SLS to ABS/MDMS Migration White Paper

Table of Contents

SLS to ABS/MDMS Migration White Paper

Introduction

The Storage Library System backup application has been in existence since the late 1980's and has served the OpenVMS customer based extremely well. With the decision to make ABS/MDMS the go forward backup application on OpenVMS operating system environments, the need has risen to provide guidance to those needing to migrate to ABS/MDMS from their SLS systems.

This document is designed to give the SLS System Manager the ability to grasp what tasks need to be accomplished during the migration. Because each environment is different, the paper can not address every possible scenario. However with an understanding of what one currently has in SLS and how it equates to ABS/MDMS, the migrations impact can be minimized.

The key to a successful migration will be to understand:

- ✓ How SLS relates to ABS
- ✓ How the current SLS environment is configured
- ✓ What changes need to be made in the current backup environment
- ✓ The objects and policies that need to be created in the ABS/MDMS environment.

Migrating from SLS to ABS/MDMS will present advantages and challenges. The goal is to capture and use as much of the new functionality as possible that eliminates the drawbacks to using SLS. Also, by identifying the challenges of the new application, it is hoped that the migration will be implemented as seamlessly as possible.

Some of the advantages that will be seen in the new ABS/MDMS application include:

- ✓ An opportunity to change current ineffective procedures.
- ✓ The use of new functionality presented in the ABS/MDMS application including Oracle, Windows, and Unix Backups.
- ✓ A more secure backup environment with less need to customize the code.
- ✓ Availability of a more robust GUI that can be used from an OpenVMS system or Windows based PC.
- ✓ Stronger scheduling options.
- ✓ Support for new devices introduced by HP.
- ✓ An object and policy driven software application.

Challenges that need to be considered are listed below. Each environment will be somewhat different so the following may or may not have an affect on your site:

- ✓ Sorting out of customized code that may be embedded into current SLS environment.
- ✓ Change in processes for current operations.
- ✓ Remembering how and why SLS was originally configured in order to translate into ABS objects and processes. For others, SLS may have been configured by another person who has long since moved on without appropriate documentation.
- ✓ Designing a testing period as well as parallel production processing time.
- ✓ Identifying a process to find and restore data that may be left in 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 will help identify areas of weakness as well as giving 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. Questions such as how often and the type of backup as well as the retention period for the data should be included. A step by step process of when backups are to run should be documented 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 review the backup policy. In some cases it may be the time to develop the document. It is also a good idea to include how the project of migration will be accomplished and an implementation summary.

The following sections are written to address the gathering of this information to help with the final configuration of ABS/MDMS. Worksheets and templates have been designed to assist in managing and organizing this data.

Overview of Differences between SLS and ABS/MDMS

Program / Coding

ABS/MDMS contains mostly executable code compared with SLS with its 60% DCL command procedures. The result of this difference is that ABS/MDMS cannot be customized to a customer’s desire as easily. Many of the SLS customizations may already be in ABS/MDMS. The modifications carried out in SLS will have to be considered and a strategy to implement this functionality put into ABS/MDMS. In some cases, a new process may have to be developed on how operations handle their backups.

Backup / Scripting

SLS uses backup scripts known as SBK files to set up the environment in which their backups will be completed. These DCL command files also make it easy to insert custom code into the backup processing in order to have SLS behave in a slightly different manner. These SBK files could also be easily built on the fly from a list of disks that may be created daily to assist in load balancing. ABS/MDMS uses a system of objects that include SAVE, ARCHIVE and ENVIRONMENT polices to complete its work. SAVE policies can be created dynamically to be run in a "one-time" environment but will require that a system of scripts be built to create this functionality. This environment of command procedures will be different than the SLS environment and will require a ground up approach. Previous SLS scripts will no longer work with ABS/MDMS.

ABS/MDMS also makes use of a RESTORE policy that can be used to schedule restores from backups as opposed to the SLS GUI driven restore screens. Any command procedures written around STORAGE RESTORE will need to be recreated using new MDMS commands.

History Sets vs. Cataloging

ABS/MDMS uses as series of files to create its catalogs. Due to the ABS/MDMS catalog complexity, processing through them may take longer than SLS does to get through its history sets. ABS/MDMS catalogs tend to grow larger in size as well as requiring a more granular catalog strategy. For example, where a simple "Image" history set in SLS may be sufficient for tracking a range of backups, ABS/MDMS, for better performance, may require that an "Image" catalog for weekly's, monthly and yearly time periods be created. With this strategy, the amount of diskspace used remains in close proximity if only one large catalog was created, however the lookup and cleanup processing time will improve.

User Interfaces

Both SLS and ABS/MDMS contain adequate DCL interfaces. Almost any command that that may be completed from an ABS/MDMS GUI can also be done from the DCL command line. SLS provided three GUIs based off of 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 and user privileges control information that will be accessible. The ABS/MDMS GUI's can not only be run on an OpenVMS workstation but also served to a Windows box.

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

ABS/MDMS has a more robust scheduler than SLS. Where SLS only had the ability to start its backup based on the day and time with slight variations, an ABS/MDMS schedule can be defined and associated with as many backups as desired.

Configuration

Configuring SLS required the editing of the SYS$MANAGER:TAPESTART.COM file and setting a series of symbols to the appropriate value to meet the needs of the environment. ABS/MDMS does not have a similar file but rather uses a series of settings on objects and one startup file, MDMS$SYSTARTUP.COM for its configuration. Following is a list of the symbols that are found in the SLS TAPESTART.COM procedure. Where available, the equivalent ABS/MDMS policy and field is referenced. Settings that need to be done in the MDMS$SYSTARTUP.COM procedure are listed as well.

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[1]
SYSCLN_RUN		
JUKEBOX	JUKEBOX	/control
JUKEBOX_1_LOWER	JUKEBOX	/cap
JUKEBOX_1_UPPER	JUKEBOX	/cap
SBARLOG	N/A	
SBARINT	N/A	
SBACLAS	N/A	

[1] CLEANUP_Q in SLS also defined the queue that it would run in as well as the hour to start. ABS/MDMS now uses SCHEDULER policies to start cleanup activities on the default queue ABS$nodename.

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

Databases

SLS made use of the following database files found in the location pointed to by the logical SLS$MASTER. Where applicable the equivalent ABS/MDMS database is listed.

