StorageTek Enterprise Library Software

Managing HSC and VTCS

Release 7.2

E37154-08

September 2016

ORACLE®

StorageTek Enterprise Library Software Managing HSC and VTCS, Release 7.2

E37154-08

Contents

4 Using the VTCS Dashboard

5 Working the VTCS Must Do (Sometime) Chores List

6 Leading Edge Solutions with Management and Storage Classes

7 Finding and Fixing VTCS Problems

A   LCM Control Statements

Index

List of Figures

List of Tables

Preface

Oracle's StorageTek Enterprise Library Software (ELS) is a solution consisting of the following base software:

- StorageTek Storage Management Component (SMC)

- StorageTek Host Software Component (HSC)

- StorageTek Virtual Tape Control Software (VTCS)

- StorageTek Concurrent Disaster Recovery Test (CDRT)

Additionally, the following software is provided with the ELS package:

- StorageTek Library Content Manager (LCM) (formerly ExLM). LCM includes an enhanced version of the product formerly known as Offsite Vault Feature.

- StorageTek Client System Component for MVS Environments (MVS/CSC)

- StorageTek LibraryStation

Audience

This guide is for Oracle or customer personnel who are responsible for managing ELS.

Prerequisites

To perform the tasks described in this guide, you should already understand the following:

- The MVS operating environment

- JES2 or JES3

- Enterprise Library Software (ELS)

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at
http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit
http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit
http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing

impaired.

About This Book

This book is designed for anybody whose job is to keep HSC and VTCS up and running. HSC/VTCS is basically a server, and like all servers, if you install and configure it correctly, construct effective policies, and provide a reasonable amount of careful management, it takes care of itself. HSC and VTCS share some common tasks, such as maintaining the CDS, and have some parallel but separate tasks, such as working with real volumes versus working with VTVs and MVCs. For information about managing the allocation member of the trio, see *Managing SMC*.

What kind of "conscientious management" *is* required? What does it take to "manage HSC and VTCS?" The following list describes the sections of this book and the corresponding management tasks:

- "The Toolkit(s)" describes, in table format, the tools you use to manage HSC and VTCS. This section describes the specific tasks and functions for these tools. For the details, you want to have within easy reach *ELS Command, Control Statement, and Utility Reference*.

- "Managing the CDS" tells how to safeguard the CD and how to recover it if you lose one or more copies of the CDS.

- "Managing a TapePlex" describes the periodic and as-needed tasks required to manage a TapePlex.

- "Using the VTCS Dashboard" talks about the tasks you need to do on an ongoing basis. In addition to monitoring the health of your VTCS system, there are several key items you want to put in place, such as a Demand MVC Space Reclamation batch job.

- "Working the VTCS Must Do (Sometime) Chores List" consists of items you do for VTCS as they need to be done.

  There is some overlap between periodic and as-needed management, just as there is some overlap between these two categories and resolving problems. Most likely, if you are scrupulous about your management tasks, you will not *have* to resolve many problems, and vice versa. "Periodic" and "As-Needed" are two general categories, and you have to adjust according to the needs of your shop.

- "Leading Edge Solutions with Management and Storage Classes" is about Management and Storage Classes, which are the basics of anything leading edge in VTCS land.

- "Finding and Fixing VTCS Problems" is about what to do when problems occur.

- "LCM Control Statements" is a handy cross-reference of things you can do with HSC that you can also do with LCM.

What's New?

This revision includes the following update:

- Added the Bulk CAP size (36 cells) to the CAP size list.

  See "Managing CAPs."

- Added some stipulations when you delete MVCs or VTVs from the configuration.

  See "Working with MVCs" and "Working with VTVs."

1

The Toolkit(s)

The tables below describe the tools you need to manage HSC and VTCS, grouped by the interface. For more information about the tools, see *ELS Command, Control Statement, and Utility Reference*.

> **Note:** **Important: Register for HIPER Notifications.** To help you stay current on maintenance and to proactively avoid problems, Oracle publishes HIPER Notifications as Knowledge Base Alert documents. You can retrieve the full list of HIPER Notifications by searching for keyword "HIPER" from the My Oracle Support (MOS) home page. You should also register to receive new HIPER Notification Alert documents automatically by MOS Hot Topics E-mail. Knowledge Base article 793436.1, searchable by number from the MOS home page, explains the Hot Topics E-mail feature and how to subscribe. Be sure to add all Oracle StorageTek products of interest to your Hot Topics E-mail profile. Simply click **Add** to include a specific product, then choose a product from the list that appears when you type **StorageTek** in the Product field on the **Add Product** box. Be sure to check the Alert document type as you select each product.

The HSC Toolkit

Table 1–1   HSC Operator Commands

Command:	Function:	See This Topic:
CDS ENABLE/DISABLE	Switches, expands, or moves CDS copies.	"Manually Switching to Another CDS Copy"
COMMPATH	Changes communications settings.	"Adding Scratch Volumes"
DISPLAY ACS	Displays the status of your ACSs, which includes hardware statuses for the ACSs, LMUs, and stations; and scratch volumes and free cells available.	"Managing ACSs"
DISPLAY ALL	All of DISPLAY ACS plus LSM and CDS status.	"Managing ACSs"
DISPLAY CAP	Displays CAP information.	"Adding Scratch Volumes"
DISPLAY EXCEPTNS	Displays library hardware problems.	"Managing Library Hardware"
DISPLAY LSM	Displays LSM status.	"Entering Cartridges"

Table 1–1 (Cont.) HSC Operator Commands

Command:	Function:	See This Topic:
DISPLAY SCRATCH	Displays scratch count information based on subpools.	"Adding Scratch Volumes"
DISPLAY THRESHLD	Displays threshold values set by the Warn command.	"Adding Scratch Volumes"
DISPLAY VOLUME	Displays physical volume information.	"Errant and Selected Volumes"
EJECT	Ejects up to 9999 cartridges.	"Entering Cartridges"
ENTER	Enters one or more volumes.	"Entering Cartridges"
MNTD AUTOCLN EJCTAUTO	Determines automatic drive cleaning and automatic ejection of cleaning cartridges that have exceeded their MAXCLEAN value.	"Managing Cleaning Cartridges"
MODIFY	Changes LSM and CAP state.	■ "Managing ACSs" ■ "Adding Scratch Volumes"
OPTION ENTDUP	Determines the response to entering duplicate volumes.	"Entering Duplicate Volumes"
RELEASE	Releases a CAP and makes it available for use.	"Freeing An Allocated CAP"
SMC RESYNCH	Redrive lost mounts.	"Adding Scratch Volumes"
SCRATCH	Scratches specific volumes.	"Adding Scratch Volumes"
UNSCRATCH	Unscratches specific volumes.	"Adding Scratch Volumes"
WARN	Establishes scratch warning threshold values.	"Adding Scratch Volumes"

Table 1–2 HSC PARMLIB Control Statements

Control Statement:	Function:	See This Topic:
CDSDEF	Specifies data set names of CDS copies.	■ "Expanding the CDS" ■ "Renaming a CDS" ■ "Relocating a CDS"

Table 1–3 HSC Utilities (SLUADMIN unless indicated otherwise)

Utility:	Function:	See This Topic:
BACKUP	Backs up the primary CDS.	"Using the BACKup Utility to Back Up the CDS"
FMTLOG	Pre-formats log files.	"Managing CDS Logging"
LOGUTIL	Uses log files to help recover the CDS.	"Restoring the CDS from a Backup Copy"
OFFLOAD LOGFILE	Offloads transactions from any current log files and activates new log files.	"Managing CDS Logging"
RESTORE	Restores the CDS from a backup copy.	"Restoring the CDS from a Backup Copy"
Scratch Conversion(SLUCONDB)	Updates the HSC scratch list from the TMS scratch list.	"Adding Scratch Volumes"

Table 1–3   (Cont.) HSC Utilities (SLUADMIN unless indicated otherwise)

Utility:	Function:	See This Topic:
SCREDIST	Balances the number of scratch volumes between specified LSMs in an ACS. Provides an option to balance by subpool.	"Adding Scratch Volumes"
SET LOGFILE	Specifies log file data sets.	"Managing CDS Logging"
SET VOLPARM	Defines all types of volumes and pools.	■ "Adding Scratch Volumes" ■ "Defining Nearline Volumes" ■ "Adding Scratch Volumes"
PITCOPY	Backs up the primary CDS (point in time).	"Using PITCOPY to back up the CDS"
UNSELECT	Resets volume selected status.	"Errant and Selected Volumes"
VOLRPT	Provides physical volume reports.	"Using a Volume Report"

Table 1–4   Diagnostics

Diagnostic Tool:	Function:	See This Topic/Document:
Generalized Trace Facility	The Generalized Trace Facility (GTF) provides a low level history of the recent past.	"Adding Scratch Volumes"
Supervisor Call and Abnormal End Dumps (ABEND)	SVC and ABEND dumps are taken where appropriate to diagnose software failures.	"Adding Scratch Volumes"
Error Recording Data Set Records	Software failures are logged into these error recording data sets. 4480 Cartridge Subsystem and 3278 Terminal Subsystem Error Recording Data Set (ERDS) records are written by MVS routines.	"Adding Scratch Volumes"
HSC LIST Command	The LIst command is used to display an HSC data structure and to provide for diagnostic support for the HSC. Use this command under the direction of Oracle support personnel.	*ELS Command, Control Statement, and Utility Reference*

Table 1–5   HSC Performance Tools

Performance Tool:	Function:	See This Topic:
Activities report	Produces a detailed report identifying performance values for various library activities.	"Optimizing Mount Performance"
Scratch Redistribution utility	Redistributes scratch volumes across the LSMs within a specific ACS.	■ "Optimizing Mount Performance" ■ "Adding Scratch Volumes"

Table 1–5 (Cont.) HSC Performance Tools

Performance Tool:	Function:	See This Topic:
WARN command	Establishes thresholds which control when HSC notifies the operator of an insufficient quantity of scratch cartridges in an ACS.	"Adding Scratch Volumes"
CAPPREF command	Specifies preference values for CAPs.	"Adding Scratch Volumes"
SMF records	Logs ACS activity and performance data for analysis.	"Optimizing Mount Performance"

The VTCS Toolkit

Table 1–6 The VTCS Toolkit

Command:	Function:	See This Topic/Document:
AUDIT	Updates the MVC and VTV information in the HSC CDS.	■ "Changing RTD Device Types" ■ "Changing RTD Device Types" ■ "Logically Dismounting VTVs in an Offline VTSS"
DELETSCR	Deletes scratch VTVs from VTSSs and unlinks any migrated VTVs from MVCs	"Doing Demand MVC Space Reclamations"
DISPLAY	Displays the status of All Things VTCS and, as such is one of your Primary tools.	■ "Checking Virtual Tape Status (Daily)" ■ "Checking Virtual Tape Status (Daily)" ■ "Doing Demand MVC Space Reclamations" ■ "Doing Demand MVC Space Reclamations" ■ "Poor VTV Mount Performance" ■ "Poor VTV Mount Performance" ■ "RTD/MVC Failures" ■ "Poor VTV Mount Performance"
MIGRATE	Demand migrates VTVs to MVCs.	"Doing Demand MVC Space Reclamations"
MVCDRAIN	Recalls all current and scratched VTVs from an MVC and, optionally, "virtually" ejects the MVC (makes it unavailable for VSM use without physically ejecting it from the library).	"Doing Demand MVC Space Reclamations"
MVCMAINT	Sets MVC attributes.	"Changing RTD Device Types"
MVCPLRPT	Manages your MVCPOOLS and Storage Classes.	*ELS Command, Control Statement, and Utility Reference*

Table 1–6   (Cont.) The VTCS Toolkit

Command:	Function:	See This Topic/Document:
MVCRPT	Reports the status of your VSM system's MVCs and is Primary Wellness Meter #2.	■ "Checking Virtual Tape Status (Daily)" ■ "Doing Demand MVC Space Reclamations" ■ "Using PITCOPY to back up the CDS" ■ "Fixing Common Problems"
RECALL	Does demand VTV recalls.	"Doing Demand MVC Space Reclamations"
RECLAIM	Does demand MVC space reclamation.	"Doing Demand MVC Space Reclamations"
RECONCIL	Reconciles VTV media and location (moves VTVs from one Storage Class to another).	"Working with VTSSs"
The RTV utility	Converts VTVs contained on MVCs to data sets on Nearline volumes (real tape volumes).	"Fixing Common Problems"
SET MIGOPT	Changes the following migration parameters: ■ Maximum and minimum concurrent automatic migration, immediate migration, and migrate-to-threshold tasks. ■ High and low AMTs.	■ "Checking Virtual Tape Status (Daily)" ■ "Doing Demand MVC Space Reclamations"
VARY RTD	Changes RTD state	■ "Checking Virtual Tape Status (Daily)" ■ "Working with VTSSs"
VARY VTSS	Changes VTSS states on all hosts.	"Working with VTSSs"

2

Managing the CDS

Managing the CDS, a key task because of the critical data that the CDS contains. The following sections tell how to safeguard the CDS and how to recover it if you lose one or more copies of the CDS. For information about planning and creating the CDS, see *Configuring HSC and VTCS*.

Backing Up the CDS

If all copies of the CDS fail (that is, you lose the Primary, Secondary, and Standby copies), you must restore the CDS from a backup. You can back up the CDS using non-ELS facilities such as disk mirroring, third party backup software such as IBM DFSMSdss and Innovation FDR, or third party Point-In-Time Copy software such as IBM FlashCopy. Oracle, however, does not recommend these methods, as they do not guarantee CDS consistency and integrity.

Instead, Oracle recommends the following for CDS backup:

- If you have facilities to create a snapshot backup, then use the HSC PITCOPY utility, which invokes these utilities after ensuring CDS consistency. Backup times, especially for a large CDS, is significantly reduced compared to the standard BACKUP utility.

- If you do not have facilities to create a snapshot backup, then use the HSC BACKUP utility.

Frequency of Backup

Oracle recommends that you back up the CDS at least daily. If using CDS Logging, you should also run a Log File Offload immediately after you back up the CDS so that during recovery, you can select the log offload file that corresponds to the backup file you use for recovery.

You should also back up the CDS before configuration changes and other significant events, including, but not limited to:

- Running the SLUADMIN SET utility
- Running the VTCS CONFIG utility
- Moving a CDS
- Expanding a CDS
- Merging CDSs.

The PITCOPY function does not require HSC to be active. However, if HSC is not active, then DD statements must be provided for the CDS data sets.

Using PITCOPY to back up the CDS

The PITCOPY utility supports the following snapshot methods:

- StorageTek SVAA SnapShot (SIBBATCH)

- DFSMSdss SnapShot / DFSMSdss FlashCopy (ADRDSSU)

- FDR-invoked SnapShot (FDRSNAP)

Consult the documentation of the utility for your environment for required JCL DD statements and control cards.

In addition to DD statements required by the SLUADMIN utility and by the snapshot utility, the PITCOPY function requires the following DD statements:

- `SLSPARMP`, a data set containing snapshot control statements to back up the data set defined in your environment as SLSCNTL.

- `SLSPARMS`, a data set containing snapshot control statements to back up the data set defined in your environment as SLSCNTL2 (if any).

- `SLSPARMB`, a data set containing snapshot control statements to back up the data set defined in your environment as SLSSTBY (if any).

- `SYSIN`, must be defined as a data set that can be both written and read by the PITCOPY utility.

The PITCOPY function does not require HSC to be active. However, if HSC is not active, then DD statements must be provided for the CDS data sets.

The following example shows general JCL required for all PIT copy methods.

```
//PITCOPY EXEC PGM=SLUADMIN,PARM='MIXED'
//STEPLIB  DD DSN=hlq.SEALINK,DISP=SHR
//* If HSC IS NOT OR MAY NOT BE ACTIVE, INCLUDE THE
//* FOLLOWING:
//SLSCNTL DD DSN=primary.cds.name,DISP=SHR
//SLSCNTL2 DD DSN=secondary.cds.name,DISP=SHR
//SLSSTBY   DD DSN=standby.cds.name,DISP=SHR
//SLSPARMP DD DSN=hlq.PARMLIB(BKPCNTL),DISP=SHR
//SLSPARMS DD DSN=hlq.PARMLIB(BKPCNTL2),DISP=SHR
//SLSPARMB DD DSN=hlq.PARMLIB(BKPSTBY),DISP=SHR
//SYSIN          DD UNIT=SYSDA,SPACE=(TRK,1)
//* THE FOLLOWING IS USED BY THE SNAPSHOT UTILITY:
//SYSPRINT  DD  SYSOUT=*
//SLSPRINT   DD SYSOUT=*
//SLSIN        DD *
 PITCOPY METHOD=(SIBBATCH|ADRDSSU|FDRSNAP)
```

Consult the user's guide for the snapshot utility you are using for additional JCL requirements.

- The control cards in SLSPARMP, SLSPARMS, and SLSPARMB are those needed to invoke the snapshot facility. Consult the user's guide for the snapshot utility you are using for the format of the required control cards. You should use a control card option that requests no enqueue on the data set, as the PITCOPY utility ensures serialization.

Using the BACKup Utility to Back Up the CDS

Running the BACKup utility:

1. Allocate the SLSBKUP data set.

It should be slightly larger than the CDSs. The size depends upon the number of discrepancies found. However, a recommended size is five percent larger than the current CDS.

The backup data set is formatted as RECFM=F, BLKSIZE=4096. The logical record length may be LRECL=0. In this case, LRECL=0 indicates to default to the block size.

2. Run the BACKup utility.

The CDS is placed in a reserve state only long enough for the data to be copied to the SLSBKUP data set. No other functions can access the CDS data sets during backup processing:

■ If HSC is not active, you must identify the primary, secondary, and standby CDSs to be used for the backup by specifying the SLSCNTL, SLSCNTL2, and SLSSTBY DD statements in the JCL.

■ If the HSC is active and you identify the primary, secondary, and standby CDSs to be used for the backup by specifying the SLSCNTL, SLSCNTL2, and SLSSTBY DD statements in the JCL, the DD statements are used even if HSC is active on the host where the utility is being executed. If the HSC is active and these DD statements are omitted, the data sets used by the active HSC on the host where the utility is being executed are used.

Example JCL for backing up the Primary, Secondary, and Standby CDSs:

```
//JOBBKUP   job (account),programmer
//S1        EXEC PGM=SLUADMIN,PARM=MIXED
//STEPLIB     DD DSN=hlq.SEALINK,DISP=SHR
//SLSBKUP   DD DSN=backup.dataset.name(+1),DISP=(NEW,CATLG,DELETE),
//            UNIT=SYSDA,SPACE=
//SLSCNTL   DD DSN=primary.dataset.name,DISP=SHR
//SLSCNTL2 DD DSN=secondary.dataset.name,DISP=SHR
//SLSSTBY DD DSN=standby.dataset.name,DISP=SHR
//SLSPRINT  DD SYSOUT=A
//SLSIN     DD *
  BACKUP
```

Managing CDS Logging

The HSC Transaction Logging Service is an internal HSC service that records information into one or, optionally, two Log Files. If all copies of the CDS fail (that is, you lose the Primary, Secondary, and Standby copies), recovery from a backup can be time consuming. When recovering from a backup, CDS logging can aid in resynchronizing the CDS with the data content of the VSM subsystem. If you are initially setting up logging in your configuration, see *Configuring HSC and VTCS*. The following sections tell how to manage CDS Logging.

> **Note:** Recovery using log files involves an analysis of the log data and construction of the required MVC audit statements to restore access to the latest versions of data. This does not recover the CDS to its exact state before the failure! For example, a VTV can exist on multiple MVCs. The audit recovers access to all copies of that VTV, but the copies may not be on exactly the same MVCs as before a CDS loss. The objective of CDS logging is to recover access to data, which is not a bit by bit CDS recovery!

Periodically Offloading Log Files

You need to periodically offload log files to keep them from filling up, which stops logging. Here are some Best Practices for doing so:

- Set up a PROC to periodically offload the log files. Experimenting for a while to see how long it takes for the log file to fill under normal or peak operations, and then run log offload at that interval.

- Also consider setting up your auto-ops package to intercept the SLS1781W message, which is displayed starting at 70% full for the Log File data sets.

- You should also schedule a log file offload to be run when you run your periodic CDS backup so that during recovery, you can select the log offload file that corresponds to the backup file you use for recovery.

- If the CDS is lost (all three copies are gone) or unreliable for any reason, do not attempt to offload the log file, instead see "Restoring the CDS."

Expanding or Reallocating Log Files

If you need to expand your log files, or re-allocate them for any reason, do the following:

1. Create and format new log files.

   The size of the allocated log files depends on the number of transactions your system generates, as well as the frequency of performing the offload. Oracle suggests you allocate and activate some test log files and determine the rate at which they fill before determining the size of the production log files.

   Sample JCL:

   ```
   //FMTLOG  JOB (account),REGION=1024K
   //S1 EXEC PGM=SLUADMIN,PARM=MIXED
   //STEPLIB DD DSN=hlq.SEALINK,DISP=SHR
   //SLSLOG1 DD DSN=hlq.CDSLOG1,DISP=(,CATLG,DELETE),
   // UNIT=SYSDA,SPACE=(CYL,100)
   //SLSLOG2 DD DSN=hlq.CDSLOG2,DISP=(,CATLG,DELETE),
   // UNIT=SYSDA,SPACE=(CYL,100)
   //SLSPRINT DD SYSOUT=*
   //SLSIN DD *
   FMTLOG
   ```

2. Run `SET LOGFILE` to specify the new log files.

   Sample JCL:

   ```
   //SETLOG  JOB (account),REGION=1024K
   //S1 EXEC PGM=SLUADMIN,PARM=MIXED
   //STEPLIB DD DSN=hlq.SEALINK,DISP=SHR
   //SLSPRINT DD SYSOUT=*
   //SLSIN DD *
   SET LOGFILE(hlq.CDSLOG1,hlq.CDSLOG2)
   ```

3. Run `OFFLOAD LOGFILE` to offload the old log files.

   ```
   //JOBOFFL job (account),programmer
   //S1        EXEC PGM=SLUADMIN,PARM=MIXED
   //STEPLIB      DD DSN=hlq.SEALINK,DISP=SHR
   //SLSOFFLD DD DSN=offload.dataset.name(+1),DISP=(NEW,CATLG,DELETE),
   //            UNIT=SYSDA,SPACE=
   //SLSCNTL   DD DSN=primary.dataset.name,DISP=SHR
   ```

```
//SLSCNTL2 DD DSN=secondary.dataset.name,DISP=SHR
//SLSSTBY DD DSN=standby.dataset.name,DISP=SHR
//SLSPRINT  DD SYSOUT=A
//SLSIN     DD *
OFFLOAD LOGFILE
```

This automatically causes HSC to switch to the new log files.

Manually Switching to Another CDS Copy

As described in "Backing Up the CDS," if you specify multiple CDS copies, HSC automatically switches to another CDS copy if the Primary CDS fails. Why would you want to manually switch to another CDS copy? Basically, you do so in either of the following situations:

- You want to move the CDS and/or expand it.

- After an HSC automatically switches CDS copies, you want to delete the old Primary, reallocate it, and repopulate it. Once you enable the reallocated Primary CDS, HSC recopies the valid CDS data to it from a backup copy.

You use the CDS ENABLE/DISABLE commands to switch, expand, or move CDS copies. For more information, see *ELS Command, Control Statement, and Utility Reference*.

Expanding the CDS

You can use the CDS EXpand command to add VOLPARM data, VTCS CONFIG data, or vaulted volume data to the CDS. To make other configuration changes that may require increasing the size of the CDS (such as adding new ACSs and LSMs when the LIBGEN did not use the FUTRACS and FUTRLSM parameters) you must use other methods instead of CDS EXpand. See "Reconfiguring a TapePlex" in *Configuring HSC and VTCS* for more information.

> **Caution:** *Oracle recommends backing up all CDS copies before issuing the CDS EXpand command.* Failures during the expand operation usually cause the CDS to become unusable. It is important to back up the CDS before invoking the CDS EXpand command to make sure that the latest copy of the CDS is available in case of a failure during the expand operation.

The following are guidelines for CDS EXpand:

- HSC must be active and tape activity can be ongoing when using CDS EXpand.

- Do not issue CDS Enable and/or Disable commands simultaneously from multiple hosts. Doing so can cause the HSC to issue error messages. If this occurs, refer to *ELS Messages and Codes* to determine if user action is required.

- Update your CDSDEF control statement in the HSC PARMLIB as necessary to make it consistent with the results of any expand operation.

- Restoring the CDS to a larger pre-allocated data set cannot be used to dynamically expand a CDS; the additional space is not formatted and cannot be used as is

- Oracle recommends that all CDSs be allocated with the same amount of DASD space. The usable space in the CDS is determined by the number of 4096 byte blocks that fit in the CDS with the smallest DASD space allocation.

To expand all CDSs:

1. Disable each CDS (one at a time) with the CDS DISABLE command.

2. Reallocate each CDS with a larger primary space quantity.

   Do not specify a secondary space quantity; all CDSs must have only a single extent.

3. Re-enable each CDS with the CDS ENABLE command.

4. Enter the CDS EXpand command.

   All CDSs that are known to the HSC (that is, are enabled) are expanded at the same time with a single invocation of the CDS EXpand command.

Rotating the CDSs

CDS names do not change as the result of a CDS automatic or manual switch. You can use a sequence of CDS Disable and Enable commands to restore the order and match the data set names to their expected roles.

To rotate the CDSs:

1. Enter a Display CDS command to view the current status and assignment of the CDSs:

```
SYS000001 = ELS.SYSCNTL2
PRIVOL = PROD02 FLAGS(40) ACTIVE
SYS000002 = ELS.SYSCNTL
SECVOL = PROD01 FLAGS(40) ACTIVE
```

   The display shows that SYSCNTL (the original Primary) is now the Secondary and SYSCNTL2 (the original Secondary) is now the Primary.

2. Disable the current Primary:

```
CDS DISABLE PRIMARY
```

   ELS.SLSCNTL2 is disabled, and ELS.SLSCNTLT becomes the new Primary.

3. Re-enable ELS.SLSCNTL2 as the new active Secondary:

```
CDS ENABLE DSN(ELS.SLSCNTL2)
```

4. Enter Display CDS command to confirm that the CDSs are rotated:

```
SYS000001 = ELS.SYSCNTL
PRIVOL = PROD02 FLAGS(40) ACTIVE
SYS000002 = ELS.SYSCNTL2
SECVOL = PROD01 FLAGS(40) ACTIVE
```

Renaming a CDS

There are two methods for renaming a CDS:

■ Use the `CDS Disable` and `Enable` commands to do the rename without stopping HSC.

■ Stop HSC and use the `BACKup` and `RESTore` utilities to do the rename.

The obvious disadvantage to using the `BACKup` and `RESTore` utilities is that HSC must be stopped; however, the advantage is that a backup of the CDS is created.

