Storage Library System for OpenVMS

Installation and Configuration Guide

Abstract

This document contains information for installing the Storage Library System for OpenVMS (SLS) software.

Revision/Update Information:	This revised document supersedes the previous release of this document
Software Version:	Storage Library System for OpenVMS Version 2.9J
Required Operating System:	OpenVMS VAX V7.3 OpenVMS Alpha V7.3-2 and V8.2

Hewlett-Packard Company
Palo Alto, California

January 2005

The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

Printed in the U.S.A.

Contents

3 Installing the SLS Software

4 After Installing SLS

A SLS Server Installation Example Log File

B Files and Logical Names

Tables

Preface

Purpose of This Document

This document contains information about installing Storage Library System for OpenVMS ™ software.

You can read the online release notes after completing the installation procedure by entering the following command:

```
$ TYPE SYS$HELP:SLSJ029.RELEASE_NOTES
```

Intended Audience

This manual is intended for experienced OpenVMS™ system managers and storage administrators. It should be used with the System Management Subkit of the OpenVMS documentation set.

Document Structure

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

Chapter 1 Identifies required software and hardware and explains tasks you need to perform before you install the SLS software.

Chapter 2 Contains detailed instructions for installing SMF software.

Chapter 3 Explains tasks you need to perform after you install the SLS software.

Appendix A Shows a sample SLS server installation.

Appendix B Shows a sample SLS client installation.

Appendix C Lists SLS files and logical names.

Appendix D Lists SLS quota and account information.

Related Documents

The following documents are related to this documentation set or are mentioned in this manual:

- *Storage Library System for OpenVMS Command Reference Guide*
- *Storage Library System for OpenVMS Guide to Backup and Restore Operations*
- *OpenVMS System Management Utilities Reference Manual: A--L)*
- *OpenVMS DCL Dictionary: A--M*
- *OpenVMS DCL Dictionary: N--Z*
- *OpenVMS License Management Utility Manual*
- *OpenVMS User's Manual*

Conventions

The following conventions are used in this document:

Convention	Description
`{}`	In format command descriptions, braces indicate required elements. You must include one of the elements.
`[ ]`	Square brackets show optional elements in command syntax. You can omit these elements if you wish to use the default response.
`...`	Horizontal ellipsis points indicate the omission of information from a sentence or paragraph that is not important to the topic being discussed.
. . .	Vertical ellipsis points indicate the omission of information from an example or command format. The information has been omitted because it is not important to the topic being discussed.
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 labeled 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 *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 are mentioned in this document:

Product	Description
ABS	ABS refers to Archive Backup System for OpenVMS™ software.
ABS OMT	ABS OMT refers to Archive/Backup System for OpenVMS Management Tools™ software.
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.

Determining and Reporting Problems

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

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

1

Introduction to the Storage Library System for OpenVMS

This chapter defines the role of HP's Storage Library System for OpenVMS (SLS) software product in storage management.

1.1 What is SLS?

Storage Library System for OpenVMS(SLS) is a layered software product that provides support for saving copies of data and restoring those copies as necessary.

Warning

Due to changes in functionality and database structure, SLS software versions 2.5B and later are incompatible with SLS software version 2.4A and earlier. If you are an existing SLS customer, and upgrade SLS on your server node to version 2.5 or higher, then you must also upgrade each of your client nodes to version 2.5 or higher. This requirement results from a change in the format of the common magazine database in SLS software version V2.5B.

Storage administrators face three primary storage management problems that SLS software helps to solve:

- Ensuring data is available in the event of a disaster
- Maintaining data in some form for regulatory and legal purposes
- Providing sufficient online storage space to meet users' needs

1.1.1 SLS Product Functions

SLS allows you to do the following things:

- Backup (save copies of) data files on an as-needed basis
- Backup groups of files or entire disks on a scheduled basis for maintenance purposes
- Restore previously-saved files or disks after a disaster occurs
- Move data from online storage to offsite storage (archive) when files no longer need to be available for frequent use
- Generate reports about saved data

1.1.2 SLS Product Features

Backup and Restore Features

SLS provides the following backup and restore features:

- Automatic, flexible scheduling for backup operations

- Support through the Remote Device Facility (RDF) for remote backup and archiving operations between devices located on different systems within a DECnet network, provided all are running SLS
- Online catalog containing information about backed up data
- Ability to restore data through the catalog without the user having to search for where the data is stored
- Integration with RMU/BACKUP and RMU/RESTORE for backing up and restoring Oracle Rdb databases
- DCL commands and menu interfaces
- Customization features to meet a variety of site- specific and operational storage needs
- Support for user-initiated as well as system-initiated backup operations

Note

RDF is not supported on OpenVMS Alpha V8.2.

Media and Device Management Features

SLS provides the following media and device management features:

- Volume database containing information about volumes that SLS can use
- Magazine database relating individual volumes to multi- volume magazines and magazine-loader jukebox devices
- Media life cycle control, including maintenance of volume retention periods, availability status, location, and scheduling of dates for volumes to move offsite and return
- Volume counters noting the number of times the volume has been mounted and the number of errors associated with the volume; these counters are used to schedule volume cleaning
- Volume access control based on user-defined volume pools
- Media and device configurations using TAPESTART.COM to define devices available for use and the media they support
- Support for a variety of media and devices, ranging from single-drive TK50 devices to robotically-controlled StorageTek silos and automated tape library systems

1.1.3 SLS Device Support

SLS supports a large array of storage devices, ranging from single-tape drives to automated optical and tape jukeboxes. See the SLS SPD for a complete list of supported devices.

1.2 Mixed Architecture Environments

Warning

Before You Install your Storage Management Software

If you are planning to install software in a mixed architecture OpenVMS Cluster, you should understand the configuration issues explained in this section. If you do not consider these configuration issues, you could spend considerable time deleting and editing files, and reinstalling the software.

This section addresses the characteristics of a mixed architecture environment and describes some fundamental approaches to installing and configuring your software to run in it. The following list identifies the topics and their purposes

- Section 1.2.1 defines the mixed architecture environment and discusses ways in which they can come about, change, then disappear. Each of these occurrences requires some consideration about how to configure your software.
- Section 1.2.2 lists the guiding principles that require you to make special considerations for mixed architecture implementation, and what these principles mean to you.
- Section 1.2.3 describes three possible approaches to implementing a mixed architecture environment.
- Section 1.2.4 explains why the documentation includes procedures for a specific approach. If you cannot use the documented procedures, you should decide on an approach before you begin installation.

1.2.1 Mixed Architecture Environments

A mixed architecture OpenVMS Cluster includes at least one VAX system and at least one Alpha system.

Creating a Mixed Architecture Configuration:

If you add an Alpha system to a homogenous VAX OpenVMS Cluster, or if you are currently running a homogenous Alpha OpenVMS Cluster and inherit a VAX system, you will have a mixed architecture environment.

Before you integrate the Alpha or VAX node into the system, you should decide an approach to take for handling mixed architecture issues.

Operating a Mixed Architecture Configuration:

If you are currently operating a mixed architecture environment, and you want to add a VAX system or an Alpha system you must integrate it into your current configuration consistently with your other applications.

You should understand the particular requirements of any new application you introduce into a mixed architecture OpenVMS Cluster.

Dissolving a Mixed Architecture Configuration:

If you remove the last VAX or Alpha system, leaving a homogenous OpenVMS Cluster, you should remove any aspects of configuration that accounted for the heterogeneous nature of the mixed architecture system. This includes (but is not limited to) removing startup files, duplicate directory structures, and logical tables.

1.2.2 Principles Guiding Mixed Architecture Configuration

Limitations:

VAX systems cannot execute image files compiled on an Alpha system, and Alpha systems cannot execute image files compiled on a VAX system. Other types of files cannot be shared, including object code files (.OBJ), and user interface description files (.UID). You must place files that cannot be shared in different locations: VAX files accessible only to VAX OpenVMS Cluster nodes, and Alpha files accessible only to Alpha OpenVMS Cluster nodes.

Data files, in most cases, must be shared between OpenVMS Cluster nodes. You should place all shared files in directories accessible by both VAX and Alpha OpenVMS Cluster nodes.

Logical names, that reference files which cannot be shared, or the directories in which they reside, must be defined differently on VAX and Alpha systems.

Files that assign logical name values must therefore be architecture specific. Such files may either reside on node-specific disks or shared only among OpenVMS Cluster nodes of the same hardware architecture.

1.2.3 Configuring Applications in a Mixed Architecture OpenVMS Cluster

This section describes three approaches to configuring applications to run in a mixed architecture OpenVMS Cluster. The one you choose depends on your existing configuration, and the needs of the particular application you are installing. These approaches are given as examples only. You should decide which you want to implement based on your own situation and style of system management.

All of these approaches have two aspects in common:

- All shared files reside in one location
- All files that cannot be shared reside in separate locations

1.2.3.1 Separate Disk Configuration

These characteristics describe the separate disk configuration:

- Product directories are installed on two separate disks.
- One of the product directories is a complete installation containing all data (and other shared) files, and all executable (and other nonshared) files for either VAX or Alpha systems.
- The other of the product directories is a partial product installation, with only those directories that contain either Alpha or VAX system executables and other nonshared files.
- The systems using the disk with the complete installation uses logicals that normally reference the product executables and shared files.
- The systems using the disk with only the nonshared files use normal product logical definitions to point to shared files and directories. System logicals that point to nonshared files are assigned to the specific device, directory and/or file names.

1.2.3.2 Separate Root Configuration

These characteristics describe the separate root configuration:

- Product directories are installed on the same disk, but at different root locations.
- One of the product directories is a complete installation containing all data (and other shared files), and all executable (and other nonshared) files for either VAX or Alpha systems.
- The other of the product directories is a partial product installation, with only those subdirectories that contain either Alpha or VAX system executables and other nonshared files.
- The systems using the directory with the complete installation uses logicals that normally reference the product executables and shared files.
- The systems using the directory with only the nonshared files use normal product logical definitions to point to shared files and directories. System logicals that point to nonshared files are assigned to the specific device, directory and/or file names.

1.2.3.3 Separate Subdirectory Configuration

These characteristics describe the separate directory configuration:

- Product directories are installed on the same disk, and under the same root.

- Any directory which would normally contain the nonshared files (under a single architecture installation) has two subdirectories: one for VAX system nonshared files, and one for Alpha system nonshared files.
- Logicals that reference nonshared files are assigned search list values that point to the directories which holds shared files, and to the architecture specific subdirectories holding nonshared files.

1.2.4 Implementation Specific Approach

This document includes specific procedures for a recommended approach based on current product configuration and the behavior of the installation software with respect to its use of logical definitions during upgrades.

If the recommended approach is inconsistent with the way you currently manage your system, you should decide on a different approach before you begin your installation procedures.

2

Before Installing SLS

This chapter explains preinstallation requirements for SLS.

2.1 Preinstallation Process

This section describes what you need to do before you run the SLS, Version 2.9J installation procedure. Table 2–1 identifies actions you need to take to prepare for installing the SLS, Version 2.9J server software (which provides all SLS functionality) or client software (which provides most SLS functionality, except for the media databases). Later sections of this chapter provide details.

Warning

Important - during Installation

Installing SLS Version 2.9J requires DECnet to be enabled throughout the process. Before you begin installing make sure DECnet is enabled.

Table 2–1 Preinstallation Tasks

Stage	Action
1.	Configure and Verify Hardware Connections
2.	Verify required quotas and OpenVMS operating system classes.
3.	Check that the system on which you intend to install SLS meets hardware and software requirements.
4.	Shut down previous versions of SLS software on all nodes of the cluster on which SLS software will be installed.
5.	Delete all existing SLS jobs from batch queues. If you do not, the jobs will try to run using the old SYSBAK.COM and will fail.
6.	Register the SLS license.

2.2 Configuring Your Hardware

Note

MDMS Component of SLS

The hardware configuration activities described in these sections refer to MDMS, which is an integral component of SLS.

This section describes how to configure automated media libraries, media loaders, and operator attended drives to enable MDMS to provide a managed media environment. This means installing, connecting, and testing the tape drives.

Each procedure depends on the kind of tape drive that you plan to use. Use the procedures appropriate for the tape drives you plan to include in the managed media environment:

- For an automated media library, refer to Section 2.2.1 Hardware Configuration For an Automated Media Library.
- For a media loader, refer to Section 2.2.2 Hardware Configuration for a Media Loader.
- For an operator attended drive, refer to Section 2.2.3 Hardware Configuration for an Operator Attended Drive.

2.2.1 Hardware Configuration For an Automated Media Library

The following procedure applies to the installation and testing of an automated media library. Configuration and testing of MDMS to manage an automated media library is described in Chapter 4.

2.2.1.1 Hardware Installation

During this procedure, refer to the drive-specific hardware installation procedure information for details. The following procedure provides only the basic steps necessary to later configure MDMS against the drive:

1. 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.
3. Make note of the drive name after you complete your connection procedure. You will need this, especially if you are going to use the Media Robot Utility software.

2.2.1.2 Test the Hardware

With an automated media library, use a utility such as the Media Robot Utility (MRU) or use the front panel of the drive and the OpenVMS system to test the connection between the host and drive.

Use the following procedure to test the hardware connection between the automated tape library and the OpenVMS system:

1. Verify the robot device name and the drive names located in the automatic tape library.

   Issue 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.

```
$ SHOW DEVICE [drive-name[:]]
```

The following is the StorageWorks Media Robot Utility command for examining the robot:

```
$ ROBOT SHOW ROBOT [robot_name]
```

If you have problems with the drive name, refer to the OpenVMS system and drive-specific hardware documentation for remedial action.

2. Place a volume in the robot.
3. Inject the volume through a port on the robot by using the front panel controls or, if you are using StorageWorks Media Robot Utility, by issuing the ROBOT INJECT command:

   ```
   $ ROBOT INJECT [PORT port_number] SLOT slot_number [ROBOT robot_name]
   ```

   Alternately, you can expose the internal storage slots and place the volume directly into the robot. You also should consider placing one or more cleaning volumes into the robot at this time.
4. Load the volume into the drive by using the front panel controls or, if you are using MRU, by issuing the ROBOT SHOW and ROBOT LOAD command:

   ```
   $ ROBOT SHOW SLOT 0-n [ROBOT robot_name]
   ```

   ```
   $ ROBOT LOAD SLOT slot_num [ROBOT robot_name
   ```
5. After loading the volume, check the OpenVMS system connection to the drive by mounting and dismounting the volume.

   Mount the volume foreign:

   ```
   $ MOUNT/FOREIGN/NOASSIST drive_name[:]
   ```

   Dismount the volume:

   ```
   $ DISMOUNT drive_name[:]
   ```
6. Remove the volume from the drive by using the front panel controls or, if you are using MRU, by issuing the ROBOT UNLOAD command:

   ```
   $ ROBOT UNLOAD SLOT slot_number [ROBOT robot_name]
   ```

After having completed this procedure, the volume remains in the robot but is not loaded in the drive.

2.2.2 Hardware Configuration for a Media Loader

The following procedure applies to the installation and testing of a media loader. Configuration and testing of MDMS to manage a media loader is described in Chapter 4.

2.2.2.1 Hardware Installation

During this procedure, refer to the drive-specific hardware installation information for details. This procedure provides only the basic steps necessary to later configure MDMS against the drive:

1. 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.
3. Make note of the drive name after you complete your connection procedure. You will need this, especially if you are going to use the Media Robot Utility software.

2.2.2.2 Test the Hardware

With a media loader, use a utility such as the Media Robot Utility (MRU) or use the front panel of the drive and the OpenVMS system to test the connection between the OpenVMS system and drive.

Note

Before you perform this procedure, you must have a compatible, initialized volume in the robot. Use this initialized volume to perform this test.

Use the following procedure to test the hardware connection between the tape loader and the OpenVMS system:

1. Verify the changer device name and the drive names.

   Issue 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:

   ```
   $ SHOW DEVICE [drive-name[:]
   ```

   The following is the MRU command for examining the robot:

   ```
   $ ROBOT SHOW ROBOT [robot_name]
   ```

   If you have problems with the drive name, refer to the drive-specific hardware documentation for remedial action.
2. Place a volume in a magazine; place the magazine into the drive.
3. Load a volume into the robot by using the front panel controls or, if you are using MRU, by issuing the ROBOT SHOW and ROBOT LOAD command:

   ```
   $ ROBOT SHOW SLOT 0-n [ROBOT robot_name]
   $ ROBOT LOAD SLOT slot_num [ROBOT robot_name]
   ```
4. After loading the volume, check the OpenVMS system connection to the drive by mounting and dismounting the volume.

   Mount the volume with the OpenVMS MOUNT/FOREIGN command:

   ```
   $ MOUNT/FOREIGN/NOASSIST drive_name[:]
   ```

   Dismount the volume with the OpenVMS DISMOUNT command. Be sure to include the / NOUNLOAD qualifier:

   ```
   $ DISMOUNT/NOUNLOAD drive_name[:]
   ```
5. Remove the volume from the drive by using the front panel controls or, if you are using MRU, by issuing the ROBOT UNLOAD command.

```
$ ROBOT UNLOAD SLOT slot_number [ROBOT robot_name]
```

After having completed this procedure, the magazine with the volume remains in the media loader, but it is not loaded in the drive.

2.2.3 Hardware Configuration for an Operator Attended Drive

This procedure applies to the installation and testing of an operator attended drive. The configuration and testing of MDMS to manage the operator attended drive is described in Chapter 4.

2.2.3.1 Hardware Installation

During this procedure, refer to the drive-specific hardware installation information for details. This procedure provides only basic steps necessary to later configure MDMS against the drive:

1. Install and connect the hardware. Apply power to the drive.
2. If you are using an HSx controller, configure it to allow the OpenVMS system to communicate with the drive.
3. Make note of the drive name after you complete your connection procedure.

