:	Yes	No

Table 2–4 provides the worksheet for your ABS/MDMS configuration. It is recommended that you make a copy of the worksheet and save the original for future use.

Table 2–4 Installation Worksheet

Server/Client	Node Name or OpenVMS Cluster Name	Disk Name	Support a Remote Tape Device?	Support a Remote Backup Operation?
ABS/MDMS Server Software				
ABS OpenVMS Client and MDMS				
ABS OpenVMS Client and MDMS				
ABS NT Client			N/A	N/A
			N/A	N/A
			N/A	N/A
			N/A	N/A
ABS UNIX Client			N/A	N/A
			N/A	N/A
			N/A	N/A
			N/A	N/A

2.5 Required Privileges

To install ABS, log on to the SYSTEM account or to an account that has SETPRV or, at a minimum, has the following privileges enabled:

- CMKRNL
- WORLD
- SYSPRV
- TMPMBX
- NETMBX

Note that VMSINSTAL turns off the BYPASS privilege at the start of the installation procedure.

2.6 Required OpenVMS Operating System Subclasses

OpenVMS operating system comes with a variety of support options, or subclasses. Subclasses include such features as networking and RMS journaling. To use ABS, your system should have the following subclasses resident:

- Programming support

- Utilities

- System programming environment

- Secure user's environment

- Network support

How to verify:
For information about verifying these components, refer to either *OpenVMS VAX Installation Procedures* or *OpenVMS Alpha Installation Procedures*.

2.7 Hardware, Software, and System Requirements

To make sure that your system is ready for the installation, verify that your system meets the following requirements:

- Hardware

- Software

- System parameters

- Process account quotas

- Processes

2.7.1 Required Hardware

To install software, you must meet the following minimum hardware requirements:

- A VAX™ system, a MicroVAX™ system, a VAXstation™, or an Alpha system.

- Please refer to the OpenVMS documentation for minimum requirements of RAM memory for running OpenVMS. We recommend that you have atleast 32 MB of memory on VAX system and 128 MB of memory on Alpha systems.

- One or more tape drives if you plan to back up your data to removable media. Refer to *Section* for instructions about configuring your hardware.

- One disk, such as a COMPAQ RD™ or RZ™ series disk.

- Adequate disk space. Refer to Table 2–5 to make sure there is adequate disk space to install ABS.

Verify that there are enough free blocks on the disk where you are installing the software.

─────────────**Note**─────────────

If you are providing remote drive support, you must answer *yes* to the remote drive question during MDMS installation procedure. This requires additional free disk space.

─────────────────────────────────

Enter the following command to show the amount of used disk space on your disk:

```
$ SHOW DEVICE disk-name
```

Disk space required during an full installation on an Alpha system is 214,500 blocks with 167,400 required after the installation. Disk space required on a VAX is 204,000 blocks with 120,000 required after the installation.

2.7.2 Required Software

Table 2–5 lists the software you must have installed on your system before you can install ABS.

Table 2–5 Required Software

Software	Purpose	System
DECnet Phase IV or DECnet Plus for Open-VMS	Provides network support **Note**: This software must be up and running *before* you start the installation procedure	ABS OpenVMS Server ABS OpenVMS Client
OpenVMS Operating System[a]	Provides OpenVMS and DCL capabilities.	ABS OpenVMS Server ABS OpenVMS Client

a. See *ABS Software Product Description (SPD)* for supported versions.

2.7.3 Patch Requirements

Table 2–6 describes the patch requirements for MDMS:

Table 2–6 Prerequisite Patches

Component	Operating System	Version	Patch
MDMS$SERVER	OpenVMS Alpha	V6.2	ALPY2K_062
	OpenVMS VAX	V6.2	VAXLIBR06_070
GUI	OpenVMS Alpha	V7.1 to V7.1-H2	ALPBASE02_071
		V7.1 to V7.1-H2	ALPACRT06_071
		V7.1 to V7.1-H2	ALPDCL01_071
		V7.1 to V7.1-H2	ALPSYSA01_071
		V7.1 to V7.1-H2	ALPSYSB02_071
		V7.1 to V7.1-H2	ALPTHREADS_03071
		V7.1-2 ONLY	VMS712_PTHREADS

If the server patches are not installed, you will see the following error while trying to start the server:

```
09-Mar-1999 10:38:16 %MDMS-I-TEXT, "10k Day" patch not installed!
```

─────────────────────**Note**─────────────────────

You can obtain these patches or the latest revision by contacting your Compaq representative. If the patches for the MDMS$SERVER are not installed, the server will not

> **start but you can successfully install MDMS, then install the patches and start the**
> **server.**

2.7.4 Install CMA Shareable Images

If you are installing MDMS on an OpenVMS V6.2 VAX system, you have to install the following three files:

• SYS$COMMON:[SYSLIB]CMA$RTL

• SYS$COMMON:[SYSLIB]CMA$OPEN_RTL

• SYS$COMMON:[SYSLIB]CMA$LIB_SHR

If these images are not installed by default, include the following lines in the SYS$STARTUP:SYSTARTUP_VMS.COM:

```
$!
$! Install CMA stuff for MDMS
$!
$ INSTALL = "$INSTALL/COMMAND_MODE"
$ IF .NOT. F$FILE_ATTRIBUTES("SYS$COMMON:[SYSLIB]CMA$RTL.EXE", "KNOWN")
$ THEN -
$   INSTALL ADD SYS$COMMON:[SYSLIB]CMA$RTL
$ ENDIF
$ IF .NOT. F$FILE_ATTRIBUTES("SYS$COMMON:[SYSLIB]CMA$OPEN_RTL.EXE", "KNOWN")
$ THEN
$   INSTALL ADD SYS$COMMON:[SYSLIB]CMA$OPEN_RTL
$ ENDIF
$ IF .NOT. F$FILE_ATTRIBUTES("SYS$COMMON:[SYSLIB]CMA$LIB_SHR.EXE", "KNOWN")
$ THEN
$   INSTALL ADD SYS$COMMON:[SYSLIB]CMA$LIB_SHR
$ ENDIF
```

2.7.5 Optional Software

Table 2–7 describes the optional software you can use with ABS software

─────────────────────**Note**─────────────────────

The following ABS features are not available with the ABS-OMT license:

• Support for OpenVMS clients
• Command line interface (CLI), you must use the GUI
• Support of STK silos and DCSC jukeboxes

Table 2–7 Optional Software

Software	Purpose	System
ABS NT Client Software	Provides the ability to save data from Windows NT systems to ABS OpenVMS server system using ABS software (provided with ABS, not purchased separately).	NT Client
C Compiler	Provides the ability to build the executable files on UNIX clients	UNIX Client
DCSC[a](Digital Cartridge Server Component)	If you have a StorageTek Automated Cartridge Server (ACS), you must install the DCSC software.	The system where the StorageTek silo resides.

TCP/IP Services for OpenVMS[a]	Provides network support for NT and UNIX clients.	ABS OpenVMS Server
eXcursion[a]	Provides the ability to display the graphical user interface (GUI) on NT client systems.	NT client system
Media Robot Utility (MRU)	Provides library and loader testing, diagnostics, and control.	The OpenVMS system where the robotic device is physically connected
Motif for X-Windows[a]	Provides graphical user interface capabilities	ABS OpenVMS server ABS OpenVMS client
Oracle Rdb[a] for VAX or Alpha systems	Provides distributed and multi streaming capabilities.	ABS OpenVMS Server

Requirements:

If you install Oracle Rdb for ABS policy database, the version of Oracle Rdb that you select must be up and running before you install. Also, the file SYS$LIBRARY:SQL$USER.OLB or SYS$LIBRARY:SQL$USER_ n.OLB (where *n* is version specific) must pre exist on your system.

If you have Oracle Rdb Version 6.0, you must apply engineering change order (ECO) number 5.

a. See *ABS Software Product Description (SPD)* for supported versions.

2.7.6 Required System Parameters

To install ABS, the system parameters must be set to the minimum value or higher. Table 2–8 lists the minimum system parameter values required for the installation procedure to run successfully. Depending on the kinds of programs and applications running at your site, you may need higher values.

Table 2–8 System Parameters – Minimum Values

System Parameter	Minimum Value	Dynamic
PQL_DENQLM	300	Y
GBLPAGES[a]	2000	N
GBLSECTIONS[a]	4	N
LOCKIDTBL	45000	N
PQL_MENQLM	300	Y
PQL_MPGFLQUOTA	10000	Y
PROCSECTCNT	100	N
PQL_MTQELM	200	Y

a. The values listed for these system parameters represent the number of free global pages and global sections required to install and run ABS, not the total number you need to operate your system and other software.

To see the current system parameter values on your system, enter the following command:

```
$ MCR SYSGEN
SYSGEN> SHOW/GEN
```

<u>**Result**</u>:

Shows the current values of all the system parameters. If you need to modify one or more of the system parameters, see the following example:

```
$ MCR SYSGEN
SYSGEN> SET GLBPAGES 2000
SYSGEN> WRITE CURRENT
SYSGEN> EXIT
```

The changed parameters should be added to SYS$SYSTEM:MODPARAMS.DAT for future changes made with AUTOGEN. You must then reboot the system so the non dynamic parameter values are recognized.

<u>**More information**</u>:

Refer to *OpenVMS System Manager's Manual: Managing System Parameters* for detailed information about required system parameters.

2.7.7 Required Process Account Quotas

The account you use to install ABS (typically the SYSTEM account) must have sufficient quotas to enable you to perform the installation. If your SYSTEM account quotas are the same as or higher than the default values provided with the OpenVMS operating system, then these values should be sufficient to install the software.

Table 2–9 summarizes the process quotas and the quotas that VMSINSTAL requires to perform the installation.

Table 2–9 Required Installing Account Process Quotas

Account Quota	Minimum Value
ASTLM	200
BIOLM	10000
BYTLM	18000
DIOLM	200
ENQLM	2048
FILLM	300
PGFLQUO	10000
TQELM	200

To see your current process quotas, see the following example:

```
$ MCR AUTHORIZE
UAF> SHOW SMITH
```

<u>**Result**</u>:

This command shows all your process quotas. If you need to increase your process account quotas, see the following example:

```
$ MCR SYS$SYTEM:AUTHORIZE
UAF> MODIFY SMITH/ENQLM=2048
UAF> EXIT
```

_____Note_____

If you are supporting NT or UNIX clients, to ensure successful save and restore operations, set the quotas to the following values from ABS OpenVMS server node:

UCX> SET PROTOCOL TCP/QUOTA=(SEND:50000,RECEIVE:50000)

If you have to reboot the machine, make sure that you reset these values after rebooting.

_____

More information:
For detailed instructions about modifying account quotas, see the description of *Authorize Utility in OpenVMS System Management Subkit.*

2.7.8 Required Processes

Before beginning the installation procedure, check to see that DECnet Phase IV or DECnet Plus and the OpenVMS Queue Manager are running. To see if these processes are active on your system, enter the following command:

```
$ SHOW SYSTEM
```

The following information is displayed for DECnet Phase IV:

```
OpenVMS V7.1 on node NODE1  8-AUG-1997 13:39:28.23Uptime 0 23:36:26
Pid          Process Name    State   Pri   I/O   CPU            Page flts Page
.
.
.
20A0022C   QUEUE_MANAGER   HIB    8     72    0 00:00:00.83    751       1210
.
.
.
20A00212   NETACP         HIB   10    285   0 00:00:02.84 338    666
```

The following information is displayed for DECnet Plus:

```
37C00215   NET$ACP        HIB    4    629   0 00:27:23.22   1894      2465
.
.
.
37C0024A   QUEUE_MANAGER   HIB    8    3333  0 00:07:45.24   1246      1766
```

If these processes are not active, follow the steps in Table 2–10.

Table 2–10 How to Start DECnet and the OpenVMS Queue Manager

Step	Action
1.	Start DECnet software. For DECnet Phase IV, enter the following command at DCL prompt: `$ @SYS$MANAGER:STARTNET.COM` For DECnet Plus, enter the following command at DCL prompt: `$ @SYS$STARTUP:NET$STARTUP.COM`
2.	Start OpenVMS Queue Manager. Enter the following command at DCL prompt: `$ START/QUEUE/MANAGER`

2.8 Overview of Hardware Installation

For your storage application to work, the hardware it depends on must be installed, connected, and configured to function with the operating system. This section provides a sequence of actions that you can use to verify that storage devices are connected and operating before you install MDMS and other application.

This procedure applies to the installation of a jukebox with drives or a standalone drive.

——————————————————————————**Note**——————————————————————————

Use an initialized volume to perform this test. To initialize a volume, consult Open VMS documentation or enter HELP INITIALIZE at an OpenVMS terminal.

——

2.8.1 Jukebox or Drive Hardware Installation

During this procedure, refer to the device specific hardware installation information for details. This procedure provides only the basic steps necessary to configure MDMS later against the jukebox and/or drives:

1. Following manufacturer's documentation, install and connect the hardware. Apply power to the drives.

2. If you are using an HSx controller, configure it to allow the OpenVMS system to communicate with the drives. Some controllers require you to configure pass through devices for drives and jukeboxes.

3. Make note of the device names after you complete connection procedures. If you are going to use Media Robot Utility software, define the logical MRU_ROBOT with the name of the jukebox device.

2.8.2 Test the Jukebox

If possible, use a utility such as Media Robot Utility (MRU) for this procedure. Otherwise, use the front panel of the drive and OpenVMS system software to test the connection between the OpenVMS system and jukebox.

Table 2–11 Testing the Jukebox Connection

Step	Action

1.	Verify the changer device name and the drive names. Enter the DCL SHOW DEVICE command and/or the ROBOT SHOW ROBOT command. If there is a problem with drive name or the connection, it becomes apparent here. Use the OpenVMS SHOW DEVICE command:
	
	`$ SHOW DEVICE [device_name[:]]`
	
	Use the following MRU command for examining the robot:
	
	`$ DEFINE MRU_ROBOT device_name`
	`$ ROBOT SHOW ROBOT`
	
	If either of these commands returns an error, make sure the device name is correct. If the device name is correct, then refer to the hardware documentation for remedial action.

2.	Place an initialized volume in the jukebox. • If your jukebox uses a magazine, place the volume in the magazine and then insert the magazine into the jukebox. • If your jukebox uses ports, place the volume into the port and then move the volume to an available slot in the jukebox. If you are using MRU, enter the following commands to accomplish this: `$ ROBOT INJECT PORT port_number SLOT slot_number`

2.8.3 Test the Drive

This test involves loading a volume and then mounting it on the drive. For jukeboxes with multiple drives, perform this procedure with each drive.

Following the successful completion of this test, you will be ready to install and test MDMS.

Table 2–12 Testing the Drive Connection

Step	Action
1.	Load the volume into the drive. If you are using MRU, enter the following command: `$ ROBOT LOAD SLOT slot_num DRIVE drive_num`
2.	Mount volume with the OpenVMS MOUNT/FOREIGN command: `$ MOUNT/FOREIGN/NOASSIST drive_name`

3.	Dismount the volume with the OpenVMS DISMOUNT command.
	
	• If you are using a stand alone drive or a tape library then enter the command with /UNLOAD qualifier:
	
	`$ DISMOUNT/UNLOAD drive_name`
	
	• If you are using a loader, include the /NOUNLOAD qualifier:
	
	`$ DISMOUNT/NOUNLOAD drive_name`

4.	If you are using a jukebox, unload the volume from the drive. Use the front panel controls, or if you are using MRU, enter the following command: `$ ROBOT UNLOAD DRIVE drive_num SLOT slot_num`
5.	If you are using a jukebox, remove the volume. Use the front panel controls, or if you are using MRU, enter the following command: `$ ROBOT EJECT PORT port_number`

2.9 Verify the Node is in the MDMS Database

If this installation is not the **initial** installation of MDMS, you need to verify that the node you are installing MDMS on is in the MDMS database. Enter the following command on a node that has MDMS already installed on it and verify that the node you are installing MDMS on is in the database:

```
$ MDMS SHOW NODE node_name_you_are_installing_on
%MDMS-E-NOSUCHOBJECT, specified object does not exist
```

If the node is not in the database, you receive the %MDMS-E-NOSUCHOBJECT error message and you should create the node using the following command:

```
$ MDMS CREATE NODE node_name_you_are_installing_on
```

See the *Archive Backup System for OpenVMS Command Reference Guide* for the qualifiers to use.

If the node you are adding is an MDMS server node, the installation procedure will create the node using the /DATABASE qualifier. In addition, you need to edit all SYS$STARTUP:MDMS$SYSTARTUP.COM files in your domain and add this node to the definition of MDMS$DATABASE_SERVERS.

2.10 Consider RDF Configuration

MDMS provides RDF software to facilitate operations that require access to remote, network connected tape drives. This allows you to copy data from a local site to a remote site, or copy data from a remote site to a local site.

RDF is not available if you are installing MDMS with the ABS-OMT license.

During the installation you will be asked questions on whether you want to install on this node, the software that will allow it to act as a server and/or client for the RDF software. You need to decide if you want the server and/or client installed on the node.

- Install the **RDF Server software** on all nodes that are connected to the tape drives used for remote operations.

- Install the **RDF Client software** on all nodes that initiate remote operations to tape drives on the RDF Server node.