Renaming a CDS Using the CDS Disable and Enable Commands

To rename a CDS using commands:

1. Disable the CDS you want to rename on all hosts.

   For example:

   ```
   CDS DISABLE DSN(ACS.DBASEOLD)
   ```

2. Rename the data set that contains the CDS.

   For example, rename `ACS.DBASEOLD` to `ACS.DBASECPY`.

3. Enable the CDS by its new name.

   For example:

   ```
   CDS ENABLE DSN(ACS.DBASECPY)
   ```

   If the `Enable` command fails for the renamed CDS, CDS definitions are restored to what they were before the command was issued. Modify the CDSDEF control statement in the HSC PARMLIB to keep it consistent with the active CDS.

   The CDS names are recorded in the Database Heartbeat (DHB) record. During HSC initialization, the data set names in the DHB are compared with the data set names specified on the CDSDEF control statement in the HSC PARMLIB.

   If a data set name specified on the CDSDEF statement does not match any of the CDS names recorded in the DHB, that CDS is disabled. If all the specified CDSs are disabled, the HSC does not initialize. The specific assignment of enabled CDSs to the primary, secondary, and standby CDS(s) is made based on the CDS assignments recorded in the Database Heartbeat record.

   If the CDS(s) needs to be renamed, use the HSC BACKup and RESTore utilities as described in the rename procedure below. Do not use MVS, TSO ISPF, or third-party utilities to rename the CDS(s) without running the BACKup and RESTore utilities.

Relocating a CDS

To relocate a CDS:

1. Disable the CDS you want to relocate on all hosts.

   For example:

   ```
   CDS DISABLE DSN(ACS.DBASECPY)
   ```

2. Allocate a new data set with the appropriate CDS attributes.

   For more information, see *Configuring HSC and VTCS*.

3. Enable the CDS the data set allocated in step 2:

   For example:

   ```
   CDS ENABLE DSN(ACS.DBASENEW)NEWLOC
   ```

   Note: MVS uses catalog services to resolve the volume and unit definitions, if not specified.

If a rename or relocate operation fails, CDS definitions are restored to what they were before the command was issued. Modify the CDSDEF control statement to keep it consistent with the active CDS.

Relocating an Uncataloged CDS Copy

Assume that ACS.DBASECPY has been disabled and deleted (or uncataloged), and ACS.NOTCATLG has been allocated and cataloged. The following command relocates an uncataloged CDS copy:

```
CDS ENABLE DSN(ACS.NOTCATLG) NEWVOL(HSC001),NEWUNIT(ABC)
```

Modify the CDSDEF control statement to make it consistent with the CDS definitions in this command. If a rename or relocate operation fails, CDS definitions are restored to what they were before the command was issued.

Swapping a CDS

In contrast with relocating a CDS using HSC commands or utilities, there are a number of third-party software products that allow disk data sets to be swapped from one location to another without affecting the users of those data sets. The HSC has been enhanced to dynamically update the unit address information stored in the CDS within a few seconds after a TDMF, FDRPAS, or P/DAS swap operation takes place so that HSC and VTCS utilities always use the new CDS location.

This enhancement does not provide an unrestricted ability to move a CDS. Moving a CDS by a means other than a PPRC or swap operation requires a change to the VOLSER of the CDS. The traditional process of a SLUADMIN BACKup and RESTore of the CDS must be used to reset both unit address and volume serial information if the VOLSER of the CDS changes.

Restoring the CDS

If all copies of the CDS fail (that is, you lose the Primary, Secondary, and Standby copies), you must restore the CDS from a backup. If you lose a single copy of the CDS (and you have multiple copies), you can use the CDS ENABLE/DISABLE commands to continue running on the valid copy and recreate the copy that has the error. For more information, see "Manually Switching to Another CDS Copy."

Caution: If you think you have lost all copies of the CDS, contact Oracle Software Support to ensure that restoring the CDS from a backup is actually required and to assist, if necessary, with the restore process.

Note:

- The RESTORE process is identical whether you have used PITCOPY or SLUADMIN BACKUP to create the CDS backup.

- Before doing a restore, you should either do a new backup of the corrupt CDS, or rename all the current CDS copies to preserve them for later analysis.

Restoring the CDS from a Backup Copy

To restore the CDS from a backup copy:

1. If HSC is still active on any host, stop the active HSC systems.

    If logging is enabled, continue with step 2 to resynchronize VSM data. Otherwise, go to step 3.

2. If the CDS can be accessed, run the normal log file offload job.

    This offload is to capture the VSM activity since your last offload.

    ```
    //JOBOFFL job (account),programmer
    //S1       EXEC PGM=SLUADMIN,PARM='execution parameters'
    //STEPLIB     DD DSN=hlq.SEALINK,DISP=SHR
    //SLSOFFLD DD DSN=offload.dataset.name(+1),DISP=(NEW,CATLG,DELETE),
    //            UNIT=SYSDA,SPACE=
    //SLSCNTL   DD DSN=primary.dataset.name,DISP=SHR
    //SLSCNTL2 DD DSN=secondary.dataset.name,DISP=SHR
    //SLSSTBY DD DSN=standby.dataset.name,DISP=SHR
    //SLSPRINT  DD SYSOUT=A
    //SLSIN     DD *
    OFFLOAD LOGFILE
    ```

    If the CDS cannot be accessed, offload the log file using the `LOGDSN` parameter. The normal CDS logging process uses log information stored in the CDS. If the CDS is inaccessible, then you need to use the logging information that is in the current log file, which you specify with the `LOGDSN` parameter.

    ```
    //JOBOFFL job (account),programmer
    //S1       EXEC PGM=SLUADMIN,PARM='execution parameters'
    //STEPLIB     DD DSN=hlq.SEALINK,DISP=SHR
    //SLSOFFLD DD DSN=offload.dataset.name(+1),DISP=(NEW,CATLG,DELETE),
    //            UNIT=SYSDA,SPACE=
    //SLSPRINT  DD SYSOUT=A
    //SLSIN     DD *
    OFFLOAD LOGFILE LOGDSN(hsc.log1)
    ```

    > **Note:** If the CDS cannot be accessed, *note that* if you attempt to run the `OFFLOAD LOGFILE` utility after restoring the CDS but without using the `LOGDSN` option, the offload may not succeed, or may create an erroneous log offload file.

3. Restore the CDS from the last good backup.

    ```
    //JOBREST job (account),programmer
    //S1       EXEC PGM=SLUADMIN,PARM='execution parameters'
    //STEPLIB     DD DSN=hlq.SEALINK,DISP=SHR
    //SLSBKUP   DD DSN=backup.dataset.name(0),DISP=SHR
    //SLSCNTL   DD DSN=primary.dataset.name,DISP=SHR
    //SLSCNTL2 DD DSN=secondary.dataset.name,DISP=SHR
    //SLSSTBY DD DSN=standby.dataset.name,DISP=SHR
    //SLSPRINT  DD SYSOUT=A
    //SLSIN     DD *
     RESTORE
    ```

> **Caution:** *It is critical that your RESTORE JCL specifies all HSC data sets in use (Primary, Secondary, Standby).*

For more information, see *ELS Command, Control Statement, and Utility Reference*.

> **Note:** Special precautions should be taken for running BACKup and RESTore when local/remote-linked libraries run CDSs independent of each other. These precautions should be followed when the link connecting the local and remote libraries is disrupted. See *ELS Legacy Interfaces Reference* for more information.

If you are using log files for recovery, continue with step 4. Otherwise, go to step 9.

4. Ensure that the restored CDS is referencing the correct log file information:

   - Run `SET LOGFILE(OFF)`.

   ```
   //SETLOG  JOB (account),REGION=1024K
   //S1 EXEC PGM=SLUADMIN,PARM='execution parameters'
   //STEPLIB DD DSN=hlq.SEALINK,DISP=SHR
   //SLSPRINT DD SYSOUT=*
   //SLSIN DD *
   SET LOGFILE(OFF)
   ```

   - Run the `FMTLG` utility to reformat the log files.

   ```
   //FMTLOG  JOB (account),REGION=1024K
   //S1 EXEC PGM=SLUADMIN,PARM='execution parameters'
   //STEPLIB DD DSN=hlq.SEALINK,DISP=SHR
   //SLSLOG1 DD DSN=hlq.CDSLOG1,DISP=(,CATLG,DELETE),
   // UNIT=SYSDA,SPACE=(CYL,100)
   //SLSLOG2 DD DSN=hlq.CDSLOG2,DISP=(,CATLG,DELETE),
   // UNIT=SYSDA,SPACE=(CYL,100)
   //SLSPRINT DD SYSOUT=*
   //SLSIN DD *
   FMTLOG
   ```

   - Run `SET LOGFILE` again, specifying the names of the newly initialized log files.

   ```
   //SETLOG  JOB (account),REGION=1024K
   //S1 EXEC PGM=SLUADMIN,PARM='execution parameters'
   //STEPLIB DD DSN=hlq.SEALINK,DISP=SHR
   //SLSPRINT DD SYSOUT=*
   //SLSIN DD *
   SET LOGFILE(hlq.CDSLOG1,hlq.CDSLOG2)
   ```

5. Run the `LOGUTIL` command to generate a data set with the `AUDIT MVC` commands required for recovery and run the commands in that data set.

   In the following example, assume that last known good CDS backup was taken on 5/12/2009 at 10 p.m:

   - The `LOGIN` statement specifies a DD statement for each offload log file, beginning with the log file created with the last known good backup and ending with the log file that you created in 2.

   - `LOGUTIL GENAUDIT` specifies `RECVCMD` as the DD name of the data set that contains the commands for recovery of the VSM data.

   ```
   //JOBLOGR job (account),programmer,REGION=1024k
   ```

```
//S1 EXEC PGM=SLUADMIN,PARM='execution parameters'
//STEPLIB      DD DSN=hlq.SEALINK,DISP=SHR
//LOGIN        DD DSN=offload.dataset.name(-2),DISP=OLD
//             DD DSN=offload.dataset.name(-1),DISP=OLD
//             DD DSN=offload.dataset.name(0),DISP=OLD
//RECVCMD      DD DSN=offload.dataset.genaudit,DISP=(,CATLG,DELETE),
//                UNIT=SYSDA,SPACE=
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
LOGUTIL FROMDATE(2009-5-12) FROMTIME(22:00:00)  LOGDD(LOGIN)
GENAUDIT COMMANDS(RECVCMD)
```

6. If `LOGPOL=REQUIRED` is currently specified in the VTCS configuration:

Specify `LOGPOL=OPTIONAL` in the `CONFIG GLOBAL` statement and run `CONFIG` to temporarily disable logging in the VTCS configuration. This is required to successfully execute the `MVCMAINT` commands while HSC/VTCS is not active.

7. Set the MVCs to be audited to read-only status:

Execute the `MVCMAINT READONLY(ON)` commands for the MVCs to be audited with HSC/VTCS inactive. This ensures that the MVCs are not used when HSC/VTCS is active.

```
//JOBAUDIT job (account),programmer,REGION=1024K
//S1 EXEC PGM=SLUADMIN,PARM='execution parameters'
//STEPLIB DD DSN=hlq.SEALINK,DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
MVCMAINT READONLY(ON) MVC( +
mvc1 +
mvc2 +
…      +
mvcn +
        )
```

8. If LOGPOL=REQUIRED is required in the VTCS configuration:

Specify `LOGPOL=REQUIRED` in the `CONFIG GLOBAL` statement and run `CONFIG` if `LOGPOL=OPTIONAL` was temporarily specified at step 6 (above).

9. Restart HSC on one hosts, but do not start tape activity.

10. To recover the non-VSM portion of the CDS, run `SLUADMIN AUDIT` on all ACSs.

```
//JOBAUDIT job (account),programmer,REGION=1024k
//S1 EXEC PGM=SLUADMIN,PARM='execution parameters'
//STEPLIB      DD DSN=hlq.SEALINK,DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
AUDIT ALL
```

> **Note:** Once the library audit is started, you can simultaneously run step 11, step 12, and step 13.

11. Audit all VTSSs.

```
//JOBAUDITV job (account),programmer,REGION=1024k
//S1 EXEC PGM=SLUADMIN,PARM='execution parameters'
//STEPLIB      DD DSN=hlq.SEALINK,DISP=SHR
```

```
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
AUDIT VTSS(VTSS0,VTSS1)
```

12. Run scratch synchronization.

   Run `SLUCONDB` to synchronize HSC scratch status with the TMS(s). For more information, see *ELS Command, Control Statement, and Utility Reference*. If you are using LCM, see LCM User's Guide for the equivalent LCM utilities.

13. If you are using logging, audit affected MVCs using the recovery commands from the data set created in step 5.

```
//JOBRMVC job (account),programmer,REGION=1024k
//S1 EXEC PGM=SLUADMIN,PARM='execution parameters'
//STEPLIB     DD DSN=hlq.SEALINK,DISP=SHR
//SLSIN DD DSN=offload.dataset.genaudit,DISP=SHR
//SLSPRINT DD SYSOUT=*
```

   If you are not using logging, contact Oracle Software Support for assistance in determining which MVCs need to be audited.

14. Start HSC on all hosts.

15. Restart tape activity.

3

Managing a TapePlex

In general, managing a TapePlex is similar to managing VTCS, which is to (1) periodically and continuously, keep tabs on the health of your TapePlex(es) and (2) do as-needed chores as the need for them occurs. This chapter, however, intermixes periodic and as-needed tasks as it addresses the physical elements of a TapePlex and what you need to do to keep them operating normally. SMC/HSC/VTCS is server software, so it is designed to be self-correcting. Similarly, there is redundancy and resilience built into the library hardware itself. As long as you frequently check the status of the software and hardware and fix any problems, that takes care of the bulk of your TapePlex management tasks.

Managing Library Hardware

One of your best status tools is the innocuously named `DISPLAY EXCEPTNS` command, where "exceptions" are less-than optimal hardware conditions. The command output you want to see is a series of messages that say "No *name of hardware component* problems were detected". If you are experiencing system problems, this kind of output tells you that the hardware is fine, but there might be software problems.

If, on the other hand, you get output such as "CAP Not Operational", "Robot Hand needs Maintenance", and "LSM is Offline", you need more information to determine if the problem is with the software or hardware. Therefore, if you get any of these messages, look them up in *ELS Messages and Codes*, where you find detailed explanations and user responses. In some cases, the response is obvious and straightforward. If a robot hand needs maintenance, you put in a call to the CSE. Sometimes the response is not so clear, however. If an LSM is offline, the obvious move is to modify it back online, unless the LSM is down for maintenance, or you are changing drive unit addresses, in which case you wait until the work is done before bringing the LSM back online.

If you have used `DISPLAY EXCEPTNS` as your first diagnostic tool, and either found no problems or found problems and resolved them and you are using the Activities and ACS daily Reports to tune up the library performance, what is next in terms of managing the library hardware? The following sections get more detailed with status commands and responses to the conditions they show.

Managing ACSs

Two other commands you should know and use are `DISPLAY ACS` and `DISPLAY ALL`. `DISPLAY ACS` shows you the status of your ACSs, which includes hardware statuses for the ACSs, LMUs, and stations; and scratch volumes and free cells available. `DISPLAY ALL` tells you all that plus LSM and CDS status. You may see components that are not in an online or operational state: ACSs disconnected, stations offline or in standby

mode, LSMs offline, and so forth. You use the MODIFY command to change the state of LSMs and CAPs, and the VARY command to change ACS state. But as described in the above discussion, know why you are changing hardware states and how to do it before you enter the appropriate command. If you need to modify an LSM offline for maintenance, make sure you have quiesced tape activity to that LSM before you do so and, correspondingly, make sure the maintenance is complete before you vary the LSM online.

Managing SL8500 Redundant Electronics

A Redundant Electronics option is available to minimize control path downtime caused by a Library Controller (LC) failure by automating a Library Controller switch. In this configuration a second Library Controller is cabled to the local area network, connecting it to the LSM(s). HSC directs all work to one Library Controller, called the active LC, while the second LC, called the standby, remains powered on as a ready backup. Designation of active and standby LC is dynamic and their roles change according to environmental conditions.

For information about the commands and utilities discussed in this section, see *ELS Command, Control Statement, and Utility Reference*.

Dual Library Controller Configuration

For this implementation, a dual LC configuration is one pair of LCs installed and network connected to one library in a string. This is the only valid configuration. Figure 3–1 shows an example of this configuration.

Figure 3–1   Dual LC Configuration
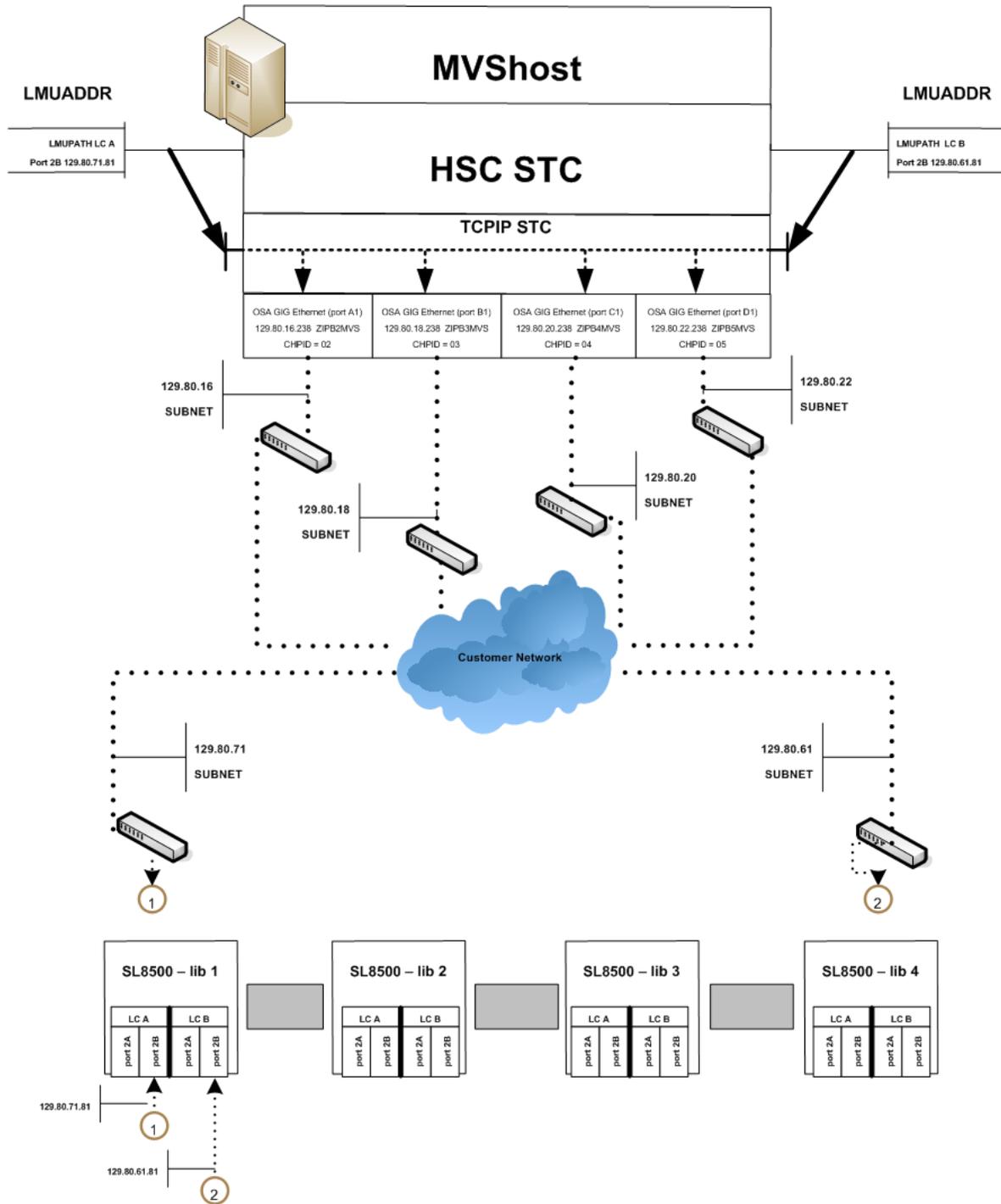
The following rules apply to a dual LC configuration. Both LCs

- Should be connected to all hosts
- Can be powered on at all times

■ Must be connected to both LAN 0 and LAN 1.

In the event of a failure from the active LC, an automated switch occurs and the standby assumes the active LC role. The HSC is informed and notifies the operator of the switch. Most outstanding requests are re-driven and all future library requests are sent to the newly activated LC. Once the failed LC is repaired and powered on, it assumes the standby role.

An operator SWitch command is provided for those occasions when it becomes desirable to dynamically activate a standby LC. The operator issues the Switch command, and the standby LC assumes the active LC role after instructing the former active LC to initiate a reboot. If the former active LC reboots successfully, it assumes the standby LC role.

> **Note:** Warnings and precautions apply to operating in a dual LC environment. See "Redundant LC Operational Overview" and become familiar with these requirements.

Network connections must be specified for both the active and standby LCs when the HSC is installed.

SL8500 Redundant Electronics Environment

Library requests cannot be automated if all network connections paths are offline to the LC, or if all network connection paths are inoperative. In a Redundant Electronics environment, if all network connection paths are inoperative to the active LC, use the HSC `SWitch` command to move the workload to the standby LC.

SL8500 LC Operation

The Library Controller (LC) responds to mount and dismount requests from the host through the HSC and passes each request to the appropriate LSM, to perform the physical action requested.

The following procedures are described in this section:

■ "Displaying ACS Status"

■ "Redundant LC Operational Overview."

> **Note:**
>
> ■ When you reboot an LC, error messages appearing before the message:
>
> ```
> ... ACS AA LSM LL ready; ...
> ```
>
> can be ignored. Do not ignore error messages that are displayed after the message.
>
> ■ Refer to the appropriate Oracle hardware operator's guide for a description of the library control panel and information on how to power on, power off, and boot a Library Controller in both stand-alone LC and redundant LC configurations.

Displaying ACS Status To determine the status of the LC, issue the following command:

```
Display Acs acsid
```

The command response lists the following:

- Current status of the ACS
- Current status of the Redundant LC (if configured) including:
  - HSC/LC compatibility levels
  - Active LCs
  - Standby LCs
  - Online, offline or standby status of each LC

Redundant LC Operational Overview Redundant Electronics configurations provide a backup LC which can take over if the active LC fails. (Refer to "Dual Library Controller Configuration" for a description of the dual LC option.) Both LCs are capable of being the active one but only one can be the active LC at any given moment. Each LC regularly checks the status of the other LC over the LAN. HSC directs all I/O to the active LC.

When an LC is powered on in a dual LC configuration, it places itself in a standby status and checks for an active LC. If an active LC exists, the second LC remains ready as the standby. If an active LC does not exist, the first LC to be fully initialized assumes the active role.

If both LCs power on successfully, the following HSC message is displayed:

```
... ACS AA: RE LIBID C1 is configured; Active is y, Standby is ready
```

where y is the configured LC identifier: "A" or "B."

LC Switch Overview The active LC functionality can be switched in the following ways:

- An automatic switch from a self-detected failure or power supply problem in the active LC.
- An operator initiated switch (using the SWitch command) after all online stations become incapable of communication.
- An operator initiated switch (using the SWitch command) while online stations remain capable of communication with the active LC.
- When the reboot switch on the active LC is pressed.
- When the active LC is powered off.

Consequences of an LC Switch When an LC switch occurs, a series of console messages track the changing status of the active and standby LCs. Critical status changes appear as highlighted messages which remain on the screen until the condition is corrected or the message is superseded.

The following message is displayed when status changes and the standby LC is ready:

```
... ACS AA: RE LIBID C1 is configured; Active is y, Standby is ready
```

The following message is displayed if there is a status change and an LC error occurs:

```
... ACS AA: : RE LIBID C1 Is Not configured; Active Is y, Standby is not ready
```

The following message is displayed if there is a status change and the standby LC is not ready:

```
... ACS AA: : RE LIBID C1 Is configured; Active Is y, Standby is not ready
```

In the messages above, y is the configured LC identifier, "A" or "B".

When an LC switch occurs, the HSC interrogates each LC to determine the current status. The status of each station is maintained as online, offline, standby, pending online, pending offline, or pending force offline. Use the `Display Acs` *acsid* command to determine the status of each station and LC.

> **Note:** Auto-mode CAPs are locked during an LC switch until the LSM completes initialization.

Automatic LC Switch LC microcode detects, reports, and reacts to LC failures as necessary to keep the ACS operational. *Note that* this happens automatically so some work may be in process within the LC. When the standby LC assumes the active role, it notifies HSC of its changed status and sends the LC Ready signal. HSC reads the recovery information provided by the new active LC concerning in transit cartridges within the various LSMs in the ACS. HSC then:

- Modifies, or marks "completed," the queued requests, reflecting the current locations of in-transit cartridges
- Sends all incomplete requests to the active LC
- Marks cartridges as errant when requests cannot be completed successfully.

> **Note:** Enter and eject operations may have to be restarted after an LC switch.

Operator Initiated LC Switch Before initiating an LC switch if some network connections are communicating, determine if the need to switch LCs is immediate, or if processing can be completed before initiating the switch. If the need to switch LCs is not critical, wait until the LSMs are relatively idle.

Initiate an LC switch for a Library ID by issuing the following command:

```
SWitch Acs acsid LIB libid
```

The command is routed through the standby LC to the disconnected LC, causing it to reboot. The following sequence of events occurs:

1. HSC issues the message

   ```
   ... ACS AA: Switch initiated, expect notification in 20 seconds
   ```

2. The standby LC assumes the active LC role ("ACTIVE" indicator illuminates).

3. Online LSMs perform quick-init.

4. As each LSM becomes ready, the HSC issues the message:

   ```
   ... ACS AA LSM LL Ready; Temp Outage Queue Will Be ReDriven
   ```

5. If the former active LC completes reboot successfully, it assumes the standby LC role.

If the SWitch command is issued while processing is active, cartridges may be in-transit in various LSMs. Recovery procedures occur automatically as described in "Automatic LC Switch." Upon receiving the LC Ready signal, HSC waits for the LSMs

to complete a quick-init and then re-drives the requests through processing the Temp Outage Queue.

CAP Manual Recovery When the LC Switch occurs and certain CAP functions are processing requests (enter/eject), manual intervention for recovery is required after the switch has completed. Below are various CAP scenarios with an appropriate procedure to resolve for each.

A switch occurs when an Enter function is active When a switch occurs and an enter function is active, after the switch completes, two possible events can occur simultaneously or individually:

- A cartridge is selected by HSC/ELS, but not moved to an LSM home cell when the switch occurred. After the switch completes, the cartridge is selected and cannot be moved to an LSM cell.

- The cartridges in the CAP to be moved to the LSM receives message SLS0699I - 03/01 error from the library during a move. The library reserves the CAP because there are carts in the CAP after the switch completes. The library retains ownership of the CAP until the cartridges are removed. This is normal behavior of the SL8500 library.

Procedure to resolve:

- Use the procedure in the SL8500 Operator's Guide to manually remove the cartridges from the CAP with the SLC.

- Issue a `Display Cap` *aa:ll:cc* command for the specific cap affected.

- If the CAP is offline, issue the `Modify CAP` *aa:ll:cc*`,online` command.

