OpenVMS Technical Journal V7

SLS to ABS/MDMS Migration

Ted Saul - HP Off-site Software Support Consultant

Introduction

The Storage Library System backup application has served the OpenVMS customer based extremely well since the late 1980's. Because ABS/MDMS is the go-forward backup application on OpenVMS operating system environments, this article provides guidance for migrating to ABS/MDMS from your SLS system.

This article describes the tasks the SLS system manager must accomplish during the migration. Each environment is different, and this paper cannot address every possible scenario. However, you can minimize the impact of the migration by understanding what you currently have in SLS and how it equates to ABS/MDMS.

The keys to a successful migration are:

- Understand how SLS relates to ABS
- Understand how the current SLS environment is configured
- Determine the changes that need to be made in the current backup environment
- Plan the objects and policies that need to be created in the ABS/MDMS environment.

Migrating from SLS to ABS/MDMS presents advantages and challenges. By using as much of the new functionality as possible, you can eliminate the drawbacks to using SLS. And by identifying the challenges of the new application, you can implement the migration as seamlessly as possible.

Some of the advantages of the ABS/MDMS application include:

- An object and policy driven software application
- A more robust GUI that can be used from an OpenVMS system or Windows based PC
- Stronger scheduling options
- An opportunity to change current procedures that are ineffective
- A more secure backup environment with less need to customize the code

- Oracle, Windows, and UNIX backups
- Support for new devices introduced by HP

Some or all of the following challenges may affect your migration:

- Sorting out customized code embedded in the current SLS environment
- Changing the processes for operations
- Remembering how and why SLS was originally configured, and translating that information into ABS objects and processes. In some cases, SLS was configured by a person who has long since moved on, without leaving appropriate documentation.
- Designing a testing period as well as parallel production processing time
- Finding and restoring data from SLS history set information.
- Understanding how objects and symbols relate to each other

A written backup policy is a vital part of developing and maintaining a secure backup environment. The process of working through this documentation helps identify operational weaknesses and gives operations the guidance they need when exceptions arise. This plan should include a current layout of your data and the strategy for backing it up. Include information about scheduling and the type of every backup, as well as the retention period for the data. Document a step by step process that describes when backups are to be run so that new personnel can easily understand processes. Exceptions and how to handle events and errors will provide guidance during off-hours when IT managers and supervisors are not available.

The migration period from SLS to ABS/MDMS is an excellent time to develop or review the backup policy. Be sure to include how the project of migration will be accomplished and a summary of the implementation.

The following sections to address how to gather the necessary information for the final configuration of ABS/MDMS. Worksheets and templates help to manage and organize this data.

Differences between SLS and ABS/MDMS

Program / Coding Differences

ABS/MDMS contains mostly executable code, as compared to SLS with its 60% DCL command procedures. As a result, ABS/MDMS cannot be customized to a customer’s environment as easily. Many of the SLS customizations are already in ABS/MDMS. Consider each modification in SLS and consider a strategy to implement this functionality in ABS/MDMS. You may have to implement a new process for operations to handle backups.

Backup / Scripting

To set up the environment in which backups will be completed, SLS uses backup scripts known as SBK files. It is easy to insert custom code into the backup processing in these DCL command files in order to make SLS behave differently. SBK files can easily be built on the fly from a list of disks that is created daily, to assist in load balancing. ABS/MDMS uses a system of policies that include SAVE, ARCHIVE, and ENVIRONMENT, to complete its work. SAVE policies can be created dynamically to be run in a “one-time” environment, but you will have to create a system of scripts to achieve this functionality. This policy-based environment of command procedures is different from the SLS environment and requires a ground up approach. SLS scripts will not work with ABS/MDMS.

ABS/MDMS also uses a RESTORE policy to schedule restores from backups, as opposed to the GUI-driven SLS restore screens. Any command procedures written for STORAGE RESTORE must be recreated using new ABS/MDMS commands.