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 there is an understanding of the differences between the SLS and ABS/MDMS the next step is to gather information about the currently running SLS system. For those long term SLS users, this may be a good time to review why SLS was setup in the manner that it is. Addendum A contains a work sheet that should prove helpful for capturing the current environment.

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

Assuming there is no other scheduler involved than SLS, one can identify current backups by searching the SLS$SYSBAK:*.com procedures using the following command:

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

Each SBK where these scheduling parameters are in use will return something similar to the following along with the SBK file name:

```
$!
$      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. Note that if the symbols are commented out, the SBK file is not getting scheduled and will need to be looked at as a potential manually submitted backup.

Step 2: Determine the type of each “scheduled” backup.

The name of the SBK should indicate the type of backup it is, whether it’s an image or an incremental. This can be checked by editing the SBK and examining the /qualifiers line for IMAGE, INCREMENTAL, SINCE, etc. Once this has been determined, record the name, start time and day of week to run on the worksheet.

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

This information can be found by in the FILES_n qualifers within each SBK. Record the disks being backed up and associate them with the SBK on the worksheet. For every disk being backed up there will be one associated FILES_n.

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

The symbol MEDIA_TYPE in the SBK will indicate which tape drives are being used and what type it is. For example, a TYPE of TK89 or DLT most likely indicates that they are these types of drives. If it is not evident then the MEDIA_TYPE may be traced back to the SYS$MANAGER:TAPESTART.COM and the associated media triplet. The DRIVES_n that is matched with MTYPE_n will be the correct device. A SHOW DEVICE /FULL from the OpenVMS command prompt will state what type of drive is being used. Record this information to the corresponding SBK information.

Step 5: Determine in which SLS history set that information about each SBK run is being recorded.

Each SBK file will have the symbol HISTORY_SET with a value defined. Record this information on the SLS data sheet. Most likely the name of the history set will correspond to the type it is. For example, IMAGE may indicate that all backups recorded here are of the image type. You will want to take this into consideration when creating new CATALOGS in ABS/MDMS.

Step 6: Check for manually submitted backups.

In some environments, the process is to start SBKs manually as opposed to automatically by using the STORAGE STARTUP SYSTEM_BACKUP command. Any SBKs that may have come up on the initial search without scheduling parameters are candidates for 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 would be hoped that any SBK that is run manually for any reason will be documented.

Log files found in the SLS$SUMMARY_FILES, SLS$MAINTENANCE_LOGS, SLS$SYSBAK_LOGS directories will give a hint as to which backups are running on anyone day. By default, SLS will place log files in these directories from all backup runs. If any manual backups are found, the same information found in steps 1-5 should be recorded.

Step 7: Check for third-party or home grown schedulers.

As with manually submitted backups, the manager of the SLS environment should have information about backups that run using a third-party scheduler. The same information found in Steps 1-5 should be documented in order to integrate them into the ABS/MDMS environment. ABS/MDMS can use most third-party schedulers rather than its own. This will need to be designated during its configuration.

Step 8: Capture other appropriate files.

Since TAPESTART.COM is the main configuration file for SLS, it will be needed to set the appropriate objects and policies in ABS/MDMS. Keep this file available. 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 will indicate days that certain backups are to be skipped. Check this file for any customer scheduling that has been done.

Step 9: Documenting customized code.

It will be important to understand any customized code that has been implemented in your SLS environment. Document what has been changed and why. Knowing what the goal of the changes will be helpful in making sure that ABS/MDMS can carry out the same behavior.

Analyze the Current Backup Environment

This step in the migration allows for making changes to the backup environment from how things were done in SLS. Most likely over a period of time, system managers will have identified more efficient methods to their backup strategies. This will be an excellent time to implement these strategies. It is also at this point that the backup policy documentation should be modified to include these new changes.

Questions that may want to be answered include:

- Are files being backed enough times for a reasonable restore?
- How long will a disk failure take to recover from?
- Are users satisfied with their backups and ability to restore?
- Are all tape devices being used to their fullest capacity?
- Is there functionality in ABS/MDMS that will solve 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?
- Are there any Windows or Unix Servers requiring backing up?
- Are there any Oracle databases that require backing?

Interviewing operators and administrators as well as users who rely on the restore of data is a good idea too. All stakeholders affected by the backup strategy should be taken into consideration. Also, with the focus on security and disaster recovery an analysis of the current environment safety is also appropriate.[2]

[2] HP offers backup audit services to evaluate your current SLS or ABS environment. Contact the support center for more information.

Understanding ABS/MDMS Policies and Objects

ABS/MDMS use objects and polices to define what work it's going to do and the environment that will accomplish it. In order to configure ABS/MDMS an understanding of what the purpose each object and policy will be necessary. See appendix for a listing of how objects and policies relate to each other and the order in which they should be created.

[3]The objects that need to be considered during the install or migration to ABS/MDMS are as follows. Each of these needs to be defined before a SAVE can be initiated. In step 6 of this document symbols in SLS will be associated with this objects.

Consider each of these objects:

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 that tapes may be placed. A hierarchy of locations may be defined as well.

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 defining of each node in the domain that will be running ABS/MDMS.

POOLS: A logical object that contains a set of volumes that can be allocated and used by a set of authorized users.

VOLUMES: A single piece of tape media that any MDMS application can use to store its data.

The SAVE, RESTORE and associated policies are then created using the above objects.[4]

[3] Archive Backup System for OpenVMS – Guide to Operations

[4] Archive Backup System for OpenVMS – Guide to Operations, Chapter 3. Saving and Restoring Data.

The purpose of this strategy is to reduce the amount of redundant data that needs to be entered. For example, where in SLS the MEDIA_FORMAT was required to be modified on every SBK, in ABS/MDMS the MEDIA FORMAT object needs only to be entered once on a policy called an ARCHIVE. This ARCHIVE can then be associated with many SAVE policies.

Following is a list and description of the policies used in the backup environment.

ARCHIVE: An ARCHIVE will define the media type as well as other characteristics about where backup is to be stored. One single ARCHIVE can then be associated with many SAVE policies.

CATALOGS: A CATALOG policy defines where data about each backup will be stored. There are different types of CATALOG polices that serve different purposes including how backup will restore. It's important to understand which type will be most efficient for your site.

ENVIRONMENTS: This policy will describe the criteria under which save and restore requests must execute. As with the ARCHIVE, one ENVIRONMENT can be associated with many SAVES.

SAVE: The SAVE policy defines unique information about the backup. This policy will have other policies associated with it.

RESTORE: The RESTORE policy defines unique information about a restore in order to return data from a tape to disk. Data can be restored by entire disk or down to a file level. The type of CATALOG defined will affect the ability to restore.

SELECTIONS: A SELECTION policy will hold information about files, disks, paths and databases to be saved or restored. A SELECTION can be automatically generated by using the /include qualifier on a SAVE. A SELECTION can also be generated manually be using the CREATE SELECTION DCL command or the MDMS GUI. The SELECTION then in turn needs to be associated with a SAVE or RESTORE.

SCHEDULES: ABS/MDMS provides a flexible scheduler. By default an associated SCHEDULE will be create along with a SAVE. SCHEDULES can also be created manually and then associated with a SAVE or RESTORE.

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 will consist 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 associated them accordingly.

Configuring ABS/MDMS using data gathered from SLS.

After following the tasks outlined in the ABS/MDMS Installation Guide in to get the application installed properly it will be time to tailor your environment to how SLS was accomplishing your backups. Be sure to follow the post-installation tasks in the manual as well.

Appendix B and C of this document contain mappings of data needed from SLS for each object and policy in ABS/MDMS. A good practice will be to go through each of the policies in the appendices and figure out what will need to be created in ABS/MDMS. It will also be necessary to decide how the fields in each will be set. In each of the tables SLS information, where appropriate, is listed. As one works through the tables 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 6 objects will need to be created. There will be two jukeboxes created, two drives and two media types.

This process should continue until all resources in the ABS/MDMS environment have been accounted for. Blank templates are also included in Appendix D that can be printed and copied off for noting what objects and policies need to be created.

ABS/MDMS Object Creation

The command procedure, MDMS$SYSTEM:MDMS$CONFIGURE.COM[5], will help you create the initial needed ten objects. You will be prompted for the information required to create the policy. Note that each policy can be modified after it has been created to fine tune to the needs of the environment. Within the command procedure, help for each prompt is available using "??". Before starting, have the following minimum required information for use with this procedure from this list:

- 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)

[5] Archive Backup System for OpenVMS – Guide to Operations. A – Configuration Example.

- OpenVMS device names – i.e. ALERAB$MKB200:
- Volume naming strategy – Should match bar code label

Once all objects have been successfully created, it will be necessary to prepare the jukebox for use by inventorying and initializing the volumes in the jukebox.

Run the following command to synchronize your ABS/MDMS with your jukebox:

```
$MDMS INVENTORY JUKEBOX your_jukebox_name
```

This command will update the MDMS database with the information it finds in the jukebox's firmware. When the inventory completes and the volumes in the jukebox need to initialized, issue the following command:

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

Once these commands have completed your volumes are ready for use with ABS/MDMS. These steps will have also tested the robotic configuration to ensure that they all work.

ABS/MDMS Policy Creation

With all the objects in place, the backup policies can be created. Since there is no true conversion utility to create these policies from SBK files, they will manually have to be entered. This step however, allows you to make changes and improve any backup inefficiency that may be taking place. The ABS/MDMS GUI can be used to create the policies as well as DCL commands. The order in which to create policies is:

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

RESTORES will be created as needed. SELECTIONS and SCHEDULES are normally created automatically with the SAVEs.

Once the CATALOGS have been created, use your worksheet from the SLS capture to create the ARCHIVE, ENVIRONMENT, and SAVE polices for previous existing SBK files. Remember that the ARCHIVE and ENVIRONMENT policies can be used to group redundant together for like backups. For example, if you have backups that use the same CATALOG policy and MEDIA TYPE, you may want to create an ARCHIVE for all these SAVES. Similar backups that may be grouped together may include image backups of a RAID array that are run on a regular basis. Information on the backups may be retained in the same CATALOG and retained for the same lengths of time. An example of data that may not be candidate to be grouped together might be a monthly backup and a weekly backup where the retention times are different. In this case, different CATALOGS may want to be used requiring a separate ARCHIVE policy. It should be evident that the policy structure needs to be well thought out and planned before beginning. Once in place, ABS/MDMS will provide an efficient and logical method for tracking backups.

Testing the environment

Testing the ABS/MDMS covers three areas, devices, databases and running the actual saves.

Devices

Test your Jukebox by issuing 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, it can be assumed that your jukebox and drives policies are correctly configured. A SHOW DRIVE/CHECK will attempt to physically touch the drive. If this command fails there is most likely a connectivity problem to the drive. Using the MRU application at this point can help diagnose if the problem is with MDMS or with the configuration of the drive.

Remember, if the drive is working with SLS, the problem most likely lies in the MDMS configuration.

Databases

Any of the $SHOW MDMS commands of objects will be an adequate test to ensure that there is database connectivity. If any of these fail or hang a call to the HP customer support center may be required. Test all the SHOW commands as unlike SLS, there are now many different ABS/MDMS database files to start data.

SAVES and RESTORES

SAVES and RESTORES can be tested on line at any point. It may be a good idea to test overall ARCHIVE, SELECTION and SCHEDULE polices by first creating “one_time_only” SAVEs that will automatically purge themselves. Also use a test

CATALOG or set the ARCHIVE with a short retention period or plan to recreate these databases before putting ABS/MDMS into production.

Once comfortable with how ABS/MDMS will schedule and release backups, the next step are to run them in parallel with your existing backups. See Additional Issues and how to run SLS and ABS/MDMS simultaneously. Check the log files found in the directory pointed to by the logical SLS$SYSBAK_LOGS for errors. Other troubleshooting tips will be noted in section 8 of this document.

A defined cut-off date of SLS backups will be needed to be decided upon for when the ABS/MDMS environment will take over responsibility for the backups.

Additional Issues

Oracle Backups

Oracle RDB backups are available in ABS/MDMS as it was in SLS. When a SAVE request in created, select the appropriate DATA_SELECT_TYPE for the version and area of RDB you wish to backup. This will automatically create the SELECTION policy for you. This backup should be scheduled as any other ABS/MDMS backup. Though not required, it would be a good practice to create a CATALOG for only these types of backups.

Note that ABS/MDMS can also backup up Oracle 8i and 9i databases. This functionality is not available in SLS however. Refer to the documentation to learn how to set up this type database.[6]

What to do with SLS data in History Files?

Currently there is no functionality for reading the SLS history files from ABS/MDMS and initiating a backup. Two scenarios are available for recovering pre ABS/MDMS data from SLS:

1). Keep an instance of SLS running for history lookups only. The SLS and ABS licenses will support this environment. Once information is located in SLS, initiate a restore manual restore command using the Backup utility.

2). Capture data from SLS volumes and feed data directly into ABS catalogs. This can prove to be a large task so it is important to consider which backups may need to be restored mostly commonly.

To catalog existing savesets, create a SAVE policy as follows:

[6] Archive Backup System – Guide to Operations “System Backup to Tape for Oracle Databases”