- Issue the `ENter` *aa:ll:cc* command to restart the enter function for the CAP.

- If the CAP is automatic, issue:

  ```
  CAPP x aa:ll:cc MANual
  CAPP x aa:ll:cc AUTO
  ```
  to reset it to auto mode.

- Insert the balance of the cartridges removed from the previous enter function into the CAP.

- If an SLS0251E message that states that volser is a duplicate, run the Unselect utility to unselect the volser. Afterwards, the volser may be entered into the LSM.

  When entering the volser, a message SLS0694D is displayed to ask you what you want to do with the duplicate, you should respond logically delete.

A switch occurs when an Eject function is active When a switch occurs and an eject function is active, after the switch completes, the cartridges in the LSM to be moved to the CAP receive an SLS0699I message - 03/01 error from the library for the move operation. The library reserves the CAP because there are carts in the CAP after the switch completes. The library retains ownership of the CAP until the cartridges are removed. This is normal behavior of the SL8500 library.

Procedure to resolve:

- Use the procedure in the SL8500 Operator's Guide to manually remove the cartridges from the CAP with the SLC.

- Issue a `Display Cap` *aa:ll:cc* command for the specific cap affected.

- If the CAP is offline, issue the `Modify CAP`,`online` command.

- Re-issue the EJECT *vol-list* or *vol-range aa:ll:cc* command or restart the eject utility to eject the balance of the cartridges.

- If the CAP was automatic before the eject function, issue the CAPP *x aa:ll:cc* manual and then re-issue is as CAPP *x aa:ll:cc* auto to reset it to automode.

A switch occurs when CAPs are set to automatic mode When a switch occurs and CAPs are set to automatic mode, when the switch completes, message SLS0699I may be presented with a return/reason code of 07/05 error from the library during a reserve of the CAP. This is because the CAPs have not initialized completely after the switch has completed. This is normal behavior of the SL8500 library.

Procedure to resolve:

- Wait until the message SLS0668I for the last LSM in the library switched is issued.

- Issue a CAPPref prefvalue *aa:ll:cc* MANual for each automatic CAP.

- Issue a CAPPref prefvalue *aa:ll:cc* AUTO for each CAP to become automatic.

Managing CAPs

DISPLAY CAP shows you CAP settings and status; MODIFY CAP changes the CAP state. DISPLAY CAP, in addition to basic status, provides some other valuable information that helps you decide whether you want to change CAP state, and if so, when:

- CAPid

- CAP size: PCAP (priority CAP); 21-cell or 14-cell 9740 (standard CAP); 40-cell (enhanced CAP); 20-cell (9360 CAP); 30-cell (9360 optional CAP); 36-cell Bulk CAP (SL8500); or 39-cell (SL8500 CAP)

- Host ID of the host owning the CAP

- Priority: CAP preference value

- CAP mode: cleaning, draining, ejecting, entering, or idle

- CAP status: active, automatic mode, manual mode, offline, online, and/or recovery needed

The CAPPREF command lets you set the CAP mode (manual or automatic) and preference for selection (0-9, where 9 is the highest preference). CAPs in manual mode require an ENTER command to unlock them; CAPs in automatic mode do not. In general, you specify CAP modes and preferences in a CAPPREF command at configuration in the HSC PARMLIB Member and run with these settings. If you do need to change a CAP's setting, you can do it dynamically by entering a CAPPREF command. For example, If your systems is experiencing high levels of enters and ejects, you may want to bump up the CAP preferences for enhanced CAPs. The caveat on CAP preferencing is that Priority CAPs (PCAPs) always have a priority of 0 (never selected unless specifically requested) so that they remain generally available.

Defining Manual RTDs

In a DR situation, you may have no automated tape resources available. To support nonlibrary VSM configurations, you must define manual RTDs in a nonexistent ACS.

To define manual RTDs:

1. Generate LIBGEN macros to define a nonexistent ACS.

   The SLIDRIVS macro defines the device address for the manual RTDs.

> **Note:** The LSM in the nonexistent ACS can be defined as an SL8500 or SL3000 to take advantage of the `MODIFY CONFIG` command.

2. Run the SLICREAT utility to format a new CDS.

3. Run SET SLIDRIVS utility to set the MODEL for the manual RTDs, for example:

   SET SLIDRIVS(400,401,402,403) MODEL(T9840C)

   > **Note:** Manual RTDs on the same panel must be the same drive type.

4. Run the VTCS CONFIG utility to define the VSM system.

5. Use the VTCS IMPORT utility to import MVCs into the new CDS to use with the manual RTDs.

   These MVCs are treated as nonlibrary volumes.

Protecting Tape Data

This section describes the ELS facilities that you can use to prevent data over-writes on both real and virtual tape volumes.

Protecting Native Tape Data

Oracle/StorageTek T9840, T9940B and T10000 tape cartridges can be ordered with VOLSAFE secure media technology which provides a non-rewritable protection solution for data residing on tape. VOLSAFE cartridges are also known as WORM (Write Once Read Many) cartridges.

Once data is written to VOLSAFE secured media, it can be appended to but never modified or deleted. These cartridges have physical, visual and electronic features to differentiate them from non-VOLSAFE cartridges.

Protecting VTV Data

VSM systems now offer VOLSAFE protection for VTVs. VTVs are assigned a protection attribute when scratch mounted and retain this attribute as long as the VTV is non-scratch. RACF `ALTER` authority on the `TAPEVOL` class is required to scratch a VOLSAFE protected VTV.

VTCS enables this protection through the `MGMTclas WRITE` parameter, which specifies the VTV VOLSAFE policy as follows:

MANY
the default, specifies no VOLSAFE write protection.

ONCE
provides read-only protection, VTVs cannot be modified once non-scratch.

Protecting MVC Data

Although HSC does not allow MVCs to be marked as scratch, consider additional methods of protecting MVC data such as:

- Establishing RACF authority to ensure only the HSC task can write to the MVC volumes.

- Setting MVCs as read-only using the MVCMAINT utility.

- Using VOLSAFE protected media as MVCs. Like native tape, once data is written to a VOLSAFE secured MVC, it can be appended to but never modified or deleted.

  - Specifying the POOLPARM MVC NORECLAM parameter for any MVC pools with VOLSAFE volumes to prevent space reclamation for the pool. This guarantees a writable EOT position.

  - All VTV recall processing is unaffected for VOLSAFE MVCs.

Managing Library Volumes

Managing library volumes has a wide range of tasks, ranging from using a Volume Report, where you start, through managing cleaning cartridges.

Using a Volume Report

The VOLRPT utility provides volume information that can help you manage the physical volumes in your library. The VOLRPT utility function uses the Unified User Interface, and therefore provides output formats of either text, XML or Comma Separated Values (CSV). See *ELS Command, Control Statement, and Utility Reference* and *ELS Programming Reference* for more information about how to request the XML and CSV formats and explanations of the outputs, as well as for command syntax details.

You may want to run the VOLRPT on a periodic basis to display and resolve volume exception conditions or just display volume status, such as described in the following sections:

- "Errant and Selected Volumes"

- "Unmatched VOLPARM and External Label Status"

- "Volumes not included in VOLPARM"

- "Managing Scratch Subpools"

- "Managing Cleaning Cartridges."

There are many other uses for VOLRPT. See *ELS Command, Control Statement, and Utility Reference* for other ways to select volumes and sort the VOLRPT output.

Errant and Selected Volumes

Sample VOLRPT control card:

```
VOLRPT SORT(LOC) INCLUDE(SEL,ERR)
```

This request produces a report of all errant and selected cartridges in location sequence.

The "selected" status is normally temporary and represents a cartridge being scratched or moved from one location to another. For each "selected" volume on the report, you should issue a DISPLAY VOLUME console command to determine if the volume has become unselected. If the volume remains selected, it may represent an abnormal termination of a host while the volume was selected. If you are certain that the volume is not currently in use, you can run the UNSELECT utility to reset the selected status.

An errant volume is one that is not in its last known location. Attempting to eject or mount an errant volume causes HSC to look for the volume in its home cell, its last

known source location and its last known destination location. If the volume is not in any of these locations, you may need to run an audit of the ACS to locate the volume.

Unmatched VOLPARM and External Label Status

Sample VOLRPT control card:

```
VOLRPT INCLUDE(NONMEDEQ)
```

This request produces a report of all volumes for which the media label on the cartridge does not match the VOLPARM media type.

The character preceding the Media indicates the type of mismatch detected. The character '*' means that the label did not match the VOLPARM media code. The character '-' means that there was no VOLPARM matching the volume. Note that these volumes also have a scratch subpool of '*NON-VOLPARM*'. The '@' character means that the volume had no physical media label.

Sample output of this command is shown in Figure 3–2.

Figure 3–2   Volume Report (Unmatched VOLPARM and External Label)

```
SLUADMIN (7.1.0)                      STORAGETEK ENTERPRISE LIBRARY SOFTWARE UTILITY          PAGE 0003
TIME 11:41:12                         VOLUME REPORT UTILTIY                                   DATE 2010-06-29

Volume              Cell Loc        Err    Ext Cln              <----Inserted----> <---Last Used---> Times MWL&
Serial  Media  Rectech AA:LL:Pa:Ro:Co Scr Sel Lbl Use Subpool ID  Date      Time     Date      Time     Selected
VOL111  *STK1R STK1R  03:00:00:31:00 Y        R        GENERAL     20050607  16:21:59 20100321  16:35:21 139 50
VOL222  *STK1R STK1R  03:00:00:20:00 Y        R        GENERAL     20100113  10:56:15 20100321  16:35:21 54 49
VOL333  -STK1R STK1R  03:00:02:06:01 Y        R        *NON-VOLPARM* 20050607 16:22:04 20100321  16:35:21 138 15
VOL444  -STK1R STK1R  03:00:03:05:02 Y        R        *NON-VOLPARM* 20050607 16:22:14 20100321  16:35:21 138 28
VOL555  @STK1R STK1R  03:00:00:04:02 Y                 *NON-VOLPARM* 20050607 16:21:45 20100321  16:35:21 138 47

0SLS0155I Condition code for utility function is 0                                                    ELS_015
```

Volumes not included in VOLPARM

If your subpool totals only show volumes in the *NON-VOLPARM* pool, you want to get a detailed list of these volumes. There is no special control card to select these volumes, but the INCLUDE(NONMEDEQ) can be used. You also see any volumes where the VOLPARM media does not match the media on the volume label.

Managing Scratch Subpools

You can obtain a report showing subpool totals by subpool, library location, and media type without listing individual volumes by running the command:

```
VOLRPT SUMM(SUBPOOL) NOVOL
```

Sample output of this command is shown in Figure 3–3.

Figure 3–3   Volume Report (Scratch Subpools)

```
SLUADMIN (7.1.0)              StorageTek Enterprise Library Software Utility       PAGE 0001
TIME 15:47:29                 Subpool Totals, all Ranges                          DATE 2010-09-29

Subpool ID    Label Type    Range Limits        DRTEST?
GENERAL       SL            N/A - N/A           NO

ACS           LSM           Non SCRATCH         SCRATCH     Media       Rectech
00            00:00         0                   0
              00:01         0                   0
                            _____           _____
              ACS Total     0                   0


01            01:00         98                  1           STK1R       STK1R
              01:01         0                   0
                            _____           _____
              ACS Total     98                  1


              Library Total 98                  1


SLUADMIN (7.1.0)              StorageTek Enterprise Library Software Utility       PAGE 0002
TIME 15:47:29                 Subpool Totals, all Ranges                          DATE 2010-09-29

Subpool ID    Label Type    Range Limits        DRTEST?
*NON-VOLPARM* SL            N/A - N/A           NO

ACS           LSM           Non SCRATCH         SCRATCH     Media       Rectech
00            00:00         21                  1           STK1R       STK1R
              00:01         5                   23          STK2P       STK2P
                            _____           _____
              ACS Total     26                  24


01            01:00         10                  3           STK1R       STK1R
              01:01         17                  8           STK2P       STK2P
                            _____           _____
              ACS Total     27                  11


              Library Total 53                  35                                ELS_012
```

Using a VOLRPT to Track Cleaning Cartridge Usage

You can produce a report containing only cleaning cartridges, sorted by use count, using the following:

```
VOLRPT VOL(CLN%%%) SORT(USE) DESCEND
```

Sample output is shown in .

Figure 3–4   Volume Report (Cleaning Cartridges)

```
SLUADMIN (7.1.0)                       STORAGETEK ENTERPRISE LIBRARY SOFTWARE UTILITY      PAGE 0002
TIME 16:23:09                          VOLUME REPORT UTILTIY                               DATE 2010-09-29

Volume                  Cell Loc       Err     Ext Cln Subpool <----Inserted----> <---Last Used---> Times MWL&
Serial Media   Rectech  AA:LL:Pa:Ro:Co Scr Sel Lbl Use ID      Date       Time     Date       Time     Selected

CLN007 STANDARD LONGITUD 00:00:03:07:04         R               20100418 16:29:11 20100425 20:11:19  26 50
CLN081 STANDARD LONGITUD 00:01:03:13:01         R               20100418 18:22:40 20100825 12:13:17  17 47
CLN082 STANDARD LONGITUD 01:00:03:03:01         R               20100419 14:31:01 20100603 11:24:27  5 75
CLN041 STK1U    STK1RABC 00:01:01:37:01         R               20100418 16:41:48 20100427 16:56:48  3 86
CLN003 STK2W    STK2P    00:00:00:00:15         R               20100418 16:29:11 20100501 14:19:20  2 83
CLN040 STK2W    STK2P    00:01:03:03:00         R               20100418 18:31:18 20100911 00:53:51  2 63
CLN028 STK2W    STK2P    00:00:01:09:01         R               20100418 16:41:51 20100717 15:52:46  2 15
                                                                                                    ELS_016
```

The clean use count is actually the field labeled "Times Selected" at the far right. The field labeled "Cln Use" in the middle of the report indicates Y when a cleaner is overuse.

Optimizing Mount Performance

"Optimal mount performance" is measured by how quickly you can get real tape volumes mounted and dismounted. Here, your best tool is the Activities Report, which pulls data from SMF records and provides information that helps you balance mount/dismount activity across ACSs and LSMs. This report shows Total Mount and Dismount activity broken out as follows:

- Scratch mount/total mounts and average time per mount

- Nonscratch mounts/total mounts, and average time per mount

- Scratch mounts in a different LSM/total mounts, average time per mount, and average number of pass-thrus

- Nonscratch mounts in a different LSM/total mounts, average time per mount, and average number of pass-thrus.

If the Activities Report uncovers performance problems due to scratch mounts, you can use the HSC Scratch Redistribution utility or LCM to redistribute scratch volumes across the LSMs within a particular ACS. If you are running low on scratch volumes, you can run scratch synchronization to scratch volumes with non-current data or enter more scratch volumes into the ACS. If you are seeing excessive passthru activity or long mount times for nonscratch volumes, you can use LCM to redistribute nonscratch volumes.

Using Volume Access Security

The HSC Volume Access User Exit (`SLSUX14`) performs security checking for volumes. When `SLSUX14` is enabled, security checking for volumes is performed for mount, eject, and scratch requests. The Volume Access User Exit is **not** called for LibraryStation `set_scratch` requests.

See *ELS Programming Reference*, Volume Access User Exit (`SLSUX14`) for programming details.

`SLSUX14` can allow the request to complete or be rejected. Coding in `SLSUX14` can issue RACROUTE macros or use other methods to determine whether the request is to proceed or be rejected. Another option is for `SLSUX14` to indicate for HSC to perform the security checks using RACROUTE and RACF definitions to determine the success/rejection of the request. One `SLSUX14` return code specifies that the TAPEVOL class is used for the volume access checks. Another `SLSUX14` return code specifies that the user exit has placed in the parameter list the RACF class to use for the volume access checks. HSC then uses that RACF class for the volume security checks.

RACF Attributes

The user issuing the following volume requests needs different RACF authority for the request to succeed when HSC performs the `RACROUTE` macro.

- MOUNT – RACF UPDATE authority allows the mount to proceed.

- MOUNT – RACF READ authority allows the mount to proceed. The volume is mounted write-protected.

- EJECT – for eject requests the user requires RACF UPDATE authority.

- SCRATCH – for scratch requests the user requires RACF CONTROL authority.

The following shows an example of commands to set RACF authorities.

```
RDEFINE TAPEVOL 111111 UACC(NONE)
PERMIT 111111 CLASS(TAPEVOL) ID(USER1) ACCESS(READ)
```

```
PERMIT 111111 CLASS(TAPEVOL) ID(USER2) ACCESS(UPDATE)
PERMIT 111111 CLASS(TAPEVOL) ID(USER3) ACCESS(CONTROL)
```

As shown in this example, USER1 can mount volume 111111 for read access (write-protected). USER2 can mount volume 111111 for write access. For all other users any mount requests for 111111 is rejected. USER3 can perform scratch requests.

Write Once/Read Many (WORM) Support

To support a WORM scenario, `SLSUX14` can be used to enforce a "never scratch" policy as shown in the updated command set in the following example.

```
RDEFINE TAPEVOL 111111 UACC(NONE)
PERMIT 111111 CLASS(TAPEVOL) ID(USER1) ACCESS(READ)
PERMIT 111111 CLASS(TAPEVOL) ID(USER2) ACCESS(UPDATE)
```

In this example, USER3 no longer has CONTROL access to volume 111111. Therefore, SCRATCH requests from all users, except the user that defines the RACF configuration and gets CONTROL authority, are rejected for volume 111111.

> **Note:** UX14 is not called for a LibraryStation `set_scratch` request. For LibraryStation set_scratch requests, Volume Access Security is controlled by the LibraryStation `VOLAUTH` and `VOLACC` parameters of the `LSINIT` command.

Managing Scratch Volumes

Managing scratch volumes basically means having enough scratch volumes, but not too many, and having them in the right place (preferably in an LSM with sufficient drives types compatible with the volume media). Table 3–1 describes the scratch volume management tasks and associated tools.

Table 3–1   Scratch Volume Management Tasks and Tools

Task	Tool	Notes
Maintain Sufficient Quantities of Scratch Cartridges	■ `Warn` command - establishes scratch warning threshold values. ■ Display THReshld command - displays scratch count and scratch threshold information by subpool name, ACS ID, LSM ID, media, and recording technique.	In general: Use the `Warn` command to set scratch warning thresholds and the Display THReshld command to monitor scratch volume levels versus the threshold. If you are running low on scratch volumes, run scratch synchronization, redistribute scratch volumes, and then add more scratch volumes, in that order. For more information, see "Adding Scratch Volumes."
Synchronizing CDS Scratch Status with TMS Status	■ HSC SLUCONDB scratch conversion utility (with SMC Tape Management Extract Utility (SMCUDBX) for client/server configurations. *OR* ■ LCM OPTIONS SYNC parameter process.	As described above, scratch synchronization can free scratch volumes. Typically, you do scratch synchronization after your daily TMS run.
Redistributing Scratch Volumes	Use the HSC Scratch Redistribution utility or LCM to redistribute scratch volumes across the LSMs within a particular ACS.	As described in "Optimizing Mount Performance," poor scratch distribution can inhibit optimum library performance. Scratch redistribution balances the number of scratch volumes between specified LSMs in an ACS, and can be done by subpool.
Scratching Specific Volumes	■ The HSC SCRAtch command scratches one or more volumes. ■ The HSC UNSCratch command unscratches one or more volumes.	Yes, if you have the command authorization, you can actually scratch and unscratch specific volumes. In general, however, *do not* explicitly scratch specific volumes unless you really know what you are doing. Let the TMS manage volume scratch status and do scratch synchronization instead.
Displaying Scratch Subpool Information	Display SCRatch	This only displays scratch subpool information for the host the command is executed on; to display all scratch subpool information use the VOLRPT utility.

Adding Scratch Volumes

Adding scratch volumes to the library is a general task that breaks down into two separate cases:

■ "Populating a New LSM"

■ "Adding Volumes to An Existing LSM"

In both cases, you first need to create `POOLPARM` and `VOLPARM` definitions for the new volumes as described in "Defining Nearline Volumes."

Defining Nearline Volumes To define Nearline volumes:

1. Create a POOLPARM statement to define the volume scratch pool.

   For example, to define a scratch pool for use by hosts `MVS1` and `MVS2`:

   ```
   POOLPARM NAME(SYS19840P22)TYPE(SCRATCH)HOSTID(MVS1,MVS2)LABEL(SL
   ```

2. Create a VOLPARM statement to define the volumes.

For example, to define a range of 9840C volumes:

```
VOLPARM VOLSER(C2000-C2999)MEDIA(STK1R)RECTECH(STK1RC)
```

3. Enter a SET VOLPARM command to apply the volume definitions.

```
SET VOLPARM APPLY(YES)
```

Populating a New LSM

After you have done your POOLPARM and VOLPARM definitions, to populate a new LSM, use one of the following methods:

■ Set one or more CAPs to automatic mode and repeatedly open and fill the CAP with new cartridges.

■ If the LSM is a non-Streamline library and is large enough for you to enter:

1. Open the LSM access door.

2. Manually place cartridges into the storage cells in the LSM.

3. Close and lock the access door.

4. Run an HSC Audit on the LSM.

A full LSM audit can take considerable time, but so can entering cartridges through the CAP, so you have to judge which method is faster.

Adding Volumes to An Existing LSM After you have done your POOLPARM and VOLPARM definitions, enter the cartridges as described in "Entering Cartridges."

Entering Cartridges Cartridges are just cartridges until they get entered into the ACS; then they become volumes in the CDS. You can enter scratch cartridges, non-scratch cartridges, cartridges to be used as MVCs, or cleaning cartridges.

To enter cartridges:

1. Verify that the LSM where you want to enter the cartridges is online.

For example, to see the status of LSM 00:00:

```
DISPLAY LSM 00:00
```

An SLS0054I message "LSM 00:00 now online" indicates that the verification was successful. Otherwise, you have to figure out what the problem is and, at a minimum, enter a MODIFY LSM *aa:cc* ONLINE command to get it back online.

2. Check the CAP mode:

For example, to see the status of the PCAP on LSM 00:00:

```
DISPLAY CAP 00:00:02
```

If the CAP mode is Automatic, go to step 4. Otherwise, continue with step 3.

3. If the CAP mode is Manual, you have two choices, depending on your site's practices:

■ Change the CAP mode to Automatic.

For example, for the PCAP in step 2, to set the CAP preference to zero (always required for PCAPs) and mode to Automatic:

```
CAPPREF 0 00:00:02 AUTO
```

■ Use ENTER to make the CAP ready for cartridge entry (mode stays Manual).

For example, for the PCAP in step 2, to make the CAP ready for entry and set scratch status on for the entered cartridges:

```
ENTER 00:00:02 SCRATCH
```

> **Note:** If all CAPs are currently in use for ejects, you can use the SENter command to schedule a CAP for cartridge entry.

4. Enter the cartridges as described in the *Operator's Guide* for your LSM type.

Ejecting Volumes

As described in the following sections, there are basically three reasons why you want to eject volumes:

■ You defined too many scratch volumes, and temporarily need some room for specific volumes; see "Ejecting Scratch Volumes."

■ You have specific volumes that are no longer in use, and you want to vault them. For more information, see *ELS Disaster Recovery and Offsite Data Management Guide*.

■ You have specific 9x40 volumes that are nearing end of Media Warranty Life. Figure 3–2 shows a volume report with the %MWL column on the extreme right. This column shows the percentage of Media Warranty Life used for the volume, where %100 means the volume has exceeded its Media Warranty Life.

> **Note:** For 9x40 volumes, the Media Information Record (MIR) stores the number of times the cartridge has been mounted during its lifetime. During dismount processing, HSC uses the mount count value read from the MIR to compute the %MWL. For HSC to report MWL% for a library-resident volume, the volume must have been mounted and dismounted at least once after it was entered into the library. When a volume is ejected, the associated CDS VAR record is deleted, and HSC can no longer report MWL% for that volume while it remains outside the library. However, the mount count is maintained in the MIR, and the next time the volume is entered into the library and mounted and dismounted at least once, HSC can again report MWL% for that volume while it resides in the library.

If a transport cannot read or write the MIR, the drive can reinitialize a new MIR elsewhere on the volume. Errors processing a MIR are rare, and although the drive reconstructs a new MIR if it must, it is possible the drive might not be able to transfer the mount count from the failed MIR to the replacement MIR, in which case the drive initializes the mount count to zero in the replacement MIR. If you observe a non-zero mount count go to zero for a volume (for example, by monitoring the MWL% on a volume report), you should examine the SYSLOG for data checks on this volume and have the tape drive examined, and you should also consider retiring this volume.

The Media Warranty Life feature requires the following:

■ Tape library LMU compatibility level 21

- T9x40 transports with 1.42 drive firmware

> **Note:** T9840B is not supported because of memory constraints.

- T10000A and T10000B with 1.38 drive firmware
- PTF SES700 L1H154V

■ You have cleaning cartridges that have exceeded their useful life and must be replaced. See "Manually Ejecting Used Cleaning Cartridges."

Ejecting Scratch Volumes

To eject scratch volumes:

1. Verify that the LSM where you want to eject the volumes is online.

   For example, to see the status of LSM 00:00:

   ```
   DISPLAY LSM 00:00
   ```

   An SLS0054I message "LSM 00:00 now online" indicates that the verification was successful. Otherwise, you have to figure out what the problem is and, at a minimum, enter a `MODIFY LSM` *aa:cc* `ONLINE` command to get it back online.

2. Use `EJECT` to start the eject operation.

   For example, to eject 100 scratch volumes with media STK1R and rectech STK1RC from pool `SYS19840P1`:

   ```
   EJECT SCRTCH SUBPOOL(SYS19840P1)VOLCNT(100) MEDIA(STK1R) RECTECH(STK1RC)
   ```

   `WAITCAP(YES)` is the default.

3. When you get operator messages to empty the CAP, remove the volumes from the CAP.

Managing Cleaning Cartridges

Hopefully, you defined sufficient cleaning cartridges when you configured ELS. If not, *Configuring HSC and VTCS* tells you how. After you define the cleaning cartridges, enter them as described in "Entering Cartridges." There are some additional things you can do with `MNTD` (either through the PARMLIB member or command) to manage cleaning cartridges.

> **Note:** To manage cleaning cartridges, HSC requires them to have external labels with prefix CLN followed by 3 numeric digits. Some LTO cartridges, however, have external labels that are *not* in this format. In this case, contact your Oracle representative for information on obtaining external labels for LTO cleaning cartridges.

Cleaning Cartridge Selection

To automatically select a cleaning cartridge when a drive needs cleaning, set the "auto clean" option on:

```
MNTD AUTOCLN(ON)
```

The drive cleaning process selects a cleaning cartridge in the LSM that is closest to the drive that needs cleaning. After the clean operation is completed, the "float" option determines whether the cartridge is returned to its home cell (float off) or assigned a new home cell in the current LSM (float on). Setting float on reduces pass-thru operations and thus enhances performance:

```
MNTD FLOAT(ON)
```

Ejecting Cleaning Cartridges

The POOLPARM MAXCLEAN parameter specifies the maximum number of cleaning operations for a given set of cleaning cartridges. For ELS to automatically eject cleaning cartridges that have exceeded their MAXCLEAN value, enter:

```
MNTD EJCTAUTO(ON)
```
If you are running ExLM, you may prefer letting ExLM manage ejecting cleaning cartridges on a scheduled basis. In this case, set "auto eject" off. This is very useful for a "lights-out" datacenter.