History Sets and Cataloging

ABS/MDMS uses a series of files to create complex catalogs that may take longer to get through history sets. ABS/MDMS catalogs tend to grow larger in size and required a more granular catalog strategy. For example, in SLS, a simple “Image” history set in SLS may be sufficient for tracking a range of backups, but in ABS/MDMS, for better performance, an "Image" catalog for weekly, monthly and yearly time periods must be created. However, if only one large catalog is created, a similar amount of disk space is used, but the lookup and cleanup processing time improves.

User Interfaces

Both SLS and ABS/MDMS contain DCL interfaces. Almost any command that that can be completed using the ABS/MDMS GUI can also be done from the DCL command line. SLS provides three GUIs based on FMS form handling. Each serves its own purpose in allowing access to data in volume database or managing other databases. ABS/MDMS, on the other hand, contains one GUI; user privileges control the information that is accessible. The ABS/MDMS GUI can be run on an OpenVMS workstation or Windows-based system.

Saveset Format

Like SLS, ABS/MDMS uses the native OpenVMS backup saveset format. This allows for restores using the backup image should it be required. ABS also uses the native BACKUP.EXE image from SYS$SYSTEM making the applying and tracking of OpenVMS ECOs easier.

Scheduling

The ABS/MDMS scheduler is more robust than that of SLS. Where SLS can start its backup based on the day and time, with slight variations, ABS/MDMS schedules can be defined and associated with as many backups as desired.

Configuration

To configure SLS, you edit the SYS$MANAGER:TAPESTART.COM file, setting a series of symbols to the appropriate values to meet the needs of the environment. On the other hand, ABS/MDMS uses MDMS$SYSTARTUP.COM, which contains a series of object settings and uses one startup file for its configuration. Table 1 lists the symbols found in the SLS TAPESTART.COM procedure. Where available, the equivalent ABS/MDMS policy and field is referenced. Settings that must be included in the MDMS$SYSTARTUP.COM procedure are listed as well.

Table 1 - SLS Symbols and Equivalent ABS/MDMS Policies

TAPESTART SYMBOLS	ABS/MDMS POLICY or Startup File	POLICY FIELD
PRI	NODE MDMS$SYSTARTUP.COM	Database MDMS$DATABASE_SERVERS
DB_NODES	NODE MDMS$SYSTARTUP.COM	Database MDMS$DATABASE_SERVERS
PRIMAST	MDMS$SYSTARTUP.COM	MDMS$ROOT:[DATABASE]
SLS$SYSBAK_LOGS	MDMS$SYSTARTUP.COM	MDMS$ROOT:[LOGFILE]
SLS$MAINTENANCE_LOGS	MDMS$SYSTARTUP.COM	MDMS$ROOT:[LOGFILE]
SLS$SUMMARY_FILES	N/A	N/A
SLS$TEMP_HISTORY	MDMS$SYSTARTUP.COM	MDMS$ROOT:[CATALOG]
NET_REQUEST_TIMEOUT	N/A	
BATN	N/A	
MTYPE_n	MEDIA TYPE	
DENS_n	MEDIA TYPE	/density, /compaction
DRIVES_N	DRIVE	
TAPE_JUKEBOXES	JUKEBOX	

MGRPRI	N/A	
VERBOSE	N/A	
PRIV_SEEANY	N/A	
PRIV_MODANY	N/A	
PRIV_MAXSCR	N/A	
PRIV_LABEL	N/A	
PRIV_CLEAN	N/A	
PRIV_MODOWN	N/A	
CRLF[0,8]	N/A	
CRLF[8,8]	N/A	
ESC[0,8]	N/A	
ESC_LOAD_BOLD	N/A	
ESC_LOAD_BLNK	N/A	
ESC_LOAD_NORM	N/A	
ESC_ALLOC_BOLD	N/A	
ESC_ALLOC_NORM	N/A	
ESC_MOUNT_OPER	N/A	
ESC_MOUNT_BOLD	N/A	
ESC_MOUNT_NORM	N/A	
LOC	DOMAIN	Onsite_Location
PROTECTION	DOMAIN	/protection
ALLOCSIZE	N/A	
LBL	N/A	
FRESTA	DOMAIN	/deallocate_state
TRANS_AGE	DOMAIN	/transition_time
ALLOCSCRATCH	DOMAIN	/scratch_time
BACKUPSCRATCH	N/A	
MAXSCRATCH	DOMAIN	/maximum_scratch_time
TAPEPURGE_WORK	N/A	
TAPEPURGE_MAIL	N/A	
VLT	DOMAIN	/offsite_location
ALLDEV	DRIVE	/shared
SELDEV	DRIVE	/shared
ALLTIM	N/A	