2.2.3.2 Test the Hardware

Use the following procedure to test the hardware installation and connection with the OpenVMS system:

1. Verify the drive names.

   Issue the DCL SHOW DEVICE command. If there is a problem with drive name or the connection, it becomes apparent here.

   ```
   $ SHOW DEVICE [drive-name[:]
   ```

   If you have problems with the drive name, refer to the drive-specific hardware documentation for remedial action.
2. Load a volume into the drive.
3. After loading the volume, check the OpenVMS system connection to the drive by mounting and dismounting the volume.

   Mount the volume with the OpenVMS MOUNT/FOREIGN command.

   ```
   $ MOUNT/FOREIGN/NOASSIST drive_name[:]
   ```

   Dismount the volume with the OpenVMS DISMOUNT command.

   ```
   $ DISMOUNT drive_name[:]
   ```
4. Remove the volume from the drive.

2.3 Quotas and OpenVMS Operating System Classes

Before beginning the installation process:

1. Log in to the SYSTEM account.

2. Enable all privileges.
3. Verify that the required quotas and OpenVMS classes are present.

2.3.1 Required Quotas

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 SLS software.

More information:
Refer to the *OpenVMS Upgrade and Installation Procedures* manual for further information.

2.3.2 Required OpenVMS Operating System Classes

Verify that the following OpenVMS operating system classes are resident on the system disk:

- Network support
- Programming support
- Secure user's environment
- System programming environment
- Utilities
- OpenVMS required save set

How to verify:
For information about verifying these classes, refer to the *Guide to Setting Up an OpenVMS System*.

2.4 Hardware Requirements

For a detailed list of supported hardware for the SLS, Version 2.9J product, see the Storage Library System for OpenVMS Software Product Description/Software Support Addendum. The following are the minimum hardware requirements for installing SLS, Version 2.9J software:

- A VAX™, MicroVAX™, VAXstation™, or Alpha™ system
- Four megabytes of memory for VAX, or sixteen megabytes of memory for Alpha
- One or more tape devices or optical disks, for stand-alone systems
- One disk, such as a Digital RD-™ or RA-™ series disk
- Disk space:

  Use the steps in Table 2–2 to ensure there is enough disk space to install SLS server or client software.

Table 2–2 Verifying Dsik Space Requirements

Stage	Action		
1.	Verify that there are enough free blocks on the disk where you are installing SLS software by entering the command: `$ SHOW DEVICE SYS$SYSDEVICE`		
	IF you are installing . . .	**RDF**	**THEN you need . . .**
	Server software	Yes	10,800 peak blocks during installation 8,400 net blocks after installation (permanent)
		No	10,200 peak blocks during installation 7,800 net blocks after installation (permanent)
	Client software	Yes	10,800 peak blocks during installation 8,400 net blocks after installation (permanent)
		No	10,200 peak blocks during installation 7,800 net blocks after installation (permanent)
2.			Note the disk name where SLS will be installed.

Warning

If your system is monitored by the Audit facility, then you must consider the system disk free block requirements. To determine this requirement, enter the command:

```
$ SHOW AUDIT/ALL
```

2.5 Client and Server Definitions

This section introduces SLS's concepts of client software and server software. You need to understand these distinctions before you install the SLS software on your system.

2.5.1 Server Software

SLS server software gives users access to the full functionality of SLS software. SLS provides this access at the node or OpenVMScluster system where the SLS software is executing. You install SLS server software on the server node.

SLS server software manages the:

- Media database-A set of RMS files that describe all removable storage media known to SLS software. These files are part of the Media and Device Management Services for OpenVMS (MDMS) portion of the SLS product. For detailed information about the MDMS databases, see the *Media and Device Management Services for OpenVMS Guide to Operations*.

  A node or OpenVMScluster system executing SLS server software provides media management services both for itself and for any client nodes connected to it. The media database contains information about volume names, locations, magazines, tape jukeboxes, and pool access authorization.

- SLS history files-Information is recorded at the local node about user files backed up or archived under SLS control.

Requirement:
To use the SLS product, you must have at least one SLS server license in the network.

2.5.2 Client Software

SLS client software:

- Provides backup scheduling and archiving services for the node or OpenVMScluster system on which it is executing
- Acts as a client of another node executing SLS server software when it requires media management services
- Provides users with full functionality when a node or OpenVMScluster system executing SLS client software is communicating with another node or OpenVMScluster system executing SLS server software

Similar to the SLS server software, SLS client software also maintains history files in which it records information about user files backed up or archived locally under its control.

The node executing SLS client software communicates using DECnet software to a server node running SLS server software. The server node maintains the media database.

Requirement:
To use media management services, each SLS client node must be connected to a server node executing SLS server software. Any number of SLS client nodes may be connected to a single SLS server node.

2.6 Software Requirements

The following software is required to install either SLS Version 2.9J server or client software:

- OpenVMS VAX operating system version 7.3 and OpenVMS Alpha operating system versions 7.3-2 & 8.2.
- DECnet™ communications software, Version 5.4 or later.

2.6.1 Server Software Requirements

Before you install the server software, check to see that the DECnet software and the OpenVMS Queue Manager are running. If they are not running, follow the steps in Table 2–3 to start them.

Table 2–3 How to Start DECnet and OpenVMS Queue Manager

Stage	Action
1.	Start the DECnet software. Enter the following command at the DCL prompt: `$ @SYS$MANAGER:STARTNET`

Table 2–3 How to Start DECnet and OpenVMS Queue Manager

2. Start the OpenVMS Queue Manager. Enter the following command at the DCL prompt:

```
$ START/QUEUE/MANAGER
```

2.6.2 Client Software Requirements

Before you install client software, the node on which you intend to install the SLS client software must be registered on the SLS server node. If it is not, the IVP fails. For more information, Section 4.3.3 Authorizing SLS Volume Database and Pool Access in Chapter 4.

2.6.3 Optional Software

- DCSC (Digital Cartridge Server Component)

  If you have a StorageTek Automated Cartridge Server (ACS[TM]), you must install the DCSC software and load the SLS-ACS license to use the ACS with SLS software.

  License name:
  SLS-ACS

- RDF (Remote Device Facility)

  SLS, Version 2.9J software provides optional installation of Remote Device Facility (RDF) software. During installation of the SLS software, you are asked if you want to access tape devices on a remote system or if you want to allow remote backups to tape devices on this system. RDF software is installed automatically if you answer YES to either question.

Note

RDF is not supported on OpenVMS Alpha V8.2 and these questions are not asked while installing SLS on OpenVMS Alpha V8.2.

2.7 Shutting Down the Previous Version of SLS Software

If you are running a previous version of SLS, then shut down SLS on all nodes in the cluster before installing the new version of SLS server or client software.

Shut down the SLS software with the following command:

```
$ @SLS$SYSTEM:SLS$SHUTDOWN
```

Note

Due to changes in functionality and database structure, SLS software versions 2.5B and 2.6 are incompatible with SLS software version 2.4A or earlier. If you are an existing SLS customer, and upgrade SLS on your server node to version 2.5 or higher, then you must also upgrade each of your client nodes to version 2.5 or higher. This requirement results from a change in the format of the common magazine database in SLS software version V2.5B.

You might also need to check the batch queues to see if any SLS jobs exist. If so, you need to delete the existing SLS jobs from the batch queues.

2.8 Registering the SLS Software License

If you are installing SLS server or client software for the first time, you must register the SLS software license.

The following sections describe:

- License and kit names
- How to unload a previous license
- How to register the server or client software license

2.8.1 License and Kit Names

Refer to Table 2–4 for the license and kit names for the server or client software.

Table 2–4 Software License Name

Software Type	License Name	Software Kit Name
Server	SLS	SLS29
Client	SLS-REMOTE	SLSCLIENT29
ACS	SLS-ACS	None. Although you load the ACS license separately, SLS support for ACS installs as part of the standard installation process.

2.8.2 Unloading the Current SLS License

If your existing SLS license has a termination date, follow the steps in Table 2–5 to unload the current SLS software license. After completing this procedure, you may register the new SLS server or client software license.

Table 2–5 How to Unload the Server or Client Software License

Step	Action	Enter the command for the appropriate software type:

1.	Obtain the authorization key value for the previous SLS software license.	Server:
		`$ LICENSE LIST SLS/FULL`
		Client:
		`$ LICENSE LIST SLS-REMOTE/FULL`

	Examine the display for a license with a status of "Active" and write down the value associated with the key authorization field.	

Table 2–5 How to Unload the Server or Client Software License

2.	Cancel the previous SLS software license.	Server:
		`$ LICENSE CANCEL SLS/AUTHORIZATION=key_value -`
		`/TERMINATION=YESTERDAY`
		Client:
		`$ LICENSE CANCEL SLS-REMOTE/AUTHORIZATION -`
		`=key_value /TERMINATION=YESTERDAY`

3.	Disable the previous SLS software license.	Server:
		`$ LICENSE DISABLE SLS/AUTHORIZATION=key_value`
		Client:
		`$ LICENSE DISABLE SLS REMOTE/AUTHORIZATION -`
		`=key_value`

4.	Unload the previous SLS software license.	Server:
		`$ LICENSE UNLOAD SLS`
		Client:
		`$_LICENSE_UNLOAD_SLS-REMOTE`

2.8.3 Registering Your New SLS License

If you need to register the new SLS license, follow the steps in Table 2–6 to register your new SLS license.

Where to find registration information:

The license registration information is contained in the Product Authorization Key (PAK). The PAK is a certificate shipped with the SLS package.

Table 2–6 How to Register your SLS Server or Client License

Step	**Action**	**Enter the command for the appropriate software type:**

1.	Enter the LICENSE REGISTER command with the product name and a dash.
	Important:
	Enter a dash at the end of each command in steps 1 through 8.
	Remember:
	To use SLS with a StorageTek ACS, you must also register the SLS-ACS license.
		Server:
		`$ LICENSE REGISTER SLS -`
		Client:
		`$ LICENSE REGISTER SLS REMOTE -`

2.	Enter the /ISSUER qualifier information, assigning the value DEC between quotation marks.	Server or Client: `_$ /ISSUER="DEC" -`

Table 2–6 How to Register your SLS Server or Client License

3.	Enter the /AUTHORIZATION qualifier information, assigning it the value from the AUTHORIZATION NUMBER entry of the PAK.	Server or Client: `_$ /AUTHORIZATION=xxxxxxx -`
4.	Enter the /PRODUCER qualifier information,assigning the value DEC in quotes.	Server or Client: `_$ /PRODUCER="DEC" -`
5.	Enter the /UNITS qualifier information,assigning the value from the UNITS entry of the PAK.	Server or Client: `_$ /UNITS=nn -`
6.	Enter the /DATE qualifiers information, assigning the product's release date value from the PRODUCT RELEASE DATE entry of the PAK.	Server or Client: `_$ /DATE=dd-mmm-yyyy -`
7.	Enter the /AVAILABILITY qualifier information, assigning the value from the AVAILABILITY TABLE CODE entry of the PAK.	Server or Client: `_$ /AVAILABILITY=x -`
8.	Enter the /OPTIONS qualifier information, assigning the value from the KEY OPTIONS entry of the PAK.	Server or Client: `_$ /OPTIONS=xxxxxx -`
9.	Enter the /CHECKSUM qualifier information, assigning the value from the CH entry of the PAK. Important: Do NOT end the entry with a dash.	Server or Client: `_$ /CHECKSUM=1-xxxx-xxxx-xxxx-xxxx`

10.	Invoke the LICENSE LOAD command with the product name.	Server:
		`$ LICENSE LOAD SLS`
		Client:
		`$ LICENSE LOAD SLS-REMOTE`

Note

Error messages during installation:

Users may need to refer to the MDMS Version 2.9J Installation Guide for trouble shooting during the SLS 2.9J installation.

3

Installing the SLS Software

This chapter describes how to install SLS, Version 2.9J software. For an example server installation, see *Appendix , SLS Server Installation Log File*.

Note

Recommendation:
We recommend you perform a backup of your system disk before you install SLS software. Refer to the *OpenVMS Backup Utility Manual* for more details.

Note

Note: Version Compatibility
SLS Version 2.9J software (client and server) is not compatible with any previous versions of SLS prior to version 2.5. If any node in a client-server configuration is running a version later than 2.5, then all nodes in the configuration must be running the same version or higher. An exception can be made if none of the SLS nodes serve a SLS$MAGAZINE_MASTER_FILE.DAT database.

3.1 Before You Begin-Questions and Answers

Before beginning the installation procedure, you may want to go over the following questions and answers:

Table 3–1 Installation Questions and Answers

Q:	How long will the installation procedure take?
A:	The installation procedure will take 15-30 minutes on a VAX system and 5-10 minutes on an Alpha system.
Q:	How can I get help?
A:	VMSINSTAL provides additional information about the decisions required to respond to the installation prompts. To get help at any installation prompt, enter < ? > and press <Return>.
Q:	Where can I find an example of the installation procedure?
A:	*Appendix, SLS Server Installation Log File* contains an example of an installation procedure for the server software.

Table 3–1 Installation Questions and Answers

Q: How can I abort the installation procedure?

A: Enter <Ctrl/Y>.

Note:
If you abort the VMSINSTAL procedure, you must restart the installation from the beginning.

3.2 Installing the SLS Software

Follow the steps in Table 3–2 to install SLS server or client software.

Table 3–2 How to Install the SLS Server or Client Software

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1.</td><td>Load and mount the media that contains the software kit.</td></tr>
<tr><td>2.</td><td>Invoke the VMSINSTAL procedure. Depending upon the type software you are installing, enter the one of the following commands:
<table>
<tr><th>IF you are installing the . . .</th><th>THEN enter the command:</th></tr>
<tr><td>Server software</td><td>$@SYS$UPDATE:VMSINSTAL SLSJ029 location: OPTIONS N</td></tr>
<tr><td>Client software</td><td>$@SYS$UPDATE:VMSINSTAL SLSCLIENTJ029 location: OPTIONS N
Where:
location is the tape device (or directory) that contains the software kit save set. OPTIONS N is an optional parameter that tells SLS to ask you to choose Release Notes options. The default behavior is to move the Release Notes to SYS$HELP.</td></tr>
</table>
</td></tr>
<tr><td>3.</td><td>If you see the following messages:
%VMSINSTAL-W-ACTIVE, The following processes are still active
.
.
.
* Do you want to continue anyway [NO]?
Enter Y and press <Return>.</td></tr>
</table>

Table 3–2 How to Install the SLS Server or Client Software

4. Confirm the state of the system backup.

```
* Are you satisfied with the backup of your system disk [YES]?
```

Press <Return>.

HP recommends you perform a full backup of your system disk before installing any software product.

5. If you did not include the location for the software to be mounted in the VMSINSTAL command line, identify the location for the software to be mounted.

```
* Where will the distribution volumes be mounted: SLS$KITS:
```

You may specify a tape drive, a local disk, or a remote disk. Enter the appropriate information and press <Return>.

6. Respond to the Release Notes option.

```
Release notes included with this kit are always copied to SYS$HELP.

Additional Release Notes Options:
1.  Display release notes
2.  Print release notes
3.  Both 1 and 2
4.  None of the above

* Select option [2]:
```

Enter the number of the desired option and press <Return>.

7. Continue the installation.

```
* Do you want to continue the installation [NO]?
```

Enter Y and press <Return>.

8. If SLS is already installed on this system, verify that you want to install this version of SLS software.

```
The installation procedure displays the following message:

Warning:

SLS V2.9J (client and server) is not compatible with any previous ver-
sions of SLS prior to V2.5.   If any node in a client-server configura-
tion is running a version later than V2.5, then all nodes in the
configuration must be running that same version or higher. An exception
can be made if none of the SLS nodes serve a
SLS$MAGAZINE_MASTER_FILE.DAT database.

* Do you want to continue with this installation [No]? Y
```

System nodes running SLS Version 2.9J software can interact with nodes running SLS Version 2.5 software. Enter Y and press <Return> if you are upgrading all nodes to SLS Version 2.9J or if some nodes are running SLS Version 2.5. If your configuration involves nodes running SLS Version 2.4 software or earlier, press <Return> to terminate the installation.

Table 3–2 How to Install the SLS Server or Client Software

9. Select to run the Installation Verification Procedure (IVP).

```
* Do you want to run the IVP after the installation [YES]?
```

Press <Return> to run the IVP, else enter No and press <Return>.

Recommendation:
Running the IVP procedure is recommended.

10. Confirm the PAK registration.

```
* Does this product have an authorization key registered and loaded?
```

Enter Y and press <Return>.

If you did not register and load the license, as explained in *Chapter 2, Registering the SLS Software License*, abort the installation and do so before continuing.

11. Purge old SLS software files if you want to recover storage space on your disk.

Custom files:
If you are upgrading from a previous version of SLS software and you have files in the SLS$ROOT:[CUSTOM] directory, these files will be placed in a directory created by the VMSINSTAL procedure. Refer to <REFERENCE>(old_serv_remove_cust).

```
* Do you want to purge files replaced by this installation [YES]?
```

Press <Return> to purge old files, else enter No and press <Return>.

12. Client software only:
You need to know the name of the SLS server node or the OpenVMScluster system alias name.

```
* Are you ready to proceed? [YES]?
* What is the name or VMScluster system alias of the SLS server node:
```

Enter the name or OpenVMScluster system alias name of the SLS server node.

13. RDF (Remote Device Facility) software:

If you want to use the RDF software, then respond Yes to the following queries:

```
* Do you want to backup disks on this system to a drive on a remote
system?
```

Enter Yes to use RDF client functionality, else enter No.

```
* Do you want to allow remote backups access to the tape drives on this
system?
```

Enter Yes to use RDF server functionality, else enter No.