Without ExLM, auto-ejection is the recommended technique, otherwise you have to manually schedule cleaning with the CLEAN command and manually eject the cartridge when it is used up (spent) as described in "Manually Ejecting Used Cleaning Cartridges." Auto-ejection requires an available CAP. If a CAP is not available, a manual eject is required.

Newer drives identify spent cleaning cartridges and notifies ELS regardless of the MAXCLEAN usage count. If "auto eject" is on, the cartridge is ejected when the library detects the "spent" condition. Otherwise, ExLM can be used to schedule the ejection of "spent" cleaning cartridges.

After you eject a cleaning cartridge, you usually want to enter a new cartridge to replace the used one. If you want to enter a cleaning cartridge whose volser is not yet defined to ELS, see *Configuring HSC and VTCS* for information on defining cleaning cartridges.

Manually Ejecting Used Cleaning Cartridges

To replace used cleaning cartridges:

1. Verify that the LSM where you want to eject the cartridges is online.

    For example, to see the status of LSM 00:00:

    ```
    DISPLAY LSM 00:00
    ```

    An SLS0054I message "LSM 00:00 now online" indicates that the verification was successful. Otherwise, you have to figure out what the problem is and, at a minimum, enter a MODIFY LSM *aa:cc* ONLINE command to get it back online.

2. Use EJECT to start the eject operation.

    For example, to eject a range of 9840 cleaning cartridges:

    ```
    EJECT (CLN300-CLN310) SEQ(YES)
    ```

    WAITCAP(YES) is the default, and the volumes are placed in sequence in the CAP.

3. When you get operator messages to empty the CAP, remove the cleaning cartridges from the CAP.

4. Check the CAP mode of the CAP where you want to enter the replacement cleaning cartridges:

For example, to see the status of the PCAP on LSM 00:00:

```
DISPLAY CAP 00:00:02
```

5. If the CAP mode is Manual, you have two choices, depending on your site's practices:

   ■ Change the CAP mode to Automatic.

   For example, for the PCAP in step 4, to set the CAP preference to zero (always required for PCAPs) and mode to Automatic:

   ```
   CAPPREF 0 00:00:02 AUTO
   ```

   ■ Use ENTER to make the CAP ready for cartridge entry (mode stays Manual).

   For example, for the PCAP in step 4, to make the CAP ready for entry and set scratch status on for the entered cartridges:

   ```
   ENTER 00:00:02 SCRATCH
   ```

Solving Problems

The following sections provide solutions to potential problems.

Recovering Errant Cartridges

A cartridge becomes errant when HSC is uncertain of the location of the volume in the LSM. The following conditions *may* cause a cartridge to become errant.

■ An operator modifies an LSM offline using the FORCE option.

■ The volume is not located in the cell location specified by the CDS.

■ An unrecoverable error occurs in the LMU.

■ An unrecoverable error occurs in the LSM.

Errant volume recovery is performed automatically:

■ Any time HSC attempts to use a cartridge that is marked as errant.

■ When an ACS is connected.

■ When an LSM is modified online.

■ When cross host recovery is performed.

HSC issues the following message:

```
... Attempting to locate errant volume VVVVVV
```

If the cartridge is located by checking its original source location or its destination location, HSC removes it from errant status.

If the cartridge cannot be located, the operator is prompted by the following message:

```
... Volume VVVVVV not found; Eject or Ignore (E|I)?
```

Respond with one of the following:

■ Reply "E" *if you are certain that the cartridge is not in the library*. The cartridge is deleted (logically ejected) from the CDS.

- Reply "I" if you do not know the location of the cartridge, to leave the cartridge in errant status. An audit trail is maintained for the cartridge, which can be helpful if it becomes necessary to manually locate the cartridge.

  If the cartridge is located at a later time, HSC recognizes it as errant and removes it from errant status. The cartridge is either returned to its home cell location, or marked selected for a request.

Entering Duplicate And Unreadable VOLSERS

The following discusses VOLSER-related issues.

Entering Duplicate Volumes

When you try to enter a volume that is already recorded in the CDS, HSC attempts to locate the original cartridge by checking its home cell location or destination location. If the volume is found, the duplicate cartridge is rejected by HSC.

If the volume is not found, the HSC response is determined by the `OPTion ENTdup` command setting.

- If `OPTion ENTdup` is set to `Auto`, HSC automatically deletes the original volume from the CDS, and enters the new volume.

- If `OPTion ENTdup` is set to `Manual`, HSC requests operator assistance by issuing the following message:

```
... ENTER of volume VVVVV encountered missing duplicate volser;
(Logically) Delete, or Ignore (D,I)?
```
Reply "D" to delete the original volume from the CDS and enter the new volume.

> **Note:**   Refer to OPTION Command and Control Statement in *ELS Command, Control Statement, and Utility Reference* for details on command syntax and parameters. You can enter duplicate volumes into a Streamline library as long as the library is partitioned and the duplicate volumes are entered into different partitions.

Entering Cartridges With Missing or Unreadable External Labels

Tape cartridges with unreadable labels cannot be entered into a library through a CAP. If you attempt to enter a cartridge without a label, or with a label the robot cannot read, the following HSC message is issued:

```
... UnNamed Cartridge in CAP CELL AA:LL:RR:CC;
reply "V,volser" or "EJECT"
```
Do one of the following:

- Reply "V,*volser*" to enter the cartridge by assigning it a VOLSER. The specified VOLSER is permanently assigned to the cartridge in the CDS.

> **Note:**   If any unlabeled cartridges or cartridges with unreadable labels are manually entered into a library, they are ejected during an audit. Oracle recommends that users provide an external media label for all cartridges.

- Reply "EJECT" and remove the cartridge from the CAP when the CAP is unlocked.

Clearing Recovery Status On A CAP

When a CAP status is RECOVERY, use the following procedure to force CAP recovery.

> **Note:**   If this procedure does not clear recovery status, try "Freeing An Allocated CAP."

To clear recovery status on a CAP:

1. Issue the following command:

    ENTER *cap-id*

    where *cap-id* is the CAP you want to recover.

    The following occurs:

    a. The LSM robot moves to the CAP and performs a CAP catalog.

    b. If cartridges are detected in the CAP, HSC issues a message instructing the operator to remove the cartridges.

    c. HSC issues the message:

        ... Open CAPid *AA:LL:CC* for entering when unlocked

    d. The CAP ENTER indicator lights up or Enter appears on the display panel.

2. Issue the DRAin command against that CAP.

    The CAP is locked and the status is set to INACTIVE.

Freeing An Allocated CAP

> **Caution:**   *Use this procedure only if the CAP is not in use.* Releasing a CAP from an active command or utility may produce unpredictable results. Examples of these are:
>
> - CAP cleanup
> - Errant cartridges
> - Cartridges still in the CAP that are unknown to HSC

If a host terminates without performing recovery while an active process is using a CAP, the CAP remains allocated to the host and is unavailable to all other hosts. If HSC is brought back up on that host, it releases the CAP after initialization. Otherwise, you can free the CAP by issuing the following command from any connected host:

RELEASE *cap-id*

where:

cap-id specifies the CAP you wish to release.

HSC issues a message prompting you to confirm that the CAP is to be released.

- Reply "N" to cancel the release.
- To proceed with the release:

1. Verify that the CAP is not in use by visually inspecting the CAP, issuing the Display Cap command on other hosts, and checking with other personnel.

2. Reply "Y" to the message.

   Another message confirms the CAP is released.

Restoring Host-To-Host Communications Services

The `COMMPath` command and control statement lets you define a tiered communications service between HSC hosts. With this type of service in place, HSC automatically performs a path switch if a host receives an error trying to send a message over the current communications path. Depending on the paths defined, an automatic switch can be to an equivalent path type (from one LMU path to another), or to a lower performance method (for example, from VTAM to LMU).

HSC notifies you when a switch occurs by issuing the following message to the console on both hosts:

```
... A communications path switched from XXXX1 to XXXX2; ...
```

where:

XXXX1 specifies the previous communications path.

XXXX2 specifies the current communications path.

The message also identifies the host that either initiated the switch or experienced the failure.

After HSC performs a downward switch, you can make an upward switch only with the `COMMPath` command.

To make an upward switch:

1. Find and correct the problem which prompted the switch.

2. Return to the previous communications method by issuing the command

   ```
   COMMPATH HOSTID(host-id) METHOD(method)
   ```

   where:

   host-id specifies the host which the command affects.

   method specifies the method you want to be made current.

Resolving Possible Lost Mount Requests

Mounts may be lost due to one of several reasons:

- If SMC is inactive while HSC is active, then MVS and/or TMS messages are not intercepted.

- If HSC is inactive while the SMC is active, then MVS and/or TMS messages are intercepted but there is no Automated Library Control system to receive the mount request.

- A hardware LMU error can occur.

In general, to resolve these problems, use the SMC `RESYNC` command to redrive all lost mount requests.

Resolving Lost Mount Responses (ORH)

Occasionally a mount request gets stuck on one of the HSC queues due to a response not being received from the LMU. When this happens the volume is left in a selected status and the operator is notified by the Overdue Response Handler (ORH).

Overdue Response Handler

Every 30 seconds, the ORH scans all queues for each ACS to identify requests with sequence numbers whose response wait time intervals have expired. When a time interval expires, HSC issues the message

```
... ORH interval expired: ...
```

and the ORH checks the request type code. For all request type codes other than mount, dismount, move, and swap, the ORH automatically extends the request's wait time another increment of 3, 5, 10, or 60 minutes, and issues the message:

```
... ORH interval extended: ...
```

Operator Response Required For Mount, Dismount, Move, and Swap

For mount, dismount, move, and swap requests, the operator is prompted by the message:

```
... ORH: ABORT/EXTEND...
```

Respond with one of the following:

- ABORT — the task that requested the operation is notified that the request has been abnormally terminated by ORH. This removes the volume from selected status and lets you issue HSC Mount or DISMount commands as necessary to keep the work moving. For transport related failures, the ABORT response should be used. Examples of these are Pass-Thru-Port, SL8500 elevator, robot, or transport failures.

- EXTEND — the ORH adds another ten minute increment to the request and leaves it on the same queue that it is currently on. There is no limit to the number of extensions you can request.

Using Software Diagnostics

Some problems are beyond your ability to resolve by yourself, and usually result in a call to Oracle Software Support, who may ask you to use one or more of the HSC software diagnostic tools described in Table 3–2.

Table 3–2   HSC Software Diagnostic Tools

Diagnostic Tool	Description
Generalized Trace Facility (GTF) Tracing (GTRACE)	The Generalized Trace Facility (GTF) provides a low level history of the recent past. The HSC issues GTRACE requests for significant events during operation. Format ID and Event ID (FID and EID) are specified as parameters on the EXEC statement (PARM=) of the HSC JCL procedure (refer to "Creating an HSC START Procedure" in the HSC Configuration Guide for an example procedure).

Table 3–2   (Cont.) HSC Software Diagnostic Tools

Diagnostic Tool	Description
Supervisor Call (SVC), Dump (SDUMP), and Abnormal End (ABEND) Dumps	SVC and ABEND dumps are taken where appropriate to diagnose software failures. Do not use SYSUDUMP, SYSMDUMP or SYSABEND DD statements in HSC JCL. These ABEND types do not provide enough information on the HSC address space.
Error Recording Data Sets (ERDS)	Software failures are logged into these error recording data sets. 4480 Cartridge Subsystem and 3278 Terminal Subsystem Error Recording Data Set (ERDS) records are written by MVS routines.
LISt Command	The `LISt` command is used to display an HSC data structure and to provide for diagnostic support for the HSC. *Use this command under the direction of Oracle support personnel.*
Display Command	The `Display` command offers several options that are useful for diagnostic purposes.

4

Using the VTCS Dashboard

"Using the VTCS Dashboard" is basically about reviewing your MVC and VTC reports. The main pieces and parts of your virtual tape configuration are VTSSs, VTDs, VTVs, RTDs, and MVCs, and so it is not surprising that a lot of your daily and weekly routine is making sure they are all operating normally.

Checking Virtual Tape Status (Daily)

Think of a VTSS as the virtual counterpart of a Nearline ACS, a VTD as the virtual counterpart of a Nearline real tape drive, and a VTV as the virtual counterpart of a Nearline volume, and it is easy to see why it is important to make sure all your VTSSs, VTDs, and VTVs are operating normally.

To check Virtual Tape status:

1. Enter a Display VTSS command.

   You should see something like Table 4–1.

   Table 4–1   Example output from Display VTSS - Good VTSS status

VTSS NAME	CAPACITY (MB)	DBU	HI AMT	LOW AMT	VTV COUNT	MX MT	MN MT	DEF ACS	AUTO MIG	STATE
HBVTSS16	56,209	55	80	60	2440	6	3	02	-	ONLINE
HBVTSS17	56,209	50	80	60	2180	6	3	02	-	ONLINE
HBVTSS18	56,209	52	80	60	2288	6	3	01	-	ONLINE
HBVTSS19	93,184	45	80	60	1900	6	3	01	-	ONLINE

   Table 4–1 shows the status of four "operating correctly" VTSSs:

   - First, all VTSSs are online, which is usually good.

   - The LAMTs are all 60 and the HAMTs are all 80, which are good ranges for optimizing VTSS usage and promoting effective automatic migration.

   - The DBUs are all well lower than the HAMT, which means there is still space to grow on the VTSS before auto-migration starts. You might even consider routing more work to these VTSSs to optimize your investment in virtual tape.

   - Assume you have eight RTDs connected to each VTSSs. The MX MTs (maximum migration tasks) are set at 6 and the MN MTs (minimum migration tasks) are set at 3, which are good numbers. The max of 6 leaves 2 RTDs for recalls/reclaims, and the minimum of 3 ensures that if a bunch of migrations suddenly start, there are enough tasks to handle the load.

What if Display VTSS *does not* look so good? Maybe something like Table 4–2.

Table 4–2   *Example Output from Display VTSS - VTSSs with Major Operational Issues*

VTSSNAME	CAPACITY (MB)	DBU	HI AMT	LOW AMT	VTV COUNT	MX MT	MN MT	DEF ACS	AUTO MIG	STATE
HBVTSS16	56,209	90	80	60	27,888	4	2	02	-	ONLINE
HBVTSS17	56,209	92	80	60	28,974	4	2	02	-	ONLINE
HBVTSS18	56,209	90	80	60	22,005	4	2	01	-	ONLINE
HBVTSS19	93,184	92	80	60	26,009	4	2	01	-	ONLINE

Table 4–2 shows the status of four VTSSs that have some major operational issues:

- At least they are all online. If not, unless you know a reason why they *ought* to be offline or in maintenance mode, enter some `Vary VTSS` commands to get them back online.

- The DBUs are all too high. Anything in the 90 plus range means the VTSSs are struggling to auto-migrate VTVs, which is not surprising because of the following.

- Assume you have eight RTDs connected to each VTSSs. The MX MTs (maximum migration tasks) are set at 4 and the MN MT (minimum migration tasks) are set at 2, which is definitely a little light for the current migration load.

    To fix the problem, go to step 2.

2. If you did not like what you saw in step 1, tune up your operating parameters.

    First, get some more migration tasks going:

    ```
    set migopt vtss(vtssname) maxmig(8) minmig(8) high(70) low(40)
    ```

    You now have all RTDs fully engaged in migration on all VTSSs. Keep it that way until you get the DBU under control, then go back to something like 6 max, 3 minimum. You also changed the AMTs to 40 low, 70 high. This gets you out of trouble, and the next time around, migration starts sooner and bring the buffer to a lower DBU.

    Next, enter a Display VTD to get a picture of your system's VTDs.

    Table 4–3 shows an example of Display VTD output.

Table 4–3   *Example output from Display VTD - Operating Correctly*

DRIVE	LOCATION	VTV	STATUS
A800	HBVTSS16	X00778	MOUNTED
A801	HBVTSS16	X00775	MOUNTED
A802	HBVTSS16	-	AVAILABLE
A803	HBVTSS16	-	AVAILABLE

Once again, you are in good shape in Table 4–3, with some VTDs in use and others ready to be used.

What if *all* VTDs showed VTVs mounted? Not so good, because then you risk job allocation failures if no drives are available. If it is because of something like the problem that occurred in step 2, then just accept it and vow that, in the future, you arrange workloads to let your VTDs better handle the influx of data. If this is a

long term problem, however, you may need more VTSSs or upgrades to VTSSs with more capacity and more VTDs.

3. Now it is time to see if you have enough scratch VTVs on hand by entering a Display SCRATCH command, which produces the output shown in Table 4–4.

Table 4–4   Example output from Display SCRATCH

SUBPOOL-NAME	SCRATCH-COUNT
VIR000	14,364
VIR0002	13,582
VIRTUAL	19,132
VIRTUAL1	9,905

In Table 4–4, what you are seeing are VTV scratch counts for HSC subpools. If you do not use HSC subpools for VTVs, you see VTV scratch counts for all VTVs defined to your system. There are no good or bad numbers for scratch counts, as long as you have some scratch VTVs available. The "right number of available scratches" depends on your shop's needs and workload.

If Table 4–4 showed 50 or fewer scratches available for each subpool, you might be a little concerned, right? If this happens, you can do one or more of the following:

- **Free VTV volsers by scratching VTVs with data that is not current.** This is the move to make if you have enough total VTVs in your system, but there are not enough scratch volumes available.

  Actually, *you* do not do the scratching, it is your TMS that is supposed to do the work, and the initial configuration should have defined your VTV volsers to your TMS. If that did not happen, go back and do it. More likely, you may have added VTV ranges through the VTCS CONFIG statement and forgot to define the new ranges to your TMS, so go back and fix that problem. This is all described in *Installing ELS*.

  Note however, that marking VTVs as scratch with your TMS is just part of the solution. What *also* has to happen is that somebody in VSM-land has to mark the VTV data non-current (and therefore, writeable), and, in the case of VTSS-resident VTVs, delete them from the buffer.

  Because actually deleting VTV data is a significant decision (data is gone), you are going to make a judgment call and say it is an "as-needed" task, if you go this route, turn to "Doing Demand MVC Space Reclamations."

- **Use POOLPARM or VOLPARM to add VTVs.** This is your next best option, and you do it when you really do not have any VTVs with non-current data. It turns out that it is not just POOLPARM or VOLPARM, you also have to do the appropriate TMS definitions, and so forth. Again, this is all described in *Installing ELS*.

- **Change your TAPEREQ statements or SMS routines to temporarily reroute tape work to Nearline HSC processing until you can define additional VTVs.** This is probably your least desirable option, because you are essentially sending data directly to Nearline tape that you originally intended to send to VSM, so cleaning up the mess is not trivial. Still, if you have Nearline resources available and you have an immediate need to write data to scratch volumes, this is the (temporary) way to go.

Checking Nearline Tape Status (Daily)

"Checking Virtual Tape Status (Daily)," talks about how important it is to ensure that your system's VTSS, VTDs, and VTVs are operating normally.

Think of the Nearline component of VSM (RTDs and MVCs), the place where VTVs are being migrated to and recalled from, with some MVC space reclamation going on in the background, and you will agree that this area deserves your full attention, too.

To check Nearline Tape status:

1. Enter a Display RTD.

   A good picture might be something like Table 4–5.

Table 4–5   Example output from a VT Display RTD command - All is well

RTD	STATUS	MOUNT	ALLOC	HOST	VTSS
B200	ONLINE/FREE	-	-	-	HBVTSS16
B201	ONLINE/FREE	-	-	-	HBVTSS16
0B79	ONLINE/FREE	-	-	-	HBVTSS16
0B7A	RECALL VTV	DMV051*	DMV051	EC20	HBVTSS16
1600	MVS1 :MIGRATE	-	-	-	-
1601	MVS1 :MIGRATE	-	-	-	-

   In Table 4–5, things are moving along smartly because there is a good balance of RTDs migrating, recalling, and available for new work, which is not true in Table 4–6.

Table 4–6   Example output from a VT Display RTD command - RTDs in Distress

RTD	STATUS	MOUNT	ALLOC	HOST	VTSS
B200	MVS1 :MIGRATE	-	-	-	-
B201	MVS1 :MIGRATE	-	-	-	-
0B79	MVS2 :MIGRATE	-	-	-	-
0B7A	MVS2 :MIGRATE	-	-	-	-
1600	MVS1 :MIGRATE	-	-	-	-
1601	MVS1 :MIGRATE	-	-	-	-

   If Table 4–6 is a result of the kind of emergency measures employed in step 2, then there is not much you can do other than wait until things quiet down. If, however, you do have other RTDs available, for example, RTDs that you are sharing manually with MVS and VSM, vary them offline to MVS and use Vary RTD to make them available to VTCS.

2. Next, see how your MVCs are doing with a Display MVCPool command.

Figure 4–1 shows an example of Display MVCPool output with no MVC pool name specified, so you see information for all your system's MVCs.

Figure 4–1   Example output from Display MVCPool (no pool name specified)

```
                         MVCPOOL INFORMATION

ACS          MEDIA      FREE-MVCS          RECLAIM-MVCS        USED-MVCS
                        VOLS     GB        VOLS     GB         VOLS     GB

00           ECART      310      248       4        1.2        100      65
00           ZCART      120      192       1        0.5        250      400

00           TOTAL      430      440       5        1.7        350      465


01           ECART      90       144       15       6.2        322      485
01           ZCART      35       700       3        11.3       43       675

01           TOTAL      125      844       18       17.5       365      1160


NON-LIB      STK2P      22       1100      0        0          12       1565
NON-LIB      TOTAL      22       1100      0        0          12       1565
```

Figure 4–1 shows that your MVC collection is in good condition. There appear to be plenty of Free MVCs (100% usable space, no migrated VTVs) and plenty of free space across multiple ACSs and MVC media types. Numbers for MVCs eligible for reclaim are relatively low, meaning that automatic space reclamation probably does not get in the way of migrate/recall activity.

Used MVCs look all right in proportion to Free MVCs, except maybe in ACS 01, ECART and ZCART media. For these MVCs, you might want to do some investigation. Find out which Storage Classes represent these MVCs, which Management Classes correspond to these Storage Classes, maybe scratch some VTVs by Management Class.

What if Display MVCPool showed you something like Figure 4–2?

Figure 4–2   Example output from Display MVCPool - Problem in ACS01

```
                         MVCPOOL INFORMATION

ACS          MEDIA      FREE-MVCS          RECLAIM-MVCS        USED-MVCS
                        VOLS     GB        VOLS     GB         VOLS     GB

00           ECART      310      248       4        1.2        100      65
00           ZCART      120      192       1        0.5        250      400

00           TOTAL      430      440       5        1.7        350      465


01           ECART      10       15        35       18.2       382      585
01           ZCART      5        100       20       511.3      53       1675

01           TOTAL      15       115       55       529.5      435      2260


NON-LIB      STK2P      22       1100      0        0          12       1565
NON-LIB      TOTAL      22       1100      0        0          12       1565
```

As you can see, things have gotten *worse* in ACS 01. What do you do now? Consider the following, probably in this order:

- Do demand reclamation to free space; for more information, see "Doing Demand MVC Space Reclamations."

- Add MVCs as described in "Changing RTD Device Types."

- Find out which Storage Classes represent these MVCs, which Management Classes correspond to these Storage Classes, maybe scratch some VTVs by Management Class.

  As a follow-up, review your current policies and adjust as needed. You may be able to create free MVCs or free space on MVCs by changing these policies.

Looking at the Big Picture (Weekly)

This one is not too complicated, and basically consists of running your other two primary status tools (MVC Report and VTV Report) on a weekly basis.

Using the VTV Report

Note: The VTV report is run using one of the following commands:

- `VTVRPT BASIC`, which shows all VTV copies migrated to MVCs using the `EXPORT` command

- `VTVRPT COPIES`, which shows all VTV copies migrated to MVCs using the `EXPORT` command, and, additionally, all VTV copies migrated with the `EEXPORT` command

Start with a VTV Report, which looks like Figure 4–3.

Figure 4–3   Example output from VTVRPT

```
SLUADMIN (7.1.0)                    STORAGETEK VTCS SYTEM UTILITY                              PAGE 0002
TIME 06:32:03                       VTCS VTV REPORT                                            DATE 2010-03-20

VTV     SIZE    COMP% <----CREATION-----> <----LAST USED----> MIGR SCRT RESD RELPL  MGMT  MVC1   MVC2   MVC3   MVC4   MAX VTSS
VOLSER  (MB)          DATE      TIME      DATE      TIME                             CLASS                                VTV NAME
X00T00 0.04     84    2010JUL16 05:02:08  2010JUL19 05:41:00  M    -    R    -      VCL4  022550 022551 022552 022553 .8  VTSS16
X00002 <MOUNT>        2010JUL14 05:02:09  2010JUL19 07:43:46  M    -    R    -      VCL4  033550 033551 033552 033553 .8  VTSS17
X00003 15.60    84    2010JUL14 05:02:10  2010JUL19 05:41:28  M    -    R    -      VCL2  044550 044551                .4  VTSS16
X00004 0.36     84    2010MAY28 05:02:11  2010JUL19 05:41:30  M    S    R    -      VCL3  022550 022551 022552        .8  VTSS16
X00005 15.60    84    2010JUL14 05:02:12  2010JUL19 05:41:31  M    -    R    -      VCL1  033550                       .4  VTSS16
X00006 15.60    84    2010JUL14 05:02:13  2010JUL19 08:45:31  C    -    -    -      VCL4  044550 044551 044552 044553 .8  VTSS17
                                                                                                                    ELS_008
```

At first glance, the VTV Report looks overwhelming and non-intuitive, a lot lines of data each describing everything you ever wanted to know about each and every VTV in your system.

What are some ways you can make a VTV Report more useful to your situation? To start with, you can run the VTVRPT utility against a list of volsers, a volser range, or even an individual volser. If you have specific VTVs in mind that you want to examine, use one of these selection options.

Second, the VTVRPT utility also provides the OPTION(UNAVAIL) parameter, which produces a report on unavailable VTVs that looks like Figure 4–4.