TOPERS	DOMAIN	/opcom_classes
QUICKLOAD	N/A	
QUICKLOAD_RETRIES	N/A	
UNATTENDED_BACKUPS	N/A	
BACKUPSIZE	MEDIA TYPE	/length
BAKFMT	SELECTION	/data_selection_type
BAKOPT	N/A	
BACKUP_DEFAULT_REEL	N/A	
BAKQUE	ABS$nodename	
BACKUP_FINISH	ENVIRONMENT	/notification
HISNAM_1	CATALOG	
HISDIR_1	CATALOG	/directory
HISTYP_1	CATALOG	/type
RESOPT	N/A	
RESQUE	ABS$nodename	
RESTORE_FINISH	ENVIRONMENT	/notification
CLEANUP_Q	MDMS$SYSTARTUP.COM	MDMS$SCHEDULED_ACTIVITIES _START_HOUR
SYSCLN_RUN		
JUKEBOX	JUKEBOX	/control
JUKEBOX_1_LOWER	JUKEBOX	/cap
JUKEBOX_1_UPPER	JUKEBOX	/cap
SBARLOG	N/A	
SBARINT	N/A	
SBACLAS	N/A	

CLEANUP_Q in SLS defines the queue that it will run in, as well as the hour to start. ABS/MDMS uses SCHEDULER policies to start cleanup activities on the default queue ABS$*nodename*.

Note that the root directory for SLS is defined in the SLS$STARTUP.COM procedure. The root directory for ABS/MDMS is defined as the logical MDMS$ROOT in the MDMS$SYSTARTUP.COM procedure. If you want to move the ABS/MDMS directory from its default location of SYS$COMMON:[MDMS.], redefine this logical.

Databases

Table 2 lists the SLS database files stored in the location pointed to by the logical name SLS$MASTER. Where applicable, the equivalent ABS/MDMS database is listed.

Table 2 - SLS and Equivalent ABS/MDMS Database Files

SLS DATABASE	ABS/MDMS EQUIVILENT	DESCRIPTION
TAPEMAST.DAT	MDMS$VOLUME.DAT	Volume Information

POOLAUTH.DAT	MDMS$POOL_DB.DAT	Pool Information
SLS$MAGAZINE_MASTER _FILE.DAT	MDMS$MAGAZINE_DB.DAT	Magazine Information
SLOTMAST.DAT		Slots outside Jukebox
HOLIDAYS.DAT		Holidays to skip
VALIDATE.DAT		Remote node access
	MDMS$ARCHIVE_DB.DAT	ARCHIVE Policy
	MDMS$DOMAIN_DB.DAT	DOMAIN Objects
	MDMS$DRIVE_DB.DAT	DRIVE Objects
	MDMS$ENVIRONMENT _DB.DAT	ENVIRONMENT Policy
	MDMS$GROUP_DB.DAT	GROUP Objects
	MDMS$JUKEBOX_DB.DAT	JUKEBOX Objects
	MDMS$LOCATION_DB.DAT	LOCATION Objects
	MDMS$MEDIA_DB.DAT	MEDIA Objects
	MDMS$NODE_DB.DAT	NODE Objects
	MDMS$REPORT_DB.DAT	REPORT Objects
	MDMS$RESTORE_DB.DAT	RESTORE Policy
	MDMS$SAVE_DB.DAT	SAVE Policy
	MDMS$SCHEDULE_DB.DAT	Schedule Objects
	MDMS$SELECTION_DB.DAT	SELECTION Objects

Capturing the SLS Environment

Now that you understand the differences between the SLS and ABS/MDMS, the next step is to gather information about the currently running SLS system. For long-term SLS users, this is a good time to review why SLS was set up as it is. Table 3 shows an example worksheet that is helpful for capturing information about the current environment.