```
MDMS CREATE SAVE mysaveset_catalog –
    /INCLUDE=yourtape:*-
    /DATA_TYPE=VMS_SAVESET-
    /ARCHIVE=my_archive-
    /ENVIRONMENT=my_environment
    /START=your_start_time
```

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

Running SLS and ABS/MDMS Simultaneously

SLS and ABS/MDMS can run simultaneously together on one system. Follow these steps and cautions to set up this environment:

1. In SYS$STARTUP:MDMS$SYSTARTUP.COM set the logical MDMS$VERSION3 to false. This logical was originally intended to have STORAGE commands look at the ABS/MDMS databases. Though this functionality is no longer applicable, setting to false will ensure no interference between the two applications.

2. Install the test ABS/MDMS environment on the appropriate node.

CAUTION: Crossing volumes between the two applications is the greatest danger when running SLS and ABS/MDMS together on the same node. Be sure that one application knows only of its volumes that it can use. This can be accomplished in one of two ways. First after identifying which volumes in the jukebox will be designated for use by SLS and by the ABS/MDMS applications:

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

- Create a pool in SLS called “ABS” and place all the volume to be used by ABS in this pool

The purpose of these steps is to set aside volumes in each application that the other will be using. This will prevent the allocating and use of SLS volume by ABS and vice verse.

A second solution to prevent use of each others application is simply to not define the SLS volumes in ABS or the ABS volumes in ABS. Though the jukebox will be loaded with both, inventory and allocations will overlook volumes they don’t know about.

Troubleshooting

- Startup Issues

The equivalent for SLS$ROOT:[000000]TAPESTARTnodename.COM is found in MDMS$LOG:MDMS$STARTUP_nodename.LOG. Look here for issues that may have come up during the startup of the product. Note too that as with SLS, turning on Opcom may reveal problems such as syntax errors and licensing issues.

- Save and Restore Issues

SLS users are used to reading log files for system backups found in the directory SLS$SYSBAK_LOGS. In the same manner ABS/MDMS will put its log files in the directory ABS$LOG. SAVEs and RESTOREs can be checked for normal completion by scanning their associated log files. The log files by default are named the same as the SAVE policy itself. A helpful trick to monitor these logs is to use the command:

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

This will show you the end of the file as the log buffer is dumped to disk.

- History / Catalog Issues

ABS$CATALOG:Catalog_n.LOG – This log tracks the processing of staging files for catalogs. Check this file if data recently backed up is not showing up in the appropriate catalog. The SLS equivalent are the SLS$SBUPDT.LOG files found in SLS$MAINTENANCE_LOGS.

ABS$CATALOG:ABS$CATALOG_CLEANUP.LOG – Information about the daily cleanup of catalogs and removal of obsolete records will be recorded here. Check this file if you suspect that your catalogs are not be cleaned as volumes free up. In SLS you would have checked the files, SLS$DATA:SYSCLN.LOG and SLS$DATA:CLEANUP.LOG.

- Miscellaneous logs (no SLS equivalents)

ABS$CATALOG:ABS$COORD_CLEANUP_nodename.LOG – The ABS coordinator is responsible for a number of different functions. Should there be a suspected problem with the coordinator, this log will be a starting point for troubleshooting.

MDMS$LOG:MDMS$LOGFILE_DBSERVER.LOG – Tracks events that have happened on the system and errors the MDMS$SERVER process may have run across.

There are other files in the MDMS$LOGFILE directory as well as additional setting for more indepth trouble shooting that are useful in particular troubleshooting situations. HP Services will guide through the use of these if the need arises.

Summary

The keys to a successful migration will include the following:

- ✓ Having a good understanding your current environment in order to capture the environment in ABS/MDMS. Without this understanding you may find yourselves reinventing how you do your backups.

- ✓ Understanding differences between SLS and ABS/MDMS in order to know what changes will be made to your current processes. ABS/MDMS has been defined to handle backups in a more efficient manner overall. Your site may have adapted to using SLS as is and may find change difficult. Be sure to have a change management plan as a part of your migration plan where your operations is explained why and the benefits of moving to ABS/MDMS.

- ✓ Knowing what you want to change with 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 simultaneously for a designated period of time will show that ABS/MDMS has the ability to backup safely. A successful migration period will reduce 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.