Note:
RDF is not supported on OpenVMS Alpha V8.2 and these questions are not asked while installing SLS on OpenVMS Alpha V8.2.

Table 3–2 How to Install the SLS Server or Client Software

14. Specify the disk device name where SLS software will reside.

```
* Enter the name of the disk device the SLS software will use [DUA0:]:
```

Enter the name of the device you noted in *Chapter 2*, *Hardware Requirements* and press <Return>.

15. Enter the directory name where SLS software will reside:

```
* Enter the name of the directory the SLS software will use
[SLS$FILES]:
```

Enter the directory name and press <Return>.

16. Specify the directory names where the summary and log files will reside.

```
* Where to place summary files from system backups
[SLS$ROOT:[SYSBAK.SUMMARY_FILES]]:

* Where to place intermediate history files
[SLS$ROOT:[SYSBAK.TEMP_HISTORY]]:

* Where to place maintenance log files
[SLS$ROOT:[SYSBAK.MAINTENANCE_LOGS]]:

* Where to place system backup log files
[SLS$ROOT:[SYSBAK.SYSBAK_LOGS]]:
```

Choose one of the following options in response to the above queries:

- Press <Return> to choose the default directory location [ ].
- Enter **SLS$ROOT:[SYSBAK]** to retain the location of SLS Version 2.1 or earlier.
- Enter the directory location of your choice. If you choose to provide a different directory location, the SLS installation will take somewhat longer than usual because it will need to move the files.

17. Select to automatically move existing log and summary files.

```
If you have been using earlier versions of SLS software, there will be
log files and summary files resident in SLS$SYSBAK from previous system
backups. This installation procedure has the ability to automatically
distribute these old files into the directories you have specified.

* Do you want to automatically move these old files? [Y]?
```

Press <Return> to move the files, else enter No to retain the files in SLS$SYSBAK.

18. Identify the SLS manager account UIC.

If you see the following message:

```
* Enter the UIC for the SLS manager account. [[1,21]]
```

Press <Return>, else, continue.

Table 3–2 How to Install the SLS Server or Client Software

19. Enter and verify the SLS account password.

    Restriction:
    The password must be at least 15 and no more than 31 characters.

    ```
    * Password:
    * Verify password:
    ```

20. Update quotas and flags for the SLS account.

    1. The installation procedure automatically updates the SLS account process quotas if the SLS account already exists. (See <REFERENCE>(apx_acct_auth) for detailed information about quotas.)

       When you see the following message:

       ```
       * Do you wish to update these quotas now [YES]?
       ```

       Updating quotas is recommended. Enter the quota values and press <Return>.

       Result:
       Updating the SLS process account quotas causes the Authorize utility to run.

    2. Update flags for the SLS account.

       Recommendation:
       When you see the following message:

       ```
       * Do you wish to update these flags now [YES]?
       ```

       Updating flags is recommended. Enter the flag values and press <Return>.

       Result:
       Choosing to update the flags causes the Authorize utility to run.

21. At this point, the installation proceeds without further operator intervention.

3.3 Configuring SLS in a Mixed Architecture Environment

HP recommends configuring according to the approach outlined in <REFERENCE>. The steps outlined in Table 3–3 implement that approach.

This configuration assumes two installations: a primary installation for which all files and directories will be used, and a secondary installation for which only architecture-compatible executables or other nonshared files are located.

With this installation, systems with logicals that point to shared locations must be defined to point to the primary installation file locations. This procedure describes how to change key logical definitions to point the primary installation files.

3.3.1 How to Configure for a Mixed-Architecture Environment

Table 3–3 shows the steps for configuring in a mixed architecture OpenVMS Cluster system

Table 3–3 How to Configure in a Mixed Architecture OpenVMS Cluster System

Step	Action
1.	Edit the SYS$MANAGER:TAPESTART.COM file logical on the secondary system so they point to the primary installation • The SLS master database location **PRIMAST** • System backup files **SLS$SYSBACK_LOGS** **SLS$MAINTENANCE_LOGS** **SLS$SUMMARY_FILES** **SLS$TEMP_HISTORY** • SLS history files **HISDIR_*n***
2.	Edit SLS$SYSTEM:LOADER.COM on the secondary system to change the definition of the logical that points to the Pool Authorization and user history update parameters **SLS$PARAMS**
3.	Shutdown and restart on the secondary system so the new logical definitions can take effect.

Table 3–4 shows logical definitions on both VAX and Alpha systems in a mixed architecture OpenVMS Cluster. In this particular configuration, the VAX system installation is the primary, the Alpha the secondary. The logicals on the Alpha system point to shared files in the VAX system directory.

Table 3–4 Comparing Logical Definitions Between VAX and Alpha Systems

VAX System Logical Definitions	Alpha System Logical Definitions
VAXSYS>SHOW LOGICAL SLS$* SLS$CUSTOM = SLS$ROOT:[CUSTOM] SLS$DATA = SLS$ROOT:[DATA.VAXSYS] SLS$DATAC = SLS$ROOT:[DATA] SLS$DEFSBACLASS = FOREVER <- insert archive class here SLS$HIST_$ALL$ = GENERIC =GENERIC_RMU_HISTORY SLS$HIST_$BACKUP$ = GENERIC SLS$HIST_$RMUBACKUP$ = GENERIC_RMU_HISTORY SLS$HIST_GENERIC = SLS$ROOT:[HIST] SLS$HIST_GENERIC_RMU_HISTORY = SLS$ROOT:[HIST.DATABASE_HISTORY] SLS$LABEL = _NL: SLS$LABEL_TEMPLATE = SLS$SYSTEM:PRINTED_LABEL_V21.TEM- PLATE SLS$MAILBOX = MBA71: SLS$MAINTENANCE_LOGS = SLS$ROOT:[SYS- BAK.MAINTENANCE_LOGS] SLS$MASTER = SLS$ROOT:[PRIMAST] SLS$OFFSITE_DATE = 0 SLS$ONSITE_DATE = 0 SLS$PARAMS = SLS$ROOT:[PARAMS] SLS$ROOT = 1DUA3:[VAX.SLS$FILES.] SLS$STARTUP_P1 = REBOOT SLS$SUMMARY_FILES = SLS$ROOT:[SYSBAK.SUMMARY_FILES] SLS$SYSBAK = SLS$ROOT:[SYSBAK] SLS$SYSBAK_LOGS = SLS$ROOT:[SYSBAK.SYSBAK_LOGS] SLS$SYSTEM = SLS$ROOT:[CUSTOM] = SLS$ROOT:[SYSTEM] SLS$TEMP_HISTORY = SLS$ROOT:[SYSBAK.TEMP_HISTORY] SLS$VAULT = DPS	ALPHASYS> SHOW LOGICAL SLS$* SLS$CUSTOM = SLS$ROOT:[CUSTOM] SLS$DATA = SLS$ROOT:[DATA.ALPHASYS] SLS$DATAC = SLS$ROOT:[DATA] SLS$DEFSBACLASS = FOREVER <- insert archive class here SLS$HIST_$ALL$ = GENERIC = GENERIC_RMU_HISTORY SLS$HIST_$BACKUP$ = GENERIC SLS$HIST_$RMUBACKUP$ = GENERIC_RMU_HISTORY SLS$HIST_GENERIC = 1DUA3:[VAX.SLS$FILES.HIST] SLS$HIST_GENERIC_RMU_HISTORY = $1$DUA3:[VAX.SLS$FILES.HIST.DATABASE_ HISTORY] SLS$LABEL = _NL: SLS$LABEL_TEMPLATE = SLS$SYSTEM:PRINTED_LABEL_V21.TEMPLATE SLS$MAILBOX = MBA36: SLS$MAINTENANCE_LOGS = $1$DUA3:[VAX.SLS$FILES.SYS- BAK.MAINTENANCE_LOGS] SLS$MASTER = 1DUA3:[VAX.SLS$FILES.PRIMAST] SLS$OFFSITE_DATE = 0 SLS$ONSITE_DATE = 0 SLS$PARAMS = 1DUA3:[VAX.SLS$FILES.PARAMS] SLS$ROOT = 1DUA3:[ALPHA.SLS$FILES.] SLS$STARTUP_P1 = REBOOT SLS$SUMMARY_FILES = 1DUA3:[VAX.SLS$FILES.SYS- BAK.SUMMARY_FILES] SLS$SYSBAK = SLS$ROOT:[SYSBAK] SLS$SYSBAK_LOGS = 1DUA3:[VAX.SLS$FILES.SYS- BAK.SYSBAK_LOGS] SLS$SYSTEM = SLS$ROOT:[CUSTOM] = SLS$ROOT:[SYSTEM] SLS$TEMP_HISTORY = $1$DUA3:[VAX.SLS$FILES.SYS- BAK.TEMP_HISTORY] SLS$VAULT = DPS

4

After Installing SLS

After installing the SLS server or client software, you must prepare it for configuration. This chapter explains how to do the following post installation activities:

- Verifying the installation
- Making the SLS software available on a systemwide basis
- Defining a network object on an OpenVMScluster system (server software only)
- Upgrading SLS to comply with changes to OpenVMS version
- Upgrading from a previous version of SLS software
- Configuring media and device support for SLS
- Configuring SLS for Remote Device Facility (RDF) support

4.1 Verifying the Installation

The Installation Verification Procedure (IVP) may be executed as part of the installation procedure by responding YES to the question

```
* Do you want to run the IVP after the installation [YES]?
```

If you did not execute the IVP during the installation procedure, then execute it immediately after installing the SLS software by entering the command:

```
$ @SYS$TEST:SLS$IVP
```

Requirements:

- The SLS software must be running on the system where the IVP is executed.
- When executing the IVP on a SLS client node, the client node must have access to the SLS server node.

4.2 Editing the Systemwide Command Files

To ensure SLS functions properly, you need to edit the system startup procedure and the system login file.

4.2.1 Automatically Starting SLS

To ensure the SLS software starts up each time the system is booted, edit the system startup procedure, SYS$MANAGER:SYSTARTUP_V5.COM or SYS$MAN-
AGER:SYSTARTUP_VMS.COM based on the OpenVMS Version. Add the following line:

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

Note

Include the startup line after the STARTNET command in the SYSTARTUP_x.COM file.

If you are using DECnet-Plus, make sure the SLS$STARTUP command entry appears before the call @SYS$MANAGER:NET$NCP_APPLICATIONS.COM.

4.2.2 Defining Global Symbols for SLS Users

To define symbols for SLS software users, edit the system login file, SYS$MANAGER:SYLOGIN.COM. Add the following line:

```
$ @SYS$MANAGER:SLS$TAPSYMBOL
```

Note

Include the symbol line in a section of the SYLOGIN.COM file that is executed by all accounts. If SLS$TAPSYMBOL.COM is not run, users will be unable to access the SLS menus.

4.3 Preparing SLS Server Software

If you have installed SLS server software for the first time, then you must:

- Define a network object on your OpenVMScluster system nodes.
- Authorize SLS volume database and pool access.

4.3.1 Defining a DECnet Network Object on an OpenVMScluster System

Follow the steps in Table 4–1 to define the SLS$DBX NCP object.

Requirements:
If you intend to implement the high-availability features of SLS software, all nodes that participate in the OpenVMScluster system alias must have the SLS$DBX NCP object defined.

Table 4–1 How to Define the SLS$DBX NCP Object

Step	Action
1.	From an account with BYPASS privilege, on the node on which SLS software was just installed, invoke the Network Control Program (NCP) utility by entering the command: `$ RUN SYS$SYSTEM:NCP`

Table 4–1 How to Define the SLS$DBX NCP Object

2. Obtain the password for the SLS account:

```
NCP> SHOW OBJECT SLS$DBX CHARACTERISTICS
```

Result:

```
Object Volatile Characteristics as of  05-Jan-2005 13:08:15

Object = SLS$DBX

Number               = 0
File id              = SYS$SYSTEM:SLS$FIND_DB_NODE.EXE
User id              = SLS
Password             = password
```

Note:
The password is displayed only when the BYPASS privilege is enabled.

3. Exit the NCP utility:

```
NCP> EXIT
```

4. Invoke the SYSMAN utility:

```
$RUN SYS$SYSTEM:SYSMAN
```

5. Set the SYSMAN environment to include the OpenVMScluster system nodes that participate in the DECnet OpenVMScluster system alias.

Participating nodes:

The participating nodes in an OpenVMScluster can be all of the nodes or only some of the nodes. To identify the participating nodes, use one of the following commands:

- To identify all nodes . . .

```
SYSMAN> SET ENVIRONMENT /CLUSTER
```

- To identify some specific nodes . . .

```
SYSMAN> SET ENVIRONMENT /NODE=(node_1,node_2,...node_n)
```

6. Define the SLS$DBX object with this command:

```
SYSMAN>DO MCR NCP DEFINE OBJECT SLS$DBX NUMBER 0 USER SLS -
_SYSMAN> FILE SYS$SYSTEM:SLS$FIND_DB_NODE.EXE PASSWORD password
```

Rule:
The password entered with this command must be the same as the password used for the SLS account during installation.

7. Read the SLS$DBX object on the node:

```
SYSMAN> DO MCR NCP SET KNOWN OBJECT ALL
```

Table 4–1 How to Define the SLS$DBX NCP Object

8. Exit the SYSMAN utility:

```
SYSMAN>EXIT
```

4.3.2 Defining the DECnet Phase V Network Object for SLS

This section lists the commands that register the SLS$DBX object on nodes running DECnet, Phase V. You can use SYSMAN if all nodes in a OpenVMS Cluster to issue these commands at first. If only some of the nodes run DECnet, Phase V, you must issue these commands on those nodes individually.

4.3.2.1 Issue the Commands on Currently Running Nodes

Once you have installed SLS, you need to register the SLS$DBX object on the node. Issue the commands shown:

```
RUN SYS$SYSTEM:NCL
NCL> CREATE NODE 0 SESSION CONTROL APPLICATION  SLS$DBX
NCL> SET NODE 0 SESSION CONTROL APPLICATION  SLS$DBX NODE SYNONYM TRUE
NCL> SET NODE 0 SESSION CONTROL APPLICATION SLS$DBX ADDRESSES = {NAME =
SLS$DBX}, -
NCL> IMAGE NAME = "SYS$SYSTEM:SLS$FIND_DB_NODE.EXE", USER NAME = "SLS"
```

4.3.2.2 Defining the SLS$DBX network object in the NET$APPLICATION_STARTUP.NCL File

To define the SLS$DBX network object using DECnet-Plus (Phase V), first ensure that you are logged into the SYSTEM account. Then, execute the following command procedure to define the network object.

```
$ @SYS$STARTUP:NET$CONFIGURE ADVANCED
Copyright (c) Hewlett-Packard Development Company, L.P 2005
All rights reserved.

DECnet-Plus for OpenVMS ADVANCED network configuration procedure
This procedure will help you create or modify the management scripts needed
to operate DECnet on this machine. You may receive help about most questions
by answering with a question mark '?'.

You have chosen the ADVANCED configuration option.  This option enables you
to do some specific tailoring of your system's network configuration by
answering some specific questions.

If you do not want to do specific tailoring of your system's network configu-
ration but instead want to quickly configure your system using most of the
default answers, you should invoke NET$CONFIGURE.COM with the BASIC configu-
ration option, ie:
@SYS$MANAGER:NET$CONFIGURE BASIC

* Do you want to continue? [YES] :
Configuration last run by SYSTEM on 05-Jan-2005 22:35:21.23
The following NCL script files have been modified since being created by this
procedure:

SYS$SPECIFIC:[SYSMGR]NET$APPLICATION_STARTUP.NCL;
SYS$SPECIFIC:[SYSMGR]NET$MOP_CLIENT_STARTUP.NCL;

This procedure may create a new version of these scripts or modify informa-
tion contained within them.  In all cases, any previously created NCL scripts
```

```
are renamed to "file.NCL-OLD", so that you may review any of your previous
changes.

Configuration Options for Node NODENM
[0]     Exit this procedure
[1]     Perform an entire configuration
[2]     Change naming information
[3]     Configure Devices on this machine
[4]     Configure Transports
[5]     Configure Timezone Differential Factor
[6]     Configure Event Dispatcher
[7]     Configure Application database
[8]     Configure MOP Client database
[9]     Configure Cluster Alias
[10]    Replace MOP Client configuration

* Which configuration option to perform? [1]: 7
* Do you want to ADD or DELETE an Application? [ADD]:
* What is the name of the Application?: SLS$DBX
* What is the destination type for 'SLS$DBX'? [NAME]:
* What is the destination name for 'SLS$DBX'?: SLS$DBX
* Do you want to specify another application address?[NO] :
* What is the name of the Client for 'SLS$DBX'? :
* What is the Image name for 'SLS$DBX'? :SYS$SYSTEM:SLS$FIND_DB_NODE.EXE
* Incoming Alias for 'SLS$DBX' enabled?[TRUE]:
* Incoming Proxy for 'SLS$DBX' enabled?[TRUE]:
* Outgoing Alias for 'SLS$DBX' enabled?[FALSE]:
* Outgoing Proxy for 'SLS$DBX' enabled?[TRUE]:
* Require node synonym for 'SLS$DBX' enabled?[TRUE]:
* What is the Incoming OSI TSEL for 'SLS$DBX'?:
* What is the User Name for 'SLS$DBX'?[SLS$D$SERVER]: SLS
* What UIC should 'SLS$DBX' use?[[200,200]]: [1,20]
* Rights identifiers for 'SLS'?:
%NET$CONFIGURE-I-MAKEACCOUNT, this procedure creates user account SLS
* Do you want to generate NCL configuration scripts?[YES] :
%NET$CONFIGURE-I-MODCHECKSUM, checksumming NCL management scripts modified by
NET$CONFIGURE
%NET$CONFIGURE-I-CONFIGCOMPLETED, DECnet-Plus for OpenVMS configuration com-
pleted

Configuration Options for Node NODENM
[0]     Exit this procedure
[1]     Perform an entire configuration
[2]     Change naming information
[3]     Configure Devices on this machine
[4]     Configure Transports
[5]     Configure Timezone Differential Factor
[6]     Configure Event Dispatcher
[7]     Configure Application database
[8]     Configure MOP Client database
[9]     Configure Cluster Alias
[10]    Replace MOP Client configuration

* Which configuration option to perform?[0] : 0
```

The above procedure adds the network object SLS$DBX to the file SYS$MANAGER:NET$APPLICATION_STARTUP.NCL. In order for the object to be defined on the running system without rebooting, issue the following command:

```
$ MCR NCL DO SYS$STARTUP:NET$APPLICATION_STARTUP
```

A correctly defined DECnet-Plus object will look similar to the following:

```
Node 0 Session Control Application SLS$DBX
at 2005-01-05-18:37:44.876-04:00Iinf
Characteristics
Client  = <Default value>
Addresses =
{
name = SLS$DBX
}
Outgoing Proxy    = True
Incoming Proxy    = True
Outgoing Alias    = False
Incoming Alias    = True
Node Synonym      = True
Image Name        = SYS$SYSTEM:SLS$FIND_DB_NODE.EXE
User Name         = "SLS"
Incoming OSI TSEL = <Default value>
OutgoingAlias Name= <Default value>
Network Priority            = 0
```

4.3.3 Authorizing SLS Volume Database and Pool Access

You need to run SLS$SYSTEM:SLS$SLSMGR.EXE to authorize access to:

- SLS server database for SLS client nodes
- SLS pools for SLS users

Follow the steps in Table 4–2 to enable client node access to the SLS volume database.