Table 3 - Backup Information Worksheet

Image SBK Name	Incremental SBK Name	Disk	Media	Start Days	History

To gather the information you need, follow this procedure:

1. Determine the actively scheduled backups and their starting times and days.

If no other scheduler than SLS is involved, you can identify current backups by searching the SLS$SYSBAK:*.COM procedures by entering the following command:

```
$ search *.com “days_”/window=4
```

For each SBK where these scheduling parameters are in use, something similar to the following, along with the SBK file name, is returned:

```
$!
$       DAYS_1 :== Monday,Tuesday,Wednesday,Thursday,Friday
$       TIME_1 :== 21:00
$       NODE_1 :==
$!
$!       DAYS_2 :==
$!       TIME_2 :==
$!       NODE_2 :==
$!
$!       DAYS_3 :==
$!       TIME_3 :==
$!       NODE_3 :==
```

In this case, a backup has been discovered that will run Monday through Friday at 9:00 PM on the current node. If the symbols are commented out, the SBK file is not scheduled and potentially is a manually-submitted backup.

2. Determine the type of each scheduled backup.

The name of the SBK usually indicates the type of backup (image or incremental). You can check this by editing the SBK and examining the /QUALIFIERS line for IMAGE, INCREMENTAL, SINCE, etc. After you determine the backup type, record the name, start time, and day of week to run, on the worksheet.

3. Determine the disks that are being backed up in each SBK.

This information can be found in the FILES_n qualifiers in each SBK. Record the disks that are backed up, and associate them with the SBK, on the worksheet. For every disk being backed up there is one associated FILES_n.

4. Determine what type of device is to be used during the backup.

The symbol MEDIA_TYPE in the SBK indicates the tape drives that are used and what types they are. For example, a TYPE of TK89 or DLT probably indicates that they are these types of drives. If the type is not evident, you can trace the MEDIA_TYPE in the SYS$MANAGER:TAPESTART.COM and the associated media triplet. The DRIVES_n that is matched with MTYPE_n is the correct device. Use SHOW DEVICE /FULL from the OpenVMS command prompt state to determine the type of drive is being used. Record this information with the corresponding SBK information.

5. Determine the SLS history set from which information about each SBK run is recorded.

Each SBK file contains the symbol HISTORY_SET with a defined value. Record this information in the worksheet. The name of the history set probably corresponds to its type. For example, IMAGE may indicate that all backups recorded here are of the image type. Take this into consideration when you create new CATALOGS in ABS/MDMS.

6. Check for manually submitted backups.

In some environments, SBKs are started manually by using the STORAGE STARTUP SYSTEM_BACKUP command. Any SBKs without scheduling parameters may be manual runs. Unfortunately, the only way to determine which SBKs are a part of the daily runs is to refer to the company's backup policy or procedures. It is to be hoped that any SBK that is run manually are documented.

Log files in the SLS$SUMMARY_FILES, SLS$MAINTENANCE_LOGS, and SLS$SYSBAK_LOGS directories tell which backups are run on any one day. By default, SLS places log files from all backup runs in these directories. If you find any manual backups, record the information that is described in steps 1 through 5.

7. Check for third-party or custom schedulers.

As with manually submitted backups, the manager of the SLS environment should have information about backups that run using a third-party scheduler. To integrate them into the ABS/MDMS environment. document the information described in Steps 1 through 5 for them. ABS/MDMS can use most third-party schedulers in place of its own. You designate the scheduler when you configure ABS/MDMS.

8. Capture other appropriate files.

Keep the SLS configuration file TAPESTART.COM in order to set the appropriate objects and policies in ABS/MDMS. Other data that is needed may come from the VALIDATE.DAT file and POOLAUTH.DAT, both of which are visible from the SYSMGR menu. The HOLIDAYS.DAT file indicates days that certain backups are to be skipped. Check this file for any custom scheduling.

9. Documenting customized code.

It is important to understand any customized code implemented in your SLS environment. Document what has been changed and why. Knowing the goal of the changes helps make sure that ABS/MDMS carries out the same behavior.