- Execution – Perhaps time is limited and one more project is difficult to begin. HP can manage and execute the process for providing documentation and instruction that is required. Any changes would be reviewed by your company to ensure compliance with its standards.

- Help sorting out current SLS environments- Perhaps it has been a very long time since your implementation of SLS 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 current backup strategy – HP engineers experienced in SLS and ABS/MDMS can take a look as a third-party to ensure that your backup strategy is as secure as you believe.

- Reviewing your plans – HP engineers can review your plans and be available for you to bounce your thoughts and ideas off.

To Contact the HP Services VAST team

- Send email to services_engineering2@hp.com or
- Call (888) 376-4737 or
- Ask one of your HP service engineers or Technical Account Managers

Appendix A

SLS Data Gathering Worksheet

SBK Backup Scripts

Image SBK Name	Incremental SBK Name	Disk	Media	Start Days	History

Appendix B

SLS to ABS/MDMS Object Mapping

DOMAIN Policy	SLS Equivalent
Description: Default MDMS Domain	None
Access Control: NONE	None
Last Updated By: ALERAB::SYSTEM	None
Mail: SYSTEM,TED,FIELD	Tapestart.com – TAPEPURGE_MAIL
Offsite Location: VAULT	Tapestart.com – VLT
Onsite Location: COMPUTER_ROOM	Tapestart.com – LOC
Check Access: NO	None
Deallocate State: FREE	Tapestart.com – FRESTA
Default Media Type: DLT	None – Media Type must be specified
Opcom Class: TAPES	Tapestart.com – TOPERS
Relaxed Access: YES	None
Request ID: 691433	None
Protection: S:RW,O:RW,G:R,W	None
DB Server Node: ALERAB	Tapestart.com – PRI
DB Server Date: 15-APR-2005 10:14:13	None
Scheduler Type: INTERNAL	None
Max Scratch Time: NONE	Tapestart.com - MAXSCRATCH
Scratch Time: 0365 00:00:00	Tapestart.com - ALLOCSCRATCH
Transition Time: 0000 14:00:00	Tapestart.com – TRANS_AGE

Drives in SLS are defined in the Tapestart.com. There is no specific record abut each defined drive however, the following list tells where drive characteristics can be found if available.

DRIVE Object	SLS Equivalent
Drive: TZ89	Tapestart.com – MTYPE_n
Description:	None
Access Control: NONE	None
Owner: ALERAB::SYSTEM	None
Device: ALERAB$MKB200	Tapestart.com – DRIVES_n
Nodes: ALERAB	None
Groups:	None
Volume:	Recorded on volume record
Load Volume:	None
Disabled: NO	None
Shared: YES	Tapestart.com – ALLDEV, SELDEV
Available: YES	None
State: EMPTY	None
Stacker: NO	None
Automatic Reply: YES	None
RW Media Types: DLT,DLT_NOCOMP,SDLT	Tapestart.com – MTYPE_n
RO Media Types:	Tapestart.com – MTYPE_n
Access: ALL	None
Jukebox: TL891	Tapestart.com – Jukebox symbols
Drive Number: 0	None
Allocated: NO	None

SLS has no method for grouping drives together.

GROUP Object	SLS Equivalent
Group: JUKEJOINT	None
Description:	None
Access Control: NONE	None
Owner: ALERAB::SYSTEM	None
Nodes: ALERAB	None

SLS does not record specific jukebox characteristics about a drive. It is only defined in the TAPESTART.COM.

JUKEBOX Object	SLS Equivalent
Jukebox: TL891	Tapestart.com – TAPE_JUKEBOXES
Description:	None
Access Control: NONE	None
Owner: ALERAB::SYSTEM	None
Nodes: ALERAB	Tapestart.com – TAPE_JUKEBOXES
Groups:	None
Location: COMPUTER_ROOM	None
Disabled: NO	None
Auto Reply: YES	Tapestart.com or SBK – QUICKLOAD
Access: ALL	None
State: AVAILABLE	None
Control: MRD	None
Threshold: 0	None
Free Volumes: 1	None
Robot: GKB0	Tapestart.com – TAPE_JUKEBOXES
Slot Count: 10	None
Usage: NOMAGAZINE	None

SLS has not specific method for creating locations other than the VLT and LOC symbols in Tapestart.com

LOCATION Object	SLS Equivalent
Location: VAULT	None
Description: Vault offsite location	None
Access Control: NONE	None
Owner:	None
Spaces:	None
In Location:	None

Note that Magazines are never required in ABS/MDMS as it was in some instances of SLS.

MAGAZINE Object	SLS Equivalent
Magazine: TL891	STORAGE ADD MAGAZINE
Description:	None
Access Control: NONE	None
Owner: ALERAB::SYSTEM	None
Placement: ONSITE COMPUTER_ROOM	None
Slot Count: 32	STORAGE ADD MAGAZINE/slot=
Jukebox:	STORAGE IMPORT MAGAZINE
Position:	STORAGE IMPORT MAGAZINE/loc=
Start Slot: NONE	None
Onsite Loc: COMPUTER_ROOM	None
Spaces:	None
Onsite Date: NONE	None
Offsite Loc: VAULT	None
Offsite Date: NONE	None

MEDIA_Type Object	SLS Equivalent
Media type: DLT	Tapestart.com – MTYPE_n
Description: Domain default media type	None
Access Control: NONE	None
Owner: ALERAB::SYSTEM	None
Density:	Tapestart.com – DENS_n
Compaction: YES	Tapestart.com – DENS_n
Capacity: 0	None
Length: 0	None

NODE Object	SLS Equivalent
Node: ALERAB	None
Description:	None
Access Control: NONE	None
Owner: ALERAB::SYSTEM	None
DECnet-Plus Name: LOCAL:.ALERAB	None
TCP/IP Fullname: ALERAB.CXO.CPQCORP.NET:2501-2510	None
Disabled: NO	None
Database Server: YES	None