Table 4–2 How to Enable Client Node Access to the SLS Volume Database

Step	Action
1.	On the SLS server node, access the SLS Storage Administrator Menu with the following DCL command: `$ RUN SLS$SYSTEM:SLS$SLSMGR`
2.	Select the Database Access Authorization option from the numeric keypad: Enter < 3 > and press <Return>.
3.	Invoke the record insertion feature from the numeric keypad. Press keypad < 8 > or Press <Insert Here>.
4.	Enter the name of the client node(s) in the Node field and press <Return>. Result: The list of client nodes is displayed with the new entry or entries.
5.	Save the changes to the database: Enter Y and press <Return>.

For more information, see Chapter 6 of the *Media and Device Management Services for OpenVMS Guide to Operations*.

4.4 Configuring SLS to Work With Specific VMS Versions

Note

Storage Library System for OpenVMS, Version 2.9J software is not supported on systems running OpenVMS, Version 7.0 operating system software.

SLS, Version 2.9J operates on systems running OpenVMS VAX Version 7.3, OpenVMS Alpha Versions 7.3-2 and 8.2;and OpenVMS i64 Version 8.2. Three SLS files are dependent on the OpenVMS VAX version running on the system. When SLS is installed, the OpenVMS VAX version is checked, and the correct version of these files is set up for subsequent use. If you upgrade your OpenVMS VAX version after you install SLS, you need to make sure you have the correct version-specific files to match the OpenVMS VAX version.

Note

There are no version-specific files for OpenVMS Alpha. If you are running SLS, Version 2.9J on an Alpha system, you can skip this section.

4.4.1 Version-Specific Files

When SLS, Version 2.9J is installed on your system, the installation procedure copies all files shown in Table 4–3 into the indicated directories. The procedure then copies the three files appropriate for your OpenVMS version to the same name, except that the version number is removed from the file name.

Example:

If you are running VMS Version 5.5-2, then SLS$SYSTEM:BACKUPCMD-V55.CLD is copied to SLS$SYSTEM:BACKUPCMD.CLD. A similar copy is performed for the other two files.

Table 4–3 SLS Files for Specific OpenVMS Versions Operating System Files

Operating System	Files
VMS 5.5	SLS$SYSTEM:BACKUPCMD-V55.CLD SLS$SYSTEM:VMSBUXX-V55.EXE SYS$MESSAGE:SLS$BACKUPMSG-V55.EXE
OpenVMS 6.2	SLS$SYSTEM:BACKUPCMD-V62.CLD SLS$SYSTEM:VMSBUXX-V62.EXE SYS$MESSAGE:SLS$BACKUPMSG-V62.EXE
OpenVMS 7.1	SLS$SYSTEM:BACKUPCMD-V71.CLD
OpenVMS 7.2	SLS$SYSTEM:BACKUPCMD-V72.CLD
OpenVMS 7.3	SLS$SYSTEM:BACKUPCMD-V73.CLD

4.4.2 Upgrading OpenVMS Versions after Installing SLS

If you install SLS, and then later upgrade your OpenVMS software version, you must copy the three SLS version dependent files to cause them to be used. A command procedure has been supplied for this purpose.

Example:
Assume you had installed SLS on a system running VMS Version 6.2. You later upgrade to OpenVMS Version 7.1. You must now execute the following command so that SLS will use the correct OpenVMS Version 7.1 version dependent files:

```
$ @SLS$SYSTEM:SLS$VERSION_CHECK_AND_COPY.COM
```

If you install a version of OpenVMS that is unknown to SLS, this command procedure displays an appropriate error of the following form:

```
%SLS-E-NOVERMATCH, No version specific SLS files exist for OpenVMS Vn.n
```

No files will be copied.

4.5 Upgrading from a Previous Version

If you upgrade SLS software from a previous version, then:

- You need to review the applicability of customized files.
- If you upgrade from SLS Version 2.0 software, you need to retain previous applicable SYS$MANAGER:TAPESTART.COM file symbol assignments.
- If you upgrade from SLS Version 2.1 software, the installation procedure modifies your current SYS$MANAGER:TAPESTART.COM file.

Note

SLS Version 2.9J software is incompatible with versions prior to SLS Version 2.5 software. This consideration applies to systems currently running SLS Version 2.4 software or earlier.

When installed, SLS Version 2.5 through Version 2.9J software converts the previous volume and magazine databases to a new format. All information is maintained in the databases, except that all magazines initially appear as "not imported" in a jukebox.

4.5.1 Reviewing Outdated Files from SLS$CUSTOM

Files placed in SLS$CUSTOM from a previous installation of SLS software may not be compatible with SLS, Version 2.9J software. These files could include patched images or modified command procedures. You need to determine if these files should be removed from SLS$CUSTOM.

Where to find old custom files:
If, during installation, files are found in SLS$CUSTOM, then the installation procedure:

1. Creates a new directory, [SLS$FILES.OLDCUSTOM_n] Each time an installation is performed, the value of n increases. This indicates a new directory has been created for the files in [SLS$FILES.CUSTOM] from the previous installation.
2. Renames and copies all the files in SLS$CUSTOM to the new directory

Recommendation:
Review the files in [SLS$FILES.OLDCUSTOM_n] before placing them in [SLS$CUSTOM]. Files that predictably executed in a previous version of SLS software may be unpredictable with SLS, Version 2.9J software.

4.5.2 Updating the Command File TAPESTART.COM

Follow the steps in Table 4–4 update the new version of command file TAPESTART.COM. For information about media and device management symbols in TAPESTART.COM, see the Media and Device Management Services for OpenVMS Guide to Operations.

Note

This procedure is required only if you upgraded from SLS Version 2.0 software.

Table 4–4 How to Update the Command File TAPESTART.COM

Step	Action
1.	Set default to the directory that contains the file TAPESTART.COM. `$ SET DEFAULT SYS$MANAGER`
2.	Ensure you have your old and new versions of TAPESTART.COM by entering the command: `$ DIRECTORY TAPESTART.*` Note: The previous version of the file TAPESTART.COM is named TAPESTART.OLD by the VMSINSTAL procedure. The new installed version is properly named TAPESTART.COM.
3.	Edit the new TAPESTART.COM file to include the symbol assignments from the TAPESTART.OLD file. For information about configuring media and devices for SLS, see the *Media and Device Management Services for OpenVMS Guide to Operations*.
4.	IF you have an OpenVMScluster node with unique requirements (such as a node with singular access to a tape device): • Place a copy of the TAPESTART.COM file in that node's SYS$SPECIFIC:[SYSMGR] directory. • Set the protection by entering the command: `$ SET PROTECTION=W:RE TAPESTART.COM` • Make the node-specific changes.
5.	Start SLS software by entering the command: `$ @SYS$STARTUP:SLS$STARTUP`

4.6 Configuring Media and Device Support

Once SLS is installed, you need to configure it to use the appropriate media and device support before you can actually save and restore data. This includes determining what physical devices exist for your system and defining appropriate symbols in SYS$MANAGER:TAPESTART.COM to identify them to SLS. For more information on configuring media and device support, see the *Media and Device Management Services for OpenVMS Guide to Operations.*

4.6.1 Configuring MDMS for an Automated Tape Library System

If your storage management environment contains an ATL, you must define it in the MDMS file SYS$MANAGER:TAPESTART.COM. You must then test the configuration to make sure it is correct.

4.6.1.1 Configure MDMS for an ATL

The following procedure requires you to edit the MDMS file named SYS$MANAGER:TAPESTART.COM. You can use any ASCII text editor to accomplish these edits:

1. Identify the primary processing node. Define the symbol PRI to the name of the node or the OpenVMS Cluster alias name executing the MDMS server.

   If you defined the symbol PRI to the OpenVMS Cluster alias, you must define the symbol DB_NODES to the names of the nodes in the OpenVMS Cluster system that you want to allow to be an MDMS servers.
2. Identify the node on which the current TAPESTART.COM file resides. Assign the node name to the NODE symbol.
3. Define the QUICKLOAD symbol to 1 to facilitate unattended backup.
4. Create the automatic tape library (ATL) definition. Use the following example as a guideline:

Note

The robot device name and the tape drive names must be prefixed with a node name.

Example: Defining the ATL

Assign the name meaningful name for the ATL (such as ABS_JUKE) to the TAPE_JUKEBOXES symbol:

```
$ TAPE_JUKEBOXES := "ABS_JUKE"
```

Define the robot device name and the tape drives to the jukebox name you assigned to the TAPE_JUKEBOXES symbol:

```
$ ABS_JUKE := "NODE01::$1$DUA810:,NODE01::$1$MUA43:,NODE01::$1$MUA44:"
```

Where:
- *NODE01::$1$DUA810:* is the node name plus the robot device name
- *NODE01::$1$MUA43:* and *NODE01::$1$MUA44* are the node names plus the names of the drives located in the ATL

5. Define the media triplets. Media triplets are a set of symbols used to associate types of media with drives capable of accessing them and define the media type, density, and corresponding drive name. Use the following example as a guideline:

   Example: Defining the Media Triplets for an ATL

```
$ MTYPE_1  :=  TK85K
$ DENS_1   :=
$ DRIVES_1 :=  $1$MUA43:,$1$MUA44:
```

Note:
During the installation of MDMS, MDMS provides an autoconfiguration utility that creates default media triplets. For each tape drive on the system from which you run the autoconfiguration utility, the utility creates a media triplet for that drive. Note that these are suggested defaults, and you can change the suggested default media triplets if you so choose. Refer to

Archive Backup System for OpenVMS Guide to Operations for details about automatically generating media triplets.

6. Shut down and restart MDMS to enable all the changes made to the TAPESTART.COM file. Use the following command:

```
$ @SLS$SYSTEM:SLS$SHUTDOWN
$ @SYS$STARTUP:SLS$STARTUP
```

4.6.1.2 Test the MDMS ATL Configuration

This procedure guides you through testing the configuration for an ATL.

Note

Before you perform the procedure, you must have an initialized volume that is compatible with the drive in the ATL. Use this initialized volume to perform this test.

If you have MRU software, you can take this opportunity to learn more about the nature of the managed media environment. Compare the results of the STORAGE and ROBOT commands, or use MRU to verify the state of the automated media library after issuing a STORAGE command.

Issue STORAGE and ROBOT commands at the DCL prompt.

1. Check the response of the automated tape library to MDMS. Issue the STORAGE SHOW JUKEBOX command:

```
$ STORAGE SHOW JUKEBOX [tape_jukebox_name]
```

   Issue the ROBOT SHOW ROBOT command and compare the results:

```
$ ROBOT SHOW ROBOT [robot_name]
```

2. Add a volume to the volume database with the STORAGE ADD VOLUME command:

```
$ STORAGE ADD VOLUME volume_id
```

3. Verify the volume you just added with the STORAGE SHOW VOLUME command:

```
$ STORAGE SHOW VOLUME volume_id
```

4. Prepare to reply to MDMS requests with the OpenVMS OPCOM REPLY command:

```
$ REPLY/ENABLE=TAPES
```

5. Insert the volume for which you have just created a volume database record into the library. Issue the STORAGE IMPORT CARTRIDGE command:

```
$ STORAGE IMPORT CARTRIDGE volume_id tape_jukebox_name
```

   Translate the location information from the STORAGE IMPORT CARTRIDGE command into a slot location and issue the ROBOT SHOW SLOT command to verify the volume is in the slot:

```
$ ROBOT SHOW [ROBOT robot_name] SLOT [[[slot_number]|[slot_number-
slot_number]],...]
```

6. Load the volume into the drive with the STORAGE LOAD command:

```
$ STORAGE LOAD drive-name volume-id
```

Using the ROBOT SHOW DRIVE command, validate that the volume has been loaded:

```
$ ROBOT SHOW [ROBOT robot_name] DRIVE [[[drv_number]|[drv_number-
drv_number]],...]
```

7. Unload the volume from the drive using the STORAGE UNLOAD command:

   ```
   $ STORAGE UNLOAD VOLUME volume-id
   ```

   Using the ROBOT SHOW SLOT command, validate the volume has been returned to the slot from which it was loaded:

   ```
   $ ROBOT SHOW [ROBOT robot_name] SLOT [[[slot_number]|[slot_number-
   slot_number]],...]
   ```

8. Remove the volume from the library with the STORAGE EXPORT CARTRIDGE command:

   ```
   $ STORAGE EXPORT CARTRIDGE volume-name
   ```

4.6.2 Configuring MDMS for a Media Loader

If your storage management environment contains a media loader, you must define the media loader in the MDMS file SYS$MANAGER:TAPESTART.COM. You must then test the configuration to make sure it is correct.

4.6.2.1 Configure MDMS for a Media Loader

The following procedure requires you to edit the MDMS file named SYS$MANAGER:TAPESTART.COM. You can use any ASCII text editor to accomplish these edits:

1. Identify the primary processing node. Define the symbol PRI to the name of the node or the OpenVMS Cluster alias name executing the MDMS server.

   If you defined the symbol PRI to the OpenVMS Cluster alias, you must define the symbol DB_NODES to the names of the nodes in the OpenVMS Cluster system that you want to act as MDMS servers.

2. Identify the node on which the current TAPESTART.COM file resides. Assign the node name to the NODE symbol.

3. Define the QUICKLOAD symbol to 1 to facilitate unattended backup.

4. Create the media loader definition. Use the following example as a guideline:

   Example: Defining a Media Loader
   Define a symbol ABS_JUKE to the names of the robot device and tape drive of the automated media library system. The following example shows how define the TAPE_ JUKEBOXES symbol:

   Assign a meaningful name (such as ABS_JUKE) to the TAPE_ JUKEBOXES symbol:

   ```
   $ TAPE_JUKEBOXES := "ABS_JUKE"
   ```

   Define the robot device name and the tape drive to the jukebox name defined by the TAPE_JUKEBOXES symbol:

```
$ ABS_JUKE := "NODE01::$1$DUA810:,NODE01::$1$MUA43:
```

Where:
o NODE01::$1$DUA810: is the robot changer name
o NODE01::$1$MUA43: is the node name plus the drive name of the media loader

Note

The robot device name and the tape drive names must be prefixed with a node name.

5. Define the media triplets. Media triplets are a set of symbols used to associate types of media with drives capable of accessing them. Use the following example as a guideline:

   Example: Defining the Media Triplets for a Media Loader

   ```
   $ MTYPE_1  :=  TK85K
   $ DENS_1   :=
   $ DRIVES_1 :=  $1$MUA43:,$1$MUA44:
   ```

   Note:
   During the installation of MDMS, MDMS provides an autoconfiguration utility that creates default media triplets. For each tape drive on the system from which you run the autoconfiguration utility, the utility creates a media triplet for that drive. Note that these are suggested defaults, and you can change the suggested default media triplets if you so choose. Refer to *Archive Backup System for OpenVMS Guide to Operations* for details about automatically generating media triplets.

6. Shut down and restart MDMS to enable all the changes made to the TAPESTART.COM file. Use the following command:

   ```
   $ @SLS$SYSTEM:SLS$SHUTDOWN
   $ @SYS$STARTUP:SLS$STARTUP
   ```

4.6.2.2 Test the MDMS Media Loader Configuration

This procedure guides you through testing the MDMS configuration.

Note

Before you perform the procedure, you must have an initialized volume that is compatible with the drive in the media loader. Use this initialized volume to perform this test.

If you have MRU software, you can take this opportunity to learn more about the nature of the managed media environment. Compare the results of the STORAGE and ROBOT commands, or use MRU to verify the state of the media loader after issuing a STORAGE command.

Issue STORAGE and ROBOT commands at the DCL prompt.