Analyze the Current Backup Environment

The migration process allows you to change the SLS backup environment. Over time, system managers often identify more efficient methods for their backup strategies. This is an excellent time to implement these strategies. In addition, the backup policy documentation should be modified to reflect these changes.

Some questions you should ask are:

- Are files backed up often enough for a reasonable restore?
- How long does it take to recover from a disk failure?
- Are users satisfied with their backups and ability to restore?
- Are all tape devices being used to their fullest capacity?
- Does ABS/MDMS solve any issues with the current backup policy?
- Can a complete disaster be recovered?
- What are the steps to bare metal restore?
- Where is the application historical data stored?

- Who is responsible for managing the DR strategy?
- How are notifications handled by applications?
- Do Windows or UNIX servers requiring backing up?
- Do Oracle databases require backing up?

Its a good idea to Interview operators and system administrators, as well as users who rely on the restore of data. All the stakeholders who are affected by the backup strategy should be taken into consideration. It is also appropriate, with the current focus on security and disaster recovery, to analyze the safety of the current environment. Refer to the *Archive Backup System for OpenVMS – Guide to Operations* for more information.

ABS/MDMS Policies and Objects

ABS/MDMS uses objects and policies to define its work and its environment. In order to configure ABS/MDMS, you need to understand the purpose each object and policy.

Table 4 describes the objects to consider when you install ABS/MDMS and migrate from SLS. Each object must be defined before a SAVE can be initiated. In Configuring ABS/MDMS using data gathered from SLS. each of the symbols in SLS are associated with these objects.

Table 4 - ABS/MDMS Objects

Object	**Description**
DRIVES:	A physical resource that can read and write data to tape volumes.
GROUPS:	A logical grouping of nodes to be allow common actions to be taken at one time.
JUKEBOXES:	Any robot-controlled device that supports automatic loading of tapes.
LOCATIONS:	An object that describes a physical location where tapes may be placed. You can define a hierarchy of locations.
MAGAZINES:	A logical object that contains a set of volumes that will need to be moved as a group.
MEDIA TYPES:	A logical object that describes certain attributes of tape volume media.
NODES:	A list of the nodes in the domain that run ABS/MDMS.
POOLS:	A logical object that contains a set of volumes that can be allocated to authorized users.
VOLUMES:	A single piece of tape media that the ABS/MDMS application can use to store its data.

A final object to consider is the DOMAIN. The DOMAIN is special because it encompasses all objects that are served by a single MDMS database. A domain consists of objects (including DRIVES, JUKEBOXES, NODES, and VOLUMES), as well as logical objects (including MEDIA TYPES, POOLS, and MAGAZINES). It is important to understand what devices in your computer room will belong to your ABS/MDMS DOMAIN and associate them accordingly.

Table 5 describes the policies used in the ABS/MDMS backup environment.

Table 5 - ABS/MDMS Policies

Policy	**Description**
ARCHIVE:	Defines the media type as well as other characteristics about where backup data is to be stored. One single ARCHIVE can then be associated with many SAVE policies.
CATALOGS:	Defines where data about each backup will be stored. Different types of

	CATALOG policies serve different purposes, including how backup will restore. It is important to understand which type will be most efficient for your site.
ENVIRONMENTS:	Describes the criteria under which save and restore requests must execute. As with the ARCHIVE, one ENVIRONMENT can be associated with many SAVES.
SAVE:	Defines unique information about the backup. This policy has other policies associated with it.
RESTORE:	Defines unique information about a restore in order to return data from a tape to disk. Data can be restored from entire disks, down to individual files. The ability to restore depends on the type of CATALOG defined.
SELECTIONS:	Holds information about files, disks, paths and databases to be saved or restored. A SELECTION can be generated automatically using the /include qualifier on a SAVE. or it can be generated manually using either the CREATE SELECTION DCL command or the MDMS GUI. Then the SELECTION needs to be associated with a SAVE or RESTORE.
SCHEDULES:	Defines backup schedules. ABS/MDMS provides a flexible scheduler. By default, an associated SCHEDULE is created with a SAVE. SCHEDULES can also be created manually and then associated with a SAVE or RESTORE.