Location: COMPUTER_ROOM	None
Opcom Classes: TAPES	None
Transports: DECNET,TCPIP	None

POOL Object	SLS Equivalent
Pool: MONTHLY	SLSMGR Menu
Description:	None
Access Control: NONE	None
Owner: ALERAB::SYSTEM	None
Authorized Users:	SLSMGR Menu
Default Users:	None
Threshold: 0	None
Free Volumes: 0	None

VOLUME Object	SLS Equivalent
Volume: 503914	SLS Volume Record
Description:	None
Access Control: NONE	None
Owner: ALERAB::ABS	SLS Volume Record
Placement: JUKEBOX TL891, SLOT 7	SLS Volume Record
Media Types: DLT	SLS Volume Record
Pool:	SLS Volume Record
Error Count: 0	SLS Volume Record
Mount Count: 6	SLS Volume Record
State: FREE	SLS Volume Record
Available State: FREE	None
Previous Volume:	SLS Volume Record
Next Volume:	SLS Volume Record
Format: BACKUP	SLS Volume Record
Last Access Date: 23-MAR-2005 14:33:08	None
Purchase Date: 19-MAY-2004 16:05:46	SLS Volume Record
Creation Date: 19-MAY-2004 16:05:46	None
Initialize Date: 23-MAR-2005 14:26:11	SLS Volume Record
Allocation Date: 07-APR-2005 13:00:31	SLS Volume Record
Scratch Date: 07-APR-2006 13:00:31	SLS Volume Record
Deallocation Date: 03-MAY-2005 13:05:24	Same as Freed Date
Transition Time: NONE	Tapestart.com – TRANS_AGE
Freed Date: 03-MAY-2005 13:05:24	SLS Volume Record – from SLSO menu
Protection: S:RW,O:RW,G:R,W	SLS Volume Record
Username: ABS	Same as owner in SLS
Owner UIC: [ABS]	SLS Volume Record
Account:	None
Job Name: MIKAEL	None
Magazine:	SLS Volume Record
Jukebox: TL891	SLS Volume Record
Slot: 7	SLS Volume Record
Drive: TZ89	SLS Volume Record
Offsite Loc: VAULT	SLS Volume Record
Offsite Date: NONE	SLS Volume Record
Onsite Loc: COMPUTER_ROOM	SLS Volume Record
Space:	None
Onsite Date: NONE	SLS Volume Record
Brand:	SLS Volume Record
Last Cleaned: NONE	SLS Volume Record
Times Cleaned: 0	None
Rec Length: 0	SLS Volume Record
Block Factor: 0	SLS Volume Record

Appendix C

SLS to ABS/MDMS Policy Mapping

SCHEDULE Policy	SLS Equivalent
Schedule: T5_SAVE_SCHED	None
Description: Internal schedule for save T5	None
Access Control: NONE	None
Owner: ALERAB::SYSTEM	None
After Schedule:	None
After When: NONE	None
Command: MDMS RUN SCHEDULE T5_SAVE_SCHED	None
Dates:	None
Days:	DAYS_n in the SBK file
Exclude:	SLS$PARAMS:HOLIDAYS.DAT
Include:	None
Months:	None
Times: 00:00	TIME_n
Weeks:	None
Last Start Date: NONE	None
Next Start Date: NONE	None

Selection Policy	SLS Equivalent
Selection: VMS	None
Description:	None
Access Control: NONE	None
Owner: ALERAB::SYSTEM	None
Agent Qualifiers:	QUALIFIERS symbol in SBK
Before Date: NONE	QUALIFIERS symbol in SBK
Conflict Options: RETAIN_VERSION	QUALIFIERS symbol in SBK
Data Select Type: VMS_SAVESET	BACKUP_TYPE symbol in SBK
Date Type: MODIFIED	QUALIFIERS symbol in SBK
Exclude:	QUALIFIERS symbol in SBK
Include: sys$login:	FILES_n symbol in SBK
Since Date: NONE	QUALIFIERS symbol in SBK
Source Node:	NODE_n in SBK

SAVE Policy	SLS Equivalent
Save: TEST_IMAGE	SBK file name
Description:	None
Access Control: NONE	None
Owner: ALERAB::SYSTEM	None
Archive: MONTHLY	None
Base Date: 11-SEP-2003 14:43:23	None
Delete Interval: NONE	None
Environment: TEST_ENV	None
Epilogue:	POST_PROCESS_EACH
Execution Nodes: ALERAB	NODE_N in SBK
Explicit Interval:	None
Frequency: ON_DEMAND	Note: Leave scheduling symbols blank in SBK file
Groups:	None
Incremental: NO	Qualifiers in SBK File
Job Number: 39	None
Prologue:	PRE_PROCESS_EACH
Schedule: TEST_IMAGE_SAVE_SCHED	None
Sequence Option: SEQUENTIAL	None
Skip Time: NONE	None
Start Date: NONE	TIME_n in SBK File
Transaction Status: ABS_XN_FAILED : ABS_XN_FAILED : Completed with failure	Summary Files
Selections: TEST_IMAGE_SAVE_SEL_DEF	FILES_n in SBK
Default Selection -	
- Data Select Type: VMS_FILES	None
- Include: SYS$SYSDEVICE:	Qualifiers in SBK File
- Exclude:	Qualifiers in SBK File
- Source Node:	NODE_n in SBK File

Restore Policy	SLS Equivalent
Restore: DKA200_IMAGE_RESTORE	Restores created from menus
Description:	None
Access Control: NONE	None
Owner: ALERAB::SYSTEM	None
Archive: MONTHLY	None
Base Date: 11-SEP-2003 15:54:13	None
Before Date: NONE	None
Catalog -	History lookup done through menu
- Name:	None
- Nodes: ABSN1,ALERAB	None
Date Archived: NONE	None
Delete Interval: NONE	None
Destination:	Output Files on menus
Environment: TEST_ENV	None
Epilogue:	None
Execution Nodes: ALERAB	None
Explicit Interval:	None
Frequency: ON_DEMAND	None
Groups:	None
Incremental: YES	None
Job Number: 513	None
Prologue:	None
Schedule: DKA200_IMAGE_RESTORE_REST_SCHED	None
Sequence Option: SEQUENTIAL	None
Since Date: NONE	None
Skip Time: NONE	None
Start Date: NONE	None
Transaction Status: ABS_FAILURE : Abnormally terminated	None
Selections: DKA200_IMAGE_RESTORE_REST_SEL_DEF	None
Default Selection	None
- Data Select Type: VMS_FILES	None
- Include: ALERAB$DKA200:	On SLS Menus
- Exclude:	On SLS Menus
- Source Node:	None

Archive Policy	SLS Equivalent
Archive: DAILY	None