1. Check the response of the media loader to MDMS. Issue the STORAGE SHOW JUKEBOX command:

   ```
   $ STORAGE SHOW JUKEBOX [tape_jukebox_name]
   ```

   Issue the ROBOT SHOW ROBOT command and compare the results:

```
$ ROBOT SHOW ROBOT [robot_name]
```

2. Add a volume to the volume database with the STORAGE ADD VOLUME command:

   ```
   $ STORAGE ADD VOLUME volume_id
   ```

3. Verify the volumes you just added with the STORAGE SHOW VOLUME command:

   ```
   $ STORAGE SHOW VOLUME volume_id
   ```

4. Prepare to reply to MDMS requests with the OpenVMS OPCOM REPLY command:

   ```
   $ REPLY/ENABLE=TAPES
   ```

5. Add a magazine to the media management database by issuing the STORAGE ADD MAGAZINE command:

   ```
   $ STORAGE ADD MAGAZINE magazine_name /SLOTS=n
   ```

6. Place the volume into the magazine and bind it with the STORAGE BIND command:

   ```
   $ STORAGE BIND volume_id magazine_name /SLOT=n
   ```

7. Verify the volume has been bound to the magazine with the STORAGE SHOW MAGAZINE command:

   ```
   $ STORAGE SHOW MAGAZINE [volume_id magazine_name]
   ```

8. Issue the STORAGE IMPORT MAGAZINE command and insert the magazine into the loader:

   ```
   $ STORAGE IMPORT MAGAZINE magazine_name tape_jukebox_name
   ```

9. Load the volume into the drive with the STORAGE LOAD command:

   ```
   $ STORAGE LOAD drive-name volume-id
   ```

   Using the ROBOT SHOW DRIVE command, validate the volume has been loaded:

   ```
   $ ROBOT SHOW [ROBOT robot_name] DRIVE [[[drv_number]],...]
   ```

10. Unload the volume from the drive using the STORAGE UNLOAD command:

    ```
    $ STORAGE UNLOAD VOLUME volume-id
    ```

    Using the ROBOT SHOW SLOT command, validate the volume has been returned to the slot from which it was loaded:

    ```
    $ ROBOT SHOW [ROBOT robot_name] SLOT [[[slot_number]|[slot_number-
    slot_number]],...]
    ```

11. Remove the magazine from the loader with the STORAGE EXPORT MAGAZINE command:

    ```
    $ STORAGE EXPORT MAGAZINE magazine_name]
    ```

4.6.3 Configuring for an Operator Attended Drive

If your storage management environment has an operator attended drive or drives, you must define the drive(s) in the MDMS file SYS$MANAGER:TAPESTART.COM. You must then test the configuration to make sure it is correct.

4.6.3.1 Configure MDMS for an Operator Attended Device

The following procedure requires you to edit the SYS$MANAGER:TAPESTART.COM file. You can use any ASCII text editor to accomplish these edits.

1. Identify the primary processing node. Define the symbol PRI to the name of the node or the OpenVMS Cluster alias name executing the MDMS server.

   If you defined the symbol PRI to the OpenVMS Cluster alias, you must define the symbol DB_NODES to the nodes names in the OpenVMS Cluster system that will be allowed to be the MDMS server should the active MDMS server fail.

2. Identify the node on which the current TAPESTART.COM file resides. Define the NODE symbol to that name.

3. Define the media triplets. Use the following example as a guideline:

   Example: Defining a Media Triplet for an Operator Attended Drive

   ```
   $ MTYPE_1  :=  TK85K
   $ DENS_1   :=
   $ DRIVES_1 :=  $1$MUA560:,$1$MUA561:,$1$MUA562:
   ```

   Note:
   During the installation of MDMS, MDMS provides an autoconfiguration utility that creates default media triplets. For each tape drive on the system from which you run the autoconfiguration utility, the utility creates a media triplet for that drive. Note that these are suggested defaults, and you can change the suggested default media triplets if you so choose. Refer to *Archive Backup System for OpenVMS Guide to Operations* for details about automatically generating media triplets.

4. Decide whether operator intervention and acknowledgment is needed then define the QUICKLOAD accordingly. If you need operator intervention, assign QUICKLOAD to 0. If you do not want operator intervention, assign QUICKLOAD to 1.

5. Configure the Volume Pools. Use the MDMS Administrators Menu. From OpenVMS, issue the SLSMGR command.

   More information:
   For detailed information about the MDMS menus, refer to the *Archive Backup System for OpenVMS Guide to Operations*.

6. Shut down and restart MDMS to enable all the changes made to the TAPESTART.COM file. Use the following command:

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

4.6.3.2 Test the MDMS Configuration

This procedure guides you through testing the MDMS configuration. Issue STORAGE commands at the DCL prompt.

1. Add a volume to the volume database with the STORAGE ADD VOLUME command:

   ```
   $ STORAGE ADD VOLUME volume_id
   ```

2. Verify the volumes you just added with the STORAGE SHOW VOLUME command:

   ```
   $ STORAGE SHOW VOLUME volume_id
   ```

3. Prepare to reply to MDMS requests with the OpenVMS OPCOM REPLY command:

   ```
   $ REPLY/ENABLE=TAPES
   ```

4. Load the volume into the drive with the STORAGE LOAD command:

   ```
   $ STORAGE LOAD drive-name volume-id
   ```

5. Unload the volume from the drive using the STORAGE UNLOAD command:

   ```
   $ STORAGE UNLOAD VOLUME volume-id
   ```

Additional Information:
For additional information about configuring MDMS to work with particular media and drives, see *Archive Backup System for OpenVMS Guide to Operations.*

4.6.4 Configuring Remote Tape or Optical Drives

If you answered "Yes" to either of the remote support prompts during the MDMS installation procedure, then MDMS automatically installed the RDF software.

Table 4–5 describes how to configure the files SLS$ROOT:[TTI_RDEV]CONFIG_*nodename*.DAT and SYS$MANAGER:TAPESTART.COM to enable remote drive selection.

Table 4–5 Enabling Remote Drive Selection

On the system where the tape or optical drive resides	On the system where the data to back up resides
Configure the file SLS$ROOT:[TTI_DEV]CONFIG_*nodename*.DAT to contain the tape drive names. See Section 4.6.4.1	N/A
Configure media triplets in the file SYS$MANAGER:TAPETART.COM and define the drives as local drives (Do not include a node name prefix). See Section 4.6.4.1	Configure media triplets in the file SYS$MANAGER:TAPESTART.COM and define the drives as remote drives (include a node name prefix). See Section 4.6.4.2

4.6.4.1 On The System Where the Tape or Optical Drive Resides

Use the following procedure to configure SLS$ROOT:[TTI_RDEV]CONFIG_*nodename*.DAT and SYS$MANAGER:TAPESTART.COM on the system where the drive resides:

1. Determine which tape or optical drive(s) will receive the backup data.
2. Check the file SLS$ROOT:[TTI_RDEV]CONFIG_*nodename*.DAT to make sure the drives are defined in this file.

The installation procedure creates a drive configuration file named SLS$ROOT:[TTI_RDEV]CONFIG_*nodename*.DAT. In this file, include the drive names you wish to use for remote backup operations.

Example of SLS$ROOT:[TTI_RDEV]CONFIG_*nodename*.DAT:

```
! The format of this configuration file is:
!
!  DEVICE  servable_device  [characteristic1,...]
!----------------------------------------------------------------
device $1$MUA200:  MUA200
device $1$MUA210:  MUA210
device $1$MUA220:  MUA220
```

3. Edit the file SYS$MANAGER:TAPESTART.COM and define media triplets that define the media type, density, and corresponding drive name. Make sure the drive names appear as local drives (do not prefix the drive name with a node name). The drive name must consist of either a physical name or the characteristic name of the drive or drives available for remote backup operations.

   Example of SYS$MANAGER:TAPESTART.COM:

```
! Media triplet definitions
$ MTYPE_8  :=  TK86K
$ DENS_8   :=
$ DRIVES_8 :=  $1$MUA200:
$!
$ MTYPE_9  :=  NODE01TK85K
$ DENS_9   :=  NOCOMP
$ DRIVES_9 :=  $1$MUA210:
$!
$ MTYPE_10 :=  NODE01TK85K_COMP
$ DENS_10  :=  COMP
$ DRIVES_10 := $1$MUA220:
```

   Restriction:
   Media type names cannot exceed 16 characters.

4.6.4.2 On The System Where the Data Resides

On the system that contains the disks with the data you want to back up, add one or more media triplets that define the remote drive names in the SYS$MANAGER:TAPESTART.COM file. To the dirve definition, include the node name where the drive is physically connected. Do not assign the node name to drives that are local to this system.

Example of SYS$MANAGER:TAPESTART.COM:

```
! Media triplet definitions
$ MTYPE_8  :=  TK86K
$ DENS_8   :=
$ DRIVES_8 :=  NODE01::$1$MUA200:
$!
$ MTYPE_9  :=  NODE01TK85K
$ DENS_9   :=  NOCOMP
$ DRIVES_9 :=  NODE01::$1$MUA210:
$!
$ MTYPE_10 :=  NODE01TK85K_COMP
$ DENS_10  :=  COMP
$ DRIVES_10 := NODE01::$1$MUA220:
```

See *Archive Backup System for OpenVMS Guide to Operations* for additional information about creating media triplets and defining remote drives.

4.7 Configuring Remote Device Facility Support

If you requested support for the Remote Device Facility (RDF) during installation, then you need to identify server and client controls.

Note

RDF is not supported on OpenVMS Alpha V8.2.

4.7.1 Adding RDF Client Support

In the SYS$MANAGER:TAPESTART.COM file, add one or more media triplets that describe the remote tape device names.

Important:
It is important to add the remote node name to each device name.

For more information:
Refer to the *Media and Device Management Services for OpenVMS Guide to Operations* for detailed instructions about modifying the SYS$MANAGER:TAPESTART.COM file to allow access to remote devices.

4.7.2 Adding RDF Server Support

The configuration file SLS$ROOT:[TTI_RDEV]CONFIG_ *nodename*.DAT is created during the installation procedure. Include the tape devices you wish to serve in this file.

For more information:

Refer to the *Media and Device Management Services for OpenVMS Guide to Operations* for detailed instructions about modifying the SLS$ROOT:[TTI_RDEV]CONFIG_*nodename*.DAT file for remote tape devices.

A

SLS Server Installation Example Log File

The following example shows an installation of the SLS server software on an Alpha system running OpenVMS Version 6.2.

A.1 SLS Server Installation Log-Part 1

```
\[MAXIMUM]

$ @SYS$UPDATE:VMSINSTAL SLSJ029 MAIN01::SYS$KITS:[SLS]

OpenVMS AXP Software Product Installation Procedure V6.2-1H2

It is 05-Jan-2005 at 13:28.

Enter a question mark (?) at any time for help.

%VMSINSTAL-W-NOTSYSTEM, You are not logged in to the SYSTEM account.
%VMSINSTAL-W-ACTIVE, The following processes are still active:
DECW$SERVER_0
DECW$MWM
VUE$USER_3
DECW$TE_00AE
_FTA6:
_FTA14:
FAL_8308
_FTA9:
* Do you want to continue anyway [NO]? Y
* Are you satisfied with the backup of your system disk [YES]? Y

The following products will be processed:

SLS V2.9J


Beginning installation of SLS V2.9J at 13:28

%VMSINSTAL-I-RESTORE, Restoring product save set A ...
%VMSINSTAL-I-RELMOVED, Product's release notes have been moved to SYS$HELP.
************************************************************************
*                          WARNING                                     *
************************************************************************
*                                                                      *
*  SLS V2.9J (client and server) is not compatible with any previous   *
*  versions of SLS prior to V2.5.   If any node in a client-server     *
*  configuration is running a version later than V2.5, then all        *
*  nodes in the configuration must be running that same version or     *
*  higher.  An exception can be made if none of the SLS nodes          *
*  serve a SLS$MAGAZINE_MASTER_FILE.DAT database.                      *
*                                                                      *
************************************************************************
* Do you want to continue with this installation [No]? Y
* Do you want to run the IVP after the installation [YES]? Y

Product:      SLS
Producer:     HP
Version:      2.9J
```

SLS Server Installation Example Log File
A.1 SLS Server Installation Log-Part 1

```
Release Date: 05-Jan-2005

* Does this product have an authorization key registered and loaded? Y
* Do you want to purge files replaced by this installation [YES]? Y

SLS provides support for remote backups.  This function allows you to backup
disks on this system to a tape drive located on some other system with SLS
installed, or to allow other systems to backup their disks to the tape drives
on this system.
```

Note: The following two questions will not be asked for OpenVMS Alpha V8.2.;

```
* Do you want to backup disks on this system to a drive on a remote system? Y

Remote Device Facility Client support will be provided.

* Do you want to allow remote backups access to the tape drives on this sys-
tem? Y

Remote Device Facility Server support will be provided.

* Enter the name of the disk device the SLS software will use [$1$DKA0:]:
* Enter the name of the directory the SLS software will use [SLS$FILES]:
* Where to place summary files from system backups [SLS$ROOT:[SYS-
BAK.SUMMARY_FILES]]:
* Where to place intermediate history files [SLS$ROOT:[SYSBAK.TEMP_HISTORY]]:
* Where to place maintenance log files [SLS$ROOT:[SYSBAK.MAINTENANCE_LOGS]]:
* Where to place system backup log files [SLS$ROOT:[SYSBAK.SYSBAK_LOGS]]:

If you have been using earlier versions of SLS software, there will be log
files and summary files resident in SLS$SYSBAK from previous system backups.
This installation procedure has the ability to automatically distribute these
old files into the directories you have specified.

* Do you want to automatically move these old files? [Y]?

In order for the SLS software to perform system management tasks, a privi-
leged account called SLS is created.

You must specify a unique UIC for this account so that it will not conflict
with other accounts on your system.

%SLS-I-EXISTS, the identifier SLS already exists with UIC [1,20]
This procedure will assume the SLS account also exists.

Please enter a password for the SLS account.  The password you
specify must contain at least 15 and no more than 31 characters and
may not refer to the account/product name, or your system
node or cluster name.

Valid characters include A through Z, 0 through 9, $ (dollar sign) and
_ (underscore).

As with the DCL SET PASSWORD command, your input will not echo on the
terminal and you will be asked to verify the password.

>
* Password:
>
>
* Verify password:
>
%SLS-I-UPDACC, Updating account SLS with specified password
%VMSINSTAL-I-ACCOUNT, This installation updates an ACCOUNT named SLS.
%UAF-I-MDFYMSG, user record(s) updated
%SLS-I-UPDACC, Updating account SLS for client network access
%VMSINSTAL-I-ACCOUNT, This installation updates an ACCOUNT named SLS.
%UAF-I-MDFYMSG, user record(s) updated
%SLS-I-UPDACC, Updating account SLS privileges and login directory
```

```
%VMSINSTAL-I-ACCOUNT, This installation updates an ACCOUNT named SLS.
%UAF-I-MDFYMSG, user record(s) updated
%VMSINSTAL-I-ACCOUNT, This installation updates an ACCOUNT named SLS.
%UAF-I-MDFYMSG, user record(s) updated
```

Refer to Section A.1 for Part 2 of a sample installation of the SLS server software on a VAX system running OpenVMS Version 6.2.