——————————————————**Note**——————————————————

If you are installing this version of ABS over a version earlier than V3.0B, a new version of RDF (V4.3) is included with this kit, and you must reboot both the VAX and Alpha systems on which you install RDF before you can use it. If you are upgrading from ABS V3.0B or later, no reboot is necessary.

2.11 Registering ABS Licenses

To use ABS software, you must register and load the licenses before you begin the installation procedure. This information is supplied in the License PAK document which is packaged along with *Archive Backup System for OpenVMS Cover Letter.*

To register a license under OpenVMS, use the following procedure:

1. Log in to the system where you will be installing the software. Log in under SYSTEM account, or enable your account with the privileges described in Table 2–6.

2. Select one of the following methods to register the licenses:

    – At DCL prompt, enter the LICENSE REGISTER command with appropriate qualifiers that correspond to License PAK information. See Table 2–13.

    – Invoke SYS$UPDATE:VMSLICENSE.COM procedure. When it prompts you for information, respond with data from your License PAK. See Table 2–14.

If you plan to use ABS on more than one node in an OpenVMS Cluster, you must load the licenses on other nodes after you install ABS. See Table 2–13, Step 10 for instructions.

**Table 2–13 How to Register Your ABS Licenses Using the LICENSE REGIS-
TER Command**

Step	Action
Step 1.	Enter the LICENSE REGISTER command with the product name followed by a dash (-): **$** LICENSE REGISTER ABS-SERVER-VAX - ! Register this license on the ABS VAX server node **$** LICENSE REGISTER ABS-SERVER-ALPHA - ! Register this license on the ABS Alpha server node **$** LICENSE REGISTER ABS-CLIENT-VAX - ! Register this license on all ABS VAX client nodes **$** LICENSE REGISTER ABS-CLIENT-ALPHA - ! Register this license on all ABS VAX Alpha nodes **$** LICENSE REGISTER ABS-NT-CLIENT-USER - ! Register this license on the ABS server node where you plan to support NT clients. **$** LICENSE REGISTER PAB-UNIX-CLIENT-USER - ! Register this license on the ABS server node where you plan to support UNIX clients. **$** LICENSE REGISTER ABS-OMT - ! Register this license on all ABS VAX server nodes where you have to install the ABS software **$** LICENSE REGISTER ABS-OMT-UPG - ! Register this license on all ABS VAX server nodes where you want to upgrade the ABS-OMT license to the full ABS product **<u>Important</u>**: Enter a dash (-) at the end of each command from Steps 1 through 8.
Step 2.	Enter the /ISSUER qualifier information, assigning the value DEC between quotation marks. _$ /ISSUER="DEC" -
Step 3.	Enter the /AUTHORIZATION qualifier information, assigning it the value from the AUTHORIZATION NUMBER[a] entry of the PAK: _$ /AUTHORIZATION=xxxxxx -
Step 4.	Enter the /PRODUCER qualifier information, assigning the value DEC in quotes: _$ /PRODUCER="DEC" -
Step 5.	Enter the /UNITS qualifier information, assigning it the value from the UNITS[a] entry of the PAK _$ /UNITS=nn -
Step 6.	Enter the /DATE qualifier information, assigning the product's release date value from the PRODUCT RELEASE DATE[a] entry of the PAK: _$ /DATE=dd-mmm-yyyy -
Step 7.	Enter the /AVAILABILITY qualifier information, assigning the value from the AVAILABILITY TABLE CODE[a] entry of the PAK: _$ /AVAILABILITY=x -
Step 8.	Enter the /OPTIONS qualifier information, assigning the value from the KEY OPTIONS[a] entry of the PAK: _$ /OPTIONS=xxxxxx -

Step 9.	Enter the /CHECKSUM qualifier information, assigning the value from the CH[a] entry of the PAK:
	_$ /CHECKSUM=1-xxxx-xxxx-xxxx-xxxx
	Important:
	Do NOT end the entry with a dash.
Step 10.	Invoke the LICENSE LOAD command with the product name:
	$ LICENSE LOAD product_name

a. This information is supplied in the LICENSE PAK document (provided with the software kit).

Table 2–14 How to Register Your ABS Licenses Using VMSLICENSE.COM

Step	Action
Step 1.	From the system prompt, enter the following command:
	$ @SYS$UPDATE:VMSLICENSE.COM
Step 2.	Select Option 1. "REGISTER a Product Authorization Key"
Step 3.	Answer the questions according to the information supplied in the LICENSE PAK document (provided with the software kit).
	The following is only an example. Supply the information provided in the PAK to the prompts:
	Type ? at any prompt for a description of the information requested. Press Ctrl/Z at any prompt to return to the main menu.
	Issuer[DEC]
	Authorization Number[]:
	Authorization Number[]:ALS-NQ-1996JUN10-181
	Product Name[]:ABS-SERVER
	Producer[DEC]:
	Number of Units[]:1050
	Version[]:
	Product Release Date[]:
	Key Termination Date[]:
	Availability Table Code[]:H
	Activity Table Code[]:
	Key Options[]:MOD_UNITS,ALPHA
	Product Token[]:
	Hardware-Id[]:
	Checksum[]:2-PIBA-KIPP=BIGE-DDHC
Step 4.	Verify the information that you entered is correct. Enter **Yes**.
Step 5.	To exit the command procedure, select Option 99.

Table 2–14 describes how to register your license using the command procedure.

For complete information about using LMF, see *OpenVMS License Management Utility Manual.*

2.12 Steps to Convert to a New Scheduling Option

If you are converting from DECSCHEDULER (default for V2.2n) to INT_QUEUE_MANAGER or EXT_QUEUE_MANAGER, you need to review your save requests before installing ABS V3.n.

Before updating ABS to V3.n, do an ABS SHOW SAVE * command.

1. If a save has a start time in the past, you may want to set the save request to the desired start time, or to NEVER using:

   ```
   ABS SET SAVE request /START_TIME="time" or
   ABS SET SAVE request /START_TIME = NEVER or
   ABS SET SAVE request /NOSTART_TIME
   ```

2. Save requests which have an explicit time, NOW or NEVER will NOT be scheduled.

3. During ABS V3.n installation, if you choose to use the INT_QUEUE_MANAGER or EXT_QUEUE_MANAGER scheduling option, all of the ABS jobs in the POLYCENTER Scheduler are placed on hold. You may wish to delete the POLYCENTER Scheduler job using SCHEDULER DELETE request/USER=ABS.

After updating to ABS V3.n, ABS will begin scheduling the save requests at their new start time.

─────────────────────────── **Note** ───────────────────────────

If you did not set the start time of old requests to a time in the future, ABS will reset the start time of very old save requests to the next due time, based on the start time in the save request. It will not run the very old saves until the next due time calculated. If the start time was within one scheduling interval (for example, within a week for a weekly interval), then ABS will execute the save request when ABS is started after the upgrade, then it will reset the start time for the next due time. To make sure ABS executes your saves exactly when you want them to execute, modify the start time as mentioned in Step 1.

──

Information about the ABS scheduling activities is logged to the ABS$LOG:ABS$POLICY_<node_name>.LOG file. To receive additional information about scheduling in the log file, you may define a logical name:

```
$ DEFINE/SYSTEM ABS_SCHEDULER_LOGGING TRUE
```

An opcom message will automatically be sent to the TAPE operator if ABS fails to schedule a request.

3

Installing ABS Software

This chapter contains instructions for installing Archive Backup System for OpenVMS software.

Before proceeding with the installation procedure, make sure you have completed all of the following preinstallation tasks:

- Did you decide where to install ABS server and client software?

- Did you set your default directory to SYS$UPDATE?

- Did you log into an account with the proper quotas and privileges?

- Did you perform a system backup operation?

- If you are doing an upgrade installation, did you shutdown ABS and MDMS?

- Did you verify the hardware and disk space requirements?

- Did you verify the software requirements?

- Did you check to see if DECnet (Phase IV) or DECnet-Plus and the QueueManager are running?

- Did you register the appropriate licenses?

- Did you follow the steps required for converting to a new scheduling option?

3.1 Installing Archive Backup System for OpenVMS Software

Now that you have successfully installed and configured MDMS software, see Table 3–1 for the stages of installing and configuring ABS software.

---------------------------------**Note**---------------------------------

Before installing ABS in a real time, storage management environment, COMPAQ recommends that you first install and configure ABS in a test environment. If you are not satisfied with the test installation, delete ABS and reinstall it.

Table 3–1 Stages of Installing ABS Software

Stage	Action
1.	Install ABS server software as described in Section 3.1.1. **Note**: If you are installing ABS in a mixed architecture environment (VAX and Alpha systems resident in a single OpenVMS Cluster), follow the installation instructions in further section.
2.	Install ABS OpenVMS client software as described in Section 3.1.2. **Note**: OpenVMS client support is not available with an ABS-OMT license.

Table 3–1 Stages of Installing ABS Software

3.	Install ABS NT client software as described in Section 3.1.3.
4.	Install ABS UNIX clients as described in Section 3.1.4.
5.	Authorize NT and UNIX clients as described in Section 3.1.5.
6.	Follow the Post-Installation steps in Chapter 4.

3.1.1 Installing ABS/MDMS Server Software

ABS/MDMS installation procedure consists of a series of questions and informational messages.

If for any reason you need to abort the installation procedure, at any time you can press CTRL/Y and the installation procedure deletes all files it has created up to that point and then exits. From this point, you can restart the installation procedure again.

Follow the steps in Table 3–2 to install ABS software.

Table 3–2 How to Install ABS Software

Step	Action
1.	**Invoke VMSINSTAL**:
	`$ @SYS$UPDATE:VMSINSTAL saveset-name drive-name OPTIONS N`
	To start the installation, invoke the VMSINSTAL command procedure from a privileged account, such as the SYSTEM account. VMSINSTAL is in the SYS$UPDATE directory. The following list defines the elements of the VMSINSTAL command procedure:
	save set name
	The installation name for the component.
	`ABS031 (for example)`
	drive-name
	The name of the drive where the media that contains the kit is located. For example, MTA0: is the device name for a tape drive or device:[directory] can be the CD-ROM drive name. It is not necessary to use the console drive for this installation. However, if you do use the console drive, you should replace any media you removed once the installation is complete.
	OPTIONS N
	An optional parameter that indicates you want to see the question on release notes. If you do not include the OPTIONS N parameter, VMSINSTAL does not ask you about the release notes. You should review the release notes before proceeding with the installation in case they contain additional information about the installation.
	<u>**Note**</u>: If you are restarting the installation and have already reviewed the release notes, you do not need to specify OPTIONS N. If you specify more than one option, separate them with commas (OPTIONS A,N).
	The following examples invoke VMSINSTAL to install ABS from the tape drive MTA0: and shows the responses. This example uses the OPTIONS N release note parameter.

	`$ @SYS$UPDATE:VMSINSTAL saveset_name MTA0: OPTIONS N`
	
	`OpenVMS VAX Software Product Installation Procedure V6.1`
	`It is 21-JUL-1996 at 10:00`
	`Enter a question mark (?) at any time for help.`

	If you do not supply either the product name or the drive name, VMSINSTAL prompts you for this information later in the installation procedure.
	<u>**Note**</u>: VMSINSTAL does not prompt you for any options, so be sure to include OPTIONS N on the VMSINSTAL command line to access the release notes during the installation procedure. See OpenVMS documentation located in the OpenVMS System Management Subkit for detailed information on these options.

3.1.2 Installing ABS OpenVMS Client Software

—————————————— Note ——————————————

OpenVMS client support is not available with an ABS-OMT license.

If you select to use the INT_QUEUE_MANAGER or EXT_QUEUE_MANAGER

> **scheduling options, you must have ABS V3.0 or greater installed on all of your ABS servers and ABS OpenVMS clients. If you are using any other scheduling option, you may do a rolling upgrade of your clients. In those cases, ABS V2.2x may be running on the clients with ABS V3.0 running on the server.**

When installing ABS software, notice that ABS does not provide two separate software kits. Instead, installation of ABS OpenVMS server or client software is determined by the OpenVMS node name that you enter during the installation procedure.

Table 3–3 describes how to install and configure an ABS OpenVMS client.

———————————————— Note ————————————————

In Table 3–3, ABS server node is referred to as SVNODE::, and ABS client node is referred to as CLNODE::

—————————————————————————————————————

Table 3–3 Installing ABS OpenVMS Client Software

Step	Action
1.	Install ABS software on the OpenVMS client node as on the server node, with one exception. When the installation procedure prompts for the node name for the server (ABS OpenVMS server node), do not accept the default node name (client node name). Instead, enter OpenVMS Cluster alias or OpenVMS node name that you entered when you installed ABS OpenVMS server software: `    * cluster or node list for ABS policy engine [CLNODE::] : SVNODE` **Result**: ABS installs only the client portion of ABS software on the node named CLNODE::.

2.	After installing ABS software on the OpenVMS client node, create a proxy account for ABS OpenVMS server node on ABS OpenVMS client node.
	
	On ABS OpenVMS client node (CLNODE::), enter the following set of DCL commands:
	
	`    $ SET DEFAULT SYS$SYSTEM`
	`    $ RUN AUTHORIZE`
	
	`    UAF> ADD/PROXY SVNODE::ABS ABS/DEFAULT`
	`    UAF> EXIT`
	
	Result:
	These commands allow CLNODE:: to submit a save or restore request from the client node.

3.	Create save and restore requests for OpenVMS clients as described in *Archive Backup System for OpenVMS Guide to Operations*.
4.	Create (or modify) storage and environment policies. *Archive Backup System for OpenVMS Guide to Operations* describes how to create those policies.
5.	Create system and user backup operations using the correct storage and environment policies. *Archive Backup System for OpenVMS Guide to Operations*, provides instructions for these tasks.

3.1.3 Installing and Configuring ABS NT Clients

Requirements:

- You must have eXcursion software installed on the NT client system if you want to use the ABS graphical user interface (GUI) on the NT client system.

- If you use FTP to copy the setup files to the NT System, be sure to copy the files in binary mode.

Table 3–4 describes the stages of installing and configuring an ABS NT client node.

Table 3–4 Stages of Installing an ABS NT Client

Stage	Action
1.	Register ABS NT license on ABS OpenVMS server node. See Section 2.11 for instructions.
2.	Install ABS NT client software on NT client system as described in Table 3-5. **Note:** You must perform a separate installation on each NT client node that you want ABS to back up.
3.	Authorize the NT client systems that you plan to back up using ABS (described in Section 3.1.5 and Appendix A).
4.	Create save and restore requests for the NT client system by using the GUI on the NT client system, or by using the GUI or DCL on ABS OpenVMS server node. See *Archive Backup System for OpenVMS Guide to Operations* and *Archive Backup System for OpenVMS Command Reference Guide* for instructions about creating save and restore requests for NT clients.

To install and configure the NT client software, use the procedure in Table 3–5.

Table 3–5 Installing and Configuring an ABS NT Client

Step	Action

1.	Copy all files from one of the following directories located on ABS OpenVMS server node to the NT client system where you plan to install the NT client software, or to a location accessible by the NT client system.
	
	The NT client software is provided during ABS server installation procedure. ABS creates the appropriate directories and places the NT client software kits for Alpha and Intel systems in the following directories:
	
	`ABS$ROOT:[CLIENTS.NT.ALPHA]`
	`ABS$ROOT:[CLIENTS.NT.INTEL]`

2.	Run SETUP.EXE from this location to install ABS NT client software. **Result:** ABS NT client installation procedure prompts for information about the ABS server and the local host port number. Answer the prompts exactly as you answered them during ABS server installation procedure: • ABS server node–Enter the node on which ABS server software has been installed and will be providing ABS services for the NT client backup operations. This node also verifies connection requests. • Local host port number–The NT client system uses a TCP/IP port to communicate with ABS server to initiate save and restore requests. The default port number is 1800. If you decide to change the port number, it is limited to a range between 1024 and 65535. This port number is arbitrary, but it must match the port number you use when you authorize NT clients, described in Appendix A. To test to see if port number 1800 is already in use, enter the following command from ABS OpenVMS server system prompt: `$ TELNET node 1800` Where: *node* is the node name of the OpenVMS server node **Note**: Make sure that the number you provide does not conflict with a previously installed software application.
3.	Authorize the NT clients you plan to back up using ABS as described in Section 3.1.5 and Appendix A.

3.1.4 Configuring ABS UNIX Clients

To allow ABS to perform backup and restore operations for UNIX clients, you must configure access between the OpenVMS systems that run ABS server software and the UNIX client systems that contain the files to be backed up. The stages of installing and configuring a UNIX client is described in Table 3–6.

—————————————Note—————————————

There is no extra software kit provided for UNIX clients.

Table 3–6 Stages of Installing and Configuring an ABS UNIX Client

Stage	Action
1.	Register ABS UNIX license on ABS OpenVMS server node. See Section 2.11 for instructions.
2.	Modify the appropriate files on the UNIX client system as described in the Section 3.1.4.1.
3.	Transfer the gtar and gzip sources from ABS OpenVMS server node to each UNIX client system that you intend to back up using ABS. See Section 3.1.4.2.

4.	Build the executables on each UNIX client system that you plan to back up using ABS as described in Section 3.1.4.3.
5.	Authorize the UNIX clients as described in Section 3.1.5 and Appendix A.
6.	Create save and restore requests for the UNIX client from the OpenVMS server node. See *Archive Backup System for OpenVMS Guide to Operations* and *Archive Backup System for OpenVMS Command Reference Guide* for instructions about creating save and restore requests for UNIX clients.