The SAVE, RESTORE and associated policies are created using ABS/MDMS objects. This strategy reduces the amount of redundant data that needs to be entered. For example, in SLS, the MEDIA_FORMAT had to be modified on every SBK. In ABS/MDMS, only the MEDIA FORMAT object needs to be entered once on the policy called an ARCHIVE. This ARCHIVE can then be associated with many SAVE policies.

Configuring ABS/MDMS using data gathered from SLS.

After you install ABS/MDMS, following the procedure in the *ABS/MDMS Installation Guide*, it is time to tailor ABS/MDMS to match the way SLS accomplished your backups. Be sure to follow the post-installation tasks in the manual, as well.

it is a good practice to go through each of the policies to determine what will you will need to create in ABS/MDMS. You must also decide how the fields in each policy will be set. The tables of objects and policies include information about SLS equivalents, where appropriate. Make a list of each object and policy and the quantity that will needed to be created. For example, if you have two jukeboxes where one contains a TZ89 drive and the other contains a DLT7000, note that a total of six objects must be created: two jukeboxes, two drives, and two media types.

Continue this process until you have accounted for all the resources in the ABS/MDMS environment.

ABS/MDMS Object Creation

The MDMS$SYSTEM:MDMS$CONFIGURE.COM command procedure creates the required objects. You are prompted for the information required to create the policy. After each policy has been created, you can fine tune it to meet the needs of the environment. For help within the command procedure, enter two question marks (??) at the prompt.

Before you start, make sure you have at least the following information:

- Media type (user definable)
- Onsite location (user definable)
- Offsite location (user definable)
- IP domain name for node (if using the TCPIP transport)
- Name of your jukebox (user definable)
- Robot name (i.e., GKB601:)
- Drive name (user definable)

- OpenVMS device names (i.e., ALERAB$MKB200:)
- Volume naming strategy (should match bar code label)

After all objects have been successfully created, prepare the jukebox for use by inventorying and initializing the volumes in the jukebox.

To synchronize your ABS/MDMS with your jukebox, enter the following command:

```
$ MDMS INVENTORY JUKEBOX your_jukebox_name
```

This command updates the MDMS database with the information it finds in the jukebox's firmware. When the inventory is complete and the volumes in the jukebox need to be initialized, enter the following command:

```
$ MDMS INIT VOLUME/JUKEBOX=your_jukebox_name/SLOT=(beginrange,
endrange)
```

Once these commands have completed, your volumes are ready for use with ABS/MDMS. These commands also test the robotic configurations to ensure that they all work.

ABS/MDMS Policy Creation

When all the objects are in place, you can create the backup policies. There is no conversion utility to create these policies from SBK files; therefore, you have to enter them manually. This step allows you to make changes and to improve any backup inefficiencies that may be taking place. You can use either the ABS/MDMS GUI or DCL commands to create the policies. Create the policies in the following order:

1. CATALOGS
2. ARCHIVES
3. ENVIRONMENTS
4. SAVEs

RESTORES are created as needed. SELECTIONS and SCHEDULES are usually created automatically with the SAVEs.

Once the CATALOGS have been created, use the SLS information on your worksheet to create the ARCHIVE, ENVIRONMENT, and SAVE policies for each SBK file. ARCHIVE and ENVIRONMENT policies can be used to group similar types of backups. For example, if you have backups that use the same CATALOG policy and MEDIA TYPE, you can create one ARCHIVE for all these SAVES. Similarly image backups of a RAID array that are run on a regular basis can be grouped together. Information about the backups can be retained in the same CATALOG for the same length of time. Data that cannot be grouped includes monthly and weekly backups where the retention times are different. In this case, use different CATALOGS with separate ARCHIVE policies. Obviously, the policy structure must be well thought out and planned before beginning the configuration of ABS/MDMS. Once it is in place, ABS/MDMS provides efficient and logical methods for tracking backups.

Testing the environment

The ABS/MDMS testing procedures cover three areas: devices, databases, and running the actual saves.