A.2 SLS Server Installation Log-Part 2

```
The SLS account is used to establish quotas for the processes in which Back-
ups are run.  The quota values should be updated in accordance with the rec-
ommendations set forth in the `VMS Backup Utility Reference Manual'.  If you
wish, this installation procedure will automatically update these quotas now.

* Do you wish to update these quotas now [YES]? Y

The following quotas will be set for the SLS account:

Maxjobs:         0  Fillm:       500  Bytlm:       128000
Maxacctjobs:     0  Shrfillm:      0  Pbytlm:           0
Maxdetach:       0  BIOlm:       128  JTquota:       1024
Prclm:          10  DIOlm:      4096  WSdef:         1024
Prio:            4  ASTlm:      4096  WSquo:        16384
Queprio:         4  TQElm:        32  WSextent:     16384
CPU:        (none)  Enqlm:      2500  Pgflquo:     100000

%SLS-I-UPDACC, Updating quotas for account SLS...
%VMSINSTAL-I-ACCOUNT, This installation updates an ACCOUNT named SLS.
%UAF-I-MDFYMSG, user record(s) updated
%VMSINSTAL-I-ACCOUNT, This installation updates an ACCOUNT named SLS.
%UAF-I-MDFYMSG, user record(s) updated
%VMSINSTAL-I-ACCOUNT, This installation updates an ACCOUNT named SLS.
%UAF-I-MDFYMSG, user record(s) updated

HP recommends that the SLS account be used only by the SLS software and not as
an interactive account.  It is recommended that the Storage Administrator and
Operators use other accounts to perform SLS functions.  If you wish, this
installation procedure will automatically set the appropriate flags to disal-
low interactive access to this account.

* Do you wish to update these flags now [YES]? Y

The following fields will be set for the SLS account:

/FLAGS=   NoAudit      DisMail          NoDisReport
DefCLI       DisNewMail       NoDisUser
NoDisCtlY    DisReconnect     NoGenPwd
/BATCH     /NOINTERACTIVE  /NOLOCAL     /NODIALUP
/NOREMOTE  /NOEXPIRATION   /PWDEXPIRED  /NETWORK

%SLS-I-UPDACC, Updating flags for account SLS...
%VMSINSTAL-I-ACCOUNT, This installation updates an ACCOUNT named SLS.
%UAF-I-MDFYMSG, user record(s) updated
%VMSINSTAL-I-ACCOUNT, This installation updates an ACCOUNT named SLS.
%UAF-I-MDFYMSG, user record(s) updated
%VMSINSTAL-I-ACCOUNT, This installation updates an ACCOUNT named SLS.
%UAF-I-MDFYMSG, user record(s) updated
%SLS-I-QUOTA, Checking for DISKQUOTAs on the SLS server device ...

There will be no further questions asked.  The installation of Storage
Library System will take an average of 30 minutes more to complete.  It may
take longer if the log files and summary files from previous system backups
are going to be automatically distributed into the system backup directories
you have specified.

%VMSINSTAL-I-RESTORE, Restoring product save set D ...
%SLS-I-PROCEED, Proceeding with saveset D
%SLS-I-LINKING, Linking privileged shareable image
```

A.2 SLS Server Installation Log-Part 2

```
The definitions of SLS$SYSBAK_LOGS, SLS$MAINTENANCE_LOGS, SLS$SUMMARY_FILES
and SLS$TEMP_HISTORY are being replaced in your TAPESTART.COM file with the
values you specified earlier in this installation.  If you have changed the
location of any of these logicals, you must ensure that appropriate files are
copied from the old location to the new location.

%SLS-I-CRTDIR, Creating SLS subdirectories...
%VMSINSTAL-I-SYSDIR, This product creates system disk directory
$1$DKA0:[SLS$FILES].
%CREATE-I-EXISTS, $1$DKA0:[SLS$FILES] already exists
%VMSINSTAL-I-SYSDIR, This product creates system disk directory
$1$DKA0:[SLS$FILES.AUDIT].
%CREATE-I-EXISTS, $1$DKA0:[SLS$FILES.AUDIT] already exists
%VMSINSTAL-I-SYSDIR, This product creates system disk directory
$1$DKA0:[SLS$FILES.CUSTOM].
%CREATE-I-EXISTS, $1$DKA0:[SLS$FILES.CUSTOM] already exists
%VMSINSTAL-I-SYSDIR, This product creates system disk directory
$1$DKA0:[SLS$FILES.DATA].
%CREATE-I-EXISTS, $1$DKA0:[SLS$FILES.DATA] already exists
%VMSINSTAL-I-SYSDIR, This product creates system disk directory
$1$DKA0:[SLS$FILES.HIST].
%CREATE-I-EXISTS, $1$DKA0:[SLS$FILES.HIST] already exists
%VMSINSTAL-I-SYSDIR, This product creates system disk directory
$1$DKA0:[SLS$FILES.PARAMS].
%CREATE-I-EXISTS, $1$DKA0:[SLS$FILES.PARAMS] already exists
%VMSINSTAL-I-SYSDIR, This product creates system disk directory
$1$DKA0:[SLS$FILES.PRIMAST].
%CREATE-I-EXISTS, $1$DKA0:[SLS$FILES.PRIMAST] already exists
%VMSINSTAL-I-SYSDIR, This product creates system disk directory
$1$DKA0:[SLS$FILES.SYSBAK].
%CREATE-I-EXISTS, $1$DKA0:[SLS$FILES.SYSBAK] already exists
%VMSINSTAL-I-SYSDIR, This product creates system disk directory
$1$DKA0:[SLS$FILES.SYSTEM].
%CREATE-I-EXISTS, $1$DKA0:[SLS$FILES.SYSTEM] already exists
%VMSINSTAL-I-SYSDIR, This product creates system disk directory
$1$DKA0:[SLS$FILES.USRBAK].
%CREATE-I-EXISTS, $1$DKA0:[SLS$FILES.USRBAK] already exists
%VMSINSTAL-I-SYSDIR, This product creates system disk directory
SLS$ROOT:[SYSBAK.SUMMARY_FILES].
%CREATE-I-EXISTS, SLS$ROOT:[SYSBAK.SUMMARY_FILES] already exists
%VMSINSTAL-I-SYSDIR, This product creates system disk directory
SLS$ROOT:[SYSBAK.TEMP_HISTORY].
%CREATE-I-EXISTS, SLS$ROOT:[SYSBAK.TEMP_HISTORY] already exists
%VMSINSTAL-I-SYSDIR, This product creates system disk directory
SLS$ROOT:[SYSBAK.MAINTENANCE_LOGS].
%CREATE-I-EXISTS, SLS$ROOT:[SYSBAK.MAINTENANCE_LOGS] already exists
%VMSINSTAL-I-SYSDIR, This product creates system disk directory
SLS$ROOT:[SYSBAK.SYSBAK_LOGS].
%CREATE-I-EXISTS, SLS$ROOT:[SYSBAK.SYSBAK_LOGS] already exists
%SLS-I-DEFFILE, Defining files that need to be moved
%SLS-I-PROVRDF, Providing Remote Device Facility Support
%VMSINSTAL-I-RESTORE, Restoring product save set E ...
%VMSINSTAL-I-SYSDIR, This product creates system disk directory
SLS$ROOT:[TTI_RDEV].
%CREATE-I-EXISTS, SLS$ROOT:[TTI_RDEV] already exists

%SLS-I-SPLITSYSBAK, Moving old log files and summary files as requested.
%SLS-I-SPLITDONE, Finished moving old log files and summary files.
------------------------------------------------------------------
The following post-installation steps are necessary.
Refer to the Installation Guide for details.

1. Edit the system startup procedure to start the SLS software
on this client.  Add
$ @SYS$STARTUP:SLS$STARTUP


2. Edit the system wide login file.  Add
$ @SYS$MANAGER:SLS$TAPSYMBOL
```

```
3. Run SLS$SYSTEM:SLS$SLSMGR.EXE to perform the following functions:
a. Authorize access to the SLS server database for SLS client nodes.
b. Authorize access to SLS pools for SLS users.

4. Customize SYS$MANAGER:TAPESTART.COM to reflect SLS environment parameters.

5. Check that SLS$DBX object is defined on all nodes participating in the
VMScluster System alias.

6. For Remote Device Facility client support, you should add one or more
media-type triplets (MTYPE_x, DENS_x, DRIVES_x) to your TAPESTART.COM to
describe the remote tape drives.  Be sure to include the remote nodename in
each drive name.

7. For Remote Device Facility server support, you should check the file
SLS$ROOT:[TTI_RDEV]CONFIG_NODE1.DAT to ensure that the tape devices you wish
to serve are included in this RDF configuration file.  This file will be cre-
ated initially when SLS is started after this installation.

8. On both Alpha and VAX systems, if you are upgrading from version 2.9B or
earlier, you MUST reboot your system before using the Remote Device Facility.

%VMSINSTAL-I-MOVEFILES, Files will now be moved to their target directo-
ries...
%SLS-I-STARTUP, Starting SLS
%RUN-S-PROC_ID, identification of created process is 2020023C
%SLS-I-WAIT, Waiting for the SLS software to start
****************************************************************************
Beginning SLS Version V2.9J Installation Verification Procedure (IVP)
****************************************************************************
Installation verification begun at 05-Jan-2005 13:47:55.94

Checking for correct version of VMS.

Checking for presence of DECnet.

Checking for SLS executable images.

Checking system-wide logical names.

Checking for SLS client process.

Checking for FMS executable images.

Checking for correct version of SLS.

Waiting for subprocess SLS$IVP to complete.

%SLS-I-VERSION, Storage Library System for OpenVMS, version V2.9J

Doing a STORAGE ADD VOLUME.

Waiting for subprocess SLS$IVP to complete.

Doing a STORAGE ALLOCATE.

Doing a STORAGE SHOW VOLUME.

Volume:      XYZZY9               Owner:      NODE1::SLS
Format:      BACKUP               Brand:
Allocated:   05-Jan-2005 13:48    Scratch:    05-Jan-2005 13:48
Purchased:   05-Jan-2005 13:48    Cleaned:    05-Jan-2005 13:48
Media type:  TK85K                Length:          0
Mounts:           0               UIC:        [SLS]
Location:    CXN                  Protection: S:RW,O:RW,G:R,W:
Notes:
Offsite:                          Onsite:
IO Errors:        0               Flag:       ALLOCATED
Next volume: *none*               Previous:   *none*
Pool:                             Rec len: 0  Block factor: 0
```

A.2 SLS Server Installation Log-Part 2

```
Initialized: 05-Jan-2005 13:48    Density:
Slot number:
Side:                             Other side:
Jukebox / slot:      /
Drive:
Volume is not in a jukebox.
Volume is not in a magazine.


Doing a STORAGE DEALLOCATE.

Doing a STORAGE REMOVE VOLUME.

Doing a STORAGE SHOW VOLUME; there should be no such volume.

%SLS-W-NOSUCHVOL, no such volume(s)
* * *  All tests have completed sucessfully. * * *

Installation verification finished at 05-Jan-2005 13:48:11.35
****************************************************************************
Successful Installation Verification of SLS Version V2.9J
****************************************************************************
Installation of SLS V2.9J completed at 13:48
Adding history entry in VMI$ROOT:[SYSUPD]VMSINSTAL.HISTORY
Creating installation data file: VMI$ROOT:[SYSUPD]SLSJ029.VMI_DATA
VMSINSTAL procedure done at 13:48
```

B

Files and Logical Names

B.1 Files

The Storage Library System (SLS) installation procedure creates a number of files on your system and defines logical names.

- Table B–1 lists and describes the files installed on server nodes.
- Table B–2 lists and describes the files installed on client nodes.
- Table B–3 lists logical name table entries.
- Table B–4 lists unload drive logicals.

Table B–1 Files Installed on Server Nodes

File Name	Description
SYS$LIBRARY: directory	
SLS$SHR.EXE	SLS shareable image; contains application interface routines for history file access
SYS$MANAGER: directory	
SLS$TAPSYMBOL.COM TAPESTART.COM TAPESTART.TEMPLATE	Normal user login symbol definitions Contains all SLS software's tailoring parameters Template file for TAPESTART.COM
SYS$MESSAGE: directory	
SLS$BACKUPMSG.EXE SLS$BACKUPMSG-Vnn.EXE SLS$MSG.EXE	Contains information for backup messages; copied from the appropriate SLS$BACKUPMSG-Vnn.EXE file for the version of OpenVMS running on your system Contains OpenVMS version-specific information for backup messages Contains text of SLS messages
SYS$STARTUP: directory	
SLS$STARTUP.COM	Startup command file
SYS$TEST: directory	
SLS$IVP.COM	Installation verification procedure
SLS$HIST_GENERIC: directory	
SBATTR.COM	System backup history attribute template file
SLS$MASTER: directory	

Table B–1 Files Installed on Server Nodes

File Name	Description
POOLAUTH.DAT	Pool authorization file
SLOTMAST.DAT	Slot master file
SLOTMAST.FDL	Slot master file definition
TAPEMAST.DAT	Volume database
TAPEMAST.FDL	FDL for TAPEMAST.DAT
SLS$MAGAZINE_MASTER_FILE.DAT	Magazine database
SLS$PARAMS: directory	
Note: The files in SLS$PARAMS: are not replaced by the installation procedure if SLS software was previously installed on your system.	
ASNUSRBAK.COM	Defines location of user history files
B1C.TEMPLATE	Distribution copy of B1C.TEMPLATE Note: The template version of this file is in SLS$SYSTEM:B1C.TEMPLATE. Use the updated version in SLS$PARAMS:.
HOLIDAYS.DAT	Defines dates of holidays for system backups
POOLAUTH.FDL	FDL for POOLAUTH.DAT
SLS$DEFAULT_B1C.DAT	File containing name of default Standby Archiving class
SLS$SYSBAK: directory	
ARCHIVE_SBK.COM	Template parameter file for archive system backups
SYSBAK.TEMPLATE	Template system backup file
SLS$SYSTEM: directory	

Table B–1 Files Installed on Server Nodes

File Name	Description
ACCTABLES.COM	Accesses LNM$SLS logical name table
ADV_MAIL.COM	Announces processing location
ALLOCATED_TAPES_MSTRPT.COM	Allocated Tapes report definition file
ALLOCATION_ORDER_MSTRPT.COM	Allocation Order report definition file
ALLOCDRIVE.COM	Allocates a tape drive
ASNUSRBAK_SHIPPED_VERSION.COM	ASNUSRBAK.COM distribution copy
BACKUPCMD-Vnn.CLD	Command Language Definition file for specific VMS versions - for Alpha: V6.2, V7.1 and V7.2; for VAX: V5.5- 2, V6.2, V7.1 and V7.2 example, BACKUPCMD-V72.CLD
BACKUPCMD.CLD	Command Language Definition file for backup; copied from BACKUPCMD-Vnn, where nn is the current VMS version level
BACKUPXEQ.COM	Batch job to do a backup job
BACKUPXXX_ASCII.COM	Batch job for ASCII backup job
BACKUPXXX_BACKUP.COM	Batch job for VMS BACKUP backup job
BACKUPXXX_EBCDIC.COM	Batch job for EBCDIC backup job
B1C.TEMPLATE	Template for archiving classes
CHECK_FOR_TODAY.COM	Automatic scheduling day checker
CLEANUP.COM	Early morning cleanup batch job
CLEAN_LIST_MSTRPT.COM	Cleanable tapes report definition file
COMING_ONSITE_MSTRPT.COM	Coming onsite report definition file
CREDEF.COM	Creates a symbol definition file (in [.DATA])
CR_SB_FILES.COM	Creates a SYSBAK history FILES file
DEFTAPES.COM	Installs DATATRIEVE information
DOWN_TAPES_MSTRPT.COM	Prints a list of down tapes control file
DO_A_BACKUP.COM	Creates a *_SBK.COM system backup definition file
DTSV.COM	Defines a LNM$SLS_VALUE logical name
FIND_SYSTEM.COM	Finds a file in SYS$SYSTEM
FREE_TAPES_MSTRPT.COM	Free tapes report definition file
GETEARLIEST.COM	Gets the earliest created file
GOING_OFFSITE_MSTRPT.COM	Going offsite report definition file
HISTLOCK.COM	Places an exclusive lock on a system backup history set
INITTAPES.COM	Operator initialize tapes utility
JOBCOMPLETE.COM	Sends mail message for backups and restores
LOADER.COM	SLS system bootstrap loader
LOGIN.COM	SLS manager account login file
MAIL_ADVANCE_RUN.COM	Procedure to run MAIL_ADVANCE
MAIL_ADVANCE_SEND.COM	Displays scratchable tape
MAKE_B1C.COM	Creates archive class authorizations
MISCMENU.COM	Miscellaneous operator menu
MOUNTREEL.COM	Mounts and validates a reel on a drive
MOVUSRBAK.COM	Moves USRBAK files to their owner
MSTRPT.COM	Master report batch job
MSTRPT_MSTRPT.COM	Master report template definition file
OPERSYMBOL.COM	Operator login symbol definitions
OPSTART.COM	Operator startup processing job
PRINTED_LABEL_V20.TEMPLATE	Template file for printed labels
PRINTED_LABEL_V21.TEMPLATE	Template file for printed labels
QUANTITY_REPORT.COM	Quantity report batch job
REBOOT.COM	System startup command file
RELEASE_TRANS.COM	Releases several tapes that are in transition
RESTORXEQ_BACKUP.COM	Normal main restore batch job
RESTORXXX_ASCII.COM	Restores an ASCII format tape
RESTORXXX_BACKUP.COM	Normal subrestore batch job
RESTORXXX_EBCDIC.COM	Restores an EBCDIC format tape
RESTORXXX_RMUBACKUP.COM	Restores an RMU backup tape

Table B–1 Files Installed on Server Nodes

File Name	Description
SBQUEUE.COM	Queues system backups for today
SBSUMPUR.COM	Removes old system backup summary files
SBUPDT.COM	Updates a system backup history batch job
SEQUENTIAL.FDL	RMS definition file to convert a system history file to a sequential file
SETPROTS.COM	Sets protection of SLS files
SETUSRBAK.COM	Sets up SLS$USRBAK logical name for a user
SET_VAULT_DATES.COM	Sets up onsite and offsite dates
SHCOMPRESS.COM	Compresses system history files
SLS.FLB	Contains all VAX FMS forms for SLS software
SLSMGR.FLB	Screens for SLSMGR.EXE
SLS$ABORT.EXE	Utility to abort a process
SLS$ALLOCATE.COM	Allocates a local or remote tape device
SLS$BACKUP.EXE	Builds a user backup request
SLS$B1C.FDL	FDL file for *_B1C.DAT
SLS$CHKCOMP.EXE	Checks tape drives for data compaction capability
SLS$COLLECT.EXE	Accounting period data collector
SLS$CREJOBTBL.EXE	Creates the LNM$JOB table with W:RE protection
SLS$CREMBX.EXE	Creates a permanent mailbox for DCL use
SLS$DBSYSBAK.EXE	System backup executable for RMU database backup operations
SLS$DBSYSBAK_SPAWN.COM	Command file to control spawned database backup process
SLS$DCSC_MGMT_EXPORT.COM	Operator Export Volume function for the ACS
SLS$DCSC_MGMT_IMPORT.COM	Operator Import Volume function for the ACS
SLS$DCSC_MGMT_INIT_VOL_SERIES.COM	Operator Initialize Volume Series function for the ACS
SLS$DCSC_MGMT_INVENTORY.COM	Operator Inventory Range of Volumes function for the ACS
SLS$DCSC_MGMT_LOAD.COM	
SLS$DCSC_MGMT_MENU.COM	Operator Load Volume Onto Drive function for the ACS
SLS$DCSC_MGMT_UNLOAD_DRIVE.COM	Operator SLS-ACS Management Menu
SLS$DCSC_MGMT_UNLOAD_VOL.COM	Operator Unload Drive function for the ACS Operator Unload Volume function for the ACS
SLS$DCSC_RELEASE_DRIVE.EXE	
SLS$DEALLOCATE.COM	Executable to release a drive allocated to DCSC
SLS$DELMBX.EXE	Deallocates a local or remote tape device
SLS$FORCEX.EXE	Deletes a permanent mailbox that CREMBX made
SLS$GETMBX.EXE	Executes a FORCEX system service call
SLS$GETUAI.EXE	Reads from a permanent mailbox from DCL
SLS$HELP.HLP	Gets information from SYSUAF.DAT
SLS$IBMNLBRDE.EXE	DCL STORAGE command help file
SLS$IBMNLBWRT.EXE	IBM [TM] nonlabeled tape reading utility
SLS$IS_DRIVE_DCSC.EXE	IBM [TM] nonlabeled tape writing utility
SLS$IS_DRIVE_NOUNLOAD.EXE	For SLS-ACS support, determines whether DCSC (Digital Cartridge Server Component) software accesses a device
SLS$KEYATR.EXE	Determines whether a drive is set to be no unload
SLS$LOAD_FINALSTS.EXE	Reads key attributes into symbols
SLS$LOADRIGHTS.EXE	Loads user address space with known exit status value
SLS$MAIL_ADVANCE.EXE	Reads process rights identifiers
SLS$MAINT.EXE	Finds tapes that need to have advance mail sent
SLS$MSTRPT.EXE	Maintenance screen executable
SLS$OPCOM_SERVER.EXE	Master report executable
SLS$PUTMBX.EXE	Image for SLS$OPCOM process; sends and receives remote OPCOM messages and device control requests
SLS$QUANTITY.EXE	Writes to a permanent mailbox from DCL
SLS$RDB_GETVER.COM	Quantity report executable
	Command procedure to determine version of RMU software