Description:	None
Access Control: NONE	None
Owner: ALERAB::SYSTEM	None
Archive Type: TAPE	None
Catalog -	HISTORY_SET in SBK File
- Name: ABS_CATALOG	HISTORY_SET in SBK File
- Nodes:	None
Consolidation -	None
- Interval: 0001 00:00:00	None
- Savesets: 0	None
- Volumes: 0	None
Destination:	None
Drives:	DRIVE_TYPE in SBK File
Expiration Date: NONE	None
Location: COMPUTER_ROOM	LOC from Tapestart.com
Maximum Saves: 1	None
Media Type: DLT	MEDIA_FORMAT from SBK file
Pool:	POOL from SBK File
Retention Days: 7	None
Volume Sets:	CONTINUATION in SBK File

Environment Policy	SLS Equivalent
Environment: DEFAULT_ENV	None
Description:	None
Access Control: ALERAB::ABS (READ, WRITE, DELETE, SET, SHOW, CONTROL)	None
Owner: ALERAB::ABS	None
Action: RECORD_DATE	QUALIFIERS in SBK
Assist: YES	UNATTENDED_BACKUPS in Tapestart.com
Compression: NONE	None
Data Safety: CRC,FULL_VERIFY,XOR	QUALIFIERS in SBK
Drive Count: 1	N_DRIVES in SBK
Epilogue:	POST_PROCESSING_LAST in SBK
Interval: NONE	None
Links Only: YES	None
Listing Option: NONE	LISTING_GEN in SBK
Lock: NO	None
Notification -	REPLY_MSG & STATUS_MAIL in SBK
- Opcom: TAPES	REPLY_MSG & STATUS_MAIL in SBK
- Type: BRIEF	REPLY_MSG & STATUS_MAIL in SBK
- When: FATAL	REPLY_MSG & STATUS_MAIL in SBK
Notification -	REPLY_MSG & STATUS_MAIL in SBK
- Mail: <REQUESTER>	REPLY_MSG & STATUS_MAIL in SBK
- Type: BRIEF	REPLY_MSG & STATUS_MAIL in SBK
- When: COMPLETE,FATAL	REPLY_MSG & STATUS_MAIL in SBK
Profile -	None
- Cluster: *	None
- Node: *	None
- Privileges:	None
- Rights:	None
- User: ABS	None
Prologue:	POST_PROCESSING_FIRST in SBK
Retry Limit: 0	None
Span Filesytems: YES	None

Catalog Policy	SLS Equivalent
Catalog Name: ABS_CATALOG	HISNAM_1 defined in Tapestart.com
Catalog Node: ALERAB	None
Access Control: NONE	None
Directory: ABS$ROOT:[CATALOG]	HISDIR_1 defined in Tapestart.com
Last Cleaned: 29-APR-2005 12:30:02	None
Owner: ABS	None
Staging : YES	None
Type: FILES	None

Appendix D

Blank Policy and Object Templates

DOMAIN Policy	SLS Equivalent
Description: Default MDMS Domain	
Access Control: NONE	
Last Updated By: ALERAB::SYSTEM	
Mail: SYSTEM,TED,FIELD	
Offsite Location: VAULT	
Onsite Location: COMPUTER_ROOM	
Check Access: NO	
Deallocate State: FREE	
Default Media Type: DLT	
Opcom Class: TAPES	
Relaxed Access: YES	
Request ID: 691433	
Protection: S:RW,O:RW,G:R,W	
DB Server Node: ALERAB	
DB Server Date: 15-APR-2005 10:14:13	
Scheduler Type: INTERNAL	
Max Scratch Time: NONE	
Scratch Time: 0365 00:00:00	
Transition Time: 0000 14:00:00	

GROUP Object	SLS Equivalent
Group: JUKEJOINT	
Description:	
Access Control: NONE	
Owner: ALERAB::SYSTEM	
Nodes: ALERAB	

DRIVE Object	SLS Equivalent
Drive: TZ89	
Description:	
Access Control: NONE	
Owner: ALERAB::SYSTEM	
Device: ALERAB$MKB200	
Nodes: ALERAB	
Groups:	
Volume:	
Load Volume:	
Disabled: NO	
Shared: YES	
Available: YES	
State: EMPTY	
Stacker: NO	
Automatic Reply: YES	
RW Media Types: DLT,DLT_NOCOMP,SDLT	
RO Media Types:	
Access: ALL	
Jukebox: TL891	
Drive Number: 0	
Allocated: NO	

JUKEBOX Object	SLS Equivalent
Jukebox: TL891	
Description:	
Access Control: NONE	
Owner: ALERAB::SYSTEM	
Nodes: ALERAB	
Groups:	
Location: COMPUTER_ROOM	
Disabled: NO	
Auto Reply: YES	
Access: ALL	
State: AVAILABLE	
Control: MRD	
Threshold: 0	
Free Volumes: 1	
Robot: GKB0	
Slot Count: 10	
Usage: NOMAGAZINE	

LOCATION Object	SLS Equivalent
Location: VAULT	
Description: Vault offsite location	
Access Control: NONE	
Owner:	
Spaces:	
In Location:	

MAGAZINE Object	SLS Equivalent
Magazine: TL891	
Description:	
Access Control: NONE	
Owner: ALERAB::SYSTEM	
Placement: ONSITE COMPUTER_ROOM	
Slot Count: 32	
Jukebox:	
Position:	
Start Slot: NONE	
Onsite Loc: COMPUTER_ROOM	
Spaces:	
Onsite Date: NONE	
Offsite Loc: VAULT	
Offsite Date: NONE	

MEDIA_TYPE Object	SLS Equivalent
Media type: DLT	
Description: Domain default media type	
Access Control: NONE	
Owner: ALERAB::SYSTEM	
Density:	
Compaction: YES	
Capacity: 0	
Length: 0	

NODE Object	SLS Equivalent
Node: ALERAB	
Description:	
Access Control: NONE	
Owner: ALERAB::SYSTEM	
DECnet-Plus Name: LOCAL:.ALERAB	
TCP/IP Fullname: ALERAB.CXO.CPQCORP.NET:2501-2510	
Disabled: NO	
Database Server: YES	
Location: COMPUTER_ROOM	
Opcom Classes: TAPES	
Transports: DECNET,TCPIP	

POOL Object	SLS Equivalent
Pool: MONTHLY	
Description:	
Access Control: NONE	
Owner: ALERAB::SYSTEM	
Authorized Users:	
Default Users:	
Threshold: 0	
Free Volumes: 0	

VOLUME Object	SLS Equivalent
Volume: 503914	
Description:	
Access Control: NONE	
Owner: ALERAB::ABS	
Placement: JUKEBOX TL891, SLOT 7	
Media Types: DLT	
Pool:	
Error Count: 0	
Mount Count: 6	
State: FREE	
Available State: FREE	
Previous Volume:	
Next Volume:	
Format: BACKUP	
Last Access Date: 23-MAR-2005 14:33:08	
Purchase Date: 19-MAY-2004 16:05:46	
Creation Date: 19-MAY-2004 16:05:46	
Initialize Date: 23-MAR-2005 14:26:11	
Allocation Date: 07-APR-2005 13:00:31	