Devices

Test your Jukebox by entering the following commands:

```
$ MDMS LOAD VOLUME/DRIVE=your_drive your_volume

$ MOUNT/FOREIGN your_drive

$ DISMOUNT your_drive

$ MDMS UNLOAD VOLUME your_volume
```

If these commands complete successfully, you can assume that your JUKEBOX and DRIVES policies are configured correctly. Use the SHOW DRIVE/CHECK command to physically access the drive. If this command fails, there is probably a connectivity problem to the drive. Use the MRU application to diagnose whether the problem is with ABS/MDMS, or with the configuration of the drive.

Remember: if the drive is working with SLS, then the problem probably lies in the ABS/MDMS configuration.

Databases

Use the $SHOW MDMS commands to ensure database connectivity. If any of these commands fail or hang, call the HP customer support center. Test all the SHOW commands because, unlike SLS, ABS/MDMS uses many different database files to store data.

SAVES and RESTORES

SAVES and RESTORES can be tested online any time. You can test general ARCHIVE, SELECTION, and SCHEDULE policies by first creating "one_time_only" SAVEs that automatically purge themselves. Use a test CATALOG, set the ARCHIVE with a short retention period, or plan to recreate these databases before putting ABS/MDMS into production.

Once you are comfortable with the way ABS/MDMS schedules and releases backups, run them in parallel with your existing SLS backups, as described in Running SLS and ABS/MDMS Simultaneously. Check the log files in the directory pointed to by the logical SLS$SYSBAK_LOGS for errors. See Troubleshooting for additional troubleshooting tips.

Define the cut-off date for SLS backups, when the ABS/MDMS environment will take over full responsibility for the backups.

Additional Issues

Oracle Backups

As in SLS, Oracle RDB backups are available in ABS/MDMS. When you create a SAVE request , select the appropriate DATA_SELECT_TYPE for the version and area of RDB to be backed up. This automatically creates the SELECTION policy. This backup should be scheduled like any other ABS/MDMS backup. Though it is not required, it is a good practice to create a separate CATALOG for these types of backups.

ABS/MDMS can also back up Oracle 8i and 9i databases. (This functionality is not available in SLS.) Refer to the ABS/MDMS documentation to learn how to set up this type of database.

What to do with SLS data in History Files?

At this time, there is no way to read the SLS history files from ABS/MDMS and initiate a backup. You can recover pre ABS/MDMS data from SLS using one of the following methods:

1. Keep an instance of SLS running for history lookups only. The SLS and ABS licenses support this environment, and once the information is located in SLS, you can initiate a manual restore command using the Backup utility.
2. Capture data from SLS volumes and feed it directly into ABS/MDMS catalogs. This can be a large task, so consider carefully which backups need to be restored mostly commonly.

To catalog existing savesets, create a SAVE policy by enter the following command:

```
# MDMS CREATE SAVE saveset_catalog–
/INCLUDE=tape:*-
/DATA_TYPE=VMS_SAVESET-
/ARCHIVE=archive-
/ENVIRONMENT=environment
/START=start_time
```

The data from the tape is written to the catalog designated by the ARCHIVE policy.

Running SLS and ABS/MDMS Simultaneously

SLS and ABS/MDMS can run simultaneously on one system. To set up this environment, follow this procedure:

1. In the SYS$STARTUP:MDMS$SYSTARTUP.COM file, set the logical MDMS$VERSION3 to False. This logical enables STORAGE commands to look at the ABS/MDMS databases. Though this specific functionality is no longer applicable, setting the logical to False ensures that the two applications will not interfere with one another.
2. Install the test ABS/MDMS environment on the appropriate node.

CAUTION

The greatest danger in running SLS and ABS/MDMS together on the same node is that you may cross volumes between the two applications. Make sure that each application knows only about the volumes that it can use. You can accomplish this by identifying which volumes in the jukebox will be designated for use by SLS and by the ABS/MDMS applications, and then do one of the following:

- o Create a pool in ABS/MDMS called “SLS” and place all of the volumes to be used by SLS in this pool.
- o Create a pool in SLS called “ABS” and place all the volume to be used by ABS in this pool

This procedure sets aside volumes in each application that the other will be using, preventing them from allocating and using one another’s volumes.

Another way to prevent use by each other’s application is to not define the volumes in ABS. Though the jukebox will be loaded with both, inventory and allocations will overlook volumes they don’t know about.