Table B–1 Files Installed on Server Nodes

File Name	Description
SLS$REQUEST.EXE	DCL request that can define a symbol
SLS$RESTOR.EXE	Normal restore screen request generator
SLS$SBLOAD.EXE	Loads system history definitions utility
SLS$SBPOLL.EXE	Polls for system backup subprocess complete
SLS$SBUPDT.EXE	Updates a system backup history executable
SLS$SETDEVOWN.EXE	Sets owner UIC of a device
SLS$SETPRV.EXE	Sets process privileges given a hexadecimal mask
SLS$SETUSR.EXE	Sets process username
SLS$SET_MESSAGE.COM	Allows SLS command procedures to print text of SLS system messages
SLS$SHUTDOWN.COM	Shuts down SLS software
SLS$SLSMGR.EXE	SLS manager's authorization menu
SLS$SLTRPT.EXE	Prints a slot report
SLS$START_OPCOM_SERVER.COM	Starts SLS$OPCOM process
SLS$START_RDF.COM	Starts Remote Device Facility (RDF) process
SLS$SYNTAX.CLD	DCL STORAGE command definition
SLS$SYSBAK.EXE	System backup executable
SLS$SYSBAK_SPAWN.COM	Command procedure to control spawned system backup process
SLS$SYSCLN.EXE	System backup history cleanup executable
SLS$SYSFRE.EXE	System backup history reel free executable
SLS$SYSRPT.EXE	Generates reports of system backup histories
SLS$TAPE.EXE	DCL TAPE command executable
SLS$TAPEPURGE.EXE	Purges scratchable tape's executable
SLS$TAPEUSAGE.EXE	Generate tape usage report executable
SLS$TAPMGRDB.EXE	DB manager executable
SLS$TAPMGRRQ.EXE	RQ manager executable
SLS$TRANSPORT.EXE	Moves volumes to vault
SLS$UBLOAD.EXE	User history load utility
SLS$USRBAK.EXE	User backup executable
SLS$USRFRE.EXE	User backup history tape free processor
SLS$USRRPT.EXE	User backup history report generator
SLS$VAULT_PROFILE_RPT.EXE	Vault profile report executable
SLS$VERSION_CHECK_AND_COPY.COM	Command procedure to verify your OpenVMS software version against the versions of BACKUPCMD.CLD and VMSBUXX.EXE; you use this procedure if you upgrade OpenVMS after you have installed SLS
SLS$VOLUME.EXE	Maintenance program for volumes in StorageTek ACS robots
STABACK_TAPES.COM	Allocates tapes for a standalone backup
STANDBY_ARCHIVE.COM	Standby archiving batch job
STANDBY_ARCHIVE_MENU.COM	Standby archiving control menu
STARTUP.COM	SLS startup command file
SWITCHLABEL.COM	Used in label printing
SYSBAK.COM	System backup batch job
SYSBAKRUN.COM	Template file to run system backups
SYSBAK_COMMAND.COM	Intercepts SYSBAK subprocess
SYSBAK_FILES.FDL	RMS definition file for system backup history .FILES files
SYSBAK_SETS.FDL	RMS definition file for system backup history .SETS files
SYSCLN.COM	System backup history cleanup batch job
SYSCLN_CONVERT.COM	Converts system history files as a Batch process
SYSCLN_MENU.COM	System backup history cleanup control menu
SYSOLDTAPE.COM	Reads old system backup tape and updates system history file
SYSREADLIST.COM	Reads backup listing and creates system history file
SYSRESTOR.COM	Defines entire disk restore job
SYSRESTOR2.COM	Restores an entire disk batch job
TAPEMENU.HLP	Help file for TUSERMENU.COM

Table B–1 Files Installed on Server Nodes

File Name	Description
TAPEPURGEX.COM	Purges scratchable tape's subprocess
TAPEPURGEX.MSG	Purges scratchable tape's mail message
TAPESTART_SHIPPED_VERSION.COM	Distribution version of TAPESTART.COM
TAPES_OFFSITE_MSTRPT.COM	Tapes offsite report definition file
TAPEUSERUN.COM	Generates tape usage report batch job
TAPMGRDBOFF.COM	DB manager shutdown command file
TAPMGRDBRUN.COM	DB manager startup command file
TAPMGRDBXEQ.COM	DB manager detached process command file
TAPMGRRQRUN.COM	RQ manager startup command file
TAPMGRRQXEQ.COM	RQ manager detached process command file
TAPMGRRUN.COM	Starts TAPMGRDB and TAPMGRRQ command files
TAPMGRUTXEQ.COM	Utility manager detached process command file
TAPMGRUT_FREE.COM	Free tape utility processor
TAPMGRUT_MIDNIT.COM	Midnight utility processor
TOPERMENU.COM	TOPER menu command file
TRANS_TAPES_MSTRPT.COM	Defines volume report for volumes in transition
TUSERMENU.COM	Another user menu
USRBAK_COMMAND.COM	Intercepts USRBAK subprocess
USRBAK_DOCDB.FDL	FDL for DOCDB_4R2 files
USRBAK_FILES.FDL	FDL for normal user history file
USRDELETE.COM	Deletes user history entries
USRRPTUSR.COM	Generates SLS$USRBAK logical name for USRRPT.EXE
VAULT.COM	Sets location of volumes in or out of vault
VAULTMENU.COM	Vault menu command file
VAULT_DATES.DAT	Onsite and offsite dates for SET_ VAULT_DATES
VAULT_PROFILE_RPT_RUN.COM	Vault profile report batch job
VMSBUXX-Vnn.EXE	Patched version of OpenVMS BACKUP for specific OpenVMS versions; for example, VMSBUXX-V.EXE
VMSBUXX.EXE	Patched version of OpenVMS BACKUP; copied from whichever VMSBUXX-Vnn.EXE is appropriate for the version of OpenVMS software

Table B–2 Files Installed on Client Nodes

File Name	Description
SYS$LIBRARY: directory	
JCS$SHR.EXE	Nonfunctional JCS shareable image; installed only if JCS software is not present
SLS$SHR.EXE	SLS shareable image; contains application interface routines for history file access
SYS$MANAGER: directory	
SLS$TAPSYMBOL.COM	Normal user login symbol definitions
TAPESTART.COM	Contains all SLS software's customization parameters
TAPESTART.TEMPLATE	Template file for TAPESTART.COM
SYS$MESSAGE: directory	
SLS$MSG.EXE	Contains text of SLS messages
SYS$STARTUP: directory	

Table B–2 Files Installed on Client Nodes

File Name	Description
SLS$STARTUP.COM	Startup command file
SYS$TEST: directory	
SLS$IVP.COM	Installation verification procedure
SLS$HIST_GENERIC: directory	
SBATTR.COM	System backup history attribute template file
SLS$PARAMS: directory	
ASNUSRBAK.COM B1C.TEMPLATE HOLIDAYS.DAT SLS$DEFAULT_B1C.DAT	Defines location of user history files Distribution copy of B1C.TEMPLATE Defines dates of holidays for system backups Default standby archiving authorization file
SLS$SYSBAK: directory	
ARCHIVE_SBK.COM SYSBAK.TEMPLATE	Template parameter file for archive system backups Template system backup file
SLS$SYSTEM: directory	
ACCTABLES.COM	Accesses LNM$SLS logical name table
ADV_MAIL.COM	Announces processing location
ALLOCATED_TAPES_MSTRPT.COM	Allocated tapes report definition file
ALLOCATION_ORDER_MSTRPT.COM	Allocation order report definition file
ALLOCDRIVE.COM	
ASNUSRBAK_SHIPPED_VERSION.CO	Allocates a tape drive
M	ASNUSRBAK.COM distribution copy
BACKUPXEQ.COM	
BACKUPXXX_ASCII.COM	Batch job to run a backup job
BACKUPXXX_BACKUP.COM	Batch job for ASCII backup job
BACKUPXXX_EBCDIC.COM	Batch job for VMS BACKUP backup job
BACKUPXXX_UASCII.COM	Batch job for EBCDIC backup job
CHECK_FOR_TODAY.COM	Batch job for UASCII backup job
CLEANUP.COM	Automatic scheduling day checker
CLEAN_LIST_MSTRPT.COM	Early morning cleanup batch job
COMING_ONSITE_MSTRPT.COM	Cleanable tapes report definition file
	Coming onsite report definition file

Table B–2 Files Installed on Client Nodes

File Name	Description
CREDEF.COM	Creates a symbol definition file (in [.DATA])
CR_SB_FILES.COM	Creates a SYSBAK history FILES file
DEFTAPES.COM	Installs DATATRIEVE information
DOWN_TAPES_MSTRPT.COM	Prints a list of down tapes control file
DO_A_BACKUP.COM	Creates an *_SBK.COM system backup definition file
DTSV.COM	Defines a LNM$SLS_VALUE logical name
FIND_SYSTEM.COM	Finds a file in SYS$SYSTEM
FREE_TAPES_MSTRPT.COM	Free tapes report definition file
GETEARLIEST.COM	Gets the earliest created file
GOING_OFFSITE_MSTRPT.COM	Going offsite report definition file
HISTLOCK.COM	Places an exclusive lock on a system backup history set
INITTAPES.COM	Operator initialize tapes utility
JOBCOMPLETE.COM	Sends mail message for backups and restores
LOADER.COM	SLS system bootstrap loader
LOGIN.COM	SLS manager account login file
MAIL_ADVANCE_RUN.COM	Procedure to run MAIL_ADVANCE
MAIL_ADVANCE_SEND.COM	Displays scratchable tape
MAKE_B1C.COM	Creates archive class authorizations
MISCMENU.COM	Miscellaneous operator menu
MOUNTREEL.COM	Mounts and validates a reel on a drive
MSTRPT.COM	Master report batch job
MSTRPT_MSTRPT.COM	Master report template definition file
OPERSYMBOL.COM	Operator login symbol definitions
OPSTART.COM	Operator startup processing job
PRINTED_LABEL_V20.TEMPLATE	Template file for printed labels
PRINTED_LABEL_V21.TEMPLATE	Template file for printed labels
QUANTITY_REPORT.COM	Quantity report batch job
REBOOT.COM	System startup command file
RELEASE_TRANS.COM	Releases several tapes that are in transition
RESTORXEQ_BACKUP.COM	Normal main restore batch job
RESTORXXX_ASCII.COM	Restores an ASCII format tape
RESTORXXX_BACKUP.COM	Normal subrestore batch job
RESTORXXX_EBCDIC.COM	Restores an EBCDIC format tape
SBQUEUE.COM	Queues system backups for today
SBSUMPUR.COM	Removes old system backup summary files
SBUPDT.COM	Updates a system backup history batch job
SEQUENTIAL.FDL	Creates a sequential file
SETUSRBAK.COM	Sets up SLS$USRBAK logical name for a user
SET_VAULT_DATES.COM	Sets up onsite and offsite dates
SHCOMPRESS.COM	Compresses system history files
SLS.FLB	Contains all VAX FMS forms for SLS software
SLS$ABORT.EXE	Utility to abort a process
SLS$ALLOCATE.COM	Allocates a local or remote tape device
SLS$BACKUP.EXE	Builds a normal backup request
SLS$B1C.FDL	FDL file for *_B1C.DAT
SLS$COLLECT.EXE	Accounting period data collector
SLS$CREJOBTBL.EXE	Creates the LNM$JOB table with W:RE protection
SLS$CREMBX.COM	Creates a permanent mailbox for DCL use
SLS$DCSC_MGMT_EXPORT.EXE	Operator Export Volume function for the ACS
SLS$DCSC_MGMT_IMPORT.COM	Operator Import Volume function for the ACS
SLS$DCSC_MGMT_INIT_VOL_SERIES.COM	Operator Initialize Volume Series function for the ACS
SLS$DCSC_MGMT_INVENTORY.COM	Operator Inventory Range of Volumes function for the ACS
SLS$DCSC_MGMT_LOAD.COM	Operator Load Volume Onto Drive function for the ACS
SLS$DCSC_MGMT_MENU.COM	Operator SLS-ACS Management Menu
SLS$DCSC_MGMT_UNLOAD_DRIVE.COM	Operator Unload Drive function for the ACS

Table B–2 Files Installed on Client Nodes

File Name	Description
SLS$DCSC_MGMT_UNLOAD_VOL.COM	Operator Unload Volume function for the ACS
SLS$DEALLOCATE.COM	Deallocates a local or remote tape device
SLS$DELMBX.EXE	Deletes a permanent mailbox that CREMBX made
SLS$FORCEX.EXE	Executes a FORCEX system service call
SLS$GETMBX.EXE	Reads from a permanent mailbox from DCL
SLS$GETUAI.EXE	Gets information from SYSUAF.DAT
SLS$HELP.HLP	DCL STORAGE command help file
SLS$IBMNLBRDE.EXE	IBM [TM] nonlabeled tape reading utility
SLS$IBMNLBWRT.EXE	IBM [TM] nonlabeled tape writing utility
SLS$IS_DRIVE_DCSC.EXE	For SLS-ACS support, determines whether DCSC (Digital Cartridge Server Component) software accesses a device
SLS$KEYATR.EXE	Reads key attributes into symbols
SLS$LOADRIGHTS.EXE	Reads process rights identifiers
SLS$MAIL_ADVANCE.EXE	Finds tapes that need to have advance mail sent
SLS$MAINT.EXE	Maintenance screen executable
SLS$MSTRPT.EXE	Master report executable
SLS$OPCOM_SERVER.EXE	Image for SLS$OPCOM process; sends and receives remote OPCOM messages and device control requests
SLS$PUTMBX.EXE	Writes to a permanent mailbox from DCL
SLS$QUANTITY.EXE	Quantity report executable
SLS$REQUEST.EXE	DCL request that can define a symbol
SLS$RESTOR.EXE	Normal restore screen request generator
SLS$SBLOAD.EXE	Loads system history definitions utility
SLS$SBPOLL.EXE	Polls for system backup subprocess complete
SLS$SBUPDT.EXE	Updates a system backup history executable
SLS$SETDEVOWN.EXE	Sets owner UIC of a device
SLS$SET_MESSAGE.COM	Allows SLS command procedures to print text of SLS system messages
SLS$SETPRV.EXE	Sets process privileges given a hexadecimal mask
SLS$SETUSR.EXE	Sets process username
SLS$SHUTDOWN.COM	Shuts down SLS software
SLS$SLTRPT.EXE	Prints a slot report
SLS$START_OPCOM_SERVER.COM	Starts SLS$OPCOM process
SLS$SYNTAX.CLD	DCL STORAGE command definition
SLS$SYSBAK.EXE	System backup executable
SLS$SYSCLN.EXE	System backup history cleanup executable
SLS$SYSFRE.EXE	System backup history reel free executable
SLS$SYSRPT.EXE	Generates reports of system backup histories
SLS$TAPE.EXE	DCL TAPE command executable
SLS$TAPEPURGE.EXE	Purges scratchable tape's executable
SLS$TAPEUSAGE.EXE	Generates tape usage report executable
SLS$TAPMGRRQ.EXE	RQ manager executable
SLS$TRANSPORT.EXE	Moves volumes to vault
SLS$UBLOAD.EXE	User history load utility
SLS$USRBAK.EXE	User backup executable
SLS$USRFRE.EXE	User backup history tape free processor
SLS$USRRPT.EXE	User backup history report generator
SLS$VAULT_PROFILE_RPT.EXE	Vault profile report executable
STABACK_TAPES.COM	Allocates tapes for a standalone backup
STANDBY_ARCHIVE.COM	Standby archiving batch job
STANDBY_ARCHIVE_MENU.COM	Standby archiving control menu
STARTUP.COM	SLS startup command file
SYSBAK.COM	System backup batch job
SYSBAKRUN.COM	Template file to run system backups
SYSBAK_COMMAND.COM	Intercepts SYSBAK subprocess
SYSBAK_FILES.FDL	Template for system backup history .FILES files
SYSBAK_SETS.FDL	Template for system backup history .SETS files

Table B–2 Files Installed on Client Nodes

File Name	Description
SYSCLN.COM	System backup history cleanup batch job
SYSCLN_CONVERT.COM	Converts system history files as a Batch process
SYSCLN_MENU.COM	System backup history cleanup control menu
SYSOLDTAPE.COM	Reads old system backup tape into history
SYSREADLIST.COM	Reads backup listing and creates system history file
SYSRESTOR.COM	Defines entire disk restore job
SYSRESTOR2.COM	Restores an entire disk batch job
TAPEMENU.HLP	Help file for TAPEMENU.COM
TAPEPURGEX.COM	Purges scratchable tape's subprocess
TAPEPURGEX.MSG	Purges scratchable tape's mail message
TAPESTART_SHIPPED_VERSION.COM	Distribution version of TAPESTART.COM
TAPES_OFFSITE_MSTRPT.COM	Tapes offsite report definition file
TAPEUSERUN.COM	Generates tape usage report batch job
TAPMGRRQRUN.COM	RQ manager startup command file
TAPMGRRQXEQ.COM	RQ manager detached process command file
TAPMGRRUN.COM	Starts TAPMGRDB and TAPMGRRQ command files
TAPMGRUTXEQ.COM	Utility manager detached process command file
TAPMGRUT_FREE.COM	Free volume utility processor
TAPMGRUT_MIDNIT.COM	Midnight utility processor
TOPERMENU.COM	TOPER menu command file
TRANS_TAPES_MSTRPT.COM	Defines volume report for volumes in transition
TUSERMENU.COM	Another user menu
USRBAK_COMMAND.COM	Intercepts USRBAK subprocess
USRBAK_FILES.FDL	FDL for normal user history file
USRDELETE.COM	Deletes user history entries
USRRPTUSR.COM	Generates an SLS$USRBAK logical name for USRRPT.EXE
VAULT.COM	Sets location of volumes in or out of vault
VAULTMENU.COM	Vault menu command file
VAULT_DATES.DAT	Onsite and offsite dates for SET_ VAULT_DATES
VAULT_PROFILE_RPT_RUN.COM	Vault profile report batch job
VMSBUXX.EXE	Patched version of VMS BACKUP

B.2 Logical Names

Table B–3 lists the logical names entered into the logical name tables LNM$SYSTEM_TABLE and LNM$SLS$TABLE when SLS software is installed. These names are defined by the product's startup file. 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 SLS software.