Figure 4–4   Example output from VTVRPT (UNAVAIL option)

```
SLUADMIN (7.1.0)              STORAGETEK VTCS SYTEM UTILITY                    PAGE 002
TIME 06:59:03                 UNAVAIL MOUNTED VTV REPORT                       DATE 2010-03-20

VTV    SIZE    COMP% <----CREATION-----> <----LAST USED----> MIGR SCRT RESD RELPL  MGMT   MVC1 MVC2 MVC3   MVC4   MAX VTSS
VOLSER (MB)           DATE      TIME      DATE      TIME                            CLASS                          VTV NAME
Y09053 <MOUNT>       2010MAR19 09:34:14  2010MAR20 05:55:44  -    -    R    -       M9                          .8  HBVTSS16


SLUADMIN (7.1.0)              STORAGETEK VTCS SYTEM UTILITY                    PAGE 003
TIME 06:59:03                 UNAVAIL RESIDENT VTV REPORT                      DATE 2010-03-20

VTV    SIZE    COMP% <----CREATION-----> <----LAST USED----> MIGR SCRT RESD RELPL  MGMT   MVC1 MVC2 MVC3   MVC4   MAX VTSS
VOLSER (MB)           DATE      TIME      DATE      TIME                            CLASS                          VTV NAME
X00007 156.24  89    2010JAN10 03:00:02  2010MAR01 04:51:47  -    S    R    -                                     HBVTSS16
X01010 3.90    0     2010MAR01 09:10:37  2010MAR01 09:10:37  -    -    R    -                                     HBVTSS16
X01014 3.90    0     2010MAR01 09:11:08  2010MAR01 09:11:08  -    -    R    -                                     HBVTSS16
X01021 3.90    0     2010MAR01 09:21:11  2010MAR01 09:21:11  -    -    R    -                                     HBVTSS16


SLUADMIN (7.1.0)              STORAGETEK VTCS SYTEM UTILITY                    PAGE 004
TIME 06:59:03                 UNAVAIL FENCED VTV REPORT                        DATE 2010-03-20
VTV    SIZE    COMP% <----CREATION-----> <----LAST USED----> MIGR SCRT RESD RELPL  MGMT   MVC1 MVC2 MVC3   MVC4   MAX VTSS
VOLSER (MB)           DATE      TIME      DATE      TIME                            CLASS                          VTV NAME
X01280 <FENCED>                                               -    -    -    -
X04762 <FENCED>                                               -    -    -    -
X04776 <FENCED>                                               -    -    -    -
X02019 <FENCED>                                               -    -    -    -
X10066 <FENCED>                                               -    -    -    -
X10068 <FENCED>                                               -    -    -    -                                        ELS_009
```

Obviously, OPTION(UNAVAIL) is the best option if you have had any reports of jobs (or VTCS) that are unable to access VTVs you thought were resident.

Also you get flexibility with XML output from the VTVRPT utility. You can generate, for selected reports and utilities, output in either Structured XML or Comma Separated Variables (CSV) XML.

What is the difference between Structured XML and CSV output? Consider this:

■   Structured XML contains all of the tags and structures shown for each command or utility (which you can then process, as desired, using a programming language of your choice).

■   CSV output lets you select only the tags (and their order) that you want. Each output line contains a fixed number of fields separated by commas, which can then be input into spreadsheets or report writers for customizable analysis or reports.

You now have two ways you can effectively customize a basic VTV Report for your shop's needs. For more information on this topic, see *ELS Programming Reference*.

Finally, you should be aware that LCM provides enhanced management and reporting capabilities that correspond to ELS/VTCS functions, including the VTCS MVC and VTV Reports. For more information, see "LCM Control Statements."

Using the MVC Report

Finally, look at an MVC Summary Report, which looks like Figure 4–5.

Figure 4–5 Example MVC summary report

```
SLUADMIN (7.1.0)              STORAGETEK VTCS SYTEM UTILITY                  PAGE 0002
TIME 09:26:54                 VTCS MVC SUMMARY REPORT                        DATE 2010-04-13

MVC     NUMBER  %USED  %AVAIL  %FRAG  MEDIA      TIMES    STATUS   <-----LAST MOUNTED----->   ACS  OWNER/
VOLSER  OF VTVS                       SIZE (MB)  MOUNTED  IBLDRUTM DATE       TIME    VTSS     ID   CONSOLIDATE TIME

EVS99   200     10.80  84.57   4.63   2000       310      I----U-M 2010MAR15 03:20:23 VTSS8    00   S1
EVS100  0       0.00   100.00  0.00   UNKNOWN    206      --L--U-- 2010MAR10 05:24:04 VTSS8    --
EVS101  1009    99.00  0.00    1.00   400        306      I----U-- 2010MAR15 03:20:23 VTSS8    00   S1
EVS102  5       8.25   91.75   0.00   400        6        I----U-- 2010MAR15 04:23:04 VTSS8    00   S3
EVS103  EXPVTV  0.12   99.88   0.00   400        194      I----J-- 2010MAR15 03:20:28 VTSS10   00   VTSS10
EVS104  0       0.00   100.00  0.00   400        5        I---RC-- 2010MAR18 03:49:14 VTSS8    00   2010APR12 03:49:14
EVS104  200     10.80  84.57   4.63   102040     254      I---RUT- 2010MAR18 04:10:09 VTSS8    00
EVS105  300     15.80  54.57   4.63   102040     154      I---RUW- 2010MAR18 04:10:09 VTSS8    00
EVS106  0       0.00   100.00  0.00   400        202      I----C-- 2010MAR18 03:49:20 VTSS8    00
EVS107  0       0.00   100.00  0.00   400        171      I---RE-- 2010MAR18 04:13:00 VTSS8    00

        137     Initialized MVCs processed
        8       Non-Initialized MVCs processed                                              ELS_010
```

The MVC Summary Report is a lot like the stock VTV Report; nice, if you know what you are looking for, but perhaps too much information otherwise.

The additional fields you get from an MVC Detailed Report may be more useful for a better picture. See Figure 4–6.

Figure 4–6 Example MVC detailed report (additional fields)

```
SLUADMIN (7.1.0)           STORAGETEK VTCS SYTEM UTILITY          PAGE 0002
TIME11:28:30               MVC EVS102 DETAIL REPORT               DATE 2010-06-03

VTV       SIZE     BLOCK       MANAGEMENT  MIGRATION  BLOCK       MESSAGE
VOLSER    (MB)     ID          CLASS       DATE       COUNT

X20041    76.00    00000000    M5          2010JAN08  10
X20043    76.00    134009C7    M5                     9
X20044    76.00    2A40138D    M5          2010JAN08  9
X20045    76.00    C6401D53    M3                     10
X20047    76.00    A5402719    M3                     10

5 VTVS FOUND FOR MVC:EVS102

WARNING VTV COUNT:5 DOES NOT MATCH MVC SUMMARY RECORD VTV
COUNT:22 FOR MVC:EVS102                                           ELS_011
```

Here is some detailed information about the VTVs on an MVC that you can do some diagnostic work with, if necessary.

Just as is now available with the VTV Report, you can also do either of the following with the MVC Report:

- Generate output in either Structured XML or Comma Separated Variables (CSV) XML as described in *ELS Programming Reference*.

- Use the corresponding LCM reports, as described in "LCM Control Statements."

In Summary

That is about all there is to using the VTCS Dashboard. There is a bunch of other tasks you *could* do, and probably will do, such as but not limited to running MVC Pool Reports if you used Named MVC Pools. But that is information for "As Needed" management tasks.

The main thing you should learn from this chapter is do your dailies as described in "Checking Virtual Tape Status (Daily)" and "Checking Nearline Tape Status (Daily)" and your weeklies discussed in "Looking at the Big Picture (Weekly)," and you will keep your VTCS system running properly.

5

Working the VTCS Must Do (Sometime) Chores List

Welcome to the "VTCS Must Do (Sometimes) Chores List," also known as the "As-Needed Tasks List." For example, if this is the week you decide to run DELETSCR to zero out a list of scratched VTVs that are tying up major amounts of your valuable VTSS and MVC space. Great, job well done. How long do you think it is going to be before you have to run the same operation? Especially if you do not change your delete on scratch policies? Answer: It might be a day, a month, or a year, but you will get to do it again.

No worries, however. Useful procedures are discussed here to help you pare down the Must Do (Sometime) List, and, as you have already seen from reading "Using the VTCS Dashboard," if you keep a close eye on your MVC and VTV reports, you may not even *need* a list, because they tell you when it is time to do the Must Do/As Needed chores.

There is also another class of "Must Do (Sometime)" chores that are almost policy decisions, but included here because (a) they are proactive in nature, which makes them doubly valuable as Best Practices "As Needed" chores, and (b) they are operating techniques you can use, back out, and reintroduce as they benefit (or not) your shop at any point in time. Having said that, lead off with three of our favorites in this category as described in "Doing Demand Space Reclamations, Migrations, and Recalls."

Doing Demand Space Reclamations, Migrations, and Recalls

These tasks are optional, but, especially in the case of Demand Space Reclamations, *highly* recommended Best Practices for reasons that soon become obvious.

Doing Demand MVC Space Reclamations

As you already know, VSM automatically reclaims MVC space on each host running reclamations, the key word being *automatically*. That means that space reclamation is always out there looking for work, and although it is a background task, if you have got a lot of fragmented MVCs, the space reclamation work can seriously interfere with migration/recall, especially during peak processing periods.

If your MVC summary report or Display MVCPool shows a high level of fragmentation on your system's MVCs (and this level is below the value specified on the CONFIG RECLAIM THRESHLD parameter or the MVCPool THRESH parameter), you may want to schedule demand MVC space reclamation as an off-hours batch job.

You do demand MVC space reclamation with RECLaim. Open up your *ELS Command, Control Statement, and Utility Reference* and you see some useful tools you can use to optimize demand reclamation and run it most efficiently:

- You can use *one only* of the MVCPOOL, STORCLAS, ACSid, or MVC parameters to filter the list of MVCs to process. Your MVC and VTV reports, as described in "Using the VTCS Dashboard," help you narrow the list of likely candidates to an MVC Pool, Storage Class, specific ACS, or range or list of MVCs. Input this list into RECLaim, and you are using the right tool for the job.

  Note that if you do not specify one of these parameters, space reclamation selects MVCs from the Named MVC Pool (if implemented) or media type (for multiple MVC media environments) most in need of free space.

- The parameters MAXMVC (max MVCs processed by a single space reclamation task), THRESH (MVC's fragmented percentage that make it a reclaim candidate), and CONMVC (the maximum number of MVCs that VTCS concurrently processes for both drain and reclaim) let you override the corresponding CONFIG RECLAIM global parameters for the demand reclamation. This gives you the ability to tune your demand migrations to be more or less aggressive than your automatic migrations.

- NOWAIT is a way of speeding up the process and CONMVC is another tuning method for influencing the number of MVCs processed at a time (see *ELS Command, Control Statement, and Utility Reference* for details).

- ELAPSE is a way to detect if no demand reclaims have happened within an interval you specify. If there are no reclaims in this period, the job stops.

- *Also note that* VTCS enforces the *strictest* limiting factor. For example, if you run RECLAIM and specify ELAPSE equal to 5 hours and MAXMVC equal to 10 *and* VTCS reclaims 10 MVCs in one hour, then VTCS terminates the reclaim before the ELAPSE value expires.

- VTCS and HSC must be active to process a RECLAIM request.

Doing Demand VTV Migrations

As already noted, VTCS/ELS is basically a server. For example, VSM automatically manages VTSS space and migrates VTVs to ensure a balance of optimum data availability, resource use, and data protection.

That is great for a stable environment, but what if you find out your VSM system is about to receive a whole lot of application data? Answer: It might be time to run a demand migration batch job to free VTSS space before the aforementioned peak tape processing event occurs.

You do demand migrates with, of course, MIGRATE, which provides the following options:

- You can migrate VTVs:

  – by volser (repeats allowed)

  – the Management Class

  – the data set name associated with the VTV (most effective).

  There is also a DELETE(YES) option that you may want to employ that deletes the VTV from VTSS space after a successful migration. You typically use DELete (YES) (the default) for VTVs not likely to be reaccessed. As an alternative, you can

specify DELete (NO) to ensure that critical data is available and quickly migrated for VTVs likely to be reaccessed.

■ The NOWAIT option helps you speed up the process. That is all using MIGRATE Format 1; see *ELS Command, Control Statement, and Utility Reference* for details.

■ As an alternative, you can use MIGRATE Format 2 to do a demand migrate-to-threshold for all or specific VTSSs. That is a nice tool for getting your DBU where you want it, and VTCS deals with the details.

Also note that, with SET MIGopt, you can lower the high AMT to effectively force a demand migration.

Doing Demand VTV Recalls

VTCS provides an automatic recall process which starts when a job requests a data set on a VTV that is migrated to tape, but not VTSS-resident. What if, however, you have the reverse of the above situation? For example, you are doing end-of-year processing, and *know* there is a bunch of jobs that want to read data from VTVs that are on tape only. Demand recall is the solution.

RECALL provides you with all the flexibility of control you need:

■ As with MIGRATE, you can recall VTVs by volser, Management Class, or associated data set name.

■ You can specify the VTSS where you want to recall the VTVs. Otherwise, the default is the VTSS of creation, and there are some considerations associated with the VTSS recall policy. See *ELS Command, Control Statement, and Utility Reference* for details.

■ RECALWER lets you specify whether you want to recall VTVs with read data checks.

■ There is a NOWAIT option to speed things up.

Working with RTDs

There is a lot of RTD management that you confine to "Finding and Fixing VTCS Problems," because they are almost exclusively error recovery scenarios. Best Practices for RTDs would be to have enough of them and keep all of the above up and running. Remember, RTDs get used for migrates, recalls, *and* reclaims, so keeping the right mix of RTDs for all of these jobs is a critical balancing act. To adjust this mix through operating parameters, see "Checking Virtual Tape Status (Daily)."

In addition to adjusting the RTD operating parameters, the other main tool you have with RTDs is the VTCS Vary RTD command, which you use to change RTD states. You can vary RTDs online, offline, or into maintenance mode if you need to do maintenance on the RTD.

The major as-needed tasks you are likely to encounter are related, and the first two use Vary RTD:

■ "Changing RTD Device Types," which is basically how to do a technology upgrade of some or all of your system's RTDs.

■ You need to consider the way you specify MVC media. Even though these are really *MVC* considerations, they occur because of a change of RTD device types. For more information, see *Configuring HSC and VTCS*.

Changing RTD Device Types

Use the following procedure to change RTD device types. *Note that* changing RTD device types requires you to *stop VTCS on all hosts*.

To change RTD device types, do the following:

1. Review your VSM policies.

   For example, you may want to review your Management Class and Storage Class definitions if this RTD device type is used for migrations.

2. Vary the old RTDs offline to VTCS.

3. If the new RTD devices use new MVS device addresses, do the following:

   ■ Define the new addresses to MVS.

   ■ Run DECOM to output your CONFIG statements.

   ■ Edit the CONFIG statements to change the RTD addresses to the new values.

   ■ Run CONFIG RESET.

   Caution: *Do not* vary the new transports online to MVS! Otherwise, they can be allocated as Nearline transports.

   ---

4. Install the new RTDs.

5. Vary the LSM(s) where transports were replaced to offline status.

6. Vary the LSM(s) where transports were replaced to online status.

7. Vary the new RTDs online to VTCS.

8. If necessary, add MVCs.

   For more information, see "Adding MVCs."

Working with MVCs

As you already know, it is somewhat difficult to limit a discussion to any one of your virtual entities. MVCs contain VTVs, so it is hard to talk about either in isolation, because you inevitably end up talking about the other. And if you are discussing VTVs, you are also talking about VTSSs and VTDs.

Having said that, the following sections are some basic procedures for doing fairly typical "as-needed" tasks with MVCs that are done for a number of reasons. For example, you might add MVCs because you are running out of space as described in the previous scenario or because you are being proactive and *do not* want to get into trouble.

Note: If you delete MVCs from the configuration as a result of either SET VOLPARM or CONFIG MVCVOL processing:

■ You cannot re-enter the volsers into the configuration as VTVs.

■ Do not use the volsers for native HSC tapes.

Message SLS6944I indicates the number of MVCs that have been deleted.

Adding MVCs

If you have not heard, ELS 7.2 just made adding volumes a *whole* lot easier. You now use the HSC `VOLPARM` and `POOLPARM` statements to define *all* volumes and their pools: native Nearline volumes, cleaning cartridges, MVCs, and VTVs, and the HSC `SET VOLPARM` utility to load them. For more information, seem *Configuring HSC and VTCS* and *ELS Command, Control Statement, and Utility Reference*.

To add MVCS:

1. Create a VOLPARM statement to define the MVCs.

   For example, to define a range of T10000 Full volumes to be encrypted:

   ```
   VOLPARM VOLSER(T10K2000-T10K2999)MEDIA(T10000T1)RECTECH(T1AE)
   ```

2. Create a POOLPARM statement to define the MVC pool.

   For example, to define the T10000 MVC Pool with reclaim parameters:

   ```
   POOLPARM NAME(SYS1MVCT1)TYPE(MVC)MVCFREE(40) MAXMVC(4) THRESH(60) START(70)
   ```

3. Create or update your MGMTCLAS or STORCLAS statements as needed.

   For example, if you added a new MVC media type, follow the recommendations in *Configuring HSC and VTCS*.

4. Update your POLICY or TAPEREQ output parameters as required.

   For example, if you created a new Management Class in step 3, update or create your `TAPEREQ` or `POLICY` statements to point to the new Management Class.

5. Do you need to define VTVs?

   If so, go to "Defining VTVs." Otherwise, go to "Validating and Applying the Volume Definitions."

Defining VTVs

To define VTVs:

1. Create POOLPARM or VOLPARM statements to define the VTVs.

   For example, to define two ranges of VTVs for use by hosts `MVS1` and `MVS2`:

   ```
   POOLPARM NAME(SYS1VTV1)TYPE(SCRATCH)
   VOLPARM VOLSER(V5000-V5499)MEDIA(VIRTUAL)
   POOLPARM NAME(SYS1VTV2)TYPE(SCRATCH)
   VOLPARM VOLSER(V5500-V5999)MEDIA(VIRTUAL)
   ```

2. Go to "Validating and Applying the Volume Definitions."

Validating and Applying the Volume Definitions

1. Run SET VOLPARM to validate the VOLPARM/POOLPARM statements.

   ```
   SET VOLPARM APPLY(NO)
   ```

   `APPLY(NO)` validates the statements without loading them. If you like the results, go to step 2. Otherwise, rework your volume definitions, then go to step 2.

2. Run SET VOLPARM to load the VOLPARM/POOLPARM statements.

   ```
   SET VOLPARM APPLY(YES)
   ```

3. Physically enter any real cartridges into the ACS.

4. For more information, see "Entering Cartridges."

Removing MVCs from the Pool

Why would you remove MVCs from the pool? A typical scenario might be that you are swapping out older drives for a technology refresh for your RTDs and you want to retire the old media, in which case, you get to add new MVCs to the pool as described in "Adding MVCs" and then remove the old media as described in "Permanently Removing MVCs."

Note that there are occasions where you might want to temporarily remove MVCs from the pool. For example, you have got some bad media or suspected bad media. You want to remove the bad media and put in replacements, basically under the same volsers as described in "Temporarily Removing MVCs."

Permanently Removing MVCs

To permanently remove MVCs from the pool, do the following:

1. Enter MVCDRain to drain the MVCs.

   For example, to run the MVCDRain to drain the MVCs in Storage Class STORCL1, virtually eject the MVCs, and return after the request is submitted, enter the following:

   `MVCDRAIN STORCLAS(STORCL1) EJECT NOWAIT`

2. If the MVCs are no longer required in an ACS, use an HSC `Eject` command to eject the MVCs from the ACS.

3. Remove the security restrictions, and tape management system restrictions you defined for the MVC.

   If you use `VOLPARM and POOLPARM` definitions and the virtual CDS level is G or above, continue with step 4. Otherwise, go to step 5.

4. If you want to reuse the tape volser for Nearline (non-VTCS) usage and use `VOLPARM/POOLPARM` definitions:

   a. Update the `POOLPARM/VOLPARM` statements for the MVCs you want to remove.

   b. Run `SET VOLPARM APPLY(YES)` on all hosts to apply the changes.

   c. Run the HSC `SCRAtch` command to scratch the volumes that are no longer MVCs.

5. If you want to reuse the tape volser for Nearline (non-VTCS) usage you do not use `VOLPARM and POOLPARM`, you need to do one of the following:

   a. Issue an HSC `EJECT` command to remove the MVCs from the ACS.

   b. Change the external bar code label on the cartridge.

      You must change the external bar code label, because the original MVC volsers are retained in the CDS, and these volsers are only available for use as MVCs.

   c. ENTER the cartridges back into the ACS.

   OR

   a. Create a new set of CDS data sets.

    b. Run the HSC `MERGECDS` utility specifying `DELVirt` to remove unwanted MVC ranges.

Note: All HSCs must be stopped when using this option since new CDS data sets are created.

Temporarily Removing MVCs

To temporarily remove MVCs from the pool:

1. Enter MVCDRain Eject for the MVC.

For example, to run the MVCDRain to drain the MVCs in Storage Class STORCL1, virtually eject the MVCs, and return after the request is submitted, enter the following:

```
MVCDRAIN STORCLAS(STORCL1) EJECT NOWAIT
```

This does the following:

- Recalls all VTVs on the MVC and remigrates them to new MVCs.

- Makes the MVC non-selectable for VTCS migrates.

2. To return the MVC to the MVC pool, enter an MVCDRain for the MVC.

Entering MVCDRain without the EJect parameter for the MVC makes it available again.

For example, to run the MVCDRain to drain the MVCs in Storage Class STORCL1 and return after the request is submitted, enter the following:

```
MVCDRAIN STORCLAS(STORCL1) NOWAIT
```

Note: As an alternative, you can use MVCMAINT to mark an MVC as read-only. This prevents VTCS from selecting the MVC for migrates but does not remove the VTVs from the MVC. You can also use MVCMAINT to turn off read-only.

If using VOLPARM/POOLPARM definitions, the NOMIGRAT option can be specified on the POOLPARM statement to prevent MVCs from being used for new migrations.

Draining MVCs

Use MVCDRain to "drain" an MVC (recall all VTVs on the MVC). You typically drain an MVC for the following reasons:

- An MVC report or Display shows data check errors for the MVC. VSM does not migrate to the MVC and you should remove it from the MVC pool.

- An MVC report or Display shows errors other than data check errors for the MVC.

- A Storage Class or Named MVC Pool is no longer in use and you want to remove or reuse the associated MVCs.

To select the MVCs to drain, you can specify one of the following parameters:

- MVCid to drain one or more MVCs by volser.

- MVCPOOL to drain the MVCs in a Named MVC Pool. See *ELS Command, Control Statement, and Utility Reference* for more information on Named MVC Pools.

- STORCLAS to drain the MVCs in a Storage Class. See *ELS Command, Control Statement, and Utility Reference* for more information on Storage Classes.

You can use the MVCDRain to override the CONFIG RECLAIM CONMVC setting. You can run the MVCDRain from each host, which starts drain tasks on that host equal to the CONMVC value. These drain tasks can run concurrently with drain tasks initiated by other hosts.

Also note the following:

- For VMVCs, `MVCDRAIN` with the `EJECT` parameter physically deletes the VTVs.

> **Caution:** If you use the `DRCHKPT` utility and/or the `CONFIG GLOBAL PROTECT` parameter to protect CDS backup content for VMVCs, specifying `MVCDR EJECT` invalidates the CDS backup's VMVC content.

- For both VMVCs and MVCs, `MVCDRAIN` without the `EJECT` parameter does not delete the VTVs, but updates the CDS record to show no VTVs on the VMVC/MVC.

For more information, see *ELS Command, Control Statement, and Utility Reference*.

Changing MVC Attributes with MVCMAINT

MVCMAINT is a similarly handy tool in VSM Land, and its parameters describe its abilities:

- First, MVC volser (range, list, individual volser) or MANIFEST are your two MVC selection criteria. MVC volser makes sense, but why MANIFEST? You create a manifest file (list of MVCs and the VTVs they contain) when you run EXPORT, which is something you might want to do when you are moving MVCs from one system to another. When you import the MVCs into the new system, it is probably a good idea if they started their new lives in read-only mode, so they do not get overwritten until you can define them properly.

- READONLY (ON or OFF). See the previous bullet. Also, remember the discussion about adding MVCs to the pool? You might want to enter them into the ACS in scratch status, but some shops bring everything in as non-scratch and then sort things out. When you need to make writeable the new MVCs, MVCMAINT READONLY(OFF) is the tool.

- LOST (ON or OFF). How does an MVC get lost? For example, can MVCs get lost? Believe it or not, they can. For example, if a VTCS-initiated mount of an MVC fails to complete (as opposed to completes with an error), VTCS marks the MVC as "lost" in the CDS, and de-preferences it.

  Multiplexed VTVs that reside on a "lost" MVC are recalled from an alternate MVC. VTCS does not attempt to use "lost" MVCs for migration unless there are no other valid MVCs. When an MVC in "lost" status is successfully mounted, the "lost" status is removed from the MVC record.

  What if you know the MVC is not *really* lost? Answer: You can use MVCMAINT to turn off the LOST status.

  There is an interesting use of MVCMAINT that deserves a mention. What if you have an LSM that is temporarily in manual mode? You might want to

(temporarily) de-preference MVC selection in that LSM, which you can do with LOST(ON). Then, when the LSM is back in automatic mode, reverse the process with LOST(OFF).

- ERROR (ON or OFF). An MVC can (erroneously) go into Error status for several reasons, for example:

  - VTCS does not recognize the volume mounted on the RTD as an MVC. This can be caused by some MVS job overwriting the MVC. Determine what happened to the MVC. If it no longer contains valid VTV data, reinitialize the volume and return it to the MVC pool.

  - The MVC is not writeable, which can be caused by the thumb wheel being set to read-only, or by the security package not allowing VTCS to write to the volume. Reset the thumb wheel, or change the rules in the security package to allow the MVC to be written to.

  - A bad block ID has been detected, and you have to (VTCS) audit the MVC to try to correct the condition.

    After you correct the error condition as described, use MVCMAINT to reset the MVC status to ERROR(OFF).

- EJECT (ON or OFF) specifies the "logical eject" status of the MVC. How does this status get set, and why might you want to change it? If you explicitly drain an MVC using MVCDRAIN, it is probably because you think the media is bad, and so you de-preference it by setting on the "logical eject" status. You then really eject the MVC, run some tests, find out it is just fine, and reenter it. At that point, use MVCMAINT) to set EJECT(OFF).

- Next, you have a group of MVC attributes specific to T9840/T9940 media, all with ON/OFF switches:

  - WARRANTY. VTCS also detects media warranty expiration and sets the WARRANTY status to ON. Alternatively, you can use SMF, LOGREC data, or your MVC and VTV reports to detect MVCs approaching end-of-life and use the MVCMAINT to manually set WARRANTY ON. Knowing that the warranty has expired lets you plan for media replacement before media end-of-life occurs (see the next bullet). What if you know an MVC was erroneously marked as warranty expired? Answer: Just use MVCMAINT to reset the warranty expired status.

  - RETIRED. VTCS also automatically detects media end-of-life and sets the RETIRED status to ON. As above, you can use SMF, LOGREC data, or your MVC and VTV reports to detect MVCs approaching end-of-life and use the MVCMAINT to manually set RETIRED ON or reset the status to RETIRED OFF for MVCs erroneously marked as retired

  - VTCS automatically detects an invalid Media Information Region (MIR) and sets the INVLDMIR status to ON. You can recover the MIR by using either the utility available through the operator panel for the transport or by using the utility available through MPST. After you recreate the MIR, you can use the MVCMAINT to set INVLDMIR OFF for the MVC.