3.1.4.1   Modifying the Appropriate UNIX Files

Table 3–7, lists the files that you need to modify on each UNIX client system that ABS is going to back up, and it describes the modifications to make for those specific files.

─────────────────────**Note**─────────────────────

UNIX is a case sensitive system. Be sure to enter the commands on the UNIX client system exactly as they are shown in Table 3–7 and Example 3–1.

──

Table 3–7 Modifying the Appropriate UNIX Files

File	Modification
/.rhosts [a]	Replace the ASCII readable internet address with ABS OpenVMS server nodes. The file format is: ``` # readable ip address account node01.vms.real.node ABS #ABS on node NODE01 ``` In this example, replace node01.vms.real.node with ABS OpenVMS server node names. The account name must stay the same (ABS), and it *must be specified in capital letters*. **Example of /.rhosts**: ``` node01.vms.real.node ABS node02.vms.real.node ABS ``` **Requirement**: You must always modify the /.rhosts file. If the file does not exist, then you must create it. Be sure that the /.rhosts file is located in the root directory and that is owned by the root account because ABS uses this directory during the backup operation.

/etc/hosts[a]	List the numeric internet address and the ASCII readable internet address of ABS OpenVMS server nodes. The file format is:
	
	`# Internet Address   Hostname  # Comments`
	`nn.nn.nn.nn   node01.vms.real.node  # example entry for hosts file`
	
	Where:
	
	• *nn.nn.nn.nn* is the numeric internet address of ABS node that executes save and restore requests.
	
	• *node01.vms.real.node* is the ASCII readable internet address of nn.nn.nn.nn. You must enter every node name on which ABS may execute.
	
	Example of /etc/hosts:
	
	`01.02.03.04 node01.vms.real.node  # node01 running ABS`
	`01.02.01.04 node02.vms.real.node  # node02 running ABS`
	
	Note:
	If ABS OpenVMS server nodes are already listed in the file /etc/hosts, you do not need to add them again.

a. File is not replaced, only modified.

3.1.4.2 Transferring the UNIX Backup Agent Sources

During the installation of ABS server software, the installation procedure creates a directory named ABS$ROOT:[CLIENTS.UNIX] on ABS server node. This directory contains the following two uncompressed sources that make up the UNIX backup agent:

• ABS$ROOT:[CLIENTS.UNIX]TAR-1_12.TAR

• ABS$ROOT:[CLIENTS.UNIX]GZIP-1_2_4.TAR

To configure a UNIX client, you must transfer the gtar and gzip sources to each UNIX client system that ABS is going to back up, build the executables, and place them in /usr/bin.

Refer to Example 3–1 to transfer the gtar and gzip sources from ABS server node to the UNIX client system.

Example 3–1 Transferring the Backup Agent Sources

```
u_node> ftp node01    # Connect to the ABS OpenVMS Server Node
Connected to node01.vms.dec.com.
220 node01 FTP Server (Version 3.3) Ready.
Name (node01:user1): user1
331 Username USER1 requires a Password.
Password:
230 User logged in.
Remote system type is VMS.

ftp> cd abs$root:[clients.unix] # Change to the directory that contains the
files
250 CWD command successful.
ftp> pwd
257 "ABS$ROOT:[CLIENTS.UNIX]" is current directory.
ftp> ls  # List the files in this directory
200 PORT command successful.
150 Opening data connection for  (16.82.16.75,1174)
gnu_general_public_license.txt;4
gnu_readme_where_to_get.txt;4
gzip-1_2_4.tar;4
tar-1_12.tar;4
226 NLST Directory transfer complete.
```

```
ftp> bin    # set the file transfer mode to binary
200 TYPE set to IMAGE.

ftp> get    # Get the sources
(remote-file) tar-1_12.tar
(local-file) tar-1_12.tar
200 PORT command successful.
150 Opening data connection for tar-1_11_8.tar (16.82.16.75,1178)
226 Transfer complete.
2662400 bytes received in 5.7 seconds (4.6e+02 Kbytes/s)

ftp> get    # Get the sources
(remote-file) gzip-1_2_4.tar
(local-file) gzip-1_2_4.tar
200 PORT command successful.
150 Opening data connection for gzip-1_2_4.tar (16.82.16.75,1494)
226 Transfer complete.
798720 bytes received in 1.8 seconds (4.3e+02 Kbytes/s)

ftp> quit
221 Goodbye
```

3.1.4.3 Building the UNIX Executables

After you have transferred the gtar and gzip sources, you are required to build the executables on the UNIX client system. With UNIX OS version upgrade, it is **mandatory** to rebuild ABSgtar and ABSgzip executables on the UNIX client.

The following sections describe how to build the tar and gzip executables.

3.1.4.3.1   Building the tar Executable –Use the following procedure to build the tar executable:

1. Use native tar to expand the tar file

2. Change directory to the tar directory

3. Enter the command ./configure --disable-nls

4. Enter the command make to build the tar image

5. Verify the tar image was created

6. Change directory to src

7. Verify that it is an executable image

8. Change to super user

9. Copy the executable from src/tar to usr/bin/ABSgtar

10. Change the protection on the image

11. Display the complete directory

12. Exit super user

13. Perform a cleanup operation

Example of tar

```
u_node> tar -xvf tar-1_12.tar  [1]
tar-1.12/README
tar-1.12/AUTHORS
        .

        .
```

```
                      .
        tar-1.12/po/sv.gmo


u_node> cd tar-1.12  2
u_node> configure --disable-nls  3
creating cache ./config.cache
checking host system type... alpha-dec-osf3.2


                .


                .


                .
creating config.h


u_node> make  4
for subdir in doc lib intl src scripts po; do \

echo making all in $subdir; \
(cd $subdir && make CC='gcc' CFLAGS='-g -O' LDFLAGS='' LIBS=''
prefix='/usr/local' exec_prefix='/usr/local'
bindir='/usr/local/bin' libexecdir='/usr/local/libexec'
infodir='/usr/local/info' infodir='/usr/local/info'
libexecdir='/usr/local/libexec' all) || exit 1; \

done
making all in doc


                .


                .


                .
make[1]: Leaving directory '/usr/users/user1/tar-1.11.8/po'


u_node> ls src 5
Makefile   checktar.sh  extract.o  list.c   open3.h  rmt.o    tar.h
                .


                .


                 .
buffer.c   diffarch.o gnu.c    names.c  rmt.c    tar  5


u_node> cd src   6
u-node> file tar  7
tar:  COFF format alpha dynamically linked, demand paged
      executable or object module not stripped
- version 3.11-8


u-node> su  8
Password:


# cp tar /usr/bin/ABSgtar  9
# chmod ugo+x /usr/bin/ABSgtar  10
# ls -l /usr/bin/ABSgtar  11
-rwxr-xr-x  1 root system 655794 Jan 24 11:07 ABSgtar
# exit  12
```

```
u-node> cd ..  13
u-node> rm -rf tar-1_12  13
%rm -f tar-1_12.tar  13
```

3.1.4.3.2   Building the gzip Executable – Use the following procedure to build the gzip executable:

1. Use the native tar to expand the gzip-1_2_4.tar image

2. Change directory to gzip-1.2.4

3. Enter the command ./configure

4. Enter the command make

5. Verify that the gzip file is there

6. Make sure it is an executable

7. Change to super user

8. Copy gzip file to /usr/bin/ABSgzip

9. Change the protection on the image

10. Exit super user

11. Perform a cleanup operation

Example of gzip

```
u_node> tar -xvf gzip-1_2_4.tar  1
gzip-1.2.4/README


        .

        .

        .
gzip-1.2.4/primos/include/sysTypes.h


u_node> cd gzip-1.2.4  2


u_node> ./configure  3
checking for gcc


        .

        .

        .
checking for gzip to derive installation directory prefix
chose installation directory prefix /usr/local
creating config.status
creating Makefile


u_node> make  4
gcc -c -DSTDC_HEADERS=1 -DHAVE_UNISTD_H=1 -DDIRENT=1 -O gzip.c


        .

        .
```

```
                    .
          gcc -O -o gzip gzip.o zip.o deflate.o trees.o bits.o unzip.o
          inflate.o util.o crypt.o lzw.o unlzw.o unpack.o unlzh.o getopt.o
          /usr/ucb/ld:
          Warning: Linking some objects which contain exception information sections
          and some which do not. This may cause fatal runtime exception handling prob-
          lems (last obj encountered without exceptions was crypt.o).

          rm -f gunzip zcat
          ln gzip gunzip
          ln gzip zcat
```

u_node> ls gzip [5]
gzip
u_node> file gzip [6]
gzip: COFF format alpha dynamically linked, demand paged executable or object
module not stripped
- version 3.11-8

u-node> su [7]
Password:
cp gzip /usr/bin/ABSgzip [8]
chmod ugo+x /usr/bin/ABSgzip [9]
ls -l /usr/bin/ABSgzip
-rwxr-xr-x 1 root system 654785 Jan 24 11:08 ABSgzip
ln -s /usr/bin/ABSgzip /usr/bin/gzip
exit [10]

u-node> cd .. [11]
u-node> rm -rf gzip-1.2.4 [11]
u-node> rm -f gzip-1_2_4.tar [11]

_____Note_____

**If you have problems transferring or building the tar or gzip files, see your UNIX sys-
tem manager.**

_____

3.1.5 Authorizing NT and UNIX Clients

After you have registered and loaded the NT or UNIX client license on ABS server node, run the
authorization executable file to authorize the NT or UNIX client node names that you intend to
back up using ABS. You must authorize access on the ABS OpenVMS server system for each
UNIX and NT system that you are going to back up using ABS.

Table 3–8 describes how to authorize NT and UNIX client nodes.

_____Note_____

**ABS NT and UNIX client licenses are sold in units according to the number of nodes
that you want to support: 1, 5, 10, 25, 50, or 100. ABS calculates the number of nodes
authorized versus the number of units the license allows. You cannot authorize more
NT or UNIX clients than the number of units allowed by the NT or UNIX client license
that you have purchased.**

_____

Table 3–8 Authorizing NT and UNIX Client Nodes

Step	Action
1.	Enter the following command from ABS server: `$ RUN ABS$SYSTEM:ABS$CLIENT_LICENSE.EXE`

2.	Add the node names of the UNIX or NT client nodes. When prompted, specify an NT or UNIX client:
	
	`Would you like to Add/Modify/Remove/Show the Client License?: ADD`
	
	`Enter Node Name: CLIENT_NODE_NAME`
	
	`Client Node Type (UNIX or NT) [UNIX]: NT`
	
	`Enter TCPIP Port Number [1800]:`
	
	Note:
	The port number is arbitrary, but it must match the port number you use when you install the NT clients.

3.	If the client node is an NT client, enter the port number. The default is 1800.
4.	Make sure that the logical named ABS$CLIENT_DB is defined as /SYSTEM/EXEC and that it translates to ABS$ROOT:[DATABASE]CLIENT_DB.DAT. This logical name should be defined during the startup procedure. You can verify that the logical name is defined correctly by entering the following command: `$ SHOW LOGICAL/FULL ABS$CLIENT_DB`

More Information:

For an example of adding, modifying, removing and showing NT or UNIX clients, see Appendix A.

4

Performing Postinstallation Tasks

Complete ABS/MDMS postinstallation tasks described in this chapter after you have success-
fully installed ABS/MDMS OpenVMS server or client software:

- Section 4.1 describes how to initialize the ABS database if this failed during the installation.

- Section 4.2 describes how to create a node object for MDMS.

- Section 4.3 describes how to verify the installation was successful.

- Section 4.4 explains how to edit the startup and shutdown files.

- Section 4.5 describes removing the SLS/MDMS V2.x Automatic Startup.

- Section 4.6 describes the requirements for an OpenVMS Cluster installation.

- Section 4.7 describes configuring MDMS.

- Section 4.8 describes configuring remote tape drives.

- Section 4.9 explains how to set up access right identifiers.

- Section 4.10 describes how to modify ABS default policy objects.

- Section 4.11 explains how to test ABS installation to perform save and restore requests.

- Section 4.12 explains how to set the quotas if you are supporting UNIX and NT clients.

- Section 4.13 describes the need of adding NT parameter, before issuing multivolume SAVE
  requests.

- Section 4.14 describes the procedure to allow ABS to access all files on NT systems.

- Section 4.15 describes the MDMS graphical user interface (GUI) installation.

- Section 4.16 describes how to run the graphical user interface (GUI).

———————————————————Note———————————————————

OpenVMS client support is not available with an ABS-OMT license.

———

4.1 Installing ABS for the First Time

If you are installing ABS as a new installation, database initialization programs may fail to run.
This results in IVP failure with errors showing the storage classes and execution environments.

To initialize the database with the default storage policies and execution policies, run the follow-
ing executable:

```
RUN ABS$SYSTEM:ABS$DB_INIT.EXE
```

4.2 Create a Node Object

If this is the initial installation of MDMS, you may need to create the node object in the MDMS node database for this node. Use the MDMS CREATE NODE command to create this initial database node. Refer to the command reference guide for the qualifiers for this command. The following is an example:

```
$ MDMS CREATE NODE NABORS -
! NABORS is the DECnet Phase IV node name or a
! name you make up if you do not use DECnet
! Phase IV in your network
/DATABASE_SERVER -
! a potential database node
! must also be defined in
! in SYS$STARTUP:MDMS$SYSTARTUP.COM
/TCPIP_FULLNAME=NABORS.SITE.INC.COM -
! the TCP/IP full node name if you
! are using TCP/IP you need this if
! you are using the GUI
/DECNET_FULLNAME=INC:.SITE.NABORS -
! this is the full DECnet Phase V node name
! do not define if you do not have DECnet Phase V on this node
! be sure to define if you have DECnet Phase V installed on this node
/TRANSPORT=(DECNET,TCPIP)
! describes the transports that listeners are
! started up on
```

4.3 Verifying ABS Installation

If you did not execute the IVP during the installation procedure, you can execute it immediately after installing ABS/MDMS software. Enter the following command at the DCL system prompt:

```
$ @SYS$TEST:ABS$IVP.COM
```

```
$ @SYS$TEST:MDMS$IVP.COM
```

_____**Note**_____

Support for installation IVP procedure is not available with an ABS-OMT license.
_____

If an error occurs during the IVP, the following message is displayed:

```
ABS Version 3.2 Installation Verification Procedure failed.

%VMSINSTAL-E-IVPFAIL, The IVP for ABS Version 3.2 has failed.
```

Errors can occur during the installation if any of the following conditions exist:

•   ABS is currently running

•   The operating system version is incorrect

•   A prerequisite software version is incorrect

•   Quotas necessary for successful installation are insufficient

•   System parameter values for successful installation are insufficient

•   The OpenVMS help library is currently in use

•   The product license has not been registered and loaded

For descriptions of the error messages generated by these conditions, see the OpenVMS documentation on system messages, recovery procedures, and OpenVMS software installation. If you

are notified that any of these conditions exist, you should take the appropriate action as described in the message.

4.4 Providing Automatic Start Up and Shut Down

You must edit the startup and shutdown files to provide automatic startup and shutdown of ABS software. To make sure that ABS automatically starts up and shuts down, follow these steps:

Step 1. Add the following command lines to the system startup file named SYS$MAN-AGER:SYSTARTUP_ VMS.COM:

```
$ @SYS$STARTUP:MDMS$STARTUP
$ @SYS$STARTUP:ABS$STARTUP
```

Step 2. Add the following line to the system shutdown file named SYS$MANAGER:SYSHUTDOWN.COM:

```
$ @SYS$MANAGER:ABSS$SHUTDOWN
$ @SYS$MANAGER:MDMS$SHUTDOWN
```

Step 3. When using MDMS with ABS, make sure that the MDMS startup is executed prior to the ABS startup. A logical name is defined by the MDMS startup which is needed by ABS.

4.5 Removing SLS/MDMS V2.x Automatic Startup

If you have been using SLS/MDMS V2.x before and all your nodes running ABS and/or HSM version now support the new MDMS make sure you remove this line from your system's start up file:

```
$ @SYS$STARTUP:SLS$STARTUP
```

If "SYS$MANAGER:TAPESYMBOL.COM" has been included in SYS$MANAGER:SYLO-GIN.COM or the User Login Command Procedure, ensure that you remove this line from the same.

If this node still needs to support clients that use SLS/MDMS V2.x refer to the Appendix **SLS/MDMS V2.x Compatibility** in the guide to operations. Until you have made all of the changes described in this appendix, you should not start up SLS/MDMS V2.x.

—————————————————————Note—————————————————————

You must have performed the conversion procedure as described before editing the TAPESTART.COM file.

——

4.6 Meeting OpenVMS Cluster Requirements

If you installed ABS server software on an OpenVMS Cluster system, perform the steps in Table 4–1 on each node in the OpenVMS Cluster (excluding the installing node) where you want to use ABS.

—————————————————————Note—————————————————————

The command line interface is not available with the ABS-OMT license, you must use the GUI.