Troubleshooting

This section describes additional troubleshooting procedures.

Startup Issues

The ABS/MDMS equivalent to the SLS command procedure SLS$ROOT:[000000]TAPESTART*nodename*.COM is MDMS$LOG:MDMS$STARTUP_*nodename*.LOG. This procedure reveals issues during the startup of the product. As with SLS, turning on OPCOM may reveal problems such as syntax errors and licensing issues.

Save and Restore Issues

SLS users can read log files for system backups from the directory SLS$SYSBAK_LOGS. Similarly, ABS/MDMS puts its log files in the directory pointed to by ABS$LOG. You can check SAVEs and RESTOREs for completion status by scanning the associated log files. By default, the log files have the same name as the SAVE policy itself. To monitor these logs, enter the following command:

```
$ TYPE/TAIL/CONT
```

This command shows you the end of the file as the log buffer is dumped to disk.

History / Catalog Issues

The ABS$CATALOG:Catalog_*n*.LOG file tracks the way files are processed and staged for catalogs. Check this file to determine whether recently backed-up data is not showing up in the appropriate catalog. The SLS equivalent file is SLS$SBUPDT.LOG in SLS$MAINTENANCE_LOGS.

The ABS$CATALOG:ABS$CATALOG_CLEANUP.LOG file contains information about the daily cleanup of catalogs and removal of obsolete records. Check this file if you suspect that your catalogs are not be cleaned up as volumes freed. In SLS, the SLS$DATA:SYSCLN.LOG and SLS$DATA:CLEANUP.LOG contain this information.

Miscellaneous logs (no SLS equivalents)

The ABS$CATALOG:ABS$COORD_CLEANUP_nodename.LOG file reveals any problems with the ABS/MDMS coordinator, which is responsible for a number of different functions. If you suspect there is a problem with the coordinator, this log is a starting point for troubleshooting.

The MDMS$LOG:MDMS$LOGFILE_DBSERVER.LOG file tracks system events and errors detected by the MDMS$SERVER process.

Other files in the MDMS$LOGFILE directory, as well as additional settings that provide more in-depth trouble shooting information, may be used at the direction of HP Services if the need arises.

Summary

The keys to a successful migration include:

- Understanding your current environment in order to capture it in ABS/MDMS. Without this understanding, you may end up reinventing how you do your backups.
- Understanding the differences between SLS and ABS/MDMS, in order to know what changes need to be made to your current processes. In general, ABS/MDMS handles backups in a more efficient manner than SLS. Your site may have adapted to using SLS a certain way and the change may be difficult. Create a change management plan as a part of your migration plan to explain the reasons for the migration and the benefits of moving to ABS/MDMS.
- Knowing how you want to change your backup policy, in order to incorporate these changes into your new ABS/MDMS environment. Now is a good time to improve the company backup plan that may have been in place for years.
- Running SLS and ABS/MDMS simultaneously for a designated period of time shows that ABS/MDMS is capable of performing necessary backups safely. A successful migration period reduces the risk to the company, as well the fears that the application won't be able to handle your required backup strategy

The HP Services Value Added Services Team is available to assist with your migrations. There are many ways and levels that HP can help including:

- Migration planning. HP can help develop your plan to ensure its success.
- Migration execution. If time is limited and it is difficult to begin this project, HP can manage and execute the process by providing the documentation and the instruction that is required. Any changes would be reviewed by your company to ensure compliance with its standards.
- Help sorting out current SLS environments. If it has been a long time since SLS implementation took place, a qualified HP engineer familiar with both SLS and ABS/MDMS can help you understand all that is being accomplished at your site by SLS. In turn a knowledge transfer can be done to allow you to continue with your migration.
- Auditing the current backup strategy. HP engineers experienced in SLS and ABS/MDMS can look as a third-party to ensure that your backup strategy is as secure as you believe it is.
- Reviewing your plans. HP engineers can review your plans and can be available for you to bounce your thoughts and ideas off of.

For more information

This white paper, complete with templates and relational addendums, is available from the HP Call Center.

To contact the HP Services VAST team, ask one of your HP service engineers or Technical Account Managers, or call (888) 376-4737.