    Note: Running MVCMAINT also produces an MVC report of the volumes affected by the MVCMAINT job.

Verifying MVCs or VMVCs

The `MEDVERify` utility does Media Verification (MV) by verifying that VTV data can be read on MVCs or VMVCs (ELS 7.1 and VLE 1.2 and above only). For VLE, `MEDVERify` ensures that deduplicated VMVCs can be "rehydrated" (reconstituted).

The utility reports on MVCs that pass or fail verification and also produces XML output. For more information on the `MEDVERify` utility, see *ELS Command, Control Statement, and Utility Reference*.

The following sections show examples of using the `MEDVERify` utility for MV.

Performing MV for a Single VMVC

```
MEDVERFY   MVC(VMC000)
```

In this example:

■   `MEDVERify` selects a single VMVC.

■   `MAXMVC` defaults to 99.

■   `CONMVC` defaults to 1 so only a single MVC is processed at a time.

■   No time out is specified.

Performing MV by MVC Pool

```
MEDVER   MVCPOOL(MP1)
```

In this example:

■   `MEDVERify` selects MVCs in MVC Pool `MP1` for processing.

■   `FREQency` is not specified and `MAXMVC` defaults to 99, so `MEDVERify` selects the best 99 MVC candidates based on time of last verification.

■   `CONMVC` defaults to 1 so only a single MVC is processed at a time.

■   No time out is specified.

Performing MV by MVC Volser

```
MEDVER   MVC(MVC000-MVC049) CONMVC(2) TIMEOUT(720)
```

In this example:

■   `MEDVERify` selects a range of 50 MVC volsers for processing.

■   `FREQency` is not specified and `MAXMVC` defaults to 99, so `MEDVERify` processes all 50 specified MVCs.

■   `CONMVC` is 2, so `MEDVERify` processes two MVCs concurrently.

■   `MEDVERify` runs for 12 hours before timing out.

Performing MV by Storage Class

```
MEDVER   STORCLAS(SC1) MAXMVC(50) FREQ(365)
```

In this example:

■   `MEDVERify` selects MVCs in Storage Class `SC1` for processing.

■   `MAXMVC` is 50 and `FREQency` specifies 365 days, so `MEDVERify` selects the best 50 MVC candidates that have not been verified for 1 year or more.

- CONMVC defaults to 1 so only a single MVC is processed at a time.

- No time out is specified.

Working with VTSSs

The main thing you can do, as needed, with VTSSs is use the VTCS Vary VTSS command or utility to vary a VTSS online, offline, or to quiesced state. Always know what you are doing, and why, when you vary a VTSS offline or to quiesced state. The why is probably that the VTSS needs maintenance or you are going to remove it from the configuration, which is talked about in "Finding and Fixing VTCS Problems."

First, however, the following chart shows you what happens when you vary a VTSS into each of its supported modes (and why you should use QUIESCED over OFFline, if at all possible).

Table 5–1   VTSS States

Vary VTSS Parameter Specified	VTSS First State	VTSS Next State
ONline	**Online Pending** - In online pending state, the online process has started but has not completed on all hosts.	**Online** - In online state, the VTSS is online, available, and accepts both front-end and back-end work. If the VTSS was offline, when it goes online, VTCS issues a warning message recommending a VTSS audit.
QUIESCED	**Quiescing** - In quiescing state, VTCS does not direct any DD allocation to the VTSS, which still accepts pending mounts to allow those long running jobs with unit=aff chains to complete. When all VTDs are no longer in use (their UCBs are not allocated on MVS), the VTSS goes to quiesced state. In quiescing state, the VTSS continues to accept and process back-end work; for example, migrates, recalls, and audits.	**Quiesced** - In quiesced state, the VTSS continues to accept and process back-end work; for example, migrates, recalls, and audits. That is, you can use the recall and migrate commands and utilities to do these operations using the quiesced VTSS.
OFFline	**Offline Pending** - In offline pending state, the offline process has started but has not completed on all hosts. VTCS immediately shuts down the VTSS and interrupts and purges all active tasks and purges all queued tasks. The VTSS server task terminates and no longer accepts new front-end and back-end work. VTCS creates new VTVs and mounts/dismounts existing VTVs only on alternate VTSSs, if they are available.	**Offline** - In offline state, The VTSS is offline to all hosts and does not accept either front-end or back-end work. If a copy of a VTV is resident on an offline VTSS and also on an MVC and a job requires the VTV, VTCS automatically recalls the VTV to an alternate VTSS, if available.

> **Note:** In a client/server environment (MVS/CSC and LibraryStation or SMC/HTTP server on client hosts), VTCS cannot determine if long running jobs are active on client hosts. After a VTSS goes to offline state, therefore, you should still either (a) explicitly vary its VTDs off line to MVS or (b) ensure that virtual tape activity on the client host has ceased.
>
> In Clustered VTSS or Cross-TapePlex Replication (CTR) configurations, the Clinks to the VTSS should be varied offline to stop replication and electronic export processing.

Quiescing a VTSS for Service

Before you service a VTSS, quiesce the VTSS as follows:

1. On every host vary the VTSS VTDs offline.

   On every host wait for all devices to go offline. *Note that* VTDs do not go through the offline process until they are no longer allocated. If a long running job is using a VTD then you must either wait for the job to complete or cancel the job.

2. Vary the VTSS QUIESCED from any VTCS system where the named VTSS is defined.

   Wait for message SLS6742I on each VTCS system indicating the VTSS is quiesced.

3. Optionally, you can migrate data from the VTSS.

4. Vary the VTSS OFFLINE from any VTCS system where the named VTSS is defined.

   Wait for message SLS6742I on each VTCS system indicating the VTSS is offline.You can now service the VTSS.

Removing a VTSS

Here is the scenario for removing a VTSS: You have two separate VSM systems, the workload for one grows while the workload diminishes for the other. Solution: take a VTSS out of System A and give it to System B. *Installing ELS* discusses how to add a VTSS, so this section is confined to what you do to remove a VTSS.

To remove a VTSS:

1. *Before* you remove the VTSS, do the following:

   ■ You do not need to empty a VTSS before deletion. What you *do* need to ensure is that all VTVs are fully migrated. Also consider changing other parameters, for example, TAPEREQ statements so that new work is not routed to the removed VTSS.

   ■ If removing all of one device type/ACS combination from a VTSS, also ensure that all VTVs are fully migrated first. As above, consider changing other parameters to reflect the changed migration capabilities of the VTSS (for example, Management Classes, which point to Storage Classes that specify ACS and media).

2. Vary the VTSS to Quiesced state.

   After it goes offline, continue with step 3.

3. Remove the VTSS, then rerun CONFIG to logically remove it.

The following shows sample JCL to run CONFIG to update the configuration to deny host access to VTSS2 that you physically removed from your configuration. In this example, you respecify the VTSS statement for VTSS2 with no parameters to deny host access to this VTSS.

```
//UPDATECFGEXEC PGM=SLUADMIN,PARM='MIXED'
//STEPLIBDD DSN=hlq.SEALINK,DISP=SHR
//SLSCNTLDD DSN=FEDB.VSMLMULT.DBASEPRM,DISP=SHR
//SLSCNTL2DD DSN=FEDB.VSMLMULT.DBASESEC,DISP=SHR
//SLSSTBYDD DSN=FEDB.VSMLMULT.DBASETBY,DISP=SHR
//SLSPRINTDD   SYSOUT=*
//SLSINDD   *
 CONFIG
 GLOBALMAXVTV=32000MVCFREE=40
 RECLAIMTHRESHLD=70MAXMVC=40  START=35
 VTSSNAME=VTSS1  LOW=70 HIGH=80  MAXMIG=3  RETAIN=5
RTDNAME=VTS18800 DEVNO=8800 CHANIF=0A
RTDNAME=VTS18801 DEVNO=8801 CHANIF=0I
RTDNAME=VTS18802 DEVNO=8802 CHANIF=1A
RTDNAME=VTS18803 DEVNO=8803 CHANIF=1I
RTDNAME=VTS18811 DEVNO=8811 CHANIF=0E
RTDNAME=VTS18813 DEVNO=8813 CHANIF=1E
VTDLOW=8900 HIGH=893F
 VTSSNAME=VTSS2
```

Working with VTVs

This section consists of the most likely tasks you might have to do on an as-needed basis: deleting scratch VTVs and changing VTV attributes.

> **Note:**   If you delete VTVs from the configuration as a result of either SET VOLPARM or CONFIG MVCVOL processing:
>
> ■   You cannot re-enter the volsers into the configuration as MVCs.
>
> ■   Do not use the volsers for native HSC tapes.
>
> Message SLS6944I indicates the number of VTVs that have been deleted.

Deleting Scratch VTVs

There are two ways you can delete scratch VTVs:

■   By policy, by specifying DELSCR(YES) on a VTV's Management Class and using HSC or LCM scratch synchronization to do the actual scratch.

■   For specific tasks, by using the DELETSCR utility. DELETSCR deletes scratch VTVs from VTSSs and unlinks any migrated VTVs from MVCs. Deleted VTVs are marked as non-initialized, although versioning information is retained.

Installing ELS addresses deleting scratch VTVs, so the information that follows discusses the "as needed" version.

Take note of the following warning:

> **Caution:**   When you use DELETSCR to delete scratch VTVs, any data on those VTVs is *gone* and cannot be recovered!

Deleting VTVs is *not* something you do because you have nothing else to do. If you have to manually delete scratch VTVs, it is because you are in trouble as shown in the scenario on page 4-1.

To prevent inadvertent VTV deletion through an operator command, DELETSCR is a SLUADMIN utility only, and has the following capabilities:

- You can specify VTVs by volser (individual volser, list, or range), Management Class, or HSC Scratch Pool. Using your MVC and VTV reports, you should already have a good idea of the best way to identify the candidates and apply the corresponding DELETSCR option. You can only specify one option (VTVid, MGMTclas, or SCRpool) and if you do not specify anything, DELETSCR deletes *all* eligible VTVs. This may be what you want, but think before you go for that method.

- The mandatory NOTREF parameter specifies the days since a VTV was referenced (1-999). NOTREF is effectively a grace period; any VTV referenced within the specified grace period **is not** deleted.

- There is a handy (optional) MAXVTV parameter that specifies the maximum number of VTVs that DELETSCR deletes. Note that this is a *maximum*, not a target. If you are running DELETSCR proactively during a non-peak period, you might not care about MAXVTV. If you are in trouble, you most certainly will.

  Note that the range for MAXVTV is 0-999. What happens if you specify 0? In this case, DELETSCR does not delete any VTVs, but the *summary* report shows *how many* VTVs have been deleted at the point at which you ran DELETSCR (that is, the report is just a snapshot).

- Finally, you can see the results of your work through the DELETSCR reports, standard or detailed (by specifying the DETAIL parameter).

DELTSCR JCL Example

The following shows sample JCL to run the DELETSCR to delete scratch VTVs in Management Class MC1 not referenced within 60 days up to a maximum of 800 VTVs and produce a detailed report.

```
//DELETSCR EXEC PGM=SLUADMIN,PARM='MIXED'
//STEPLIBDD DSN=hlq.SEALINK,DISP=SHR
//SLSPRINTDD SYSOUT=*
//SLSINDD *
  DELETSCR MGMTCLAS(MC1) NOTREF(60) MAXVTV(800) DET
```

Changing VTV Attributes with VTVMAINT

VTVMAINT is another handy tool, this time for VTV maintenance, which does the following:

- Selects VTVs by volser (range, list, or individual volser as you choose).

- Unlinks VTVs from MVCs, because it is likely you are going to want to do this if you change a VTV's Management Class as described in "Changing VTV Management Class and Unlinking VTVs from MVCs."

- Changes the VTV's Management Class, which you do when you want the VTV to be managed differently. There are other ways to do this, but the best tool is clearly VTVMAINT, as described in "Changing VTV Management Class and Unlinking VTVs from MVCs."

- Logically dismounts specified VTVs in an offline VTSS. This is best explained in "Logically Dismounting VTVs in an Offline VTSS."

- "Managing VTVs Replicated Through Cross-TapePlex Replication (CTR)."

Note: And do not forget, running VTVMAINT also produces a VTV report of the volumes affected by the VTVMAINT job.

Changing VTV Management Class and Unlinking VTVs from MVCs

You can use VTVMAINT to change a VTV's Management Class. If the new Management Class specifies a different Storage Class, the VTV's current location on MVCs is incorrect. The following procedure tells how to use VTVMAINT to change a VTV's Management Class and Storage Class.

To change a VTV's Management Class and unlink it:

1. Recall the VTV.

   The VTV must be VTSS-resident for the unlink to succeed in step 2.

2. Use VTVMAINT ULINKMVC to unlink the VTV from the MVC(s) where it is located.

3. Use VTVMAINT MGMTclas to assign a new management class.

4. Remigrate the VTV to place it on the correct MVCs, or see "Changing VTV Storage Class with RECONcil" for procedures for moving VTVs to MVCs as needed.

Logically Dismounting VTVs in an Offline VTSS

If a VTV is mounted when a VTSS goes offline and a copy of the VTV exists on an MVC, VTCS does not recall the migrated VTV to an alternate VTSS because the VTV is in mounted status on the offline VTSS. In this situation, you can use the VTVMAINT to logically dismount VTVs in the offline VTSS (turn off the "mounted" bit in the CDS), then recall the VTV to an alternate VTSS. VTCS records each successful VTV dismount in the SMF14STA field of the SMF Subtype 14 record. The VTVRPT (UNAVAIL) option reports the status of unavailable VTVs in an offline VTSS. For more information, see *ELS Command, Control Statement, and Utility Reference*.

Do not dismount an unavailable VTV in an offline VTSS unless you are absolutely sure that the MVC copies, if any, of the VTV, are identical in content to the unavailable VTV! Otherwise, you risk recalling a VTV with back-level data to an alternate VTSS! For example, a VTV mounted for read is probably safe to dismount for recall to an alternate VTSS. A VTV mounted for write, however, is probably not safe to dismount because it has probably been updated and the MVC copies are therefore back-level.

The following procedure provides the general steps you use to logically dismount a VTV and access that VTV from a different VTSS.

To logically dismount a VTV and access that VTV from a different VTSS:

1. Vary the VTSS offline to VTCS with the following command:

```
VT VARY VTSS(name) OFFLINE
```

   If I/O was active and the VTSS failed, MVS should box the VTDs and dismount any mounted VTVs *from the MVS perspective*. However, if communication with the VTSS failed before the VTSS actually dismounted any mounted VTVs, they may still be online to VTCS. Therefore, you first need to vary the VTSS offline to VTCS.

If MVS boxed the VTDs and dismounted any mounted VTVs, go to step 3.
Otherwise, continue with step 2.

2. Dismount the VTV (MVS perspective).

You cannot remount the VTV on a VTD in another VTSS if MVS still considers it
mounted in the offline VTSS. Do either of the following:

- Use the MVS UNLOAD command to dismount the VTV.

- Use the VARY OFFLINE to vary offline the VTD where the VTV is mounted,
  which also dismounts the VTV.

3. Run VTVMAINT, specifying the offline VTSS and VTV(s) you want to logically
dismount.

For example, to logically dismount VTVs VV6823, VV6825, and VV6688 in offline
VTSS01, code the following SLSIN DD statement in your JCL:

```
VTVMAINT DISMOUNT VTV(VV6823,VV6825,VV6688) VTSS(VTSS01)
```

If migrated copies of the dismounted VTVs exist that an online VTSS can access,
you can now use this VTSS to access the VTVs.

Caution: If the VTV copy mounted in the offline VTSS was modified
and not migrated, the MVC copy that you recall to an alternate VTSS
is not current! Therefore, Oracle *strongly recommends* that you do not
recall these non-current MVC copies!

Tip: When the offline VTSS is ready to be brought back online,
Oracle *strongly recommends* that you audit the VTSS before running
production jobs that use the VTSS. Also ensure that you clear any
boxed VTD conditions before issuing the VTSS VARY ONLINE
command.

Managing VTVs Replicated Through Cross-TapePlex Replication (CTR)

You can use VTVMAINT to change the status of VTVs replicated through CTR as follows:

- Use VTVMAINT OWNRPLEX to change the VTV's owning TapePlex.

- Use VTVMAINT DELEXpot to remove the name of a TapePlex that references a VTV.

- Use VTVMAINT ADDEXpot to add the name of a TapePlex that references a VTV.

For more information, see *ELS Disaster Recovery and Offsite Data Management Guide*.

Changing VTV Storage Class with RECONcil

As described in "Changing VTV Management Class and Unlinking VTVs from MVCs,"
you can use VTVMAINT to change a VTV's Management Class, which could, of
course, change its Storage Class. Or what if you want to *explicitly* move the VTV from
one Storage Class to another? Answer: use RECONcil.

Before you submit your first RECONcil job (SLUADMIN utility only), figure out *why*
you want to change a VTV's Storage Class. There are basically three reasons:

- As above, you are explicitly changing the VTV's Management Class/Storage
  Class.

- The VTVs are on the wrong media, in the wrong ACS, or both.

■ An ACS is unavailable for a considerable period of time, then is brought back online. In this case, you first change the MIGpol parameter on the MGMTclas statement for the affected VTVs to point to a different ACS (and media, if desired). When the original ACS comes back online, you then change the MIGpol parameter on the MGMTclas statement to point to the original ACS, and run RECONcil specifying the updated MGMTclas (or STORclas) statement(s) to move the VTVs to the original ACS.

Notice that this discussion talks about using RECONcil to reconcile a VTV's *incorrect* Storage Class (incorrect MVC media, ACS location, or both). What if you wanted to move VTVs whose data is now less frequently accessed from access-centric media (such as T9840 cartridges) to storage-centric media (such as T9940 cartridges) and an Extended Store ACS or offsite? In that case, you typically set up an Archive Policy using the ARCHAge/ARCHPol parameters of the MGMTCLAS statement, and the VTV movement then occurs automatically according to the ARCHPol specification when the ARCHAge value is exceeded *and when the VTV is recalled and remigrated*.

An automatic Archive Policy, therefore, is like an automatic migration. Both happen over time, and time is what you *do not* have if one or more VTVs are truly in the wrong place. In this case, use RECONcil.

Running a RECONcil Job

To change VTV ACS/media with RECONcil:

1. To select the VTVs to validate (that is, Do they need reconciliation or not?), you can specify one of the following RECONcil parameters:

   STORclas - Specifies one or more Storage Classes. Here, RECONcil does the following:

   ■ Looks up the ACS and media definition for the specified Storage Class(es).

   ■ Scans the MVCs *currently in* the Storage Classes. Does the MVC ACS and media match the Storage Class definition? If not, list the MVCs/VTVs in error.

   MVC - Specifies a list or range of MVCs. RECONcil does the following:

   ■ Determines the actual ACS and media for the specified MVCs.

   ■ Does the actual MVC ACS/media match the Storage Class definition for the MVC? If not, list the MVCs/VTVs in error.

   MGMTclas - Specifies one or more Management Classes. RECONcil does the following:

   ■ Looks up the ACS and media definition as specified on the MGMTclas MIGpol parameter.

   ■ Scans the VTVs *currently in* the specified Management Classes. Is the VTV on an MVC with ACS/media that matches the MGMTclas MIGpol specification? If not, list the VTVs on the MVCs in error.

   VTV - list or range of VTVs. RECONcil does the following:

   ■ Determines the Management Class(es) for the specified VTVs.

   ■ Looks up the ACS and media definition as specified on the MGMTclas MIGpol parameter.

   ■ Scans the VTVs *currently in* the specified Management Classes. Is the VTV on an MVC with ACS/media that matches the MGMTclas MIGpol specification? If not, list the VTVs on the MVCs in error.

> **Note:** And as you can imagine, if you do not specify *any* of the selection parameters, VTCS validates *all* VTVs. More information about this appears in step 2.

2. Accept the default when you first run RECONcil, which is to generate a report only, as you can imagine, does no data movement, but merely reports the VTVs that are candidates for reconciliation.

> **Caution:** Because reconciling VTVs can be resource intensive, Oracle *strongly recommends* that you run RECONcil *without* MOVEVTV first, then adjust the job as needed before specifying MOVEVTV!

3. If needed, adjust the RECONcil job.

    For example, if you ran the report in step 2, and it looks like you will be reconciling for a long time, consider doing the following:

    ■ Run RECONcil during non-peak processing periods, just as you would a demand MVC space reclamation.

    ■ Use the RECONcil utility parameters to override the CONFIG RECLAIM THRESHLD, MAXMVC, and CONMVC settings to optimize reconcile performance.

    ■ Specify the maximum time for the reconcile in minutes on the ELAPSE parameter.

> **Note:** There are *multiple* limiting factors that influence reconciliations (for example, MAXMVC and ELAPSE). VTCS enforces the *strictest* limiting factor. For example, if you run RECONcil and specify ELAPSE equal to 5 hours and MAXMVC equal to 10 *and* VTCS reconciles 10 MVCs in one hour, then VTCS terminates the reconciliations before the ELAPSE value expires.

    ■ There is also a RECONcil POLICYdd option that is also available on the ARCHive utility and can be a useful diagnostic. POLICYdd, which enforces producing only a report, points to a file that contains an alternate set of MGMTclas statements.

> **Tip:** This is basically a *valuable* "what if" tool that says if you changed some VTV Management Classes discussed in "Changing VTV Management Class and Unlinking VTVs from MVCs" (including their Storage Class specifications) and then ran RECONcil, what does *that* look like? Now you can find out *before* you actually change a VTV's Management Class.

> **Note:** VTCS and HSC must be active to process a RECONcil request *except* when you specify the POLICYdd parameter.

4. You have done all the "what ifs," fine tuning, and off-peak schedule required.

Now it is time to make it happen. The following shows example JCL to run RECONcil:

- Reconcile VTVs in Management Classes LOCALPROD1 and LOCALPROD2.

- Set MAXMVC to 60, CONMVC to 8, and ELAPSE to 60 for the RECONcil job.

```
//RECONCIL    EXEC PGM=SLUADMIN
//STEPLIBDD DSN=hlq.SEALINK,DISP=SHR
//SLSPRINTDD SYSOUT=*
//SLSINDD *
  RECON MGMT (LOCALPROD1,LOCALPROD2) MAXMVC(60) CONMVC(8)
ELAPSE(360) MOVEVTV
```

And, of course, you get an after-action RECONcil report that tells you how smoothly (or not) things went so you can readjust and rerun the process if necessary.

Using FOR_LOSTMVC to Recover VTVs

You can use the `LOGUTIL FOR_LOSTMVC` statement to recover VTVs that resided on lost or damaged MVCs. How does the `LOGUTIL FOR_LOSTMVC` statement work and how do you use it most effectively?

The `FOR_LOSTMVC` utility scans the CDS and the log file structure (if necessary) to identify all VTVs on the lost or damaged MVCs whose volsers you specify and to determine the recovery method from an alternate VTV copy as described in Table 5–2. `LOGUTIL FOR_LOSTMVC` generates a report showing all VTVs that existed on the lost or damaged MVCs and how they are recovered, plus summary information for each lost or damaged MVC.

Table 5–2   Alternate VTV Copy and Recovery Process

Alternate VTV Copy Category	Recovery Process
Category 1: Currently VTSS resident.	Recovery is from the resident copy. If you requested recovery commands, `VTVMAINT ULINKMVC` commands are generated to unlink the VTV from the lost or damaged MVCs.
Category 2: Currently linked to one or more alternate MVC copies.	Recovery is from the best alternate MVC based on four factors: ■ Does an MVC record exist in the CDS for the MVC? ■ Is the MVC in lost status? ■ Is the MVC in broken status? ■ Has a data check occurred on the MVC? If you requested recovery commands, `VTVMAINT ULINKMVC` and `RECALL` commands are generated to unlink the VTV from the lost or damaged MVCs and then recall the MVC.

Table 5–2   (Cont.) Alternate VTV Copy and Recovery Process

Alternate VTV Copy Category	Recovery Process
Category 3: Has been Cross TapePlex Replicated.	The first remote TapePlex encountered that contains a copy of the VTV is used to recover the VTV.
	If you requested recovery commands, `EEXPORT ULINKMVC` commands are generated. These commands must be run from the remote TapePlex where the VTV currently resides. Comments in the COMMANDS data set show the TapePlex where these commands must be run. The command unlinks the VTV from the lost or damaged MVCs and then Cross TapePlex Replicates the VTV back to the local TapePlex.
Category 4: Was previously linked to one or more MVC copies that may still contain the VTV data.	One of the previously linked MVCs is selected as the recovery MVC. These MVC copies were found in the log files and may still contain a copy of the VTV. You must audit the selected recovery MVC. The best previously linked MVC copy to do the recovery from is selected based on the same factors as alternate MVCs.
	If you requested recovery commands, `AUDIT` commands are generated to audit the MVC and to attempt to link it to the VTV.
	`MVCMAINT READONLY(ON)` commands are generated for `AUDIT` MVCs.
Category 5: Is unrecoverable.	Unrecoverable, copies only existed on the lost or damaged MVCs.

Note: If you requested recovery commands, `MVCMAINT` commands are also generated for Categories 1, 2, and 3, and 4. These statements mark the lost or damaged MVCs as readonly and broken so that they are no longer selected for recalls or migrates.

FOR_LOSTMVC Recovery Procedure

Note: In this procedure, the JCL examples do not show DD statements for the CDS copies, which is valid if HSC is active and you want to use the active CDS on the system where you are running LOGUTIL. Otherwise, you must specify the DD statements for the CDS copies.

To recover VTVs using FOR_LOSTMVC:

1. First, run the `LOGUTIL FOR_LOSTMVC` command with only the volsers of the lost or damaged MVCs.

    For example, the following example shows:

    ■   The logging data set is `LOGIN`.

> **Note:** You can run `LOGUTIL FOR_LOSTMVC` with a dummy `LOGDD` specified to allow recovery on systems where CDS Logging has not been activated. The recovery is limited to the data in the CDS but may still be useful if all VTVs are either resident, on an alternate MVC copy or exported through Cross Tape Replication.

- The volser of the damaged MVC is `DMV509`.