——

Table 4–1 Updating the DCL Tables

Step	Action

1.	Run the common file ABS$STARTUP and MDMS$STARTUP from each node that you want to use ABS. This ensures ABS$/MDMS$ logical names are defined the same across all nodes in the OpenVMS Cluster:
	
	`$ @SYS$STARTUP:ABS$STARTUP.COM`
	`$ @SYS$STARTUP:MDMS$STARTUP.COM`

2.	Update the DCL table on each node in the OpenVMS Cluster (excluding the installing node). Enter the following command on each node: `$ INSTALL REPLACE SYSLIBRARY:DCLTABLES.EXE`
3.	Have all system users log off and log on again to enable them to use the DCL ABS/MDMS commands (unless performing an upgrade).

4.7 Configure MDMS

Now that you have installed MDMS you need to configure the MDMS database. MDMS provides a command procedure that you can use to configure the MDMS database in a new installation. The procedure is completely self-documenting with a help facility, and all key object attributes are defined. The procedure guides you through the following configuration objects:

- The domain (defining default values for other objects)

- Locations

- Nodes and groups

- Jukeboxes and associated drives

- Standalone drives and stackers

- Volumes, media types and pools

The procedure is activated using the following command:

```
$ @ MDMS$SYSTEM:MDMS$CONFIGURE.COM
```

Please refer to the *Archive Backup System for OpenVMS Guide to Operations* appendix "Sample Configuration of MDMS" for an example of using this procedure. If you are upgrading from SLS or ABS V2.9x, you should use the procedure MDMS$CONVERT_V2_TO_V3 instead. Refer to the *Archive Backup System for OpenVMS Guide to Operations* Appendix "Converting SLS.MDMS V2.x to V3".

Once MDMS is installed, and any conversions are performed, you may wish to adjust your configuration prior to performing MDMS operations

4.8 Configure Remote Tape Drives

If you installed the RDF software, you need to configure the remote tape drives. RDF is not available if you are installing MDMS with the ABS-OMT license.

For each tape drive served with RDF Server software, make sure there is a drive object record in the MDMS database that describes it. Refer to the chapters on MDMS configuration in the guide to operations and the MDMS CREATE DRIVE command in the command reference guide.

For each node connected to the tape drive, edit the file TTI_RDEV:CONFIG_node.DAT and make sure that all tape drives are represented in the file. The syntax for representing tape drives is given in the file.

During startup of MDMS, the RDF client and server are also started. The RDF images are linked on your system. If you see the following link errors on Alpha V6.2, this is not an RDF bug. The problem is caused by installed VMS patches ALPCOMPAT_062 and ALPCLUSIO01_062.

```
%LINK-I-DATMISMCH, creation date of 11-FEB-1997 15:16 in
shareable image SYS$COMMON:[SYSLIB]DISMNTSHR.EXE;3
differs from date of 4-MAY-1995 22:33 in shareable image library
SYS$COMMON:[SYSLIB]IMAGELIB.OLB;1
.
.
.
```

This is a known problem and is documented in TIMA. To correct the problem, issue the following DCL commands:

```
$ LIBRARY/REPLACE/SHARE SYS$LIBRARY:IMAGELIB.OLB SYS$SHARE:DISMNTSHR.EXE
$ LIBRARY/REPLACE/SHARE SYS$LIBRARY:IMAGELIB.OLB SYS$SHARE:INIT$SHR.EXE
$ LIBRARY/REPLACE/SHARE SYS$LIBRARY:IMAGELIB.OLB SYS$SHARE:MOUNTSHR.EXE
```

4.9 Granting the Appropriate ABS/MDMS Access Right Identifiers

When ABS Server installation procedure is complete, the user account that performed the installation (typically the SYSTEM account) is granted the following ABS access rights identifiers:

- ABS_CREATE_STORAGE_CLASS–Users who are granted this access right identifier can create a storage class (applicable only on ABS server system).

- ABS_CREATE_EXECUTION_ENV–Users who are granted this access right identifier can create an execution environment (applicable only on ABS server system).

- ABS_SHOW_ALL–Users who are granted this access right identifier can show all ABS policy objects (applicable only on ABS server system).

- ABS_LOOKUP_ALL–Users who are granted this access right identifier can look up all ABS saved data from any catalog (applicable on any ABS node).

- ABS_CREATE_REMOTE_JOBS–Users who are granted this access right identifier can submit a save or restore request that will be executed on a remote client node (applicable only on ABS server system).

  Requirement:
  To create NT or UNIX save requests, the requester (creating process) must have the ABS_CREATE_REMOTE_JOBS access rights identifier enabled.

- ABS_BACKUP_JOB–Users who are granted this access right identifier can submit a save or restore request that runs under their user name (a user backup). The user must also have read and write access control to the storage policy and environment policy intended for user backups.

- ABS_BYPASS–Users who are granted this access right identifier can perform any ABS function (applicable only on ABS server system). This includes creating, deleting, modifying, or showing any ABS policy objects.

Before any user can use MDMS, you must grant MDMS rights to those users. Refer to the MDMS Rights and Privileges Appendix for an explanation of MDMS rights and how to assign them.

4.9.1 Enabling an Access Rights Identifier

To grant an access rights identifier to a user's account, run the AUTHORIZE utility.

Example:

```
$ SET DEFAULT SYS$SYSTEM
$ RUN AUTHORIZE

UAF>GRANT/IDENTIFIER ABS_LOOKUP_ALL USER1

%UAF-I-GRANTMSG, identifier ABS_LOOKUP_All granted to USER1

UAF>EXIT
```

———————————————————Note———————————————————

Do not enable the access rights identifier by performing a SET RIGHTS_LIST at the DCL prompt.

———

4.9.2 Revoking An Access Rights Identifier

To remove an access right identifier, run AUTHORIZE utility and revoke the identifier from the user's account:

```
$ SET DEFAULT SYS$SYSTEM
$ RUN AUTHORIZE

UAF>REVOKE/IDENTIFIER ABS_BYPASS USER1
%UAF-I-REVOKEMSG, identifier ABS_BYPASS revoked from USER1

UAF>EXIT
```

4.10 Modifying The ABS Default Policy Objects

ABS provides a set of default policy objects so you can start using ABS for your backup operations immediately after installation.

 lists ABS default policy objects that are resident on your system when the installation procedure has successfully completed.

Table 4–2 Default ABS Policy Objects

Storage Policies	Environment Policies
ABS_ARCHIVE	ABS_ARCHIVE_ENV
DISASTER_RECOVERY	DISASTER_RECOVERY_ENV
SYSTEM_BACKUPS	SYSTEM_BACKUPS_ENV
USER_BACKUPS	USER_BACKUPS_ENV
UNIX_BACKUPS	UNIX_BACKUPS_ENV
	DEFAULT ENV

Table 4–3 lists the default policies that are available with an ABS-OMT license.

Table 4–3 Default ABS Policy Objects With An ABS-OMT License

Storage Policies	Environment Policies
OMT_BACKUPS	OMT_BACKUPS_ENV

4.10.1 Default Policy Object Attributes

Each of ABS default policy objects have the following attributes at the completion of the installation procedure:

- The owner of a default policy object is the user name of the account who installed the software (typically SYSTEM).

- Access to a default policy object is allowed only from ABS server node and by ABS account.

Upon completion of ABS installation procedure, ABS default objects enable you to create save and restore requests objects only from ABS server node. At this point in time, the default access controls are set to allow access only from ABS server system; you cannot create a save or restore request from an ABS client system. Attempting to run a save request from an ABS client system would cause an access violation to the default storage policies, with the exception of the USER_BACKUPS storage policy. The USER_BACKUPS storage policy default attributes allows users from ABS client systems to create save and restore requests.

─────────────────── **Note** ───────────────────

OpenVMS client support is not available with an ABS-OMT license.

──

To enable access to the other default ABS policy objects by ABS client systems, you must modify the default policy objects as described Section

4.10.2 Modifying Default Policy Objects

Because the installing account has been granted ABS_BYPASS access right identifier, only this account can access ABS default policy objects at this point in time. For maintenance purposes, modify the policy objects provided by ABS as shown in Table 4–4.

─────────────────── **Note** ───────────────────

The command line interface is not available with an ABS-OMT license, you must use the GUI.

OMT_BACKUPS storage policy and OMT_BACKUPS_ENV environment policy are the only policies available with the ABS-OMT license.

──

Table 4–4 Modifying ABS Provided Policy Objects

Step	Action
1.	Add the storage administrator as a user to the storage and environment policies provided by ABS, enable CONTROL access in order to modify these policy objects. **DCL Example**: ``` $ ABS SET STORAGE SYSTEM_BACKUPS - _$ /ACCESS=(USER_ID=SMITH,ACCESS="CONTROL") $ ABS SET ENVIRONMENT SYSTEM_BACKUPS_ENV - _$ /ACCESS=(USER_ID=SMITH,ACCESS="CONTROL") ```
2.	To enable access to ABS account from an ABS client system, see the following example: **DCL Example**: ``` $ ABS SET STORAGE SYSTEM_BACKUPS/ACCESS=(USER_ID=*::ABS, ACCESS="READ, WRITE") $ ABS SET ENVIRONMENT SYSTEM_BACKUPS_ENV/ACCESS=(USER_ID=*::ABS, ACCESS="READ, WRITE") ``` Where: *:: enables *any node* to access the storage and environment policy. ABS is the *account name* allowed to access the storage and environment policy. ``` $ ABS SET STORAGE SYSTEM_BACKUPS/ACCESS=(USER_ID=node::ABS, ACCESS="READ, WRITE") $ ABS SET ENVIRONMENT SYSTEM_BACKUPS_ENV/ACCESS=(USER_ID=node::ABS, ACCESS="READ, WRITE") ``` Where: node is the client node.

You can modify the other policy objects provided by ABS in the same manner. See *Archive Backup System for OpenVMS Guide to Operations* for instructions about adding users and enabling access controls.

Before using your storage policy, you may need to modify the MDMS related information in the policy. For example, you may wish to use a different media type than the default media type from your MDMS domain. When ABS is installed, the storage policies are initialized with the defaults from the domain. Issue an MDMS SHOW DOMAIN command to see the defaults. Make sure that your storage policy contains the desired settings before executing a save request.

4.11 Performing a Save, Lookup, and Restore Operation

To make sure that you can use ABS for your backup and restore operations, use the following procedure to test your installation:

—————————————————**Note**————————————————

The command line interface is not available with the ABS-OMT license, you must use the GUI to test the save, lookup, and restore.

——

1. Log into ABS server node under the same user account where you installed the software (typically SYSTEM).

2. Create a save request for the ABS server node and specify a file name in your directory. Enter the following command:

```
$ ABS SAVE filename/NAME=TEST_SAVE/STORAGE=SYSTEM_BACKUPS
```

3. Verify that the file you saved is recorded in the ABS catalog by entering ABS LOOKUP command:

    ```
    $ ABS LOOKUP filename
    ```

4. Restore the file back to its original location using ABS. Enter the following command:

    ```
    $ ABS RESTORE filename/NAME=TEST_RESTORE/STORAGE=SYSTEM_BACKUPS/-
    _$ CONFLICT_OPTIONS=NEW_VERSION
    ```

4.12 Verifying NT and UNIX Client Quotas

If you are supporting NT or UNIX clients, to ensure successful save and restore operations, set the quotas to the following values on ABS OpenVMS server node:

```
UCX> SET PROTOCOL TCP /QUOTA=(SEND:50000,RECEIVE:50000)
```

4.13 Adding NT Parameter

Issuing multivolume SAVE requests to NT client requires you to modify the Registry Path

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Sevices\Tcpip\Parameters
```

with the following NT parameter (20 or greater is recommended).

```
TcpMaxDataRetransmissions    REG_DWORD    20
```

This change to the default built in Windows NT parameter/subkey ensures that the TCP/IP connection is not prematurely terminated with send failure.

———————————————Note———————————————

After making the changes to the parameter, you need to reboot the system to allow the changes to take effect.

4.14 Allowing ABS Access to All Files on the NT Systems

ABS must have access to all the files you wish to backup on your NT system. There are two ways to do this:

*   Set access on the files for the SYSTEM account

    To set the file access,

    –   Select the file from a fileview window.

    –   Select Properties from the File pulldown.

    –   Click the Security tab

    –   Select Permissions.

    –   Select Add and highlight SYSTEM. Add the type of access (full control is best, so you can restore files).

    –   Click the Add button.

    This gives the SYSTEM account access to the files.

*   Set the Account that ABS uses for the Backups

    ABS, by default, uses the SYSTEM account to backup the files. If you wish to change the account used by ABS, you may do this by modifying the properties of the service:

    –   Click on Services in the Control Panel.

    –   Highlight the ABS service

    –   Click on Startup.

    –   In the Log On As section, select This Account.

    –   Enter the account name to be used and the password for that account.

    –   Click OK.

The account that you select should be a member of the Administrator group. The administrator group should be able to access all the files on your NT system, unless you set access denied for Administrator on a file.

4.15 MDMS Graphical User Interface (GUI) Installation

This section describes how to install and run the Graphical User Interface (GUI) on various platforms.
As the GUI is based on Java, you must have the Java virtual machine installed on the system you run the MDMS GUI on. If you do not have Java installed on your system, these sections describe what is needed and where to get it.

This installation procedure extracts files from the MDMS kit and places them in MDMS$ROOT:[GUI...]. You can then move the files to your Windows system and install them.

——————————————————————**Note**——————————————————————

For the GUI to communicate with the MDMS server, you must have TCP/IP services on the node where you have the MDMS server running. After installation be sure to refer to Section 4.16 to run the GUI.

——

4.15.1 Requirements

The GUI requires the following in order to run:

Virtual Machine – Since the MDMS GUI is a Java application, it requires the platform specific Java Virtual Machine. The availability of each Java Virtual Machine is described in the following sections. The best way of getting a Java Virtual Machine is to down load the platform-specific kit from the given URLs. If this is not possible, the MDMS package also contains a copy for your convenience. Issues concerning availability and installation of the Java Virtual Machine can be directed to:

```
http://www.sun.com/java/products for Windows NT and
http://www.compaq.com/java/download/jdk_ovms/1.1.8/index.html for OpenVMS
```

A Java Virtual Machine is included in this MDMS kit for the purpose of completeness. MDMS provides both the pointers (URLs) of downloading a Java Virtual Machine and the actual files of the Java Virtual Machine in the release package. However, the downloading approach is encouraged.
Memory - The hard drive space requirement is 6 MB for Java Virtual Machine and 2 MB for MDMS GUI. The main memory space requirement for running MDMS GUI is 10 MB.

4.15.2 Installation on OpenVMS Alpha V7.1-2 and V7.2-1

The following steps describe how to install and run the MDMS GUI on OpenVMS Alpha:

If you are installing the GUI on OpenVMS Alpha V7.1-2, then the following patches (or their latest equivalent) need to be installed on the system. Please contact your Compaq representative for obtaining these patches.

Table 4–5 Patches Required for OpenVMS V7.1 for JAVA

Patch Required	For OpenVMS Version	Fix Description
ALPBASE02_071	V7.1 to V7.1-H2	Fixes needed to enable ALPACRT06_071 and ALPSYSA01_071. **Must be installed first**.
ALPACRT06_071	V7.1 to V7.1-H2	DECC fixes-fork, exec
ALPDCL01_071	V7.1 to V7.1-H2	Fixes for multiple kernel threading problem
ALPSYSA01_071	V7.1 to V7.1-H2	Higher-priority thread blocking
ALPSYSB02_071	V7.1 to V7.1-H2	IEEE arithmetic
ALPTHREADS_03071	V7.1 to V7.1-H2	DECthreads; support for Java, selected fixes
VMS712_PTHREADS	V7.1-2 ONLY	DECthreads; support for Java, selected fixes

These patches are not required for installation on OpenVMS Alpha V7.2-1.

1. Extract the files for the OpenVMS Java Virtual Machine.

   You may use the Java kit provided with the MDMS kit or download files from the Web. If you want to install from the MDMS kit, answer YES to the following question:

   ```
   Do you want the OpenVMS Java kit extracted [NO]?
   ```

   If you install from the MDMS kit, a file called:

   ```
   MDMS$ROOT:[GUI.VMS]DEC-AXPVMS-JAVA-V0101-81-1.PCSI_DCX_AXPEXE
   ```

   is created. Use this file to install Java as in step 4.

2. In the MDMS installation, the following question is asked.

   ```
   Do you want the MDMS GUI installed on Alpha OpenVMS [YES]?
   ```

   Reply 'Yes' to the question if you want to install the GUI on OpenVMS. Files will be moved to MDMS$ROOT:[GUI.VMS] and the GUI installation will be completed.

3. Following the MDMS installation, you should install Java by first extracting the PCSI file for the Java installation using the following commands:

   ```
   $ SET DEFAULT MDMS$ROOT:[GUI.VMS]

   $ RUN DEC-AXPVMS-JAVA-V0101-81-1.PCSI_DCX_AXPEXE
   ```

   Extract and read the Release Notes for additional information on how to use this product in an OpenVMS environment:

   ```
   $ PRODUCT EXTRACT RELEASE_NOTES JAVA-
   /SOURCE=[directory_where_you_put_the_PCSI_file]-
   /FILE=[directory_where_you_want_it]JDK118_VMS_RELEASE_NOTES.HTML-
   /BASE_SYSTEM=AXPVMS
   ```

   Install the JDK1.1.8 from the .PCSI file obtained:

   ```
   $ PRODUCT INSTALL JAVA-
   /SOURCE=[directory_where_you_put_the_PCSI_file]/BASE_SYSTEM=AXPVMS
   ```

   The following files are installed by PCSI (POLYCENTER Software Installation utility) with file attribute of ARCHIVE:

```
SYS$MANAGER:JAVA$SETUP.COM
SYS$MANAGER:JAVA$STARTUP.COM
SYS$SYSROOT:[JAVA.LIB]FONT.PROPERTIES
SYS$SYSROOT:[JAVA.LIB]FONT_PROPERTIES.JA
```

If a file having any of these names already exists on the system, the installation process renames it to a new   name with the file type ending '*_OLD*', before loading the new copy from the kit. Only the latest version of the existing file is preserved (by being renamed to file.type_old) before PCSI deletes all remaining versions.

For example, an existing SYS$MANAGER:JAVA$SETUP.COM
is renamed to SYS$MANAGER:JAVA$SETUP.COM_OLD before the new copy is copied from the kit. If you have previously personalized any of these files, you might need to merge your personalizations with the new copy.

The JDK documentation is installed on your system at the following location:

```
SYS$COMMON:[SYSHLP.JAVA]INDEX.HTML
```

4. After installation you must do the following:

```
$ EDIT SYS$STARTUP:JAVA$SETUP.COM
```

and include the following logical name definition at the end of the file:

```
$ DEFINE JAVA$CLASSPATH -
MDMS$ROOT:[GUI.VMS]MDMS.ZIP,-
MDMS$ROOT:[GUI.VMS]SYMANTEC.ZIP, -
MDMS$ROOT:[GUI.VMS]SWINGALL.JAR, -
SYS$COMMON:[JAVA.LIB]JDK118_CLASSES.ZIP, []
```

5. Run JAVA$SETUP.COM to establish defaults for the logical names CLASSPATH and JAVA$FILENAME_CONTROLS, and to define symbols that determine whether Java will interpret commands as either foreign commands or DCL commands:

```
$ @SYS$MANAGER:JAVA$SETUP.COM
```

Add the above command line to SYS$COMMON:[SYSMGR]SYLOGIN.COM so that when users login, they will have the Java definitions.

─────────────────────────────────**Note**───────────────────────────────────

Make sure that the logical JAVA$USE_DCL is not defined or the GUI will not work.
───

6. The JAVA$SETUP.COM procedure calls:

```
SYS$SYSROOT:[SYSHLP.JAVA]JAVA$FILENAME_CONTROLS.COM
```

to establish the JAVA$FILENAME_CONTROLS default values. You can edit this file to see what defaults are being used and how to change them. (This information is also in the "UNIX Style Filenames on an OpenVMS System" section of the JDK release notes.)

7. Rename the file SYS$COMMON:[JAVA.LIB]FONT.PROPERTIES to another name. This file remaps the fonts and makes the MDMS GUI appear incorrect. Renaming the file to another name will cause Java to use the default fonts, which are necessary to run the MDMS GUI.

8. If you are running the GUI using a non-PC keyboard, for example an OpenVMS keyboard, issue the following command on the system running the GUI:

```
$ MCR decw$utils:xmodmap -e "keysym Delete = BackSpace Delete"
```

9. Note that this command affects all DECwindows sessions. In order to get the proper opera-
    tion on both the GUI session and other DECwindows sessions, the following sequence
    should be employed:

    ```
    $ MCR decw$utils:xmodmap -e "keysym Delete = BackSpace Delete"
    $ MDMS/INTERFACE=GUI
    $ MCR decw$utils:xmodmap -e "keysym Delete = Delete"
    ```

    This resets the behavior of all the other windows to normal while the Java GUI still retains
    the earlier definition. This way, both the GUI and the other DECwindows can co-exist, both
    retaining the required functionality.

10. The Java kit and MDMS GUI are provided in zipped files. The Java Virtual Machine is
    capable of reading zipped files directly. **Do not unzip** any of the zipped files provided with
    the GUI.

4.15.3 Installation on Intel Windows NT/95/98

The following describes how to install the MDMS GUI on Intel platforms running Windows NT/
95/98:

1. In the MDMS installation, the following question is asked.

    ```
    Do you want files extracted for Microsoft Windows NT/95/98 on Intel [YES]?
    ```

    Reply YES if you want to install the GUI on Intel Windows NT/95/98.

2. Install the Java Virtual Machine - If Java Virtual Machine is not already installed on your
    PC, down load JRE 1.1.8 from:

    ```
    http://www.javasoft.com/products/jdk/1.1/jre or
    http://www.sun.com/developers/developers.html
    ```

    and follow the instructions to perform a default installation. You may use other versions of
    JRE, preferably 1.1.8 or later. If a Java Virtual Machine is not available, you may use
    MDMS$ROOT:[GUI.INTEL]JRE117WINTEL.EXE.
    Simply double-click on this file to install Java, and follow the setup instructions.

3. Install the MDMS GUI:

    Make MDMS$ROOT:[GUI.INTEL]SETUP_INTEL.EXE available to the target machine
    (Intel PC running Windows NT/95/98)

    Run SETUP_INTEL.EXE on the target machine.

4.15.4 Installation on Alpha Windows NT

The following describes how to install the MDMS GUI on an Alpha platform running Windows
NT:

1. In the MDMS installation, the following question is asked. Reply YES if you want to install
    the GUI on Alpha NT.

    ```
    Do you want the MDMS GUI files extracted for Alpha NT [YES] ?
    ```

2. Install the Java Virtual Machine - If Java Virtual Machine is not already installed on your
    Alpha, down load JRE 1.1.8 from:

    ```
    http://www.compaq.com/java/download/jre_nt/1.1.8/jre118_down.html
    ```

    and follow the instruction to perform a default installation. If a Java Virtual Machine is not
    available, you may use:

```
MDMS$ROOT:[GUI.ALPHA_NT]JRE118ALPHANT.EXE.
```

3. Install FX!32 if not installed and make sure FX!32 is enabled.

4. Install the MDMS GUI:

   – Make MDMS$ROOT:[GUI.ALPHA_NT]SETUP_ALPHA_NT.EXE available to the target machine (Alpha PC running Windows NT)

   – Run SETUP_ALPHA_NT.EXE on the target machine.

   – If the 'Unzip To' folder has been modified to anything other than default directory, remember to modify the MDMS_GUI.BAT.

4.16 Running the GUI

Now that you have installed the GUI, you have to make sure the server node is configured to accept communications from the GUI. The server node for the GUI must have:

• TCP/IP enabled
  and

• the MDMS rights enabled in the SYSUAF record for the user

To enable TCP/IP communications on the server, you have to set the *TCP/IP Fullname* attribute and enable the TCPIP transport. See the command reference guide for information about setting these attributes in a node.

MDMS rights for the user must be enabled in the SYSUAF record to log into the server using the GUI. Refer to the command reference guide for information about MDMS rights.

The following sections describe how to run the GUI on various platforms.

4.16.1 Running the GUI on OpenVMS Alpha

To use the MDMS GUI on OpenVMS Alpha systems, use the following commands:

```
$ @SYS$STARTUP:JAVA$SETUP.COM
$ SET DISPLAY/CREATE/NODE=node_name/TRANSPORT=transport
$ MDMS/INTERFACE=GUI
```

For the SET DISPLAY command, the node_name is the name of the node on which the monitor screen exists. This allows you to use the GUI on systems other than those running OpenVMS Alpha V7.1-2 or higher. The transport must be a keyword of:

• LOCAL – if you are running the GUI on the same node as the monitor

• DECNET – if you are running the GUI on a monitor connected to another node and you wish to use DECnet protocol between the monitor node and the GUI Java node.

• TCPIP – if you are running the GUI on a monitor connected to another node and you wish to use TCPIP protocol between the monitor node and the GUI Java node.

4.16.2 Running the GUI on Intel Windows NT/95/98

To use the MDMS GUI on Intel Windows NT/95/98 platforms, double click MDMS_GUI\MDMS_GUI.BAT.

4.16.3 Running the GUI on Alpha Windows NT

To use the MDMS GUI on Alpha Windows NT, do one the following:

1. If both the Java Virtual Machine and the MDMS GUI were installed with default selection, then double click MDMS_GUI\MDMS_GUI.BAT

2. Otherwise in MDMS_GUI\MDMS_GUI.bat, replace the Java Virtual Machine directory and MDMS_GUI directory as necessary, double click MDMS_GUI\MDMS_GUI.BAT.

A

Examples of Authorizing NT and UNIX Clients

This appendix contains examples of how to authorize NT and UNIX clients. This includes adding, modifying, showing, and removing NT and UNIX client licenses.

To use the license command as shown in the example in this appendix, you can define the following symbol at the system prompt:

```
$ LICENSE := $ABS$SYSTEM:ABS$CLIENT_LICENSE.EXE
```

All of the examples in the following sections use this symbol definition.

A.1 Adding Client Licenses

Example A–1 shows how to add a UNIX or NT client license

Example A–1 Adding Client Licenses

```
$ LICENSE


Would you like to Add/Modify/Remove/Show the Client License?: ADD
Enter Node Name: NTNODE
Client Node Type (UNIX or NT) [UNIX]: NT
Enter TCPIP Port Number [1800]: 1800
Client NTNODE successfully ADDED to License Database
License Count: 1 used of 6 total


$ LICENSE


Would you like to Add/Modify/Remove/Show the Client License?: ADD
Enter Node Name: UNIX_1
Client Node Type (UNIX or NT) [UNIX]: UNIX
Client UNIX_1 successfully ADDED to License Database
License Count: 1 used of 25 total
```

A.2 Modifying Client Licenses

Example A–2 shows how to modify the port number of an NT or UNIX client   license.

─────────────────────────── **Note** ───────────────────────────

The port number that you enter here must match the port number you assigned during the NT client installation.

──

Example A–2 Modifying a Client License

```
$ LICENSE


Would you like to Add/Modify/Remove/Show the Client License?: MODIFY
Enter Node Name: NTNODE
```

```
Client Node Type (UNIX or NT) [UNIX]: NT
Current values
Type: Windows NT
Transport:TCP/IP
Port:1800

New Port #?: 1800
Client NTNODE successfully MODIFIED in License Database

License Count: 1 used of 6 total


$ LICENSE


Would you like to Add/Modify/Remove/Show the Client License?: MODIFY
Enter Node Name: UNIX_1
Client Node Type (UNIX or NT) [UNIX]: UNIX
Current values
Type: UNIX
Transport: TCP/IP
Port: 514

New Port #?: 1800
Client UNIX_1 successfully MODIFIED in License Database

License Count: 1 used of 25 total
```

A.3 Showing Client Licenses

Example A–3 illustrates how to show an NT or UNIX client license.

Example A–3 Showing Client Licenses

```
$ LICENSE


Enter Node Name or [ALL]: ALL
Client Node Type (UNIX or NT) [UNIX]:  NT


Node Name           Type           Transport     Port
---------           ----           ---------     ----
NTNODE              Windows NT     TCP/IP        1800


License Count: 1 used of 6 total

$ LICENSE


Enter Node Name or [ALL]: ALL
Client Node Type (UNIX or NT) [UNIX]:  UNIX


Node Name           Type           Transport     Port
---------           ----           ---------     ----
UNIX_1              UNIX            TCP/IP        1800


License Count: 1 used of 25 total
```

A.4 Removing Client Licenses

Example A–4 shows how to remove NT or UNIX client licenses:

Example A–4 Removing Client Licenses

```
$ LICENSE


Would you like to Add/Modify/Remove/Show the Client License?: REMOVE
Enter Node Name: NTNODE
Client Node Type (UNIX or NT) [UNIX]: NT
```