Scratch Date: 07-APR-2006 13:00:31	
Deallocation Date: 03-MAY-2005 13:05:24	
Transition Time: NONE	
Freed Date: 03-MAY-2005 13:05:24	
Protection: S:RW,O:RW,G:R,W	
Username: ABS	
Owner UIC: [ABS]	
Account:	
Job Name: MIKAEL	
Magazine:	
Jukebox: TL891	
Slot: 7	
Drive: TZ89	
Offsite Loc: VAULT	
Offsite Date: NONE	
Onsite Loc: COMPUTER_ROOM	
Space:	
Onsite Date: NONE	
Brand:	
Last Cleaned: NONE	
Times Cleaned: 0	
Rec Length: 0	
Block Factor: 0	

SCHEDULE Policy	SLS Equivalent
Schedule: T5_SAVE_SCHED	
Description: Internal schedule for save T5	
Access Control: NONE	
Owner: ALERAB::SYSTEM	
After Schedule:	
After When: NONE	
Command: MDMS RUN SCHEDULE T5_SAVE_SCHED	
Dates:	
Days:	
Exclude:	
Include:	
Months:	
Times: 00:00	
Weeks:	
Last Start Date: NONE	
Next Start Date: NONE	

Selection Policy	SLS Equivalent
Selection: VMS	
Description:	
Access Control: NONE	
Owner: ALERAB::SYSTEM	
Agent Qualifiers:	
Before Date: NONE	
Conflict Options: RETAIN_VERSION	
Data Select Type: VMS_SAVESET	
Date Type: MODIFIED	
Exclude:	
Include: sys$login:	
Since Date: NONE	
Source Node:	

SAVE Policy	SLS Equivalent
Save: TEST_IMAGE	
Description:	
Access Control: NONE	
Owner: ALERAB::SYSTEM	
Archive: MONTHLY	
Base Date: 11-SEP-2003 14:43:23	
Delete Interval: NONE	
Environment: TEST_ENV	
Epilogue:	
Execution Nodes: ALERAB	
Explicit Interval:	
Frequency: ON_DEMAND	
Groups:	
Incremental: NO	
Job Number: 39	
Prologue:	
Schedule: TEST_IMAGE_SAVE_SCHED	
Sequence Option: SEQUENTIAL	
Skip Time: NONE	
Start Date: NONE	
Transaction Status: ABS_XN_FAILED : ABS_XN_FAILED : Completed with failure	
Selections: TEST_IMAGE_SAVE_SEL_DEF	
Default Selection -	
- Data Select Type: VMS_FILES	
- Include: SYS$SYSDEVICE:	
- Exclude:	
- Source Node:	

Restore Policy	SLS Equivalent
Restore: DKA200_IMAGE_RESTORE	
Description:	
Access Control: NONE	
Owner: ALERAB::SYSTEM	
Archive: MONTHLY	
Base Date: 11-SEP-2003 15:54:13	
Before Date: NONE	
Catalog -	
- Name:	
- Nodes: ABSN1,ALERAB	
Date Archived: NONE	
Delete Interval: NONE	
Destination:	
Environment: TEST_ENV	
Epilogue:	
Execution Nodes: ALERAB	
Explicit Interval:	
Frequency: ON_DEMAND	
Groups:	
Incremental: YES	
Job Number: 513	
Prologue:	
Schedule: DKA200_IMAGE_RESTORE_REST_SCHED	
Sequence Option: SEQUENTIAL	
Since Date: NONE	
Skip Time: NONE	
Start Date: NONE	
Transaction Status: ABS_FAILURE : Abnormally terminated	
Selections: DKA200_IMAGE_RESTORE_REST_SEL_DEF	
Default Selection	
- Data Select Type: VMS_FILES	
- Include: ALERAB$DKA200:	
- Exclude:	
- Source Node:	

Archive Policy	SLS Equivalent
Archive: DAILY	
Description:	
Access Control: NONE	
Owner: ALERAB::SYSTEM	
Archive Type: TAPE	
Catalog -	
- Name: ABS_CATALOG	
- Nodes:	
Consolidation -	
- Interval: 0001 00:00:00	
- Savesets: 0	
- Volumes: 0	
Destination:	
Drives:	
Expiration Date: NONE	
Location: COMPUTER_ROOM	
Maximum Saves: 1	
Media Type: DLT	
Pool:	
Retention Days: 7	
Volume Sets:	

Environment Policy	SLS Equivalent
Environment: DEFAULT_ENV	
Description:	
Access Control: ALERAB::ABS (READ, WRITE, DELETE, SET, SHOW, CONTROL)	
Owner: ALERAB::ABS	
Action: RECORD_DATE	
Assist: YES	
Compression: NONE	
Data Safety: CRC,FULL_VERIFY,XOR	
Drive Count: 1	
Epilogue:	
Interval: NONE	
Links Only: YES	
Listing Option: NONE	
Lock: NO	
Notification -	
- Opcom: TAPES	
- Type: BRIEF	
- When: FATAL	
Notification -	
- Mail: <REQUESTER>	
- Type: BRIEF	
- When: COMPLETE,FATAL	
Profile -	
- Cluster: *	
- Node: *	
- Privileges:	
- Rights:	
- User: ABS	
Prologue:	
Retry Limit: 0	
Span Filesytems: YES	

Catalog Policy	SLS Equivalent
Catalog Name: ABS_CATALOG	
Catalog Node: ALERAB	
Access Control: NONE	
Directory: ABS$ROOT:[CATALOG]	
Last Cleaned: 29-APR-2005 12:30:02	
Owner: ABS	
Staging : YES	
Type: FILES	

Appendix E

Object and Policy Relationship

Following is a list of all ABS/MDMS objects and policies along with associated and objects and policies.

Object and Policies	Associate Object and Policies
Domain	Location Node Media Type
Drive	Jukebox Group Drive Node Media Type
Group	Node
Jukebox	Node Group Location
Location	
Node	Location
Magazine	Jukebox Location
Media_Type	
Pool	
Volume	Media_Type Pool Magazine Jukebox Drive Location
Schedule	
Selection	
Save	Environment Group Selection Archive Schedule
Restore	Environment Group

    - Selection
    - Archive
    - Schedule
- Archive
    - Catalog
    - Drive
    - Location
    - Media Type
    - Pool
- Environment
- Catalog

Order in which objects and policies need to be created.

1. Location
2. Node
3. Groups (optional)
4. Jukebox
5. Magazine (optional)
6. Media_type
7. Drive
8. Pool
9. Volumes
10. Catalog
11. Selections (optional)
12. Archive
13. Environment
14. Save
15. Restore
16. Schedule (optional)
17. Domain (Note that the domain is automatically created during initial startup. Adjustments may want to be made after creating other objects)