- The recovery commands are logged in data set `RECVCMD`.

```
//JOBLOGR job (account),programmer,REGION=1024k
//S1 EXEC PGM=SLUADMIN,PARM=MIXED
//STEPLIB     DD DSN=hlq.SEALINK,DISP=SHR
//LOGIN       DD DSN=FEDB.VSMLMULT.LOGFILE.OFFLOAD(-2),DISP=OLD
//            DD DSN=FEDB.VSMLMULT.LOGFILE.OFFLOAD(-1),DISP=OLD
//            DD DSN=FEDB.VSMLMULT.LOGFILE.OFFLOAD(0),DISP=OLD
//RECVCMD     DD DSN=FEDB.VSMLMULT.RECVCMD,DISP=(CATLG,DELETE),
//               UNIT=SYSDA,SPACE=(CYL,(1,1),RLSE),
//               DCB=(RECFM=FB,LRECL=80,BLKSIZE=27920)
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
LOGUTIL LOGDD(LOGIN)
FOR_LOSTMVC MVC(DMV509) COMMANDS(RECVCMD)
```

2. Review the `LOGUTIL FOR_LOSTMVC` report from step 1.

   Select the VTVs you want to recover and rerun `LOGUTIL FOR_LOSTMVC`, specifying the VTVs you want to recover from the lost or damaged MVC. For example:

```
//JOBLOGR job (account),programmer,REGION=1024k
//S1 EXEC PGM=SLUADMIN,PARM=MIXED
//STEPLIB     DD DSN=hlq.SEALINK,DISP=SHR
//LOGIN       DD DSN=FEDB.VSMLMULT.LOGFILE.OFFLOAD(-2),DISP=OLD
//            DD DSN=FEDB.VSMLMULT.LOGFILE.OFFLOAD(-1),DISP=OLD
//            DD DSN=FEDB.VSMLMULT.LOGFILE.OFFLOAD(0),DISP=OLD
//RECVCMD     DD DSN=FEDB.VSMLMULT.RECVCMD,DISP=(CATLG,DELETE),
//               UNIT=SYSDA,SPACE=(CYL,(1,1),RLSE),
//               DCB=(RECFM=FB,LRECL=80,BLKSIZE=27920)
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
LOGUTIL LOGDD(LOGIN)
FOR_LOSTMVC MVC(DMV509) VTV(DX009) COMMANDS(RECVCMD)
```

> **Note:** If you specify a VTV that was not on the lost or damaged MVC, this VTV is ignored.

   If you want to recover all specified VTVs on the damaged MVC, continue with step 3.

3. To recover the specified VTV(s), run the commands in the recovery data set specified in step 2.

> **Note:**
>
> - The commands in the recovery data set should be run (using
>   standard `SLUADMIN` JCL) as soon as possible after running `FOR_`
>   `LOSTMVC` to ensure their accuracy.
>
> - Oracle recommends that you run the recovery commands in the
>   `COMMANDS` file in the following order:
>
>   1. All `EEXPORT ULINKMVC` commands.
>
>   2. All `MVCMAINT READONLY(ON)` commands.
>
>   3. All `AUDIT` commands.
>
>   4. If there were `EEXPORT ULINKMVC` or `AUDIT` commands, then re-run `FOR_`
>   `LOSTMVC`. With the new run, there should be no `EEXPORT` or `AUDIT`
>   commands in the newly generated `COMMANDS` file. If there are, then return
>   to step 1.
>
>   5. All `MVCMAINT READONLY(ON) ERROR(ON)` commands.
>
>   6. All `ULINKMVC` commands.
>
>   7. All `RECALL` commands.
>
>   8. The `RECONcil` utility
>
>   `MVCMAINT` commands are generated for all specified lost or
>   damaged MVCs that exist in the CDS and that have at least one
>   qualifying VTV on them. The `MVCMAINT` commands set the
>   readonly and error/broken bits on for the lost or damaged MVCs
>   to prevent them from being allocated for recalls or migrates. A
>   maximum of approximately 3000 MVCs is included on each
>   `MVCMAINT` command.

4. Run the RECONcil utility to ensure the correct number of MVC copies are created
   for each VTV.

   For example:

```
//JOBLOGR job (account),programmer,REGION=1024k
//S1 EXEC PGM=SLUADMIN,PARM=MIXED
//STEPLIB     DD DSN=hlq.SEALINK,DISP=SHR
//LOGIN       DD DSN=FEDB.VSMLMULT.LOGFILE.OFFLOAD(-2),DISP=OLD
//            DD DSN=FEDB.VSMLMULT.LOGFILE.OFFLOAD(-1),DISP=OLD
//            DD DSN=FEDB.VSMLMULT.LOGFILE.OFFLOAD(0),DISP=OLD
//RECVCMD     DD DSN=FEDB.VSMLMULT.RECVCMD,DISP=(CATLG,DELETE),
//               UNIT=SYSDA,SPACE=(CYL,(1,1),RLSE),
//               DCB=(RECFM=FB,LRECL=80,BLKSIZE=27920)
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
RECONCIL VTV(DX009)
```

6

Leading Edge Solutions with Management and Storage Classes

You can use this section to create VTCS Management and Storage Classes and perform common tasks associated with them.

What are Management and Storage Classes?

VTCS Management and Storage Classes, which are the building blocks of a lot of leading edge implementations, do the following:

- VTCS Management Classes specify *how* VTCS manages VTVs. The HSC MGMTclas control statement defines a Management Class and its attributes. For example, the DELSCR parameter of the MGMTclas statement specifies whether VTCS deletes scratched VTVs from the VTSS.

- Management Classes can also point to VTCS Storage Classes. VTCS Storage Classes, which specify *where* migrated VTVs reside. The HSC STORclas control statement defines a Storage Class and its attributes. For example:

```
MGMT NAME(PAYROLL) MIGPOL(LOCAC,REMAC)
STORCLAS NAME(LOCAC) ACS(00) MEDIA(STK1R)
STORCLAS NAME(REMAC) ACS(01) MEDIA(STK2P,ZCART)
```

This combination of Management and Storage Classes says "For Management Class PAYROLL, migrate duplexed to separate MVCs in the local and remote ACSs. In the local ACS, put it on 9840 media so you can get it back in a hurry if you need it. In the remote ACS, prefer 9940 over ZCART media, but *definitely* put it in Deep Storage there."

Start with "Creating and Using VTCS Management and Storage Classes: The Basics," which is a basic procedure you can adjust for any of your business needs. Next, it is on to "Leading Edge Techniques You Can Do with Management and Storage Classes." Think of this section as a gallery where you get to see a variety of options, then pick out the one that works best for you.

Creating and Using VTCS Management and Storage Classes: The Basics

You see this pattern a lot (TAPEREQ > POLICY > MGMTclas > STORclas), so get to know it, because it is the foundation for all the information in "Leading Edge Techniques You Can Do with Management and Storage Classes."

To create and use VSM Management and Storage Classes:

1. Determine the definition data set that contains the STORclas and MGMTclas statements.

   MGMTclas and STORclas statements *must* reside in the same data set for cross-validation.

2. Define Storage Classes through the STORclas control statement.

3. As desired, adjust the migration policies with `MIGRSEL` and `MIGRVTV` statements.

4. Define Management Classes with the MGMTclas control statement.

   Note that the MGMTclas control statement specifies Storage Classes on various parameters.

5. Load the control statements with the HSC MGMTDEF command.

6. Specify a tape policy on the SMC POLICY command.

7. Specify the Policy name to VTCS on either of the following:

- The SMC TAPEREQ statement.

- SMS routines that you write to the StorageTek DFSMS interface.

Maintaining Management and Storage Classes

Note the following:

- Always use an SMC POLICY command to assign the Management Class to mounts.

- You can specify a policy with either the TAPEREQ statement or SMS routines.

- Use POLICY VALIDATE to ensure that your SMC POLICY statements all reference VALID MGMTCLAS names.

- You can use the VTVMAINT utility to change a VTV's Management Class. Also note that while you *cannot* use VTVMAINT to directly change a VTV's Storage Class, you *can* use VTVMAINT to change a VTV's Management Class, which can reference a different Storage Class.

- Use only the minimum Storage Classes required to define the policies you want to implement. Excessive Storage Classes can impact VSM performance due to the MVC mount/dismount overhead incurred. In addition, an MVC can only contain VTVs in a single Storage Class, so excessive Storage Classes can underuse MVC space.

- *If you decide to delete a Management Class definition*, run a VTV Report to make sure that the Management Class is no longer assigned to any VTVs, otherwise unpredictable results occur!

Leading Edge Techniques You Can Do with Management and Storage Classes

The following is not an exhaustive list, just the most common tasks you can do with Management and Storage Classes:

- Using the STORclas MEDIA Parameter for MVC Media Preferencing. There are defaults for MVC media preferencing, but you can adjust them any way you want. For more information, see *Managing HSC and VTCS*.

- "Grouping Multiple Workloads on Shared MVCs." This was our introductory example, and is a good one if your company owns the data center, and you want to optimize use of your available resources by:

  - Duplexing critical data to separate MVCs in the local and remote ACSs. In the local ACS, put it on 9840 media so you can get it back in a hurry if you need it. In the remote ACS, prefer 9940 over ZCART media for deep storage on high-capacity media.

  - Giving two critical jobstreams (payroll and accounting) access to these Management/Storage Classes. Result: all your payroll and accounting data is duplexed local and remote, grouped on the same set of MVCs of the appropriate media discussed in the Storage Class specifications.

  - Production data is also critical, but you want it on a separate set of MVCs from the ones used for payroll and accounting data. No problem, just create another Management Class/Storage Class combination for production data, and you are done.

- "Segregating Individual Workloads on Separate Sets of MVCs." All you service groups, study this one carefully, because you will probably use it a lot. Ever want to give each of your clients his/her own set of resources for billing/security purposes? Segregating workloads is the key.

- "Archiving Data." In this scenario, you can use VTCS to mimic HSM in the StorageTek automated tape/virtual tape environment only. That is, you can use the ARCHAge and ARCHPol parameters of the MGMTclas statement to set an *Archive Policy* for VTVs in a Management Class.

  Information Lifecycle Management (ILM), a StorageTek storage management strategy, has as its central concept the idea that data should be stored on media that matches its importance to the business enterprise and its pattern of reuse. Active, important data goes on fast-access media, and has multiple copies, while inactive, less important data is archived to high-capacity, cheaper media. Automating this process is the most cost-effective way to manage data storage. Archiving implements ILM by letting you archive inactive data. Using VTCS archiving, you can move VTVs to different media (for example, from fast-access 9840 media to high-capacity 9940 media) and a different location (for example, from a local ACS to a remote ACS for ejection/vaulting). For the details, see "Archiving Data."

- "Reconciling VTV Media and Location." Think of archiving as a proactive move. You put the data on the right media at the beginning of the ILM cycle, then move it to other media as the data ages. What happens if the data ends up on the wrong media? Answer: use the RECONcil utility to move it from one Storage Class to another.

- **Controlling VTV Migration**. ELS allows very fine control over VTV migration, including deleting scratch VTVs from the VTSS buffer, specifying the immediate migrate delay interval, and specifying the maximum VTV residency interval. For more information, see *Configuring HSC and VTCS* and *ELS Disaster Recovery and Offsite Data Management Guide*.

Grouping Multiple Workloads on Shared MVCs

You can use Storage and Management Classes to group multiple workloads on a shared set of MVCs. For example, the STORclas statements below define Storage Classes LOC1, LOC2, REM1, and REM2.

```
STORCLAS NAME(LOC1) ACS(00) MEDIA(STK1R)
```

```
STORCLAS NAME(LOC2) ACS(00) MEDIA(STK1R)
STORCLAS NAME(REM1) ACS(01) MEDIA(STK2P,ZCART)
STORCLAS NAME(REM2) ACS(01) MEDIA(STK2P,ZCART)
```

- Management Classes PAY and ACCOUNT both specify the LOC1 and REM1 Storage Classes on the MIGPOL parameter. The VTVs in PAY and ACCOUNT, therefore, are duplexed and grouped on the MVCs defined by Storage Classes LOC1 and REM1.

- Management Class PROD specifies the LOC2 and REM2 Storage Classes on the MIGPOL parameter. The VTVs in PROD, therefore, are duplexed and grouped on the MVCs defined by Storage Classes LOC2 and REM2, which are separate from those for PAY and ACCOUNT.

```
MGMT NAME(PAY) MIGPOL(LOC1,REM1)
MGMT NAME(ACCOUNT) MIGPOL(LOC1,REM1)
MGMT NAME(PROD) MIGPOL(LOC2,REM2)
```

The following defines tape policies that specify virtual media and respectively assign Management Classes PAY, ACCOUNT, and PROD.

```
POLICY NAME (PPAY) MEDIA(VIRTUAL) MGMT(PAY)
POLICY NAME (PACCOUNT) MEDIA(VIRTUAL) MGMT(ACCOUNT)
POLICY NAME (PPROD) MEDIA(VIRTUAL) MGMT(PROD)
```

Finally, this example consists of TAPEREQ statements that assign policies as follows:

- Data sets with qualifiers of PAYROLL.** are assigned policy PPAY.

- Data sets with qualifiers of ACCOUNTS.** are assigned policy PACCOUNT.

- All other data sets are assigned policy PPROD.

```
TAPEREQ DSN(PAYROLL.**) POLICY(PPAY)
TAPEREQ DSN(ACCOUNTS.**) POLICY(PACCOUNT)
TAPEREQ DSN(**) MEDIA(VIRTUAL) POLICY(PPROD)
```

Once an MVC is used for a Storage Class it remains exclusively assigned to that Storage Class while it contains current VTV copies. This grouping of VTVs on MVCs are retained even after the MVCs undergo reclamation processing.

> **Caution:** You cannot use the default Storage Class (the name of the last VTSS that wrote to the MVC for reclamation or migration) to group workloads.

Segregating Individual Workloads on Separate Sets of MVCs

You can use Storage and Management Classes to segregate individual workloads on separate sets of MVCs. For example, the STORclas statements below define Storage Classes LOC, CUSTA, and CUSTB1, and CUSTB2.

```
STORCLAS NAME(LOC) ACS(00) MEDIA(STK1R)
STORCLAS NAME(CUSTA) ACS(00) MEDIA(STK1R)
STORCLAS NAME(CUSTB1) ACS(00) MEDIA(STK1R)
STORCLAS NAME(CUSTB2) ACS(01) MEDIA(STK2P)
```

The example below defines the following Management Classes:

- Management Class CUSTA specifies the CUSTA Storage Class on the MIGPOL parameter. VTCS simplexes VTVs in this Management Classes to *only* the CUSTA

Storage Class (9840 media in the local ACS), because that is what this customer wants.

■ Customer B wants more protection, namely duplexing to the local and remote ACSs, so Management Class CUSTB points to *both* the CUSTB1 and CUSTB2 Storage Classes.

■ Finally, local ACS/9840 media is just fine for your own production data, so that is what Management Class PROD does. What you probably also do is set up an Archive Policy for this Management Class (see "Archiving Data") so you can move it to Deep Storage eventually.

```
MGMT NAME(CUSTA) MIGPOL(CUSTA)
MGMT NAME(CUSTB) MIGPOL(CUSTB1,CUSTB2)
MGMT NAME(PROD) MIGPOL(LOC)
```

This example defines tape policies that specify virtual media and respectively assign Management Classes PAY, ACCOUNT, and PROD.

```
POLICY NAME (PCUSTA) MEDIA(VIRTUAL) MGMT(CUSTA)
POLICY NAME (PCUSTB) MEDIA(VIRTUAL) MGMT(CUSTB)
POLICY NAME (PPROD) MEDIA(VIRTUAL) MGMT(PROD)
```

Finally, the example below shows the corresponding TAPEREQ statements and policy assignments:

■ Data sets with HLQ CUSTA are assigned policy PCUSTA.

■ Data sets with HLQ CUSTB are assigned policy PCUSTB.

■ All other data sets are assigned policy PPROD.

```
TAPEREQ DSN(CUSTA.**) POLICY(PCUSTA)
TAPEREQ DSN(CUSTB.**) POLICY(PCUSTB)
TAPEREQ DSN(**) POLICY(PPROD)
```

> **Caution:** You cannot use the default Storage Class (the name of the last VTSS that wrote to the MVC for reclamation or migration) to segregate workloads.

Archiving Data

You can use the ARCHAge and ARCHPol parameters of the MGMTclas statement to set an *Archive Policy* for VTVs in a Management Class. When the VTV's age exceeds the ARCHAge value, the VTV is eligible for archive for the Storage Class(es) specified on the ARCHPol parameter. The actual archive occurs in one of two ways:

■ Automatically the next time the VTV is recalled and remigrated.

■ On demand using the ARCHIve utility.

A "what if" for this might be to ensure compliance. It turns out that you have data that you have to retain for 7 years for external auditors, but your internal auditors might like to look at it once a year as well. Here is what that solution looks like:

```
TAPEREQ DSN(COMPLY.**) POLICY(PCOMPLY)
POLICY NAME(PCOMPLY) MEDIA(VIRTUAL) MGMT(COMPLY)
MGMT NAME(COMPLY) IMMMED(DELETE) MIGPOL(LOC1) -
ARCHAGE(365) ARCHPOL(REMDEEP)
STOR NAME(LOC1) ACS(00) MEDIA(STK1R)
STOR NAME(REMDEEP) ACS(01) MEDIA(STK2P)
```

For this scenario, the following occurs:

- All compliance data is immediately migrated to the local ACS and grouped on 9840 media. After the migration succeeds, the VTVs are deleted from the VTSS. The "archive age" for this data is 365 days, in case the internal auditors want to see it in the next year.

- After that, the data is eligible to be archived (moved to) 9940 media in the remote ACS.

Result: Compliance, at the best possible cost, while optimizing virtual resources.

Archive Usage Notes

As noted above, you have two methods to use for the actual archive: wait until the VTV is recalled and migrated, or do it on demand using the ARCHIve utility. The problem with waiting for remigration is that this is data that is not likely to be accessed. The chances are, the best way to archive VTVs is to run the ARCHIve utility periodically or as-needed.

Here are some tips for using the ARCHive utility:

- To select the VTVs to archive, you can specify one of the following parameters:

  - MGMTclas to archive the VTVs in Storage Class(es) specified by the ARCHAge/ARCHPol parameter of the specified Management Class(es).

  - VTV to archive a list or range of VTVs for the Management Class(es) for those VTVs.

  Note: If you do not specify a value for MGMTclas or VTV, VTCS scans all VTVs. You should probably do business through Management Class, but you can see situations where you want to do it by VTV volser or all VTVs.

- By not specifying the MOVEVTV parameter, you can get a report (only) that is a valuable "what if" picture of how many VTVs, MVCs, and total MBs you process with an archive request. Oracle *strongly recommends*, therefore, that you first run ARCHIve *without* MOVEVTV, then adjust the job as needed before specifying MOVEVTV. For more information, see *ELS Command, Control Statement, and Utility Reference*.

- Because demand archive can be resource intensive, you typically run ARCHIve during non-peak processing periods. You can also use the ARCHIve utility to override the CONFIG RECLAIM THRESHLD, MAXMVC, and CONMVC settings to optimize archive performance. You can also specify the maximum time for the archive in minutes on the ELAPSE parameter. Note that there are several limiting factors that influence archives (for example, MAXMVC and ELAPSE). VTCS enforces the strictest limiting factor. For example, if you run ARCHIve and specify ELAPSE equal to 5 hours and MAXMVC equal to 10 *and* VTCS archives 10 MVCs in one hour, then VTCS terminates the archive before the ELAPSE value expires.

- VTCS and HSC must be active to process a ARCHIve request except when you specify the POLICYdd parameter. POLICYdd (which forces "report only" mode) also provides an enhanced "what if" capability. You can create one or more alternate MGMTclas statements with different Archive Policies (different ARCHAge and ARCHPol values), and use POLICYdd to view the Archive Policy and resource use for each scenario.

- The RECONcil utility looks like ARCHive because RECONcil also moves VTVs from one Storage Class to another (that is, moves them from one MVC media to another and/or moves them from one ACS to another). Think of ARCHive as proactive and RECONcil as reactive, and you see the difference, as described in "Reconciling VTV Media and Location."

If the 365 days is up, and the internal auditors have not shown up, time to go to archive. The example below shows JCL to run ARCHive as follows:

- Archive VTVs in Management Classes COMPLY to 9940 media in the remote ACS.

- Set MAXMVC to 60, CONMVC to 8, and ELAPSE to 60 for the ARCHive job.

```
//ARCHIVE     EXEC PGM=SLUADMIN
//STEPLIBDD DSN=hlq.SEALINK,DISP=SHR //SLSPRINTDD SYSOUT=*
//SLSINDD *
ARCH MGMT(COMPLY) MAXMVC(60) CONMVC(8) ELAPSE(360) MOVEVTV
```

> **Tip:** The MOVEVTV parameter also gives you a report, so you can see how well (or not) you did. If your tuning parameters did not archive everything you wanted to archive, then adjust your job and rerun.

Reconciling VTV Media and Location

Using RECONcil to reconcile VTV media and location basically means moving VTVs from one Storage Class to another. Is this at all like archiving data with ARCHive? In terms of data movement, yes. In terms of the reasons why you are doing it, it is a reactive rather than proactive move. You typically reconcile VTVs when:

- The VTVs are on the wrong media, in the wrong ACS, or both.

- An ACS is unavailable for a considerable period of time, then is brought back online. In this case, you first change the MIGpol parameter on the MGMTclas statement for the affected VTVs to point to a different ACS (and media, if desired). When the original ACS comes back online, you then change the MIGpol parameter on the MGMTclas statement to point to the original ACS, and run RECONcil specifying the updated MGMTclas (or STORclas) statement(s) to move the VTVs to the original ACS.

For details of the reconciling process, see "RECONcil Example."

RECONcil Example

If you want to reconcile VTVs that are on the wrong media and in the wrong ACS, how would you know this? Peruse your VTV reports on a weekly basis as described in *Managing HSC and VTCS*. This week, you notice that all the VTVs in your production (PROD) Management Class are on the wrong media *and* in the wrong ACS! And the Storage Class does not look like the correct one, either.

How could this have happened? You *thought* you did the following:

```
STORCLAS NAME(LOC) ACS(00) MEDIA(STK1R)
STORCLAS NAME(CUSTA) ACS(00) MEDIA(STK1R)
STORCLAS NAME(CUSTB1) ACS(00) MEDIA(STK1R)
STORCLAS NAME(CUSTB2) ACS(01) MEDIA(STK2P
MGMT NAME(CUSTA) MIGPOL(CUSTA)
MGMT NAME(CUSTB) MIGPOL(CUSTB1,CUSTB2)
MGMT NAME(PROD) MIGPOL(LOC)
```

According to this example, everything in Management Class PROD should have ended up on 9840 media in the local ACS, but in fact, they are all on 9940 media in the remote ACS, almost as if they were in the wrong Storage Class.

On closer examination, the production Management Class actually looks like this:

```
MGMT NAME(PROD) MIGPOL(CUSTA)
```

This is not good for another reason, because it means the production data is co-resident on the same MVCs that are supposed to be dedicated to one of your customers. Time to run RECONcil, right? Perhaps not. RECONcil only moves VTVs out of the wrong Storage Class, and right now, according to the way you wrote your Management Class statement, CUSTA is the right Storage Class! Before you run RECONcil, you have to go back and fix the Management Class as follows:

```
MGMT NAME(PROD) MIGPOL(LOC)
```

Now you can run RECONcil as shown below:

■ Move VTVs in Management Class PROD to their (updated) correct location in Storage Class LOC.

■ Set MAXMVC to 60, CONMVC to 8, and ELAPSE to 60 for the RECONcil job.

```
//RECONCIL    EXEC PGM=SLUADMIN
//STEPLIBDD DSN=hlq.SEALINK,DISP=SHR
//SLSPRINTDD SYSOUT=*
//SLSINDD *
RECON MGMT(PROD) MAXMVC(60)
CONMVC(8) ELAPSE(360) MOVEVTV
```

RECONcil Usage Notes

To select the VTVs to reconcile, you can specify one of the following parameters:

■ MGMTclas to move the VTVs to the Storage Class(es) specified by the MIGpol parameter. This is what was done in "RECONcil Example." Management Class points to the wrong Storage Class, get it pointed toward the right Storage Class, run RECONcil against the updated Management Class.

■ STORclas to move the VTVs to the specified Storage Class(es). You probably use this when an ACS is unavailable for a considerable period of time.

■ MVC to reconcile the VTVs on a list or range of MVCs. The VTVs are moved to the Storage Class(es) specified by the MIGpol parameter of the MGMTclas statement(s) for the VTVs. You can use this first followed by the VTV option.

■ VTV to reconcile a list or range of VTVs. The VTVs are moved to the Storage Class(es) specified by the MIGpol parameter of the Management Class(es) for the VTVs.

Note:

- If you do not specify a value for MGMTclas or VTV, VTCS scans all VTVs.

- Because reconciling VTVs can be resource intensive, you typically run RECONcil during non-peak processing periods. You can also use the RECONcil utility to override the CONFIG RECLAIM THRESHLD, MAXMVC, and CONMVC settings to optimize reconcile performance. You can also specify the maximum time for the reconcile in minutes on the ELAPSE parameter.

  Note that there are several limiting factors that influence reconciliations (for example, MAXMVC and ELAPSE). VTCS enforces the strictest limiting factor. For example, if you run RECONcil and specify ELAPSE equal to 5 hours and MAXMVC equal to 10 *and* VTCS reconciles 10 MVCs in one hour, then VTCS terminates the reconciliations before the ELAPSE value expires.

- By not specifying the MOVEVTV parameter, you can get a report (only) that is a valuable "what if" picture of how many VTVs, MVCs, and total MBs you process with a reconcile request. Oracle *strongly recommends*, therefore, that you first run RECONcil *without* MOVEVTV, then adjust the job as needed before specifying MOVEVTV.

  For more information, see *ELS Command, Control Statement, and Utility Reference*.

- VTCS and HSC must be active to process a RECONcil request except when you specify the POLICYdd parameter. POLICYdd (which forces "report only" mode) also provides an enhanced "what if" capability. You can create one or more alternate MGMTclas statements with different reconciliation scenarios (different MIGpol values), and use POLICYdd to view the VTVs reconciled and resource use for each scenario.

- VTCS and HSC must be active to process a RECONcil request.

Named MVC Pools or Not?

Named MVC Pools are the right tool for the job for all of your service groups: You can use Named MVC Pools to give an application ownership of the MVCs in the named pool. For example, a service group might elect to use Named MVC Pools where its customers have legal requirements to buy and own a group of MVCs.

However, if you do not have a specific requirement for Named MVC Pools but you do want to group or segregate client data on MVCs, Oracle *strongly advises* that you do not use Named MVC Pools. Instead, use the methods described in the following sections:

- "Grouping Multiple Workloads on Shared MVCs"

- "Segregating Individual Workloads on Separate Sets of MVCs."

The above sections tell how to use Storage Classes to group or segregate data on MVCs that are selected from the system-wide MVC pool. In this case, you need to manage a only a single MVC pool.

If you create Named MVC Pools, you must *explicitly* manage each pool, which includes ensuring that each pool has sufficient free MVCs and available MVC space,

and potentially includes setting different policies for each pool using the MVCPool MVCFREE, MAXMVC, THRESH, and START parameters.

If you elect to use Named MVC Pools, go to "Creating and Using Named MVC Pools."

Creating and Using Named MVC Pools

To create and use Named MVC Pools, do the following:

1. Modify existing POOLPARM statements and/or add additional statements to define the Named MVC Pools.

   If you do not specify the POOLPARM NAME parameter, VTCS does not create a Named MVC Subpool and assigns the specified volumes to the default pool (DEFAULTPOOL). You cannot create Named MVC Pools with the reserved names DEFAULTPOOL and ALL.