```
Client NTNODE successfully REMOVED from License Database
License Count: 0 used of 6 total
```

B

File and Logical Names

The installation procedure produces several files on your system and defines numerous logical names. The following sections are provided so you can verify that the appropriate file names and logical names are resident on your system when the installation procedure is complete.

Section B.1 lists the names of the files installed. Section B.2 lists the logical names that are added to the system logical name table.

B.1 ABS File Names

Table B–1 lists the names of ABS files created on your system after ABS is successfully installed.

Table B–1 ABS Installed Files

File Name	File Name
SYS$HELP	
ABS032.RELEASE_NOTES	ABS$HELP.HLB
SYS$LIBRARY	
ABS$COSISHR.EXE	ABS$SHR.EXE
ABS$SLS_SERVICES.EXE	ABS$USSSHR_62.EXE
ABS$USSSHR_71.EXE	ABS$USSSHR_72.EXE
ABS$USSSHR_73.EXE	
SYS$MANAGER	
ABS$SHUTDOWN.COM	ABS$SYSTARTUP.COM
SYS$STARTUP	
ABS$STARTUP.COM	
SYS$SYSTEM	
ABS$DCL.EXE	
SYS$TEST	
ABS$IVP.COM	
ABS$ROOT[000000]	
LOGIN.COM	
ABS$ROOT[CATALOG]	

Table B–1 ABS Installed Files

File Name	File Name
ABS_CATALOG_BAOE.DAT	ABS_CATALOG_BAOE_INSNC.DAT
ABS_CATALOG_BTLE.DAT	ABS$CATALOG_OBJECTS.DAT
DISASTER_RECOVERY_BAOE.DAT	DISASTER_RECOVERY_BTLE.DAT
DISASTER_RECOVERY_BAOE_INSNC.DAT	

ABS$ROOT[CLIENTS.UNIX]	
GNU_GENERAL_PUBLIC_LICENSE.TXT	GNU_README_WHERE_TO_GET.TXT
GZIP-1-2-4.TAR	TAR-1_12.TAR

ABS$ROOT[CLIENTS.NT.ALPHA]	
DATA.Z;1	SETUP.EXE;1
SETUP.INS;1	SETUP.PKG;1
INST32A.EX;1	_ISDEL.EXE;1
_SETUP.DLL;1	_SETUP.LIB;1

ABS$ROOT[CLIENTS.NT.INTEL]	
DATA.Z;1	SETUP.EXE;1
SETUP.INS;1	SETUP.PKG;1
INST32I.EX;1	_ISDEL.EXE;1
_SETUP.DLL;1	_SETUP.LIB;1

ABS$ROOT[DATABASE]	
CLIENT_DB.DAT	EPCOT.DB1
EPCOT.DB2	EPCOT.DB3

ABS$TEMPLATES	

Table B–1 ABS Installed Files

File Name	File Name
GTAR-1.AGENT_INFORMATION;1	RMU_BACKUP_60-2.FULL_RESTORE_TEMPLATE;1
GTAR-1.FULL_ARCHIVE_TEMPLATE;1	RMU_BACKUP_60-2.INCREMENTAL_ARCHIVE_TEMPLATE;1
GTAR-1.FULL_RESTORE_TEMPLATE;1	RMU_BACKUP_60-2.INCREMENTAL_RESTORE_TEMPLATE;1
GTAR-1.INCREMENTAL_ARCHIVE_TEMPLATE;1	RMU_BACKUP_60-2.PARSE_TEMPLATE;1
GTAR-1.INCREMENTAL_RESTORE_TEMPLATE;1	RMU_BACKUP_60-2.SELECTIVE_ARCHIVE_TEMPLATE;1
GTAR-1.PARSE_TEMPLATE;1	RMU_BACKUP_60-2.SELECTIVE_RESTORE_TEMPLATE;1
GTAR-1.SELECTIVE_ARCHIVE_TEMPLATE;1	RMU_BACKUP_61-2.AGENT_INFORMATION;1
GTAR-1.SELECTIVE_RESTORE_TEMPLATE;1	RMU_BACKUP_61-2.FULL_ARCHIVE_TEMPLATE;1
NT_GTAR-1.AGENT_INFORMATION;1	RMU_BACKUP_61-2.FULL_RESTORE_TEMPLATE;1
NT_GTAR-1.FULL_ARCHIVE_TEMPLATE;1	RMU_BACKUP_61-2.INCREMENTAL_ARCHIVE_TEMPLATE;1
NT_GTAR-1.FULL_RESTORE_TEMPLATE;1	RMU_BACKUP_61-2.INCREMENTAL_RESTORE_TEMPLATE;1
NT_GTAR-1.INCREMENTAL_ARCHIVE_TEMPLATE;1	RMU_BACKUP_61-2.PARSE_TEMPLATE;1
NT_GTAR-1.INCREMENTAL_RESTORE_TEMPLATE;1	RMU_BACKUP_61-2.SELECTIVE_ARCHIVE_TEMPLATE;1
NT_GTAR-1.PARSE_TEMPLATE;1	RMU_BACKUP_61-2.SELECTIVE_RESTORE_TEMPLATE;1
NT_GTAR-1.SELECTIVE_ARCHIVE_TEMPLATE;1	RMU_BACKUP_70-1.AGENT_INFORMATION;1
NT_GTAR-1.SELECTIVE_RESTORE_TEMPLATE;1	RMU_BACKUP_70-1.FULL_ARCHIVE_TEMPLATE;1
RMU_BACKUP_42-2.AGENT_INFORMATION;1	RMU_BACKUP_70-1.FULL_RESTORE_TEMPLATE;1
RMU_BACKUP_42-2.FULL_ARCHIVE_TEMPLATE;1	RMU_BACKUP_70-1.INCREMENTAL_ARCHIVE_TEMPLATE;1
RMU_BACKUP_42-2.FULL_RESTORE_TEMPLATE;1	RMU_BACKUP_70-1.INCREMENTAL_RESTORE_TEMPLATE;1
RMU_BACKUP_42-2.INCREMENTAL_ARCHIVE_TEMPLATE;1	RMU_BACKUP_70-1.PARSE_TEMPLATE;1
RMU_BACKUP_42-2.INCREMENTAL_RESTORE_TEMPLATE;1	RMU_BACKUP_70-1.SELECTIVE_ARCHIVE_TEMPLATE;1
RMU_BACKUP_42-2.PARSE_TEMPLATE;1	RMU_BACKUP_70-1.SELECTIVE_RESTORE_TEMPLATE;1
RMU_BACKUP_42-2.SELECTIVE_ARCHIVE_TEMPLATE;1	VMS_BACKUP-2.AGENT_INFORMATION;1
RMU_BACKUP_42-2.SELECTIVE_RESTORE_TEMPLATE;1	VMS_BACKUP-2.FULL_ARCHIVE_TEMPLATE;1
RMU_BACKUP_51-2.AGENT_INFORMATION;1	VMS_BACKUP-2.FULL_RESTORE_TEMPLATE;1
RMU_BACKUP_51-2.FULL_ARCHIVE_TEMPLATE;1	VMS_BACKUP-2.INCREMENTAL_ARCHIVE_TEMPLATE;1
RMU_BACKUP_51-2.FULL_RESTORE_TEMPLATE;1	VMS_BACKUP-2.INCREMENTAL_RESTORE_TEMPLATE;1
RMU_BACKUP_51-2.INCREMENTAL_ARCHIVE_TEMPLATE;1	VMS_BACKUP-2.PARSE_TEMPLATE;1
RMU_BACKUP_51-2.INCREMENTAL_RESTORE_TEMPLATE;1	VMS_BACKUP-2.SELECTIVE_ARCHIVE_TEMPLATE;1
RMU_BACKUP_51-2.PARSE_TEMPLATE;1	VMS_BACKUP-2.SELECTIVE_RESTORE_TEMPLATE;1
RMU_BACKUP_51-2.SELECTIVE_ARCHIVE_TEMPLATE;1	VMS_SAVESET-1.AGENT_INFORMATION;1
RMU_BACKUP_51-2.SELECTIVE_RESTORE_TEMPLATE;1	VMS_SAVESET-1.PARSE_TEMPLATE;1
RMU_BACKUP_60-2.AGENT_INFORMATION;1	VMS_SAVESET-1.SELECTIVE_ARCHIVE_TEMPLATE;1
RMU_BACKUP_60-2.FULL_ARCHIVE_TEMPLATE;1	

Table B–1 ABS Installed Files

File Name	File Name
ABS$SYSTEM	
ABS$BACKUP.CLD;1	ABS$BACKUP.EXE;1
ABS$CONVERT_CATALOG.COM;1	ABS$CLEAN_DB_UTIL.EXE;1
ABS$CONVERT_TO_RMS.COM;1	ABS$CONVERT_TO_DB1.EXE;1
ABS$COORD_CLEANUP.FDL;1	ABS$CONVERT_TO_DB2.EXE;1
ABS$EXT_QUEUE_MANAGER.COM;1	ABS$CONVERT_TO_DB3.EXE;1
ABS$EXT_QUEUE_MANAGER.TEM-PLATE;1	ABS$COORD_CLEANUP.EXE;1
	ABS$POLICY_ENGINE.EXE;1
ABS$EXT_SCHEDULER.COM;1	ABS$POLICY_ENGINE_190_ORG.EXE;1
ABS$EXT_SCHEDULER.TEMPLATE;1	ABS$POLICY_MAINT.EXE;1
ABS$POLICY_CONFIG.DAT;1	ABS$UBS.EXE;1
ABS$START_CATALOG_CLEANUP.COM;1	ABS$CATALOG_OBJECT.EXE;1
ABS$START_DB_CLEANUP.COM;1	ABS$CATALOG_UPGRADE.EXE;1
ABS$START_POLICY_ENGINE.COM;1	ABS$CATALOG_CLEANUP.EXE;1
ABS$CATALOG_OBJECTS.FDL;1	ABS$CLIENT_LICENSE.EXE;1
ALPHA.DIR;1	ABS$CATALOG_UNPACK_STG.EXE;1
AOE_BRIEF.FDL;1	ABS_UI.EXE;1
AOE_INSNC_BRIEF.FDL;1	ABS_UI_EXECUTE.UID;1
CONVERT_CATALOG_V30.COM;1	ABS_UI_LOOKUP.UID;1
COORDINATOR.COM;1	ABS_UI_MAIN.UID;1
ABS$COORDINATOR.COM	ABS_UI_RESTORE.UID;1
COORD_CLEANUP.DAT;1	ABS_UI_SAVE.UID;1
DB_EPCOT_DB1.FDL;1	ABS_UI_STORAGE.UID;1
DB_EPCOT_DB2.FDL;1	ABS$COORDINATOR.EXE;1
DB_EPCOT_DB3.FDL;1	ABS$DB_INIT.EXE;1
DELETE_ABS.COM;1	
EXECCMDLINE.DAT;1	
RESTORECMDLINE.DAT;1	
SAVECMDLINE.DAT;1	
SLSTOAABS031.A;1	
SLSTOVABS031.A;1	
START_COORD_CLEANUP.COM;1	
TLE_BRIEF.FDL;1	

B.2 ABS Logical Names

The following logical names are entered into the system logical name table when ABS installation procedure is complete. These names are stored in the startup file, SYS$STARTUP:ABS$STARTUP.COM. They are automatically entered into the system logical name table whenever the system reboots or whenever the software is invoked. Table B–2 describes the logical names.

Table B–2 ABS Logical Names

Logical Name[a]	Description
ABS$CATALOG	This logical name points to the directory containing. ABS catalogs.
ABS$CLIENT_DB	Stores the license information for UNIX and NT clients.
ABS$DATABASE	This logical name points to the Rdb/VMS database used by ABS to store policy objects.

Table B–2 ABS Logical Names

Logical Name[a]	Description
ABS$GUI	This logical name points to the executable image of ABS graphical user interface (GUI).
ABS$LISTINGS	This logical name points to the directory where listing files produced by ABS will reside when requested by the customer.
ABS$LOG	This logical name points to the directory where all log files for save and restore operations are placed.
ABS$ROOT	This logical name defines the top of the directory tree used by ABS to store its files.
ABS$SLS_CONVERSION	
ABS$SYSTEM	This logical name points to the directory where all ABS system files and images reside.
ABS$TEMPLATES	This logical name points to the directory where the templates used to control backup agents are stored. Recommendation: It is not recommended that you modify these templates. The behavior of ABS in regard to its backup agents is defined by these templates.
ABS_CATALOG_CLEANUP	Defined when the catalog cleanup utility is executed.
ABS_UID_LOC	Defines the location of the .UID files used by the GUI.
ABS$USSSHR	This logical points to the appropriate ABS$USSSHR image for the current version of the Operating System.

a. For portability purposes, each logical name also has a corresponding "underscore"
    version of the logical name. For example, ABS$CATALOG also translates as
    ABS_CATALOG.

The following logical names can be used for special circumstances.

MOUNT_RETRY_INTERVAL	Defines the amount of time that ABS waits before attempting to retry the next save or restore operation.
ABS_DIAGNOSTIC_LEVEL	Typically, ABS provides all diagnostic levels in the log files. You can limit the number of messages output to the log file by setting this to a smaller number.
ABS$storage_call_LOAD_TIMEOUT	Defines how long a load request should remain outstanding before being cancelled.

C

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 C.1 lists names of the files installed and Section C.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.

———

C.1 MDMS File Names

Table C–1 contains the names of all MDMS files created on the system after MDMS V3.2 is successfully installed.

—————————————————Note—————————————————

Some files may not be installed depending on the options you select.

———

Table C–1 MDMS Installed Files

File Name	File Name
SYS$HELP	
MDMS.HLP	MDMSV32.RELEASE_NOTES
MDMSV32.RELEASE_NOTES.PS	
SYS$LIBRARY	
MDMS$SHR.EXE	
SYS$MESSAGE	
MDMS$MSG.EXE	RDF$MSG.EXE
SYS$MANAGER	
MDMS$SYSTARTUP.COM	MDMS$SYSTARTUP.TEMPLATE
SYS$SHARE	
MDMS$SHR.EXE	MRD$RTL.EXE

Table C–1 MDMS Installed Files

File Name	File Name

SYS$STARTUP

MDMS$SHUTDOWN.COM	MDMS$STARTUP.COM
MDMS$UNINSTALL.COM	

SYS$SYSTEM

MDMS$SERVER.EXE,	MDMS$DCL.EXE
MDMS$CONVERT_V3_TO_V2.EXE,	MDMS$SERVER.EXE
MDMS$CONVERT_V2_TO_V3.EXE	

SYS$TEST

MDMS$IVP.COM

MDMS$ROOT:[DATABASE]

MDMS$DOMAIN_DB.DAT	MDMS$DRIVE_DB.DAT
MDMS$GROUP_DB.DAT	MDMS$JUKEBOX_DB.DAT
MDMS$LOCATION_DB.DAT	MDMS$MAGAZINE_DB.DAT
MDMS$MEDIA_DB.DAT	MDMS$NODE_DB.DAT
MDMS$POOL_DB.DAT	MDMS$VOLUME_DB.DAT

MDMS$ROOT:[GUI.VMS]

ALPDCL01_071.A-DCX_AXPEXE	ALPBASE02_071.A-DCX_AXPEXE
ALPACRT08_071.A-DCX_AXPEXE	ALPSYSA02_071.A-DCX_AXPEXE
ALPSYSB02_071.A-DCX_AXPEXE	ALPTHREADS_03071.A-DCX_AXPEXE
DEC-AXPVMS-JAVA-V0101-6-1.PCSI_DCX_AXPEXE	DEC-AXPVMS-VMS712_PTHREADS-V0100--4.PCSI-DCX_AXPEXE
MDMS.INI	MDMS.ZIP
MDMS_GUI_HELP.HTML	SYMANTEC.ZIP

MDMS$ROOT:[GUI.VMS.GRAPHICS]

CONFWIZ4.GIF	CONFWIZ5.GIF
DOMAIN2.GIF	DRIVE.GIF
GROUP.GIF	JUKEBOX.GIF
LOCATION.GIF	MAGAZINE.GIF
MEDIATYPE.GIF	NODE.GIF
POOL.GIF	REQUESTS.GIF
SERVJUKE.GIF	SPLASH.GIF
VOLROT.GIF	VOLUME.GIF

MDMS$ROOT:[GUI.ALPHA_NT]

JRE116ALPHANT.EXE	SETUP_ALPHA_NT.EXE

MDMS$ROOT:[GUI.WINTEL]

JRE117WINTEL.EXE	SETUP_INTEL.EXE

MDMS$ROOT:[PATCHES.ALPHA]

ALPY2K02_062.A	ALPY2K02_062.CVRLET_TXT

Table C–1 MDMS Installed Files

File Name	File Name

MDMS$ROOT:[PATCHES.VAX]

File Name	File Name
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

MDMS$ROOT:[SYSTEM]

File Name	File Name
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

MDMS$ROOT:[TTI_RDEV.ALPHA]

File Name	File Name
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	RDFREE.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	

MDMS$ROOT:[TTI_RDEV.VAX]

File Name	File Name
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_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

Table C–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

C.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 C–2describes the logical names in the system table

Table C–2 MDMS Logical Names

Logical Name	Definition and Description
MDMS$DATABASE_LOCATION	This logical points to the location of the MDMS database files.
MDMS$DATABASE_SERVERS	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. This list of names must be DECnet, DECnet-Plus, or TCP/IP node names. They can be a mix of different protocols or the same. For example the node list could look like this: `NODE1,NODE2.SITE.INC.COM,INC:.SITE.NODE3` The above example shows that to communicate with: • NODE1 use DECnet • NODE2 use DECnet-Plus • NODE3 use TCP/IP
MDMS$LOGFILE_LOCATION	This logical name points to the location of the MDMS log files.
MDMS$ROOT	This logical name points to the device and directory of the root for the MDMS files.

Table C–2 MDMS Logical Names

Logical Name	Definition and Description
MDMS$SUPPORT_PRE_V3	This logical name enables or disables support of SLS/MDMS V2.9x remote servers. When this logical is TRUE, a SLS/MDMS V2.9x client can communicate with a MDMS V3.2 server. If you do not have any SLS/MDMS V2.9x clients, define this logical as FALSE.
MDMS$SCHEDULED_ACTIVITIES_START_HOUR	This logical name is the start hour (0-23) for the following scheduled activities: • Deallocate volumes that have passed their scratch date • Free volumes that have passed their freed date • Scheduled magazine rotation • Scheduled volume rotation If this logical is not defined, these activities are scheduled for 1 AM.
MDMS$SYSTEM	This logical name points to the location of MDMS utilities.
MDMS$TCPIP_SND_PORTS	This logical name is the range of port numbers for outgoing TCP/IP connections. The defaults are ports 601 to 1023.
MDMS$VERSION3	This logical name is TRUE when ABS or HSM should use MDMS V3.x. If ABS or HSM is not supposed to use MDMS V3.x this logical will not be defined. The MDMS server should not be running if ABS or HSM is not supposed to use MDMS V3.x

D

MDMS V3 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 fol-
lowing 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.

D.1 MDMS Rights - Types

MDMS has the following rights:

- High-level rights

- Low level rights

- ABS rights

D.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 D–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 *hidden* 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 **default right**. By default, there are no low-level rights assigned to the *default* right. 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.

High Level Rights

You can disable the mapping of SYSPRV to MDMS_ALL_RIGHTS using a SET DOMAIN command

D.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.
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 D–2 Low Level Rights

Low Level Right Name	Allows Privilege to:
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_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

D.1.3 ABS Rights

MDMS can be defined to recognize ABS rights and map them to MDMS rights. This capability is disabled by default and can be enabled with a SET DOMAIN command. The exact mapping for ABS rights is as in :

Table D–3   ABS Rights

ABS RIGHTS	MDMS RIGHTS
ABS_BACKUP_JOB	MDMS_USER
ABS_BYPASS	MDMS_ALL_RIGHTS
ABS_CREATE_EXECUTION_ENV	MDMS_CREATE_ALL MDMS_SET_ALL MDMS_SHOW_ALL
ABS_CREATE_REMOTE_JOB	None
ABS_CREATE_STORAGE_CLASS	MDMS_CREATE_ALL MDMS_SET_ALL MDMS_SHOW_ALL
ABS_LOOKUP_ALL	None
ABS_SHOW_ALL	MDMS_SHOW_ALL

D.2 Default High-Level to Low-Level Mapping

This section defines the default high to low-level mapping for each high-level right.

D.2.1   MDMS_USER:

Table D–4 MDMS_USER Rights

MDMS User...	Allows privilege to...
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_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

D.2.2 MDMS_OPERATOR Rights:

Table D–5 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

D.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 D–1

```
SET DOMAIN /OPERATOR_RIGHTS=MDMS_SET_PROTECTED /ADD
```

This command adds the MDMS_SET_PROTECTED right to the operator rights list.

E

Sample Configuration of MDMS

This appendix shows a sample configuration of Media and Device Management System (MDMS) including examples for the steps involved.

E.1 Configuration Order

Configuration - which involves the creation or definition of MDMS objects, should take place in the following order:

1. Location
2. Media type
3. Node
4. Jukebox
5. Drives
6. Pools
7. Volumes

Creating these objects in the above order ensures that the following informational message, does not appear:

```
%MDMS-I-UNDEFINEDREFS, object contains undefined referenced objects
```

This message appears if an attribute of the object is not defined in the database. The object is created even though the attribute is not defined. The sample configuration consists of the following:

- Four nodes

```
SMITH1 - ACCOUN cluster node
SMITH2 - ACCOUN cluster node
SMITH3 - ACCOUN cluster node
JONES - a client node
```

- TL826 Jukebox with robot 1DUA560 and the following six drives:

```
$1$MUA560
$1$MUA561
$1$MUA562
$1$MUA563
$1$MUA564
$1$MUA565
```

Sample Configuration of MDMS
E.1 Configuration Order

The following examples illustrate each step in the order of configuration.

E.1.1 Configuration Step 1 Example - Defining Locations

This example lists the MDMS commands to define an offsite and onsite location for this domain.

```
$ !
$ !   create onsite location
$ !
$ MDMS CREATE LOCATION BLD1_COMPUTER_ROOM -
  /DESCRIPTION="Building 1 Computer Room"
$ MDMS SHOW LOCATION BLD1_COMPUTER_ROOM
     Location: BLD1_COMPUTER_ROOM
 Description: Building 1 Computer Room
       Spaces:
 In Location:
$ !
$ !  create offsite location
$ !
$ MDMS CREATE LOCATION ANDYS_STORAGE -
  /DESCRIPTION="Andy's Offsite Storage, corner of 5th and Main"
$ MDMS SHOW LOCATION ANDYS_STORAGE
     Location: ANDYS_STORAGE
 Description: Andy's Offsite Storage, corner of 5th and Main
       Spaces:
 In Location:
```

E.1.2 Configuration Step 2 Example - Defining Media Type

This example shows the MDMS command to define the media type used in the TL826.