Table B–3 Logical Name Table Entries

Logical Name	Definition and Description
LNM$SYSTEM_TABLE Entries	
SLS$CUSTOM	SLS$ROOT:[CUSTOM] This directory contains customized SLS files for your site.
SLS$DATA	SLS$ROOT:[DATA.nodename] This directory stores all the log files for your current SLS system.

Table B–3 Logical Name Table Entries

SLS$DATAC	SLS$ROOT:[DATA]
	This is the top level directory for log file storage.
SLS$DEFAULT_JUKEBOX	JCS$DEFAULT_JUKEBOX
	The default jukebox used by SLS software.
SLS$DEFSBACLASS	FOREVER <- insert archive class here
	This logical points to the default archive class. Refer to the Storage Library System for OpenVMS Storage Administrator's Guide for information on defining standby archiving classes.
SLS$HIST_$ALL$	GENERIC
	The default system backup history set.
SLS$HIST_GENERIC	SLS$ROOT:[HIST]
	The location of the GENERIC history set.
SLS$LABEL	filespec or _NL: or device:
	This logical points to the current label file or printer where labels are printed.
SLS$MAILBOX	mbx_id
	The mailbox used for communication to the SLS$TAPMGRRQ process.
SLS$MASTER	SLS$ROOT:[PRIMAST]
	This directory points to the node's database, whether this is your current node or a node remote to you.
SLS$PARAMS	SLS$ROOT:[PARAMS]
	This directory contains the system dependent parameter files for SLS software.
SLS$ROOT	device:[SLS$FILES.]
	This directory points to the top level directory for SLS files.
SLS$STARTUP_P1	REBOOT
	Used internally during startup.
SLS$SYSBAK	SLS$ROOT:[SYSBAK]
	This directory contains the *_SBK.COM history files.
SLS$SYSTEM	SLS$ROOT:[CUSTOM] SLS$ROOT:[SYSTEM]
	This directory contains all the files for running SLS software.
SLS$VAULT	DPS
	The location for volumes sent offsite.
LNM$SLS$TABLE Entries	

Table B–3 Logical Name Table Entries

PRINODNAME	nodename Name of node where primary volume database resides.
TAPMGRDB_PROCID	proc_id Points to the process ID of the database manager process.
TAPMGRRQ_PROCID	proc_id Points to the process ID of the request manager process.
LNM$JOB Entries	
SLS$B1C_FILE	SLS$PARAMS:SLS$DEFAULT_B1C.DAT Location of default standby archiving authorization file.
SLS$USRBAK	SLS$ROOT:[USRBAK]username This points to the SLS backup procedure that uses the VMS BACKUP utility to back up user files.
SLS$NET_id_MBX	mbx_id Network mailbox used by SLS software for user. One or more may be listed.

More logical names for SLS software can be found in the logical name table LNM$SLS$VALUES. These logicals are defined whenever TAPESTART.COM executes. Sample output from the DCL command SHOW LOGICAL/TABLE=LNM$SLS$VALUES is shown in Example 4–1.

Example 4–1 Sample List of LNM$SLS$VALUES Entries

```
$ SHOW LOGICAL/TABLE=LNM$SLS$VALUES <Return>

        (LNM$SLS$VALUES)

"BACKUP_DEFAULT_REEL" = " "
"BACKUP_FINISH" = "MAIL"
"BACKUP_QUEUE" = "SLS$THUNDR/NOSPOOL"
"CLEANUP_Q" = "SLS$THUNDR/AFTER=03:00"
"DEFAULT_ALLDEV" = "THUNDR$MUB0,THUNDR$MUC0,THUNDR$MUC2,
THUNDR$MUA3,THUNDR$MUA1 "
"DEFAULT_ALLOCSCRATCH" = "365 0:0:0"
"DEFAULT_ALLOCSIZE" = "0"
"DEFAULT_ALLTIM" = "0 0:0:30"
"DEFAULT_BACKUPSCRATCH" = "365 0:0:0"
"DEFAULT_BACKUPSIZE" = "0"
"DEFAULT_BAKFMT" = "BACKUP"
"DEFAULT_BAKOPT" = "YYYYYY"
"DEFAULT_BAKUIC" = "0"
"DEFAULT_BATCHQUEUE" = "SLS$THUNDR"
"DEFAULT_DENS_1" = "6250"
"DEFAULT_DENS_10" = " "
"DEFAULT_DENS_11" = " "
```

```
"DEFAULT_DENS_12" = " "
"DEFAULT_DENS_13" = " "
"DEFAULT_DENS_14" = " "
"DEFAULT_DENS_15" = " "
"DEFAULT_DENS_16" = " "
"DEFAULT_DENS_17" = " "
"DEFAULT_DENS_18" = " "
"DEFAULT_DENS_19" = " "
"DEFAULT_DENS_2" = "1600"
"DEFAULT_DENS_20" = " "
"DEFAULT_DENS_21" = " "
"DEFAULT_DENS_22" = " "
"DEFAULT_DENS_23" = " "
"DEFAULT_DENS_24" = " "
"DEFAULT_DENS_25" = " "
"DEFAULT_DENS_26" = " "
"DEFAULT_DENS_27" = " "
"DEFAULT_DENS_28" = " "
"DEFAULT_DENS_29" = " "
"DEFAULT_DENS_3" = "800"
"DEFAULT_DENS_30" = " "
"DEFAULT_DENS_31" = " "
"DEFAULT_DENS_32" = " "
"DEFAULT_DENS_4" = "COMP"
"DEFAULT_DENS_5" = "NOCOMP"
"DEFAULT_DENS_6" = " "
"DEFAULT_DENS_7" = " "
"DEFAULT_DENS_8" = " "
"DEFAULT_DENS_9" = " "
"DEFAULT_DRIVES_1" = "THUNDR$MUB0"
"DEFAULT_DRIVES_10" = " "
"DEFAULT_DRIVES_11" = " "
"DEFAULT_DRIVES_12" = " "
"DEFAULT_DRIVES_13" = " "
"DEFAULT_DRIVES_14" = " "
"DEFAULT_DRIVES_15" = " "
"DEFAULT_DRIVES_16" = " "
"DEFAULT_DRIVES_17" = " "
"DEFAULT_DRIVES_18" = " "
"DEFAULT_DRIVES_19" = " "
"DEFAULT_DRIVES_2" = "THUNDR$MUB0"
"DEFAULT_DRIVES_20" = " "
"DEFAULT_DRIVES_21" = " "
"DEFAULT_DRIVES_22" = " "
"DEFAULT_DRIVES_23" = " "
"DEFAULT_DRIVES_24" = " "
"DEFAULT_DRIVES_25" = " "
"DEFAULT_DRIVES_26" = " "
"DEFAULT_DRIVES_27" = " "
"DEFAULT_DRIVES_28" = " "
"DEFAULT_DRIVES_29" = " "
"DEFAULT_DRIVES_3" = " "
"DEFAULT_DRIVES_30" = " "
"DEFAULT_DRIVES_31" = " "
```

```
"DEFAULT_DRIVES_32" = " "
"DEFAULT_DRIVES_4" = "THUNDR$MUA4 "
"DEFAULT_DRIVES_5" = "THUNDR$MUA3"
"DEFAULT_DRIVES_6" = "THUNDR$MUC0,THUNDR$MUC2,THUNDR$MUA1"
"DEFAULT_DRIVES_7" = " "
"DEFAULT_DRIVES_8" = " "
"DEFAULT_DRIVES_9" = " "
"DEFAULT_FRESTA" = "2"
"DEFAULT_MAXSCRATCH" = "31-DEC-9999 23:59:59.99"
"DEFAULT_MTYPE_1" = "9TRACK"
"DEFAULT_MTYPE_10" = " "
"DEFAULT_MTYPE_11" = " "
"DEFAULT_MTYPE_12" = " "
"DEFAULT_MTYPE_13" = " "
"DEFAULT_MTYPE_14" = " "
"DEFAULT_MTYPE_15" = " "
"DEFAULT_MTYPE_16" = " "
"DEFAULT_MTYPE_17" = " "
"DEFAULT_MTYPE_18" = " "
"DEFAULT_MTYPE_19" = " "
"DEFAULT_MTYPE_2" = "9TRACK"
"DEFAULT_MTYPE_20" = " "
"DEFAULT_MTYPE_21" = " "
"DEFAULT_MTYPE_22" = " "
"DEFAULT_MTYPE_23" = " "
"DEFAULT_MTYPE_24" = " "
"DEFAULT_MTYPE_25" = " "
"DEFAULT_MTYPE_26" = " "
"DEFAULT_MTYPE_27" = " "
"DEFAULT_MTYPE_28" = " "
"DEFAULT_MTYPE_29" = " "
"DEFAULT_MTYPE_3" = "9TRACK"
"DEFAULT_MTYPE_30" = " "
"DEFAULT_MTYPE_31" = " "
"DEFAULT_MTYPE_32" = " "
"DEFAULT_MTYPE_4" = "TA90E"
"DEFAULT_MTYPE_5" = "TA90E"
"DEFAULT_MTYPE_6" = "RV02K"
"DEFAULT_MTYPE_7" = " "
"DEFAULT_MTYPE_8" = " "
"DEFAULT_MTYPE_9" = " "
"DEFAULT_PROTECTION" = "307"
"DEFAULT_RESOPT" = " Y NY"
"DEFAULT_SELDEV" =
"THUNDR$MUB0,THUNDR$MUC0,THUNDR$MUC2,THUNDR$MUA3,THUNDR$MUA
1 "
"DEFAULT_USRRSS" = "128"
"DEFAULT_VERBOSE" = "1"
"ESC_ALLOC_BOLD" = ".   .[1m"
"ESC_ALLOC_NORM" = ".   .[m"
"ESC_LOAD_BLNK" = ".[5m.[7w"
"ESC_LOAD_BOLD" = ".[1m.[7w"
"ESC_LOAD_NORM" = ".[m.[w"
"ESC_MOUNT_BOLD" = ".   .[1m"
```

```
"ESC_MOUNT_NORM" = ".   .[m"
"ESC_MOUNT_OPER" = "...[1m.#6 OPERATOR:"
"LOCATION" = "HEADQUARTERS"
"MGRPRI" = "8"
"NET_REQUEST_TIMEOUT" = "15"
"NODENAME" = "THUNDR"
"PRIV_CLEAN" = "OPER"
"PRIV_LABEL" = "OPER"
"PRIV_MAXSCR" = "OPER"
"PRIV_MODANY" = "OPER"
"PRIV_MODOWN" = "TMPMBX"
"PRIV_SEEANY" = "OPER"
"PRIV_V_CLEAN" = "18"
"PRIV_V_LABEL" = "18"
"PRIV_V_MAXSCR" = "18"
"PRIV_V_MODANY" = "18"
"PRIV_V_MODOWN" = "15"
"PRIV_V_SEEANY" = "18"
"QUICKLOAD" = "1"
"QUICKLOAD_RETRIES" = "80"
"REMOTE_RETRIES" = "5"
"RESTORE_FINISH" = "MAIL"
"RESTOR_QUEUE" = "SLS$THUNDR/NOSPOOL"
"SLS$UNATTENDED_BACKUPS" = "0"
"STANDBY_ARCHIVE_INT" = "00:00:30"
"STANDBY_ARCHIVE_LOG" = "SBARLOG"
"STANDBY_ARCHIVE_START" = "0"
"SYSCLN_RUN" = "WED=03,FRI=03,SAT=51 "
"TAPEPURGE_MAIL" = "SYSTEM "
"TAPEPURGE_WORK" = "MAIL "
"TOPERMASK" = "1024"
"TOPERSTRING" = "TAPES"
"TRANS_AGE" = "14 0:0:0"
"USERBACK_FIRSTLINE" = "$!"
"USERNAME" = "SLS"
"VERSION" = "V2.9"
"VMSBU" = "VMSBUXX"
"VMSV5" = "1"
```

Table B–4 lists logical names that control whether or not SLS software unloads a drive when it dismounts a volume. In some cases, you may not want the drive to unload. These logical names are defined in the logical name table LNM$SLS$VALUES.

There are two options:

- To have the drive unload, define the logical as /UNLOAD.
- To have the drive not unload, define the logical as /NOUNLOAD

.

Table B–4 Unload Drive Logicals

Logical in LNM$SLS$VALUES Table	Description
ARCHIVE_DISM_QUALS	For standby archiving operations executed with the STORAGE SAVE command
BACKUPXEQ_DISM_QUALS	For user backup operations executed with the STORAGE SAVE command
RESTORXEQ_DISM_QUALS	For user restore operations executed with the STORAGE RESTORE command
SYSBAK_DISM_QUALS	For system backup operations
SYSRES_DISM_QUALS	For system restore operations

C

SLS Account User Authorization Data

C.1 Authorization Data

Table A-1 lists AUTHORIZE qualifiers used to update the SLS account with values recommended for SLS software. They are also used when the account is being created.

Table C–1 SLS Account User Authorization Data

Authorization field and value	Description
/MAXJOBS=0	Maximum number of interactive, batch, and detached processes that can be active at one time with this user name is unlimited.
/MAXACCTJOBS=0	Maximum number of interactive, batch, and detached processes on the same account that can be active at one time is unlimited.
/MAXDETACH=0	Maximum number of detached processes with this user name that may be active at one time is unlimited.
/PRCLM=10	Total number of subprocesses that may be active at one time.
/PRIORITY=4	Default base priority.
/QUEPRIO=4	Default queue priority.
/CPU=(none)	Maximum amount of CPU time is unlimited.
/FILLM=500	Total number of files and network logical links that may be active at one time.
/SHRFILLM=0	Maximum number of shared files that may be open at one time is unlimited.
/BIOLM=128	Maximum number of buffered I/O operations that may be active at one time.
/DIOLM=4096	Maximum number of direct I/O operations that may be active at one time.

Table C–1 SLS Account User Authorization Data

Authorization field and value	Description
/ASTLM=4096	Maximum number of pending AST requests that may be queued for the user 0.
/ENQLM=1000	Total number of lock requests that may be outstanding at the same time.
/BYTLM=128000	Total number of bytes that may be specified for a buffered I/O operation.
/PBYTLM=0	Paged pool byte count limit (unlimited).
/JTQUOTA=1024	Byte count quota of the job-wide logical name table.
/WSDEF=1024	Initial limit of a working set for the user process.
/WSQUO=16384	Maximum to which the user's process may raise the working set limit when system memory is in demand.
/WSEXTENT=16384	Maximum to which the user's process may raise its working set limit when there is free memory available.
/PGFLQUO=100000	Total pages that this process can use in the system paging file.
/BATCH	Allows unrestricted access hours for batch jobs for this user name.
/NOINTER	Disallows interactive logins for this user name.
/NOLOCAL	Disallows interactive logins for this user name through a local terminal.
/NODIAL	Disallows interactive logins for this user name through dialup terminals.
/NOREM	Disallows interactive logins for this user name through network remote terminals.
/NOEXPIRATION	Account is not expired.
/PWDEXPIRED	Password is expired

Table C–1 SLS Account User Authorization Data

Authorization field and value	Description
/FLAGS=	NODISUSER-Does not disable account completely. NOAUDIT-Does not audit all security relevant actions. DEFCLI-Prevents user from changing default CLI or CLI tables. NODISCTLY-Allows Ctrl/Y interrupts. DISMAIL-Prevents mail delivery to this user. DISNEWMAIL-Suppresses "New Mail..." announcements. DISRECONNECT-Disables automated reconnections. NODISREPORT-Disables time of last login and other security reports. NOGENPWD-Does not require user to use generated passwords. Note: These should be enclosed in parentheses and, if more than one is specified, separated by commas.
/UIC=	UIC as specified by the installer.
/OWNER="SLS_MANAGER"	Name of the account owner.

Index