   You can use the optional MVCFREE, MAXMVC, THRESH, and START parameters to specify values for the Named MVC Pool that override the global values specified on CONFIG.

   For example, the following VOLPARM and POOLPARM statements defines a range of T10000 Full volumes to be encrypted for Named Pool SYS1MVCT1 with reclamation parameter values that override the CONFIG global values:

   ```
   VOLPARM VOLSER(T10K2000-T10K2999)MEDIA(T10000T1)RECTECH(T1AE)
   POOLPARM NAME(SYS1MVCT1)TYPE(MVC)MVCFREE(40) MAXMVC(4) THRESH(60) START(70)
   ```

2. Run the SET VOLPARM to apply the volume and pool definitions:

   ```
   SET VOLPARM APPLY(YES)
   ```

3. Define Storage Classes and associate them with Named MVC Pools.

   For example, the following STORclas statement defines STORCL1 and associates this Storage Class with Named MVC Pool CUST1POOL. Requests to use MVCs for storage class STORC1 result in MVCs being selected only from the named pool SYS1MVCT1.

   ```
   STOR NAME(STORCL1) MEDIA(T!AE) MVCPOOL(SYS1MVCT1)
   ```

4. Create Management Classes that specify the Storage Classes you defined in step 3 and specify these Management Classes when you route data to the Named MVC Pool.

   For more information, see "Creating and Using VTCS Management and Storage Classes: The Basics."

5. Specify the Management Class name to VTCS on any of the following:

   - The SMC TAPEREQ statement.

   - SMS routines that you write to the StorageTek DFSMS interface. For more information, see *SMC Configuration and Administration*.

7

Finding and Fixing VTCS Problems

This section is about what to do when things go wrong. You have already done your dailies as described in "Using the VTCS Dashboard," and the as-needed tasks in "Restoring the CDS from a Backup Copy," and things are *still* not going well. Here is the place where you find out how to get VTCS working again when problems occur, starting with the simple problems you probably run across in "Fixing Common Problems."

> **Note:** Recovering the CDS is primarily an HSC task, but it also has a VSM side. For more information, see "Using PITCOPY to back up the CDS."

Fixing Common Problems

"Common," in this context, just means things that are likely to go wrong despite your best efforts. The way you find out about trouble often doubles back to taking another look at your VTCS Dashboard, and the fixes often reside in your as-needed tasks.

Before you begin with VTV mount performance problems, these are common problems that you can *generally* diagnose and fix on your own. After making a reasonable effort, however, if things still are not working out, it is time to call for help from Customer Support. There are also some tools that are not discussed here, like traces, because you basically only want to use them under the direction of Oracle service.

Poor VTV Mount Performance

If VTV mounts occur very slowly or not at all, check the following:

- Are mounts failing on a single VTD? This usually occurs because a host requests a mount of an MVC–resident VTV that VSM cannot recall. If so, do the following:

  - Enter a Display Queue DETail command to check the queued recalls. If a recall is queued waiting for an MVC, it may be in use by another VTCS process, which you can check with Display Active DETail.

  - If the MVC is *not* in use, next enter an HSC DISPLAY VOLUME command. Is the MVC actually in the ACS? If not, you must reenter the MVC to complete the recall.

  - Next, are RTDs available to mount the MVC to recall the VTV? Enter Display RTD to check RTD availability. If no RTDs are available, use Display on all hosts to check active and queued processes.

If necessary, use Cancel to cancel processes and free an RTD so the recall can complete. With Cancel, VTCS tries to stop processes without affecting system resources or information; therefore, the cancellation may not occur immediately. For example, VTCS may wait for hardware time out periods before terminating a process using a specific RTD.

Note: If you cancel a parent request, you stop the parent and all child requests. If you cancel a child request, the parent request continues processing.

Caution: If you cancel a task associated with migration scheduler (either with the MIGrate parameter or by specific process ID), this task terminates but migration scheduler starts another migration task at its next timer interval. You can, however, use migrate-to-threshold to stop automigration by specifying a value greater than the current DBU.

Tip: Setting the MGMTclas statement IMMEDmig parameter to either KEEP or DELETE preferences migration processing (and RTD use for migration) and may increase I/O to the RTDs.

Also note that you can change the CONFIG MAXMIG and MINMIG parameter settings to rebalance automatic migration tasks with other tasks (such as recall and reclaim) for the RTDs you have defined for each VTSS.

- Are the mounts failing on multiple VTDs? If so, check the following:

  - Check VTD status with Display VTD.

  - Enter Display Active. If there are no active processes, ensure that VTCS, HSC, all VTSSs, and all communications are functioning normally.

  - Ensure that you have sufficient VTSS space.

  - Check to see if your system is running out of available MVCs or usable MVC space.

  - Raising the low AMT tends to keep more VTVs resident in VTSS space, which may help prevent virtual mounts from failing.

- If a VTV mount fails, even if VTDs are online, use the MVS VARY command to vary VTDs online, use the MVS UNLOAD command to clear the VTDs, then use the HSC MOUNT and DISMOUNT commands to retry the operation.

Poor Migration Performance

If VTV migration occurs very slowly, check the following:

- Start with Display MIGrate, which shows you, in broad strokes, how well or poorly your various migration tasks are doing. You may be able to rearrange the furniture (for example, raise the MAXMIG/MINMIG values) to get things moving.

- Ensure that your supply of RTDs and MVCs is in good shape as described in "Checking Virtual Tape Status (Daily)." If you want to get down to bits and bytes, also use Display Queue DETail to check the status of queued processes. If many

processes are waiting for RTDs, and you are sharing RTDs with MVS, you may want to vary transports offline to MVS and online to VSM.

Note: In the JES3 environment, VTV mounts may fail if you have not created and installed the correct User Exit modifications.

Migration Failures

There is only one thing worse than poor migration performance, and that is no migration at all. Fortunately, VTCS provides detailed information about migration failures as described in the following sections:

- "Messages Enhancements"
- "Display STORCLas"
- "Enhanced MVC Pool Validation"
- "Enhanced Storage Class Validation."

Messages Enhancements

To provide greater detail about migration failures, message SLS6700E is replaced by the following messages:

- SLS6853E Migration failed Storage Class:*stor-clas-name* ACS:*acs-id* VTSS:*vtss-name* - MVCPool *poolname* is not defined
- SLS6854E Migration failed Storage Class:*stor-clas-name* ACS:*acs-id* VTSS:*vtss-name* - no MVCs found for specified media
- SLS6855E Migration failed Storage Class:*stor-clas-name* ACS:*acs-id* VTSS:*vtss-name* - no MVCs found for specified media/SC/ACS
- SLS6856E Migration failed Storage Class:*stor-clas-name* ACS:*acs-id* VTSS:*vtss-name* - no usable MVCs found for specified media/SC/ACS
- SLS6857E Migration failed Storage Class:*stor-clas-name* ACS:*acs-id* VTSS:*vtss-name* - no RTDs for requested media and ACS
- SLS6858E Migration failed Storage Class:*stor-clas-name* ACS:*acs-id* VTSS:*vtss-name* - all RTDs for requested media and ACS are offline
- SLS6859E Migration failed Storage Class:*stor-clas-name* ACS:*acs-id* VTSS:*vtss-name* - unknown reason (X'*xx*')

In addition, message SLS6860I is always output after any of the preceding messages are issued to provide details of the Storage Class. If applicable, SLS6860I also reports any errors with regard to satisfying the migration requirements:

- If the MVC Pool is undefined.
- If the MVC Pool contains none of the specified media.
- If the MVC Pool contains no free MVCs of the specified media.
- If the VTSS/ACS has no suitable RTD defined to write the migration MVC.
- If all suitable RTDs are offline.

The result is that you are now getting more detailed information, and more specific recommendations for fixes when migration failures *do* occur.

Display STORCLas

Display is enhanced with the STORCLas parameter, whose output is:

- The characteristics of the Storage Class (ACS, MVC Pool, and Media).

- VTVs waiting migration to the Storage Class from any VTSS.

- Requirements of the MVCs to be used for migration.

- The device type(s) of the RTDs needed to write to the migration MVCs.

- Any errors with regard to satisfying the migration requirements.

Once again, VTCS provides information about a critical element (Storage Classes) in the migration scenario.

Enhanced MVC Pool Validation

Validation of MVC Pools is enhanced to check for common set-up errors:

- Has at least one valid MVC Pool been defined? If not, message SLS6845E is issued. VTCS functionality is severely degraded because no migrations can occur. If you receive this message, you must define appropriate MVC Pools. See the next bullet.

- Does the default MVC Pool (DEFAULTPOOL) exist? DEFAULTPOOL is used when migrating to a Storage Class that does not specify a Named MVC Pool and in error situations with Storage Class !ERROR. If DEFAULTPOOL does not exist, message SLS6846W is issued.

  You indicate migrations to a Storage Class should use a particular MVC Pool by coding MVCPool(*pool-name*) on the STORCLAS statement. If MVCPool(*pool-name*) is **not** coded, VTCS treats the STORCLAS as though MVCPool(DEFAULTPOOL) was coded.

Enhanced Storage Class Validation

To continue in this theme, validation of Storage Classes is enhanced to check for common set-up errors:

- If you specify a Named MVC Pool on a Storage Class (STORCLAS NAME(*stor-clas-name*) MVCPOOL(*poolname*)), VTCS checks that the Named MVC Pool is defined. Therefore, if you code STORCLAS NAME(*stor-clas-name*) MVCPOOL(*poolname*),ensure that the Named MVC Pool exists. If not, VTCS issues message SLS6848W. If you get this message, define the Named MVC Pool, change your Storage Class definition, or both.

- Similarly, if you do *not* specify a Named MVC Pool on a Storage Class (STORCLAS NAME(*stor-clas-name*), VTCS checks that the DEFAULTPOOL is defined. Therefore, if you code STORCLAS NAME(*stor-clas-name*), ensure that there is *at least* one MVCPOOL statement that does not create a Named MVC Pool. If not, VTCS issues message SLS6846W. If you get this message, code at least one MVCPOOL statement that does not create a Named MVC Pool, change your Storage Class definition, or both.

- If you specify an MVC media on a Storage Class (STORCLAS NAME(*stor-clas-name*) MEDIA(*media-type*)), VTCS checks that the MVC Pool contains media of type *media-type* (if a Named MVC Pool is not specified, DEFAULTPOOL is implied). If not, message VTCS issues message SLS6849W. Ensure that the media type exists in the corresponding pool, change your Storage Class definition, or both.

■ If you specify an ACS and media type on a Storage Class (STORCLAS NAME(*stor-clas-name*) ACS(*acs-id*) MEDIA(*media-type*)), VTCS checks that there are RTDs in the specified ACS compatible with the specified media type. If not, message VTCS issues message SLS6851W. Ensure that required RTD type exists in the specified ACS, change your Storage Class definition, or both.

■ If you specify media type without a specific ACS on a Storage Class (STORCLAS NAME(*stor-clas-name*) MEDIA(*media-type*)), VTCS checks that there are RTDs in the configuration compatible with the specified media type. If not, message VTCS issues message SLS6851W. Ensure that required RTD type(s) exist in the configuration, change your Storage Class definition, or both.

RTD/MVC Failures

At first, you may not know if you are looking at a media or drive failure. That is, if VTCS detects read/write errors on an MVC, VTCS swaps the MVC to another RTD. If VTCS detects no further read/write errors on the MVC, VTCS assumes that the first RTD is in error.

Message SLS6662A indicates that an RTD is in maintenance mode, and this status is also reported on Display RTD output. An RTD in maintenance mode is typically in error and requires assistance from your hardware operations or service personnel. Note that an RTD in recovery mode is initializing (when varied online, for example), and typically is not in error.

If a failed RTD cannot be quickly repaired or if the failed RTD is attached to a remote ACS, you may want to remove the RTD from your configuration to prevent attempts to allocate that RTD. Remove the RTD statement for the RTD and rerun CONFIG.

> **Caution:** In a dual-ACS configuration (two ACSs connected to a single VTSS), ensure that you *do not* allow all RTDs in either ACS to be unavailable to the VTSS for an extended period. If no RTDs are available in that ACS, migrates to or recalls from that ACS cannot occur, and the VTSS space can fill up. In addition, this condition can also cause stalled migrations to RTDs in the other ACS.

In a dual-ACS configuration, therefore, if you must make all RTDs in an ACS unavailable for an extended period, remove the RTDs from the configuration as described above.

Is it a Bad MVC?

If you ran through the check list for RTD problems, above, and that is not the problem, and you also did all the things you can reasonably do to make more MVC space available, and compared the volsers on the MVC Summary Report to an HSC Volume Report, the MVCs actually were in the ACS. Otherwise, you either reenter or replace any MVCs not listed on the HSC Volume Report.

It really *does* look like a media problem. You see what kind of media problem looking at the MVC and VTV reports described in "Checking Virtual Tape Status (Daily)." That section talks about some of the fixes for the most straightforward MVC anomalies. The following is an exhaustive list of the MVC statuses you *do not* want to see on your MVC and VTV reports, and what to do about them:

BROKEN

This is a generic error that indicates the MVC, drive, or combination of the two has a problem. VTCS attempts to de-preference MVCs with this state. *In general,* to clear this state:

If the MVC caused the problem, use a DRAIN(EJECT) command to remove the MVC from service.

If the RTD caused the problem, use the MVCMAINT utility to reset the MVC state.

Note also that one or more of the following messages is issued for BROKEN status: SLS6686, SLS6687, SLS6688, SLS6690. For detailed recovery procedures for these messages, see *VTCS Messages and Codes*.

DATA CHECK

A data check condition has been reported against this MVC. VTCS attempts to de-preference MVCs with this state. To clear this state:

If all VTVs on the MVC are duplexed, use MVCDRain on the MVC without the Eject option. This recovers all VTVs and removes the MVC from service.

If all VTVs on the MVC are not duplexed, VTCS AUDIT the MVC. The audit probably fails. After the audit, do an MVCDRAIN (no eject). This recalls the VTVs before the data-check area in ascending block-id order and the VTVs after the data-check area in a descending block-id order. Processing the VTVs in this sequence ensures that VTCS recovers as many VTVs as possible from the media. You then need to recreate the data for any VTVs still on the MVC.

After clearing data checks, remove and replace MVCs with data check errors as described in "Permanently Removing MVCs." This procedure also tells how to remove an MVC from VTCS use and return it to Nearline operations.

DRAINING

The MVC is either currently being drained or has been the subject of a failed MVCDRain.

IN ERROR

An error occurred while the MVC was mounted.

INITIALIZED

The MVC has been initialized.

LOST - FAILED TO MOUNT

VTCS attempted to mount an MVC and the mount did not complete within a 15-minute time-out period. VTCS is attempting to recover from a situation that may be caused by hardware problems, HSC problems, or by the MVC being removed from the ACS. VTCS attempts to de-preference MVCs with this state.

If VTCS does perform a subsequent successful mount of an MVC with LOST(ON) state, VTCS sets the state to LOST(OFF).

Determine the cause of the error and fix it. You can also use the VTCS MVCMAINT utility to set LOST(OFF) for the following events:

LOST(ON) was set due to LSM failures or drive errors that have been resolved.

LOST(ON) was set because the MVC was outside the ACS and has been reentered.

MARKED FULL

The MVC is full and is not a candidate for future migrations.

MOUNTED

The MVC is mounted on an RTD.

NOT-INITIALIZED

The MVC has been defined through the CONFIG utility, but has not ever been used.

READ ONLY

The MVC has been marked read-only because of one of the following conditions:

- The MVC being the target of an export or consolidation process. The read-only state protects the MVC from further updates.

- The MVC media is set to file protect. Correct the error and use the MVCMAINT utility to set READONLY(OFF).

- The MVC does not having the appropriate SAF rules set to enable VTCS to update the MVC. Correct the error (for more information, see "Defining A Security System User ID for HSC, SMC, and VTCS" in *Installing ELS* and use the MVCMAINT utility to set READONLY(OFF).

BEING AUDITED

The MVC is either currently being audited or has been the subject of a failed audit. If the audit failed, VTCS does not use the MVC for migration. To clear this condition, rerun the AUDIT utility against this MVC.

LOGICALLY EJECTED

The MVC has either been the subject of an MVCDRain Eject or the MVC was ejected for update by a RACROUTE call. The MVC is not be used again for migration or recall. To clear this condition, use MVCDRain against the MVC without the Eject option.

RETIRED

The MVC is retired. VTCS recalls from, but does not migrate to, the MVC. Replace the MVC as soon as possible.

WARRANTY HAS EXPIRED

The MVC's warranty has expired. VTCS continues to use the MVC. You should start making plans to replace the MVC when it reaches Retired state.

INVALID MIR

VTCS has received status from an RTD to indicate the MIR (media information record) for a 9x40 media is invalid. An invalid MIR does not prevent access to data but may cause significant performance problems while accessing records on the tape. The MVC is not capable of high-speed searches on areas of the tape that do not have a valid MIR entry.

VTCS attempts to de-preference MVCs with this condition. For recalls, if the VTV resides on multiple MVCs, VTCS selects MVCs with valid MIRs ahead of MVCs with invalid MIRs. VTCS avoids using MVCs with invalid MIRs for migration, unless the migration is at the beginning of the tape. Migrating from the beginning of tape corrects the MIR.

VTCS detects the invalid MIR condition at either mount time or dismount time. If detected at mount time and the operation can be completed with another MVC, VTCS dismounts the first MVC and selects the alternate MVC. *Note that* VTCS has only a limited ability to switch to an alternate MVC. That is, it is mainly used for migrate and virtual mount.

For MVCs with invalid MIRs, determine the cause of the error, which may be caused by media or drive problems, and fix the error.

To recover an MVC with an invalid MIR, run the `INVENTRY` utility. For example, to recover `MVC707`, enter:

```
INVENTRY MVCID(MVC707)
```

Recovering an MVC with a Data Check

This is a very specific instance of the general "bad MVC" woes, and you know it is required when you see an MVC data check error on your MVC and VTV reports.

To recover an MVC with a Data Check:

1. Run an MVC audit against the MVC.

   The audit attempts to read the VTV metadata sequentially from the MVC. The audit fails when it encounters the data check, which leaves the MVC in an auditing state. This prevents VTCS from selecting this MVC for output.

2. Run an MVCDRain Eject for the MVC.

   This causes all the available VTVs to be recalled to a VTSS and then remigrated to a new error-free MVC. This logically removes the MVC from the MVC pool.

   > **Note:**
   >
   > - Due to the error status of the MVC, VTCS recalls VTVs from alternate MVCs if possible.
   >
   > - If VTVs must be recalled from the MVC in error (no other copies available), then:
   >
   >   - VTVs *before* the data check area are recalled in ascending block ID order.
   >
   >   - VTVs *after* the data check area are recalled in descending block ID order.

3. Determine if any VTVs could not be recovered from the MVC.

   Run an MVC Detail report for the MVC. If any VTVs are still reported as being on the MVC, then these VTVs are not recoverable; you must use other methods to recover your data.

4. Manage the defective MVC by doing one of the following:

   Replace the defective MVC with an initialized tape volume with the same internal and external labels:

   a. Enter the HSC `EJECT` command for the defective MVC.

   b. Enter the HSC `ENTER` command for the replacement MVC.

   c. Initialize the tape as required.

   d. Enter HSC `AUDIT` for the new MVC.

   e. Run an `MVCDRAIN` (no EJECT) to return the MVC to the MVC pool.

   Remove the MVC from the system:

   a. Enter the HSC `EJECT` command for the defective MVC.

   b. Edit the MVC pool definitions to remove the defective MVC from the pool.

    c. Enter a VT MVCDEF on all active hosts to activate the new MVC pool definitions.

Using the RTV Utility

The RTV utility is another item you are probably only going to use after talking with Oracle service, because RTV is designed to read VTV data directly from an MVC without any assistance from VTCS, for example, in the case that you really *have* lost the CDS.

RTV is a standalone utility, and the way it works is to read a VTV from an MVC, decompress the VTV, then write the data to a single output tape (real tape volume) so the data can be read by user applications. Because RTV utility is a stand–alone utility; you can run RTV when VSM is down but the MVS system is up.

What the RTV Utility Can Recover

The RTV utility can recover:

- All or specified VTVs from a specified MVC. If you do not know the location of the most current version of a VTV on the MVC, specify only the VTV volser, and RTV converts the most current version of the VTV it finds on this MVC.

- A VTV at a specified block ID on a specified MVC. The LISTONLY parameter listing supplies a Block ID value that you can use as input to the RTV utility to convert a VTV to a Nearline volume. Specifying the volser and Block ID speeds positioning time.

- A VTV specified by logical data set number on a specified MVC. Specifying the volser and logical data set number will have a much longer positioning time compared to specifying volser and Block ID. Using volser and Block ID is the preferred method to access a single VTV.

> **Note:** If more than one VTV is specified, or if no block-id or FILEnum parameter is specified, the entire MVC is read and the MVC contents displayed as part of the output. Reading of the entire MVC is necessary to insure that only the most current copy of a VTV is decompressed.

General Usage Guidelines

- The output volume that contains the converted VTV(s) must be at least the size of your maximum VTV size (400 Mb, 800 Mb, 2 Gb, or 4 Gb) to ensure that it can contain an individual VTV.

- The VTCS MVC and VTV reports provide information to specify which copy of a VTV you want RTV to recover. Ensure that you have a current copy of these reports before you run the RTV utility. In addition, to help identify the VTVs you want to convert, you can use the LISTONLY parameter to produce a list of the VTVs on an MVC.

  Because multiple copies of the same VTV can exist on the same or different MVCs, *study carefully* your VTV and MVC reports and LISTONLY listings to ensure that you are using the correct MVC to convert the most current copy of a VTV!

- The RTV utility does not update the system catalog or TMC with information about the converted volumes; you must do this manually.

Security Considerations

- You must have read access both to the VTVs you want to convert and to the MVC that contains these VTVs or your system's security application cannot be running. Otherwise, the conversion fails.

- Ensure that you APF authorize the RTV utility load library.

- RTV makes no attempts to bypass any TMS protection. All RTV tape mounts are subject to full TMS control.

> **Note:** Because the RTV utility must be capable of rewriting the tape standard labels on the output unit and positioning over label information on the input unit, Dynamic Allocation is used to invoke bypass label processing (BLP) on the tape volumes. This requires that the library that contains the SWSRTV executable code be APF authorized.

JCL Examples

The following displays JCL examples using the RTV utility.

Listing the VTVs on an MVC

The following shows example JCL to lists the VTVs on MVC MVC001.

```
//JOBVRECJOB(account),programmer
//RUNRTV EXEC PGM=SWSRTV,PARM='MIXED'
//STEPLIBDD DSN=hlq.SEALINK,DISP=SHR
//SLSPRINTDD SYSOUT=A
//SLSINDD *
RTV  MVC(MVC001)INUNIT(/1AB4) LISTONLY
/*
//
```

Converting a Single VTV by Specifying Its Volser

The following shows example JCL to run the RTV utility to convert VTV VTV200 on MVC MVC001, which is mounted on a 3490E transport. The output (converted VTV VTV200) goes to the output volume mounted on transport 280, and RTV copies the VTV VOLID from the VTV to the output volume.

```
//JOBVRECJOB(account),programmer
//RUNRTV EXEC PGM=SWSRTV,PARM='MIXED'
//STEPLIBDD DSN=hlq.SEALINK,DISP=SHR
//SLSPRINTDD SYSOUT=A
//SLSINDD *
  RTV  MVC(MVC001) INUNIT(3490E) VTV(VTV200) CPYVOLID OUTUNIT(280)
/*
//
```

Converting a Single VTV by Specifying Its Volser and Block ID

The following shows example JCL to run the RTV utility to convert VTV VTV200 at block ID x'8EA484AB' on MVC MVC001, which is mounted on a 3490E transport. The output (converted VTV VTV200) goes to the output volume mounted on transport 480.

```
//JOBVRECJOB(account),programmer
//RUNRTV EXEC PGM=SWSRTV,PARM='MIXED'
```

```
//STEPLIBDD DSN=hlq.SEALINK,DISP=SHR
//SLSPRINTDD SYSOUT=A
//SLSINDD *
  RTV  MVC(MVC001) INUNIT(3490E) VTV(VTV200) BLOCK(8EA484AB) OUTUNIT(480)
/*
//
```

A

LCM Control Statements

LCM control statements are listed in this appendix.

Table A–1   LCM Control Statements

LCM Control Statement	Status	HSC/VTCS Utility Command
ACTION CONSOLIDATE	Unchanged	CONSOLIDATE
ACTION DRAIN	Enhanced	MVCDRAIN
ACTION EJECT	Unchanged	EJECT
ACTION EXPORT	Unchanged	EXPORT
ACTION MIGRATE	Unchanged	MIGRATE
ACTION MOVE	Unchanged	MOVE
ACTION RECALL	Unchanged	RECALL
ACTION RECLAIM	Enhanced	RECLAIM
ACTION SCRATCH	Unchanged	SCRATCH
ACTION UNSCRATCH	Unchanged	UNSCRATCH, REPLACEALL
ACTION VAULT ASSIGN	Enhanced	
ACTION VAULT RELEASE	Enhanced	
ACTION VAULT RETURN	Enhanced	
DATASET	Unchanged	
LOCATION	Unchanged	
MANAGE PHYSICAL	Enhanced	
MANAGE VIRTUAL	Unchanged	
METHOD	Unchanged	
OPTIONS	Enhanced	
OPTIONS SYNC and SYNCVTV MANAGE PHYSICAL and MANAGE VIRTUAL	Enhanced	SLUCONDB (and SMC equivalent)
PULLLIST	Unchanged	
REPORT CELLCNT	Enhanced	
REPORT CONSOLIDATE	Unchanged	
REPORT DATASET	Unchanged	

LCM Control Statement	Status	HSC/VTCS Utility Command
REPORT EJECT	Unchanged	
REPORT ENTER	Unchanged	
REPORT EXPORT	Unchanged	
REPORT LSM	Enhanced	
REPORT MIGRATE	Unchanged	
REPORT MULTIPLE	Unchanged	MVCRPT
REPORT MVC_VTV	Enhanced	MVCRPT
REPORT NONSCRCNT	Enhanced	
REPORT OPERATOR	Unchanged	
REPORT PHYSICAL	Unchanged	VOLRPT
REPORT RECALL	Unchanged	
REPORT SCRCNT	Enhanced	
REPORT SUMMARY	Enhanced	
REPORT VIRTUAL	Unchanged	VTVRPT
REPORT VOLUME	Enhanced	
SET METHOD	Unchanged	
SUBPOOL	Unchanged	
SUBPOOL and MANAGE PHYSICAL BALSCR	Unchanged	SCRATCH REDISTRIBUTION
TMS CA1	Unchanged	
TMS COMMON	Unchanged	
TMS CTT	Unchanged	
TMS CUSTOM	Unchanged	
TMS OPEN	Unchanged	
TMS RMM	Unchanged	
TMS TLMS	Unchanged	
UNMANAGED	Unchanged	

Index