```
 !
$ !  create the media type
$ !
$ MDMS CREATE MEDIA_TYPE TK88K -
  /DESCRIPTION="Media type for volumes in TL826 with TK88 drives" -
  /COMPACTION      ! volumes are written in compaction mode
$ MDMS SHOW MEDIA_TYPE TK88K
  Media type: TK88K
 Description: Media type for volumes in TL826 with TK88 drives
     Density:
  Compaction: YES
    Capacity: 0
      Length: 0
```

E.1.3 Configuration Step 3 Example - Defining Domain Attributes

This example shows the MDMS command to set the domain attributes. The reason this command is not run until after the locations and media type are defined, is because they are default attributes for the domain object. Note that the deallocation state (transition) is taken as the default. All of the rights are taken as default also.

```
$ !
$ !  set up defaults in the domain record
$ !
$ MDMS SET DOMAIN -
  /DESCRIPTION="Smiths Accounting Domain" - ! domain name
  /MEDIA_TYPE=TK88K -                        ! default media type
  /OFFSITE_LOCATION=ANDYS_STORAGE -          ! default offsite location
  /ONSITE_LOCATION=BLD1_COMPUTER_ROOM -      ! default onsite location
  /PROTECTION=(S:RW,O:RW,G:RW,W)             ! default protection for volumes
$ MDMS SHOW DOMAIN/FULL
     Description: Smiths Accounting Domain
```

```
                Mail: SYSTEM
    Offsite Location: ANDYS_STORAGE
     Onsite Location: BLD1_COMPUTER_ROOM
     Def. Media Type: TK88K
   Deallocate State: TRANSITION
         Opcom Class: TAPES
            Priority: 1536
          Request ID: 2576
          Protection: S:RW,O:RW,G:RW,W
      DB Server Node: SPIELN
      DB Server Date: 1-FEB-1999 08:18:20
   Max Scratch Time: NONE
        Scratch Time:  365 00:00:00
     Transition Time:   14 00:00:00
     Network Timeout:    0 00:02:00
          ABS Rights: NO
      SYSPRIV Rights: YES
  Application Rights: MDMS_ASSIST
                      MDMS_LOAD_SCRATCH
                      MDMS_ALLOCATE_OWN
                      MDMS_ALLOCATE_POOL
                      MDMS_BIND_OWN
                      MDMS_CANCEL_OWN
                      MDMS_CREATE_POOL
                      MDMS_DEALLOCATE_OWN
                      MDMS_DELETE_POOL
                      MDMS_LOAD_OWN
                      MDMS_MOVE_OWN
                      MDMS_SET_OWN
                      MDMS_SHOW_OWN
                      MDMS_SHOW_POOL
                      MDMS_UNBIND_OWN
                      MDMS_UNLOAD_OWN
      Default Rights:
     Operator Rights: MDMS_ALLOCATE_ALL
                      MDMS_ASSIST
                      MDMS_BIND_ALL
                      MDMS_CANCEL_ALL
                      MDMS_DEALLOCATE_ALL
                      MDMS_INITIALIZE_ALL
                      MDMS_INVENTORY_ALL
                      MDMS_LOAD_ALL
                      MDMS_MOVE_ALL
                      MDMS_SHOW_ALL
                      MDMS_SHOW_RIGHTS
                      MDMS_UNBIND_ALL
                      MDMS_UNLOAD_ALL
                      MDMS_CREATE_POOL
                      MDMS_DELETE_POOL
                      MDMS_SET_OWN
                      MDMS_SET_POOL
         User Rights: MDMS_ASSIST
                      MDMS_ALLOCATE_OWN
                      MDMS_ALLOCATE_POOL
                      MDMS_BIND_OWN
                      MDMS_CANCEL_OWN
                      MDMS_DEALLOCATE_OWN
                      MDMS_LOAD_OWN
                      MDMS_SHOW_OWN
                      MDMS_SHOW_POOL
                      MDMS_UNBIND_OWN
                      MDMS_UNLOAD_OWN
```

E.1.4 Configuration Step 4 Example - Defining MDMS Database Nodes

This example shows the MDMS commands for defining the three MDMS database nodes of the cluster ACCOUN. This cluster is configured to use DECnet-PLUS.

Sample Configuration of MDMS
E.1 Configuration Order

Note that a node is defined using the DECnet node name as the name of the node.

- If the node has DECnet-PLUS installed, the DECnet Fullname attribute must be the DEC-net-PLUS full name.

- If the node uses TCP/IP, the TCP/IP attribute should be defined.

- If you use the GUI, you must define the TCP/IP attribute and include TCPIP in the Trans-ports attribute.

```
$ !
$ !   create nodes
$ !   database node
$ MDMS CREATE NODE SMITH1 -             ! DECnet node name
  /DESCRIPTION="ALPHA node on cluster ACCOUN" -
  /DATABASE_SERVER -                    ! this node is a database server
  /DECNET_FULLNAME=SMI:.BLD.SMITH1 -    ! DECnet-Plus name
  /LOCATION=BLD1_COMPUTER_ROOM -
  /TCPIP_FULLNAME=SMITH1.SMI.BLD.COM - ! TCP/IP name
$ MDMS SHOW NODE SMITH1
            Node: SMITH1
     Description: ALPHA node on cluster ACCOUN
DECnet Fullname: SMI:.BLD.SMITH1
TCP/IP Fullname: SMITH1.SMI.BLD.COM:2501-2510
        Disabled: NO
Database Server: YES
        Location: BLD1_COMPUTER_ROOM
   Opcom Classes: TAPES
      Transports: DECNET,TCPIP
$ MDMS CREATE NODE SMITH2  -            ! DECnet node name
  /DESCRIPTION="ALPHA node on cluster ACCOUN" -
  /DATABASE_SERVER -                    ! this node is a database server
  /DECNET_FULLNAME=SMI:.BLD.SMITH2 -    ! DECnet-Plus name
  /LOCATION=BLD1_COMPUTER_ROOM -
  /TCPIP_FULLNAME=SMITH2.SMI.BLD.COM - ! TCP/IP name
  /TRANSPORT=(DECNET,TCPIP)             ! TCPIP used by JAVA GUI and JONES
$ MDMS SHOW NODE SMITH2
            Node: SMITH2
     Description: ALPHA node on cluster ACCOUN
DECnet Fullname: SMI:.BLD.SMITH2
TCP/IP Fullname: SMITH2.SMI.BLD.COM:2501-2510
        Disabled: NO
Database Server: YES
        Location: BLD1_COMPUTER_ROOM
   Opcom Classes: TAPES
      Transports: DECNET,TCPIP
$ MDMS CREATE NODE SMITH3 -            ! DECnet node name
  /DESCRIPTION="VAX node on cluster ACCOUN" -
  /DATABASE_SERVER -                    ! this node is a database server
  /DECNET_FULLNAME=SMI:.BLD.SMITH3  -   ! DECnet-Plus name
  /LOCATION=BLD1_COMPUTER_ROOM -
  /TCPIP_FULLNAME=CROP.SMI.BLD.COM -    ! TCP/IP name
  /TRANSPORT=(DECNET,TCPIP)             ! TCPIP used by JAVA GUI and JONES
$ MDMS SHOW NODE SMITH3
            Node: SMITH3
     Description: VAX node on cluster ACCOUN
DECnet Fullname: SMI:.BLD.SMITH3
TCP/IP Fullname: CROP.SMI.BLD.COM:2501-2510
        Disabled: NO
Database Server: YES
        Location: BLD1_COMPUTER_ROOM
   Opcom Classes: TAPES
      Transports: DECNET,TCPIP
```

E.1.5 Configuration Step 5 Example - Defining a Client Node

This example shows the MDMS command for creating a client node. TCP/IP is the only transport on this node.

```
$ !
$ !    client node
$ !    only has TCP/IP
$ MDMS CREATE NODE JONES -
  /DESCRIPTION="ALPHA client node, standalone" -
  /NODATABASE_SERVER -                  ! not a database server
  /LOCATION=BLD1_COMPUTER_ROOM -
  /TCPIP_FULLNAME=JONES.SMI.BLD.COM -  ! TCP/IP name
  /TRANSPORT=(TCPIP)                    ! TCPIP is used by JAVA GUI
$ MDMS SHOW NODE JONES
            Node: JONES
     Description: ALPHA client node, standalone
DECnet Fullname:
TCP/IP Fullname: JONES.SMI.BLD.COM:2501-2510
        Disabled: NO
Database Server: NO
        Location: BLD1_COMPUTER_ROOM
   Opcom Classes: TAPES
      Transports: TCPIP
```

E.1.6 Configuration Step 6 Example - Creating a Jukebox

This example shows the MDMS command for creating a jukebox

```
$ !
$ !    create jukebox
$ !
$ MDMS CREATE JUKEBOX TL826_JUKE -
  /DESCRIPTION="TL826 Jukebox in Building 1" -
  /ACCESS=ALL -            ! local + remote for JONES
  /AUTOMATIC_REPLY -       ! MDMS automatically replies to OPCOM requests
  /CONTROL=MRD -           ! controled by MRD robot control
  /NODES=(SMITH1,SMITH2,SMITH3) - ! nodes the can control the robot
  /ROBOT=$1$DUA560 -       ! the robot device
  /SLOT_COUNT=176          ! 176 slots in the library
$ MDMS SHOW JUKEBOX TL826_JUKE
      Jukebox: TL826_JUKE
  Description: TL826 Jukebox in Building 1
        Nodes: SMITH1,SMITH2,SMITH3
       Groups:
     Location: BLD1_COMPUTER_ROOM
     Disabled: NO
       Shared: NO
   Auto Reply: YES
       Access: ALL
        State: AVAILABLE
      Control: MRD
        Robot: $1$DUA560
   Slot Count: 176
        Usage: NOMAGAZINE
```

E.1.7 Configuration Step 7 Example - Defining a Drive

This example shows the MDMS commands for creating the six drives for the jukebox.
This example is a command procedure that uses a counter to create the six drives. In this example it is easy to do this because of the drive name and device name. You may want to have the drive name the same as the device name. For example:

Sample Configuration of MDMS
E.1 Configuration Order

```
$ MDMS CREATE DRIVE $1$MUA560/DEVICE=$1$MUA560
```

This works fine if you do not have two devices in your domain with the same name.

```
$ COUNT = COUNT + 1
$ IF COUNT .LT. 6 THEN GOTO DRIVE_LOOP
$DRIVE_LOOP:
$ MDMS CREATE DRIVE TL826_D1 -
  /DESCRIPTION="Drive 1 in the TL826 JUKEBOX" -
  /ACCESS=ALL -            ! local + remote for JONES
  /AUTOMATIC_REPLY -       ! MDMS automatically replies to OPCOM requests
  /DEVICE=$1$MUA561 - ! physical device
  /DRIVE_NUMBER=1 - ! the drive number according to the robot
  /JUKEBOX=TL826_JUKE -    ! jukebox the drives are in
  /MEDIA_TYPE=TK88K -      ! media type to allocate drive and volume for
  /NODES=(SMITH1,SMITH2,SMITH3)! nodes that have access to drive
$ MDMS SHOW DRIVE TL826_D1
            Drive: TL826_D1
      Description: Drive 1 in the TL826 JUKEBOX
           Device: $1$MUA561
            Nodes: SMITH1,SMITH2,SMITH3
           Groups:
           Volume:
         Disabled: NO
           Shared: NO
        Available: NO
            State: EMPTY
          Stacker: NO
 Automatic Reply: YES
  RW Media Types: TK88K
  RO Media Types:
           Access: ALL
          Jukebox: TL826_JUKE
     Drive Number: 1
        Allocated: NO
                :
                :
                :
$ MDMS CREATE DRIVE TL826_D5 -
  /DESCRIPTION="Drive 5 in the TL826 JUKEBOX" -
  /ACCESS=ALL -            ! local + remote for JONES
  /AUTOMATIC_REPLY -       ! MDMS automatically replies to OPCOM requests
  /DEVICE=$1$MUA565 - ! physical device
  /DRIVE_NUMBER=5 - ! the drive number according to the robot
  /JUKEBOX=TL826_JUKE -    ! jukebox the drives are in
  /MEDIA_TYPE=TK88K -      ! media type to allocate drive and volume for
  /NODES=(SMITH1,SMITH2,SMITH3)! nodes that have access to drive
$ MDMS SHOW DRIVE TL826_D5
            Drive: TL826_D5
      Description: Drive 5 in the TL826 JUKEBOX
           Device: $1$MUA565
            Nodes: SMITH1,SMITH2,SMITH3
           Groups:
           Volume:
         Disabled: NO
           Shared: NO
        Available: NO
            State: EMPTY
          Stacker: NO
 Automatic Reply: YES
  RW Media Types: TK88K
  RO Media Types:
           Access: ALL
          Jukebox: TL826_JUKE
     Drive Number: 5
        Allocated: NO
$ COUNT = COUNT + 1
$ IF COUNT .LT. 6  THEN GOTO DRIVE_LOOP
```

E.1.8 Configuration Step 8 Example - Defining Pools

This example shows the MDMS commands to define two pools: ABS and HSM. The pools need to have the authorized users defined.

```
$ !
$ !   create pools
$ !
$ mdms del pool abs
$ MDMS CREATE POOL ABS -
  /DESCRIPTION="Pool for ABS" -
  /AUTHORIZED=(SMITH1::ABS,SMITH2::ABS,SMITH3::ABS,JONES::ABS)
$ MDMS SHOW POOL ABS
            Pool: ABS
     Description: Pool for ABS
Authorized Users: SMITH1::ABS,SMITH2::ABS,SMITH3::ABS,JONES::ABS
   Default Users:
$ mdms del pool hsm
$ MDMS CREATE POOL HSM -
  /DESCRIPTION="Pool for HSM" -
  /AUTHORIZED=(SMITH1::HSM,SMITH2::HSM,SMITH3::HSM)
$ MDMS SHOW POOL HSM
            Pool: HSM
     Description: Pool for HSM
Authorized Users: SMITH1::HSM,SMITH2::HSM,SMITH3::HSM
   Default Users:
```

E.1.9 Configuration Step 9 Example - Defining Volumes using the /VISION qualifier

This example shows the MDMS commands to define the 176 volumes in the TL826 using the /VISION qualifier. The volumes have the BARCODES on them and have been placed in the jukebox. Notice that the volumes are created in the UNINITIALIZED state. The last command in the example initializes the volumes and changes the state to FREE.

```
$ !
$ !  create volumes
$ !
$ !  create 120 volumeS for ABS
$ !    the media type, offsite location, and onsite location
$ !    values are taken from the DOMAIN object
$ !
$ MDMS CREATE VOLUME -
   /DESCRIPTION="Volumes for ABS" -
   /JUKEBOX=TL826_JUKE -
   /POOL=ABS -
   /SLOTS=(0-119) -
   /VISION
$ MDMS SHOW VOLUME BEB000
     Volume: BEB000
 Description: Volumes for ABS
  Placement: ONSITE BLD1_COMPUTER_ROOM
 Media Types: TK88K                            Username:
        Pool: ABS                             Owner UIC: NONE
 Error Count:          0                        Account:
 Mount Count:          0                       Job Name:
       State: UNINITIALIZED                    Magazine:
 Avail State: UNINITIALIZED                     Jukebox: TL826_JUKE
Previous Vol:                                      Slot: 0
    Next Vol:                                      Drive:
      Format: NONE                          Offsite Loc: ANDYS_STORAGE
  Protection: S:RW,O:RW,G:RW,W              Offsite Date: NONE
    Purchase:  1-FEB-1999 08:19:00            Onsite Loc: BLD1_COMPUTER_ROOM
    Creation:  1-FEB-1999 08:19:00                 Space:
        Init:  1-FEB-1999 08:19:00           Onsite Date: NONE
```

```
               Allocation: NONE                            Brand:
                 Scratch: NONE               Last Cleaned:  1-FEB-1999 08:19:00
            Deallocation: NONE                Times Cleaned: 0
              Trans Time:    14 00:00:00         Rec Length: 0
                   Freed: NONE                  Block Factor: 0
             Last Access: NONE
            $ !
            $ ! create 56 volumes for HSM
            $ !
            $ MDMS CREATE VOLUME -
                /DESCRIPTION="Volumes for HSM" -
                /JUKEBOX=TL826_JUKE -
                /POOL=HSM -
                /SLOTS=(120-175) -
                /VISION
            $ MDMS SHOW VOL BEB120
                  Volume: BEB120
             Description: Volumes for HSM
               Placement: ONSITE BLD1_COMPUTER_ROOM
             Media Types: TK88K                       Username:
                    Pool: HSM                        Owner UIC: NONE
             Error Count:          0                    Account:
             Mount Count:          0                    Job Name:
                   State: UNINITIALIZED                 Magazine:
             Avail State: UNINITIALIZED                  Jukebox: TL826_JUKE
            Previous Vol:                                   Slot: 120
                Next Vol:                                  Drive:
                  Format: NONE                      Offsite Loc: ANDYS_STORAGE
              Protection: S:RW,O:RW,G:RW,W          Offsite Date: NONE
                Purchase:  1-FEB-1999 08:22:16        Onsite Loc: BLD1_COMPUTER_ROOM
                Creation:  1-FEB-1999 08:22:16             Space:
                    Init:  1-FEB-1999 08:22:16       Onsite Date: NONE
              Allocation: NONE                            Brand:
                 Scratch: NONE               Last Cleaned:  1-FEB-1999 08:22:16
            Deallocation: NONE                Times Cleaned: 0
              Trans Time:    14 00:00:00         Rec Length: 0
                   Freed: NONE                  Block Factor: 0
             Last Access: NONE
            $ !
            $ ! initialize all of the volumes
            $ !
            $ MDMS INITIALIZE VOLUME -
                /JUKEBOX=TL826_JUKE -
                /SLOTS=(0-175)
            $ MDMS SHOW VOL BEB000
                  Volume: BEB000
             Description: Volumes for ABS
               Placement: ONSITE BLD1_COMPUTER_ROOM
             Media Types: TK88K                       Username:
                    Pool: ABS                        Owner UIC: NONE
             Error Count:          0                    Account:
             Mount Count:          0                    Job Name:
                   State: FREE                          Magazine:
             Avail State: FREE                           Jukebox: TL826_JUKE
            Previous Vol:                                   Slot: 0
                Next Vol:                                  Drive:
                  Format: NONE                      Offsite Loc: ANDYS_STORAGE
              Protection: S:RW,O:RW,G:RW,W          Offsite Date: NONE
                Purchase:  1-FEB-1999 08:19:00        Onsite Loc: BLD1_COMPUTER_ROOM
                Creation:  1-FEB-1999 08:19:00             Space:
                    Init:  1-FEB-1999 08:19:00       Onsite Date: NONE
              Allocation: NONE                            Brand:
                 Scratch: NONE               Last Cleaned:  1-FEB-1999 08:19:00
            Deallocation: NONE                Times Cleaned: 0
              Trans Time:    14 00:00:00         Rec Length: 0
                   Freed: NONE                  Block Factor: 0
             Last Access: NONE
```

F

Upgrading ABS From the ABS-OMT License

When upgrading ABS after using the ABS-OMT license, you only have to perform a couple of steps. Follow these steps, to upgrade ABS:

1. Shutdown ABS by issuing the command

   ```
   $ @SYS$MANAGER:ABS$SHUTDOWN.COM
   ```

2. Register and load the ABS-OMT-UPG license. You need both the ABS-OMT and ABS-OMT-UPG license.

3. Startup ABS by issuing the command:

   ```
   $ @SYS$STARTUP:ABS$STARTUP.COM
   ```

4. Run the following executable to load the default ABS storage and environment policies:

   ```
   $ RUN ABS$SYSTEM:ABS$DB_INIT.EXE
   ```

   All features of ABS are now enabled.

G

How to Delete ABS/MDMS From Your System

To delete ABS software from your system, shut down ABS and delete it from the system:

```
$ @SYS$STARTUP:ABS$SHUTDOWN
$ @ABS$SYSTEM:ABS$DELETE_ABS
```

To delete MDMS software from your system, shut down MDMS and uninstall it from your system:

```
$ @SYS$STARTUP:MDMS$SHUTDOWN
$ @SYS$STARTUP:MDMS$UNINSTALL.COM
```

Glossary

This glossary contains terms defined for the Archive Backup System for OpenVMS (ABS). It also contains terms associated with the following products when related to ABS:

- Media and Device Management Services for OpenVMS (MDMS)

- Storage Library System for OpenVMS (SLS)

absolute time

A data-entry format for specifying the date or time of day. The format for absolute time is [dd-mmm-yyyy[:]][hh:mm:ss.cc]. You can specify a date and time, or use the keywords TODAY, TOMORROW, or YESTERDAY.

access port

The port on a DCSC-controlled silo where volumes can be inserted into the silo.

active server process

The MDMS server process that is currently active. The active server process responds to requests issued from an MDMS client process.

allocate

To reserve something for private use. In MDMS software, a user is able to allocate volumes or drives.

allocated

The state of a drive or volume when a process is granted exclusive use of that drive or volume. The drive or volume remains allocated until the process gives up the allocation.

allocated state

One of four volume states. Volumes that are reserved for exclusive use by a user (such as ABS) are placed in the allocated state. Allocated volumes are available only to the user name (such as ABS) assigned to that volume.

ANSI

The abbreviation for the American National Standards Institute, an organization that publishes computer industry standards.

ANSI-labeled

A magnetic tape that complies with the ANSI standards for label, data, and record formats. The format of VMS ANSI-labeled magnetic tape volumes is based on Level 3 of the ANSI standard for magnetic tape labels and file structure.

archive

A repository of data that consists of

- Volumes that contains zero or more archive files.

- One or more catalogs that contain information about archived data that is stored on volumes.

- A set of services used to define the storage environment configuration and site policy. These services are also used to move data between the ABS client and the MDMS volume.

archive file system

The file system that contains the archived data.

archive object

The data object that resides in offline storage.

archiving

Saving data for the purpose of long-term storage.

ASCII

The abbreviation for the American Standard Code for Information Interchange.

This code is a set of 8-bit binary numbers representing the alpha- bet, punctuation, numerals, and other special symbols used in text representation and communications protocols.

back up

To make duplicate copies of one or more files, usually onto different media than the original media. This provides the availability to restore the original data if it is lost or corrupted.

backup agent

The client or utility that performs the actual save or restore operation. Examples are the VMS BACKUP Utility and the RMU Backup Utility.

backup engine

The backup engine moves data to and from the storage policy. Examples of backup engines: VMS BACKUP, RMU BACKUP, and UBS.

BACKUP format

Standard OpenVMS BACKUP format. The BACKUP format is the recording format used by the VMS Backup utility to back up data to save sets.

backup management domain

A node or OpenVMS Cluster system that has control over creating save requests. A backup management domain is usually controlled by a single storage administrator.

bind

The act of logically binding volumes into a magazine. This makes the volumes a logical unit that cannot be separated unless an UN- BIND operation is done on the volumes.

blocking factor

The number of records in a physical tape block. The length of a physical block written to magnetic tape is determined by multiplying the record length by the blocking factor. For example, if a record length of 132 and a blocking factor of 20 are specified, the length of each physical block written to tape will be 2640 bytes (or characters).

The blocking factor is only used when MDMS software is writing an EBCDIC tape.

catalog

Contains records of data movement operations. Each time a save request is initiated, the history of the data movement operation is recorded in an associated ABS   central security domain: The node or OpenVMS Cluster system where ABS policy server is installed. This domain controls all ABS pol- icy objects, particularly storage and environment policies.

client node

Client nodes send database requests to the server node.

combination time:

A data-entry format for specifying date and time. Combination time consists of an absolute time value plus or minus a delta time value.

Examples:

"TODAY+7-"	Indicates current date plus seven days
"TODAY+7"	Indicates current date plus seven hours
"TOMORROW-1"	Indicates current date at 23:00 hours

command

An instruction, generally an English word, entered by the user at a terminal. The command requests the software to perform a pre- defined function.

CRC

The acronym for cyclic redundancy check. It is a verification process used to ensure data is correct.

consolidation count

The criteria under which ABS creates new volume sets.

consolidation interval

The number of days (in VMS time format) between the creation of new volume sets.

consolidation size

The desired maximum number of volumes allowed in a volume set.

data object

A data object specification, such as an OpenVMS file name or an Rdb/VMS database file name.

data movement request

Either a save or restore request initiated through either the DCL command interface or ABS graphical user interface.

deallocate

To relinquish ownership of a drive, volume, or volume set.

- When a drive is deallocated, it is then available for allocation by other processes.
- When a volume set is deallocated, it is either immediately available for allocation by other users or moved into a transition state.

deassign date

The day on which an allocated volume is scheduled to go into the transition state or the free state.

default

A value or operation automatically included in a command or field unless the user specifies differently.

density

The number of bits per inch (bpi) on magnetic tape. Typical values are 6250 bpi and 1600 bpi.

device

A physical device, such as a tape drive or disk device.

down state

One of four volume states. Volumes that are either damaged, lost, or temporarily removed from the MDMS volume database for cleaning are placed in the down state.

EBCDIC

Extended Binary Coded Decimal Interchange Code. EBCDIC is an unlabeled IBM recording format. Volumes in EBCDIC format do not have records in the MDMS volume database.

environment policy

ABS policy object that defines the environment in which data ABS save and restore requests occur.

expiration

The date and time at which an archived data is no longer considered useful. The archived data can be deleted and its space removed.

format

See recording format.

free state

The volume state that allows volumes to be selected by users or other software applications.

GUI

Graphical User Interface

in port

The physical opening in a jukebox where volumes can be imported.

interface

A shared physical or logical boundary between computing system components. Interfaces are used for sending and/or accepting information and control between programs, machines, and people.

inventory

The act of automatically updating the MDMS database. MDMS can mount each volume located in a magazine and update the MDMS volume database through this process.

I/O station

A jukebox component that enables an operator to manually insert and retrieve volumes. The I/O station consists of an I/O station door on the outside of the jukebox and an I/O station slot on the inside. See also I/O station door and I/O station slot.

I/O station door

An actual door on the outside of the jukebox that can be opened and closed. Behind the I/O station door is the I/O station slot.

I/O station slot

An I/O slot that holds a volume when it is entering or leaving the jukebox.

label

- Information recorded at a fixed location on the volume that identifies the volume to software.
- The physical printed label attached to the outside of the volume to identify it.

labeled

A recording format that includes a volume label.

LEBCDIC

Labeled EBCDIC format. See also EBCDIC.

load

The process which makes a volume physically available to the computer system, such as for read or write operations.

local symbol

A symbol meaningful only to the module or DCL command procedure that defines it.

log file

Any file into which status and error messages are written to reflect the progress of a process.

MDMS server node

The active server node to which all MDMS database requests are sent to be serviced. In a high-availability configuration, when the active server node fails, another node (see MDMS standby server process) in the OpenVMS Cluster system becomes the active server node.

MDMS software

The MDMS software is an OpenVMS software service that enables you to implement media and device management for your storage management operations. MDMS provides services to SLS, ABS, and HSM.

MDMS standby server process

Any MDMS server process that is not currently active. The standby server process waits and becomes active if the active server process fails.

magazine

A physical container that holds from 5 to 11 volumes. The magazine contains a set of logically bound volumes that reside in the MDMS database.

magazine database

The MDMS database that contains the magazine name and the volume names associated with that magazine.

media

A mass storage unit. Media is referred to in this document as a volume. Volumes provide a physical surface on which data is stored. Examples of physical volumes are magnetic tape, tape cartridge, and optical cartridge.

media type

A set of site-specific names associated with volume densities and drives.

nearline storage

Storage in which file headers are accessible through the operating system, but accessing data requires extra intervention.

Nearline storage employs a robotic device to move volumes between drives and volume storage locations. Nearline storage is less costly for each megabyte of data stored. Access times for data in nearline storage may vary. Access to data may be nearly instantaneous when a volume containing the data is already loaded in a drive. The time required for a robotic device to move to the most distant storage location, retrieve a volume, load it into a drive, and position the volume determines the maximum access time.

The devices of nearline storage technology include, but are not limited to, automated tape libraries and optical jukeboxes.

offline storage

Storage in which neither the file headers nor the data is accessible by the operating system and requires extra intervention.

Offline storage requires some type of intervention to move volumes between drives and the volumes' storage location. Offline storage is the least costly for each megabyte of data stored. Access times for data in offline storage vary for the same reasons as described for nearline storage. For archive data stored in a remote vault, access time can take more than a day.

The devices of offline storage technology include, but are not limited to, standalone tape drives, optical disk drives, and sequential stack loader tape drives.

online storage

Storage in which file headers and data can be accessed through the operating system. Online storage is the most costly for each megabyte of data stored.

As a trade off, online storage also offers the highest access performance. Online storage devices offer continuous service. The devices of online storage technology include disk storage and electronic (RAM) storage that uses disk I/O channels.

OPCOM

OpenVMS Operator Communication Manager. An online communication tool that provides a method for users or batch jobs to request assistance from the operator, and allows the operator to send messages to interactive users.

OPER privilege

The level of privilege required by a system operator to suspend an MDMS operation and to perform a variety of maintenance procedures on volumes, as well as archive files and saved system files.

out port

The physical opening in a jukebox where volumes can be exported from the jukebox.

policy

The decisions and methods in which you implement your ABS policy. This includes when and how often you back up or archive data from online to nearline or offline storage.

policy engine

The component in ABS that makes intelligent decisions based upon the implementation of your ABS policy.

policy objects

The method in which ABS enables you to implement your ABS policy. ABS provides the following policy objects:

- Storage policy

- Environment policy

- Save request

- Restore request

policy server

ABS server component. Placement of this component determines the central security domain (CSD).

pool

A set of volumes in the free state. Those volumes can be allocated by users who have access to the volume pool. The storage administrator creates and authorizes user access to pools.

record

A set of related data treated as a unit of information. For example, each volume that is added to the MDMS volume database has a record created that contains information about the volume.

record length

The length of a record in bytes. See also blocking factor.

recorded label

The label recorded on the volume.

recording format

The unique arrangement of data on a volume according to a predetermined standard. Examples of recording format are BACKUP, EBCDIC, and ANSI.

restore process

The method by which the contents of a file or disk are recovered from a volume or volumes that contain the saved data. ABS software will restore data by querying ABS catalog for the file or disk name specified in the restore request, and then locate the BACKUP save sets from one or more volumes, extract the data from those save sets, and place the information onto a Files-11 structured disk where the restored data can be accessed by a user.

restore request

A request to restore data from the archives to either its original location or an alternate location. Restore re- quests are initiated either through the DCL command interface or ABS graphical user interface.

requester

The user who creates a save or restore request.

requester profile

The requester profile is the profile of the user who is creating the save or restore request. This profile is captured at the time the request is created.

restore request

ABS policy object that defines the request for the restoration of data.

robot device

A tape or optical drive that provides automatic loading of volumes, such as a TF867 or a TL820.

save process

The method by which copies of files are made on magnetic or optical volumes for later recovery or for transfer to another site.

For BACKUP formatted volumes, an ABS save operation creates BACKUP save sets on magnetic tape volume, a system disk, or optical volume.

save request

ABS policy object that defines the request for saving data.

save set

A file created by the VMS Backup Utility on a volume. When the VMS Backup Utility saves data, it creates a file in BACKUP format called a save set on the specified output volume. A single BACKUP save set can contain numerous files. Only BACKUP can interpret save sets and restore the data stored in the save set.

slot

A vertical storage space for storing a volume. The storage racks and cabinets used in data centers contain multi-row slots that are labeled to easily locate stored volumes.

storage administrator

One or more privileged users responsible for installing, configuring, and maintaining ABS software. This user has enhanced ABS authorization rights and privileges and controls the central security domain (CSD) by creating and maintaining ABS storage and environment policies.

storage policy

ABS policy object that defines where to store data saved using ABS.

SYSPRV privilege

The level of privilege required to install the software and add user names to the system.

system backup

An ABS system typically saves the system disk, also known as a full disk backup. The system backup can direct ABS software to perform automotive save operations on a predetermined schedule.

tape

See volume.

transition state

Volumes in the transition state are in the process of being deallocated, but are not yet fully deallocated. The transition state provides a grace period during which a volume can be reallocated to the original owner if necessary.

UASCII

Unlabeled ASCII format. See also ASCII.

UIC

User identification code. The pair of numbers assigned to users, files, pools, global sections, common event flag clusters, and mailboxes. The UIC determines the owner of a file or ABS policy object. UIC-based protection determines the type of access available to the object for its owner, members of the same UIC group, system accounts, and other (world) users.

UID

A globally unique identifier for this instance of an object.

unbind

The act of unbinding a volume or volumes from a magazine.

unlabeled

A recording format that does not include a recorded label.

user backup

A save request created by an individual user (not the system) when they would like to make copies of a file or set of files for later recovery or for transfer to another site.

user profile

The set of information about a user that defines the user's right to access data or the user's right to access an ABS policy object. For ABS on OpenVMS, this includes the following information:

- User name

- UIC

- Privileges

- Access right identifiers

vault

An offsite storage location to where volumes are transferred for safekeeping.

VMS Backup Utility

An OpenVMS Operating System utility that performs save and restore operations on files, directories, and disks using the BACKUP recording format.

volume

A *physical* piece of media (volume) that is known *logically* to the MDMS volume database. A volume can be a single magnetic tape or disk, or as in the case of an optical cartridge, can refer to one side of double-sided media. A volume is assigned a logical name, known as the volume label.

volume ID

The volume's internal identification used to verify that the correct volume has been selected. The volume label should be the same as the volume ID.

volume name

Same as volume ID.

volume set

One or more volumes logically connected in a sequence to form a single volume set. A volume set can contain one or more save sets. ABS adds volumes to a volume set until the storage policy's consolidation criteria has been met or exceeded.

volume state

A volume status flag. In MDMS software, volumes are placed in one of the following states:

- Free

- Allocated

- Transition

- Down

wildcard character

A nonnumeric or nonalphanumeric character such as an asterisk (*) or percent sign (%) that is used in a file specification to indicate "ALL" for a given field or portion of a field. Wildcard characters can replace all or part of the file name, file type, directory name, or version number.

Index